Class: III BCOM (CA) Course Code: 16CCU601A Course Name: INTERNET AND WEB DESIGN

Unit -V Semester: VI

Year: 2016-2019 Batch

UNIT-V- XML (EXTENSIBLE MARKUP LANGUAGE) SYLLABUS

Introduction to XML: Introduction – The Syntax of XML – XML Document Structure – Document Type Definitions – Namespaces – XML Schemas – Displaying Raw XML Documents – Displaying XML Documents with CSS – XML Processors.

Introduction to XML:

Introduction

XML—EXTENSIBLE MARKUP LANGUAGE

The eXtensible Markup Language is the language that another version of HTML is based on. Like HTML, XML is also based off of SGML. It is less strict than SGML and stricter than plain HTML. XML provides the extensibility to create various different languages.

XML is a language for writing markup languages. For example, if you are working on genealogy, you might create tags using XML to define the father, mother, daughter, and son in your XML like this: <father> <mother> <daughter> <son>. There are also several standardized languages already created with XML: MathML for defining mathematics, SMIL for working with multimedia, XHTML, and many others.

XHTML—EXTENDED HYPERTEXT MARKUP LANGUAGE

XHTML 1.0 is HTML 4.0 redefined to meet the XML standard.

XHTML has been replaced in modern web design with HTML5 and the changes that have come since. You are unlikely to find any newer sites using XHTML, but if you are working on a much older site, you may still encounter XHTML out there in the wild.

There aren't a lot of major differences between HTML and XHTML, but here is what you will notice:

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 1/31

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATOREClass: III BCOM (CA)Course Name: INTERNET AND WEB DESIGN

Course Code: 16CCU601A Unit -V Semester: VI Year: 2016-2019 Batch

XHTML is written in lower case. While HTML tags can be written in UPPER case, MiXeD case, or lower case, to be correct, XHTML tags must be all lower case. (Note - many web professionals write HTML in all lowercase, even though it is not technical required).

• All XHTML elements must have an end tag. Elements with only one tag, such as and need a closing slash (/) at the end of the tag:

<hr />

• All attributes must be quoted in XHTML. Some people remove the quotes around attributes to save space, but they are required for correct XHTML.

• XHTML requires that tags are nested correctly. If you open a bold () element and then an italics (<i>) element, you must close the italics element (</i>) before you close the bold ().

(Note that both of these elements have been deprecated because they are visual elements. HTML

now uses and in place of these two)

• HTML Attributes must have a name and a value. Attributes that are stand-alone in HTML must be declared with values as well, for example, the HR attribute would be written noshade="noshade".

What is XML?

- XML stands for eXtensible Markup Language
- XML is a markup language much like HTML
- XML was designed to store and transport data
- XML was designed to be self-descriptive
- XML is a W3C Recommendation

Example

<?xml version="1.0" encoding="UTF-8"?>

<note>

<to>Tove</to>

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 2/31

Class: III BCOM (CA)	Cour	se Name: INTERNE	T AND WEB DESIGN
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch
<from>Jani</from>			
<heading>Reminder<td>ing></td><td></td><td></td></heading>	ing>		
<body>Don't forget me this</body>	weekend! <td>~></td> <td></td>	~>	
XML Tags are Case Sensit	ve		
XML tags are case sensitive.	The tag <letter></letter>	is different from the	tag <letter>.</letter>
Opening and closing tags n	ust be written w	ith the same case:	
<message>This is incorrect<</message>	/message>		
<message>This is correct<td>nessage></td><td></td><td></td></message>	nessage>		
"Opening and closing tags" a	re often referred	to as "Start and end ta	ags". Use whatever you prefer.
is exactly the same thing.			
VML Flomonts Must be Pr	onorly Nostad		
AML Elements Must be FI	operty Nesteu		
In HTML, you might see im	properly nested el	ements:	
<i>This text is bold and</i>	italic	<u>_</u>	
In XML, all elements must b	e properly nested	within each other:	
<i>This text is bold and</i>	italic		
In the exemple shows "Drop	rly nested" simpl	y means that since th	a civalament is anonad inside

XML Attribute Values Must be Quoted

the *<*b> element, it must be closed inside the *<*b> element.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 3/31

Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN				
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch	
XML elements can have attributed	ites in name/va	lue pairs just like in H	ITML. In XML, the attribute	
values must always be quoted.				
INCORRECT:				
<note date="12/11/2007"></note>				
<to>Tove</to>				
<from>Jani</from>				
CORRECT:				
<pre><note data="12/11/2007"></note></pre>				
<1101e uale= 12/11/2007 >				
<to>Tove</to>				
<to>Tove</to> <from>Jani</from>				
<to>Tove</to> <from>Jani</from> 				
<to>Tove</to> <from>Jani</from> XML Elements				

What is an XML Element?

An XML element is everything from (including) the element's start tag to (including) the element's end tag.

<price>29.99</price> An element can contain:

```
• text
```

- attributes
- other elements
- or a mix of the above

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 4/31

Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN					
Cours	se Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch	
Comn	nents in XML				
The sy	ntax for writing commen	ts in XML is si	imilar to that of HTM	L.	
T]</td <td>his is a comment></td> <td></td> <td></td> <td></td>	his is a comment>				
Two d	lashes in the middle of a c	comment are no	ot allowed. Not allow	ed:	
T!</td <td>his is a comment -></td> <td></td> <td></td> <td></td>	his is a comment ->				
Strang	ge, but allowed:				
T!</td <td>his is a comment></td> <td></td> <td></td> <td></td>	his is a comment>				
White	-snace is preserved in X	ML			
XML white-	does not truncate multiple	white-spaces	(HTML truncates mu	ltiple white-spaces to one sing	
XML white- XML:	does not truncate multiple space): Hello	e white-spaces	(HTML truncates mu	ltiple white-spaces to one sing	
XML white- XML: HTMI	does not truncate multiple space): Hello	e white-spaces	(HTML truncates mu Tove	ltiple white-spaces to one sing	
XML white- XML: HTMI The D	does not truncate multiple space): L: Hello	e white-spaces Tove and HTML:	(HTML truncates mu Tove	ltiple white-spaces to one sing	
XML white- XML: HTMI The D	does not truncate multiple space): Hello L: Hello HTML HTML	e white-spaces Tove and HTML:	(HTML truncates mu Tove	Itiple white-spaces to one sing	
XML white- XML: HTMI The D	does not truncate multiple space): Hello L: Hello HTML HTML is used to displ	e white-spaces Tove and HTML:	(HTML truncates mu Tove	Itiple white-spaces to one sing	

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 5/31

languages.

XML is case sensitive.

XML provides a **framework to define markup**

2)

3)

itself.

HTML is a **markup language**

HTML is **not case sensitive**.

Class: III BCOM (CA) Co			urse Name: INTERNET AND WEB DESIGN	
Cours	e Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch
5)	HTML has its own pr	redefined Y	You can define tags a	eccording to your need.
	tags.			
		Σ	KML makes it mand	atory to use a closing tag.
7)	HTML is static because	se it is used to Σ	KML is dynamic beca	ause it is used to transport data
	display data.			
		X	ML preserve whites	paces.
			Freedor (Contraction	F m 21

The XML language has no predefined tags.

The tags in the example above (like <to> and <from>) are not defined in any XML standard. These tags are "invented" by the author of the XML document. HTML works with predefined tags like , <h1>, , etc.

Most XML applications will work as expected even if new data is added (or removed).

Imagine an application designed to display the original version of note.xml (<to> <from> <heading> <data>).

Then imagine a newer version of note.xml with added <date> and <hour> elements, and a removed <heading>.

The way XML is constructed; older version of the application can still work:

<note>

<date>2015-09-01</date>

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 6/31

Class: III	BCOM (CA)	Cours	se Name: INTERNI	ET AND WEB DESIGN
Course Co	ode: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch
<hour>08</hour>	30			
<to>Raja<</to>	z/to>			
<from>Ra</from>	m			
<body>Do</body>	on't forget me this w	eekend!	>	
Design go	als for XML			
•	XML shall be straig	htforwardly usal	ble over the Internet.	
•	XML shall support	a wide variety of	applications.	
•	XML shall be comp	atible with SGM	IL.	
•	It shall be easy to w	rite programs w	hich process XML de	ocuments.
•	The number of opti-	onal features in X	KML is to be kept to	the absolute minimum, ideally
zero.				
•	XML documents sh	ould be human-l	egible and reasonabl	y clear.
•	The XML design sh	ould be prepared	l quickly.	
•	The design of XML	shall be formal	and concise.	
•	XML documents sh	all be easy to cre	eate.	
•	Terseness in XML	narkup is of min	imal importance.	
A Family	of Standards and I	Recommendatio	ns:	
XML is no	ot a single standard.	The core XML 1	.0 syntactic definitio	on is enhanced by the following
standards	that are either comp	lete or under dev	elopment:	
Recomme	endations			
•	XML 1.0: February	1998 (Revised (October 2000)	
•	XML Namespaces:	January 1999		
•	XSLT: November 1	999 (part of XS	L)	
•	XPath: November 1	999 (used by XS	SLT and XPointer)	
•	XHTML (HTML re	edefined as XML	.): January 2000	
•	XHTML Basic: De	cember 2000 (Ba	ase subset)	
•	Canonical XML: M	arch 2001		

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 7/31

	KARPAGAM ACA	DEMY OF H	IGHER EDUCATIO	DN, COIMBATORE
Class: III	I BCOM (CA)	Cours	se Name: INTERNE	T AND WEB DESIGN
Course C	Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch
•	XML Schemas Parts 1	and 2: May 2	2001	
•	XLink and XBase: Jun	ne 2001		
•	XSL: October 2001			
•	XML Information Set	: October 200	1	
•	XML-Signature: Febr	uary 2002		
Proposed	Recommendations			
•	Exclusive XML Canor	nicalization: N	/lay 2002	
Candidat	te Recommendations			
•	XML Fragment Interc	hange: Februa	ary 2001	
•	XPointer: September 2	2001		
•	XInclude: February 20)02		
•	XML Signature Decry	ption: March	2002	
•	XML Encryption: Ma	rch 2002		
Working	Drafts			
•	XML Events: October	2001		
•	XML Protocol: Decen	nber 2001		,
•	XForms 1.0: January 2	2002		
•	XML Key Manageme	nt: March 200	02	
•	XKMS: March 2002			
•	XPath 2.0: April 2002			
•	XQuery: Arpil 2002			
•	XML 1.1: April 2002			
Importar	nt XML Standards			
This tutor	rial will also dig deep in	to the followi	ng important XML sta	andards:
•	XML AJAX			
•	XML DOM			

- XML XPath
- XML XSLT

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 8/31

Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN					
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch		
• XML XQuery					
• XML DTD					
• XML Schema					
• XML Services					
Following is a partial list of re	elated XML-re	lated areas being wo	orked on at W3C:		
- Associating Style	e Sheets with X	ML documents Versi	on 1.0 Recommendation W30		
[June 29, 1999]					
Standardized syntax for using a	n xml- styleshe	et processing instruct	ion to associate an XML		
document with an XSL or CSS	stylesheet.				
- Authoring Tool A	Accessibility G	uidelines Version 1.0			
Recommendation W3C [Feb.	3, 2000] Guid	elines for web authori	ng tool developers to assist in		
the design of authoring tools th	at produce acce	ssible Web content a	nd in the creation of an		
accessible authoring interface.					
- CSS1 Recomme	ndation (Revise	ed) W3C [Jan. 11, 1	1999] CSS1 is a simple style		
sheet mechanism that allows au	thors and reade	ers to attach style (e.g.	fonts, colors and spacing) to		
HTML documents.					
- CSS2 Recommen	ndation W3C	[May 12, 1998] Leve	el 2 of the Cascading Style Shee		
mechanism					
- DOM Level 1 Re	commendation	W3C [Oct. 1, 1998	8] Document Object Model		
Level 1					
- Namespaces in X	ML Recomme	ndation W3C [Jan.	14, 1999] Provide a simple		
method for qualifying element	and attribute na	mes used in XML doo	cuments. This is accomplished		
by associating a particular tag s	et by associatin	g a prefix with a URI	reference.		
- PICS Recommen	dation W3C	[May 27, 1998] Stand	lard format for making digitally		
signed, machine-readable asser	tions about a pa	articular information r	resource.		
- PICS Rules Reco	ommendation	W3C [Dec. 29, 1997] ·RDF Model and Syntax		
Recommendation W3C [Feb	22, 1999] A fo	oundation for processi	ng metadata: it provides		

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 9/31

KARPAGAM ACA	KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE						
Class: III BCOM (CA)	Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN						
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch				

interoperability between applications that exchange machine-understandable information on the Web.

- XHTML 1.0 Recommendation -- W3C [Jan. 26, 2000] A reformulation of HTML 4 as an XML 1.0 application and three DTDs corresponding to the ones defined by HTML 4.

- XML Path Language (XPath) Version 1.0 Recommendation -- W3C [Nov. 16, 1999] Provide a common syntax and semantics for querying and addressing the contents of XML documents that could be used by XSLT (XSL Transformation Language), XLink, and XPointer. XML v. 1.0 DTD Revised XML Recommendation DTD -- W3C [Sept. 10, 1998] A revised version of the XML Recommendation's DTD.

• XML XPointer Requirements W3C [Feb. 24, 1999] •XSL Transformations (XSLT) Specification Version 1.0 Recommendation -- W3C [Nov. 16, 1999] A language used to "transform" (or reconstruct the structure of) the data structures contained within XML documents.

• XInclude Inclusion Proposal (proposed for XML version 2.0) Fragments of content from external resources are included in documents content along with the content residing at the actual URL being accessed (sort of like a server-side include).

XML Syntax :

- XML documents must contain one root element that is the parent of all other elements.
- All XML Elements must have their corresponding Closing Tag.

Xml syntax:

```
<?xml
version = "version_number"
encoding = "encoding_declaration"
standalone = "standalone_status"
?>
```

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 10/31

Course Name: INTERNET AND WEB DESIGN

Course Code: 16CCU601A

Class: III BCOM (CA)

Semester: VI

Year: 2016-2019 Batch

Example:

<?xml version = "1.0"?>

<contact-info>

<name>Tanmay Patil</name>

<company>TutorialsPoint</company>

one>(011) 123-4567

</contact-info>

XML Syntax rules:

- All XML elements must have a closing tag.
- It is illegal to omit the closing tag when you are creating XML syntax.

Unit -V

- XML tags are case sensitive.
- All XML elements must be properly nested.
- All XML documents must have a root element.
- Attribute values must always be quoted.
- •

XML Declaration

• The XML declaration is case sensitive and must begin with "**<?xml**>" where "**xml**" is written in lower-case.

• If document contains XML declaration, then it strictly needs to be the first statement of the XML document.

- The XML declaration strictly needs be the first statement in the XML document.
- An HTTP protocol can override the value of encoding that you put in the XML declaration.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 11/31



XML Document Structure:

A well-formed XML document complies with the syntactic rules of XML markup. These rules are strict but basically quite simple.

- documents must be self-describing (they start with an XML declaration)
- a document must contain one or more elements
- documents must contain a single root element
- start and end tags must be used to identify elements (must have closing tags)
- empty elements must be marked as such (with a self-closing />)
- attribute values must be quoted (single or double quotes are fine)
- Element names and attributes names are case sensitive.
- element tags must be correctly nested (must not overlap)
- Element names (attribute names) must start with a letter

XML Document consists of many parts. XML Document has 2 main parts:

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 12/31

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE Class: III BCOM (CA) **Course Name: INTERNET AND WEB DESIGN** Course Code: 16CCU601A Unit -V

Semester: VI Year: 2016-2019 Batch

1. XML Prolog

2. XML Body.

XML document structure

This section introduces the terms used to describe the parts of an XML document, starting at the top level and working down to the smallest parts.

XML documents are made up of two parts, called the prolog and the body

XML prolog

The prolog contains optional information such as the XML version the document conforms to, information about the character encoding used to encode the contents of the document, and a reference to either a document type definition (DTD) or XML Schema document which describes the grammar and vocabulary of the document. The XML Schema document is the more modern way to describe XML grammar and vocabulary. XML Schemas and DTDs are usually stored in external documents and the prolog can reference both XML Schemas and DTDs.

This simple example illustrates the prolog:

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE maillist SYSTEM "maillist.dtd">

XML body

The body contains a single top-level element called the root element, which contains all other elements and other markup information.

This simple example illustrates the prolog and body, where addresses is the root element and encloses the entire body, or content, of the XML document:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

<!DOCTYPE maillist SYSTEM "maillist.dtd">

<addresses>

<address>...</address>

<address>...</address>

<address>...</address>

</addresses>

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 13/31

Class: III BCOM (CA)	Cour	se Name: INTERN	ET AND WEB DESIGN
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch
Example:			
<pre><?xml version="1.0"? <contact-info></pre>	<pre>> .1 .sPoint</pre>	ompany>]→	Document Prolog Document Elements
Describing Structures in XM	L		
XML was describing a tree stru	ucture of data. A	And tree structures ha	we one root, child elements,
branches, attributes, values. Fo	llowing are sim	ple XML structures.	
<root></root>			
<element></element>			
<subelement></subelement>			
XML Document Example			
Therefore a complete well form	ned XML docur	ment may look like:	
xml version="1.0"?			
<students></students>			
<student></student>			
<name>ram</name>			
<rollno>1</rollno>			
<program>Mca</program>	>		
<student></student>			
<pre><name>shyam</name></pre>			

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 14/31



Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 15/31

Class: III BCOM (CA)	ET AND WE	B DESIGN		
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 201	6-2019 Batch
Raw XML Content	Type of No	de Value		Number of Children
This is an order	Comment	This is a	an order	
<name></name>	Element			1
Scott Bradley	Text	Scott Br	radley	
<orderitems></orderitems>	Element			1
<item id="1"></item>	Element wi Attribute	th		1
Lord of the Rings	Text	Lord of	the Rings	

Document Type Definitions:

DTD stands for **Document Type Definition**. It defines the legal building blocks of an XML

document. It is used to define document structure with a list of legal elements and attributes.

DTD - Syntax

An XML DTD can be either specified inside the document, or it can be kept in a separate document and then the document can be linked to the DTD document to use it.

Syntax

Basic syntax of a DTD is as follows -

<!DOCTYPE element DTD identifier

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 16/31

| Class: III BCOM (CA) | Cours | se Name: INTERNE | T AND WEB DESIGN |
|---|--------------------|-------------------------|-----------------------------------|
| Course Code: 16CCU601A | Unit -V | Semester: VI | Year: 2016-2019 Batch |
| [| | | |
| declaration1 | | | |
| declaration2 | | | |
| | | | |
|]> | | | |
| | | | • |
| Purpose of DTD | | | |
| Its main purpose is to define th | e structure of ar | n XML document. It c | contains a list of legal elements |
| and defines the structure with t | he help of them | | |
| Checking Validation | | | |
| Before proceeding with XML I | OTD, you must | check the validation. | An XML document is called |
| "well-formed" if it contains the | correct syntax. | | |
| A well-formed and valid XML | document is on | e which has been val | idated against DTD. |
| Example: | | | |
| employee.xml | | | |
| xml version="1.0"? | | | |
| employee SYST</td <td>EM "employee.</td> <td>dtd"></td> <td></td> | EM "employee. | dtd"> | |
| <employee></employee> | | | |
| <firstname>vimal<td>></td><td></td><td></td></firstname> | > | | |
| <lastname>jaiswal<td>></td><td></td><td></td></lastname> | > | | |
| <email>vimal@javatpoint.com</email> | n | | |
| | | | |
| In the above example, the DOC | CTYPE declarat | ion refers to an extern | al DTD file. The content of the |
| file is shown in below paragrap | oh. | | |
| description of dtd | | | |
| | | | |
| <: DOCTYPE employee : It de | efines that the re | pot element of the doo | cument is employee. |
| ELEMENT employee: It de</td <td>fines that the en</td> <td>nployee element cont</td> <td>ains 3 elements "firstname,</td> | fines that the en | nployee element cont | ains 3 elements "firstname, |
| lestneme and ameil" | | | |

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 17/31

Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN

Course Code: 16CCU601AUnit -VSemester: VIYear: 2016-2019 Batch

<**!ELEMENT firstname:** It defines that the firstname element is **#PCDATA** typed. (parse-able data type).

<**!ELEMENT lastname:** It defines that the lastname element is **#PCDATA** typed. (parse-able data type).

<!ELEMENT email: It defines that the email element is #PCDATA typed. (parse-able data type).

XML DTD with entity declaration

A doctype declaration can also define special strings that can be used in the XML file. An entity has three parts:

- 1. An ampersand (&)
- 2. An entity name
- 3. A semicolon(;)



Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 18/31

Class: III BCOM (CA)	Cour	se Name: INTERNI	ET AND WEB DESIGN
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch
types			
DTD can be classified on its	declaration basis	in the XML documer	nt, such as –
• Internal DTD			
• External DTD			
When a DTD is declared with	nin the file it is ca	alled Internal DTD a	and if it is declared in a separate
file it is called External DTI) .		
Features			
Following are some importan	t points that a D	ΓD describes –	
• The elements that can	appear in an XM	IL document.	
• The order in which th	ey can appear.		
• Optional and mandate	ory elements.		
• Element attributes and	d whether they ar	e optional or mandate	ory.
• Whether attributes ca	n have default va	lues.	
Advantages of using DTD			
• Documentation – Ye	ou can define you	r own format for the	XML files. Looking at this
document a user/developer ca	an understand the	structure of the data.	
• Validation – It gives	a way to check the	he validity of XML fi	iles by checking whether the
elements appear in the right of	order, mandatory	elements and attribut	es are in place, the elements and
attributes have not been inser	ted in an incorrec	ct way, and so on.	
Disadvantages of using DTD			
• It does not support the	e namespaces. Na	amespace is a mechar	nism by which element and
attribute names can be assign	ed to groups. Ho	wever, in a DTD nam	nespaces have to be defined
within the DTD, which viola	tes the purpose of	f using namespaces.	
• It supports only the te	xt string data typ	e.	
• It is not object oriente	d. Hence, the cor	ncept of inheritance ca	annot be applied on the DTDs.
• Limited possibilities	o express the car	dinality for element.	

Class: III BCOM (CA)	Course Name: INTERNET AND WEB DESIGN		
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch
Namespaces			
• A Namespace is a set o	f unique name	s. Namespace is a me	chanisms by which element a
attribute name can be assigned	to a group. The	e Namespace is identi	fied by URI(Uniform Resour
Identifiers).			
Namespace Declaration			
• A Namespace is declare	ed using reserve	ed attributes. Such an	attribute name must either
be xmlns or begin with xmlns :	shown as below	<i>w</i> –	
Syntax			
• The Namespace starts w	with the keywor	rd xmlns .	
• The word name is the N	Jamespace pref	fix.	
• The URL is the Names	pace identifier.		
<element <="" td="" xmlns:name="URL"><td>"></td><td></td><td></td></element>	">		
Example			
xml version = "1.0" encoding</td <td>g = "UTF-8"?></td> <td></td> <td></td>	g = "UTF-8"?>		
<cont:contact xmlns:cont="w</td><td>ww.tutorialspo</td><td>int.com/profile"></cont:contact>			
<cont:name>Tanmay Patil<td>cont:name></td><td></td><td></td></cont:name>	cont:name>		
<cont:company>TutorialsPoi</cont:company>	nt <td>any></td> <td></td>	any>	
<cont:phone>(011) 123-4567</cont:phone>		,	
Namespace Declaration			
Write a namespace declaration	according to th	ne following descripti-	on method, describing the



XML Naming Rules:

- Element names are case-sensitive
- Element names must start with a letter or underscore

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 20/31

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE **Course Name: INTERNET AND WEB DESIGN Class: III BCOM (CA) Course Code: 16CCU601A** Unit -V Semester: VI Year: 2016-2019 Batch

- Element names cannot start with the letters xml (or XML, or Xml, etc)
- Element names can contain letters, digits, hyphens, underscores, and periods
- Element names cannot contain spaces

Naming Styles

There are no naming styles defined for XML elements. But here are some commonly used:

	Style	Exan	nple	Description
	Upper	case <fir< td=""><td>STNAME></td><td>All letters upper case</td></fir<>	STNAME>	All letters upper case
	Pascal	case <firs< td=""><td>tName></td><td>Uppercase first letter in each word</td></firs<>	tName>	Uppercase first letter in each word
nple nar	nespacin	g:		
mple nar	nespacin	g:		
	nespacin	g:	sectionTabl	c d
nple nar	nespacin • secID	g: name	sectionTabl	e name
nple nar	nespacin • secID s001	g: name John Smith	sectionTabl	e name Sales
hple nar	e secID S001 S002	g: John Smith Ichiro Tanaka	sectionTabl	e name Sales Development
ple nar	e secID S001 S002 S002	g: John Smith Ichiro Tanaka Jiro Suzuki	sectionTabl secID S001 S002 S003	e name Sales Development Administrative
byceTabl npID 100001 100002 100003 100003 100004	secID soo1 soo2 soo2 soo3	g: John Smith Ichiro Tanaka Jiro Suzuki Saburo Takahasi	sectionTabl secID S001 S002 S003 hi	e name Sales Development Administrative
nple nar	e sociD sooi sooi sooi sooi sooi sooi sooi so	g: John Smith Ichiro Tanaka Jiro Suzuki Saburo Takahasi e.empID,section section .secID = sectio	section Table social soci social social social social social social social social soci	e <u>name</u> Sales Development Administrative
ployeeTabl mpID 2000001 2000002 2000003 2000004 SELEC FROM & WHERE	secID S001 S002 S002 S003 F employee employee,s employee e.secID =	g: John Smith Ichiro Tanaka Jiro Suzuki Saburo Takahasi e.empID,section section .secID = sectio s.secID = sectio	sectionTable secID S001 S002 S003 hi	e name Sales Development Administrative
Ployee Table mpID 0000001 0000002 00000003 00000004 SELEC FROM & WHERE SELEC FROM & WHERE	e secID S001 S002 S002 S003 F employee,s employee,s employee e secID =	g: John Smith Ichiro Tanaka Jiro Suzuki Saburo Takahasi saburo Takahasi section section section section section s.secID = section s.secID	section Table Social So	e Sales Development Administrative
Apple nar	e secID S001 S002 S002 S003 F employee, employee e.secID = e.secID =	g: John Smith Ichiro Tanaka Jiro Suzuki Saburo Takahasi e.empID,section section .secID = section s.name,e.name e.section s s.secID s.name	section Table Soci Soci Soci Soci Soci Soci Soci Soci	e <u>name</u> Sales Development Administrative
ple nar	e sociD Sooi Sooi Sooi Sooi Sooi Sooi Sooi So	g: John Smith Ichiro Tanaka Jiro Suzuki Saburo Takahasi e.empID,section .secID = section s.name,e.name e.section s s.secID s.name s.secID s.name s.secID	section Table Social So	e Name Sales Development Administrative
select FROM of WHERE	es secID S001 S002 S002 S003 Femployees employes employees employees employees	g: John Smith John Smith Ichiro Tanaka Jiro Suzuki Saburo Takahasi e.empID,section s.secID = section s.secID = section s.name,e.name a,section s s.secID s.name s.secID s.name Sales Development	section Table Soo1 Soo2 Soo3 hi Anname, employer n.secID	e Name Sales Development Administrative se.name
Ployee Table	es secID S001 S002 S002 S003 Femployees employes	g: John Smith Ichiro Tanaka Jiro Suzuki Saburo Takahasi e.empID,section section s.secID = section s.secID = section s	section Table Soo1 Soo2 Soo3 hi Anname, employer n.secID	e Name Sales Development Administrative

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 21/31

Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN			
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch
XML Namespaces -Attribute			
When using prefixes in XML, a r	namespace fo	or the prefix must be	defined.
The namespace can be defined by	y an xmlns at	ttribute in the start tag	g of an element.
The namespace declaration has the	ne following	syntax.	
<root></root>			
<h:table xmlns:h="http://www.w</td><td>3.org/TR/htn</td><td>nl4/"></h:table>			
<h:tr></h:tr>	C		
<h:td>Apples</h:td>			
<h:td>Bananas</h:td>			
<f:table xmlns:f="https://www.w</td><td>/3schools.cor</td><td>n/furniture"></f:table>			
<f:name>African Coffee Table<</f:name>			
<f:width>80</f:width>			
<f:length>120</f:length>			

XMLschema is alanguagewhich is usedforexpressing constraint about XML documents.

Schema syntax:

The <schema> element is the root element of every XML Schema:

<?xml version="1.0"?>

<xs:schema>

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 22/31

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE			
Class: III BCOM (CA)	Cour	T AND WEB DESIGN	
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch
XML Schema Example			
xml version="1.0" encoding=</td <td>="UTF-8"?></td> <td>,</td> <td></td>	="UTF-8"?>	,	
<xs:schema xmlns:xs="http://w</td><td>ww.w3.org/20</td><td>01/XMLSchema"></xs:schema>			
<xs:element name="Author"></xs:element>			
<xs:complextype></xs:complextype>			
<xs:sequence></xs:sequence>			
<xs:element name="FirstN</td><td>lame" type="x</td><td>s:string"></xs:element>			

</xs:sequence>

```
</xs:complexType>
```

</xs:element>

</xs:schema>

Description of XML Schema:

<xs:element name="employee"> : It defines the element name employee.

<xs:complexType> :Itdefinesthattheelement'employee' is complextype.

<**xs:sequence**> : Itdefinesthatthecomplextype is asequenceofelements.

<**xs:element name=''firstname'' type=''xs:string''/>**:It defines that the element 'firstname' is of string/text type.

<xs:element name=''lastname'' type=''xs:string''/> :Itdefinesthattheelement'lastname' is of string/text type.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 23/31

Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN

Course Code: 16CCU601A Unit -V Semester: VI Year: 2016-2019 Batch

<**xs:element name=''email'' type=''xs:string''/>** : It defines that the element 'email' is of string/text type.

Checking Validation

An XML document is called well-formed if it contains the correct syntax. Awell-formed and valid XML document is one which has been validated against Schema.

There are two types of data types in XML schema.

- 1. Simple Type
- 2. Complex type

Simple Type

The simple Type allows you to have text-based elements. It contains less attributes, child elements, and cannot be left empty.

Complex Type

The complex Type allows you to hold multiple attributes and elements. It can contain additional sub elements and can be left empty.

XML Schemas Support Data Types

One of the greatest strength of XML Schemas is the support for data types.

- It is easier to describe allowable document content
- It is easier to validate the correctness of data
- It is easier to define data facets (restrictions on data)
- It is easier to define data patterns (data formats)
- It is easier to convert data between different data types

Schema components:

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 24/31

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE				
Class: III BCOM (CA)	Cour	se Name: INTERNE	T AND WEB DESIGN	
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch	

The main components of a schema are:

• Element declarations, which define properties of elements. These include the element name and target namespace. An important property is the type of the element, which constrains what attributes and children the element can have. In XSD the type of the element may be conditional on the values of its attributes. An element may belong to a substitution group; if element E is in the substitution group of element H, then wherever the schema permits H to appear, E may appear in its place. Elements may have integrity constraints: uniqueness constraints determining that particular values must be unique within the subtree rooted at an element, and referential constraints determining that values must match the identifier of some other element. Element declarations may be global or local, allowing the same name to be used for unrelated elements in different parts of an instance document.

• Attribute declarations, which define properties of attributes. Again the properties include the attribute name and target namespace. The attribute type constrains the values that the attribute may take. An attribute declaration may also include a default value or a fixed value (which is then the only value the attribute may take.)

• **Simple and complex types**. These are described in the following section.

• **Model group** and **attribute group** definitions. These are essentially macros: named groups of elements and attributes that can be reused in many different type definitions.

• An **attribute use** represents the relationship of a complex type and an attribute declaration, and indicates whether the attribute is mandatory or optional when it is used in that type.

• An **element particle** similarly represents the relationship of a complex type and an element declaration, and indicates the minimum and maximum number of times the element may appear in the content. As well as element particles, content models can include **model group** particles, which act like non-terminals in a grammar: they define the choice and repetition units within the sequence of permitted elements. In addition, **wildcard** particles are allowed, which permit a set of different elements (perhaps any element provided it is in a certain namespace).

Other more specialized components include annotations, assertions, notations, and the **schema component** which contains information about the schema as a whole.

Displaying Raw XML Documents:

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 25/31

Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN

Course Code: 16CCU601AUnit -VSemester: VIYear: 2016-2019 Batch

Raw XML files can be viewed in all major browsers.

XML is now mature enough that recent versions of the more popular web browsers support it

natively. At the time of writing, the most recent versions of these browsers include:

- Microsoft Internet Explorer 6 (http://www.microsoft.com/windows/ie/)
- Mozilla 1.7 and Mozilla Firefox 0.9 (http://www.mozilla.org)
- Netscape 7.1 (http://channels.netscape.com/ns/browsers/download.jsp)
- Opera 7.51 (http://www.opera.com)
- Apple's Safari 1.2 (http://www.apple.com/safari/)

Raw XML Files

<?xml version="1.0" encoding="UTF-8"?>

- <note>

```
<to>Tove</to>
```

<from>Jani</from>

<heading>Reminder</heading>

```
<body>Don't forget me this weekend!</body>
```

</note>

. Displaying XML file using CSS:

CSS can be used to display the contents of the XML document in a clear and precise manner. It gives the design and style to whole XML document.

Basic steps in defining a CSS style sheet for XML :

For defining the style rules for the XML document, the following things should be done :-

1. Define the style rules for the text elements such as font-size, color, fontweight, etc.

2. Define each element either as a block, inline or list element, using the display property of CSS.

3. Identify the titles and bold them.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 26/31

Course Name: INTERNET AND WEB DESIGN

Course Code: 16CCU601A Unit -V Semester: VI Year: 2016-2019 Batch

Linking XML with CSS :

In order to display the XML file using CSS, link XML file with CSS. Below is the syntax for linking the XML file with CSS:

<?xml-stylesheet type="text/css" href="name_of_css_file.css"?>

Example:

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/css" href="Rule.css"?>

<books>

Class: III BCOM (CA)

<heading>Welcome To GeeksforGeeks </heading>

<book>

<title>Title -: Web Programming</title>

<author>Author -: Chrisbates</author>

<publisher>Publisher -: Wiley</publisher>

<edition>Edition -: 3</edition>

<price> Price -: 300</price>

</book>

<book>

<title>Title -: Internet world-wide-web</title>

<author>Author -: Ditel</author>

<publisher>Publisher -: Pearson</publisher>

<edition>Edition -: 3</edition>

<price>Price -: 400</price>

</book>

<book>

<title>Title -: Computer Networks</title>

<author>Author -: Foruouzan</author>

<publisher>Publisher -: Mc Graw Hill</publisher>

<edition>Edition -: 5</edition>

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 27/31

Class: III BCOM (CA)	Cours	se Name: INTERNI	ET AND WEB DESIGN
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch
<price>Price -</price>	: 700		
<book></book>			
<title>Title -: 1</title>	DBMS Concept	s	
<author>Author</author>	or -: Navath <th>ithor></th> <td></td>	ithor>	
<publisher>Pu</publisher>	blisher -: Oxfor	d	
<edition>Editi</edition>	on -: 5 <th>></th> <td></td>	>	
<price>Price -</price>	: 600		
<book></book>			
<title>Title -:]</title>	Linux Programr	ning	
author Auth	or · Subbitab D		
<autility autility<="" td=""><td>oi Subilitab D</td><th>as</th><td></td></autility>	oi Subilitab D	as	
<pre><pre>caution > Autor <pre>publisher>Pu</pre></pre></pre>	blisher -: Oxfor	as d	
<pre><pre>caution>Aution <pre>Pu <pre>cedition>Edition</pre></pre></pre></pre>	on -: 8 <th>as rd ></th> <td></td>	as rd >	
<pre>caution>Aution <publisher>Pu <edition>Editi <price>Price -</price></edition></publisher></pre>	bl -: Subintab D blisher -: Oxfor on -: 8: 300	as rd >	
<pre>caution>Autor <publisher>Pu <edition>Editi <price>Price - </price></edition></publisher></pre>	on -: Subintab D on -: 8: 300	as rd >	

- CSS is used in XML or HTML to decorate the pages.
- CSS is used for interactive interface, so it is understandable by user.
- CSS enable multiple pages to share formatting, and reduce complexity and repetition in the structural content. So page loader is faster.

Disadvantages of displaying XML using CSS:

- Using CSS, no transformation can be applied to the XML documents.
- CSS uses different dimensions with different browsers. So the programmer has to run the code in different browser and test its compatibility to post it live.
- CSS have different level of versions, so it is confusing for the browser and user.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 28/31

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATOREClass: III BCOM (CA)Course Name: INTERNET AND WEB DESIGNCourse Code: 16CCU601AUnit -VSemester: VIYear: 2016-2019 Batch

Xml processor:

When a software program reads an XML document and takes actions accordingly, this is called processing the XML. Any program that can read and process XML documents is known as an XML processor. An XML processor reads the XML file and turns it into in-memory structures that the rest of the program can access.

The most fundamental XML processor reads an XML document and converts it into an internal representation for other programs or subroutines to use. This is called a parser, and it is an important component of every XML processing program.

Types

XML processors are classified as validating or non-validating types, depending on whether or not they check XML documents for validity. A processor that discovers a validity error must be able to report it, but may continue with normal processing.

A few validating parsers are – xml4c (IBM, in C++), xml4j (IBM, in Java), MSXML (Microsoft, in Java), TclXML (TCL), xmlproc (Python), XML::Parser (Perl), Java Project X (Sun, in Java). A few non-validating parsers are – OpenXML (Java), Lark (Java), xp (Java), AElfred (Java), expat (C), XParse (JavaScript), xmllib (Python).

Components of XML processor

Parser:- Every XML processor has a parser. An XML parser converts an XML document into an XML DOM object - which can then be manipulated with a JavaScript. The parser's job is to translate XML markup and data into a stream of bite-sized nuggets, called tokens, to be used for processing. A token may be an element start tag, a string of character content, the beginning delimiter of a processing instruction, or some other piece of markup that indicates a change between regions of the document.

Stage 1: In this stage, the application parses and validates the source document; recognizes and searches for relevant information based on its location or its tagging in the source document; extracts the relevant information when it is located; and, optionally, maps and binds the retrieved information to business objects.

Stage 2: Business logic handling. This is the stage in which the actual processing of the input information takes place. It might result in the generation of output information.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 29/31

KARPAGAM ACA	KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE			
Class: III BCOM (CA)	Cour	se Name: INTERNI	ET AND WEB DESIGN	
Course Code: 16CCU601A	Unit -V	Semester: VI	Year: 2016-2019 Batch	
Stage 3: XML output processin	g. In this stage	e, the application cons	structs a model of the document to	
be generated with the Documen	t Object Mode	el (DOM). It then eith	er applies XSLT style sheets or	
directly serializes to XML.				
X X				

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 30/31

Class: III BCOM (CA) Course Code: 16CCU601A

Unit -V Semester: VI

Course Name: INTERNET AND WEB DESIGN

Year: 2016-2019 Batch

POSSIBLE QUESTIONS – UNIT- V

PART A (1 Mark)

(Online Examinations)

PART B (2 Marks)

1. What is XML?

2. Differentiate the XML & HTML

3. What are the Recommendations in XML? Mention its Name.

4. Write the XML Syntax and structure with an example

5. Define XML element

6. What is Document Type Definitions?

7. What is the purpose of Document Type Definitions?

8. What are all the XML naming rules?

9. What is XML Namespaces?

10. Define XML processor.

Part-C (6 Marks)

1. List the XML standard and explain with an example

2 Elucidate the XML syntax rules with an example

3. Describe the XML document structure with an example

4. Explain the namespace with an example.

5. Discuss the Document Type Definitions with an example

6 Write the advantages and disadvantages of displaying XML using CSS

7. Explain the Design goals for XML.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 31/31



Enable | Enlighten | Enrich (Deemed to be University) (Under Section 3 of UGC Act 1956) KARPAGAM ACADEMY OF HIGHER EDUCATION (Deemed to be University Established Under Section 3 of UGC Act 1956) Coimbatore – 641 021. (For the candidates admitted from 2016 onwards)

DEPARTMENT OF COMMERCE (CA)

	Syllabus	Semester 6
16CCU601A INTERNET AND WE	B DESIGNNING	L T P C 4 4

SCOPE : Internet and Web Design enables the students with the application of concept of Internet and designing of websites. It equips the students with the skill in the application of scripting languages like HTML, Java script and ASP, Net.

OBJECTIVES:

- To know the working of Internet, uses of search engines and procedures to develop a web page.
- To make the student expertise in creating web page. And to ensure that the student know the concepts of Internet and design a web page.

UNIT- I

Introduction to Internet: Internet basics – World wide Web – browser portability – Features of Internet Explorer. Introduction to HTML ; Mark up languages – common Tags – Headers – text styling – Linking – Images – formatting text – special characters – Horizontal rules and Line text – unrecorded lists – nested unordered lists- basic HTML tables – Intermediate HTML tables.

UNIT- II

Formatting – Basic HTML forms – creating and using Image maps – meta tags – frameset – nested frameset. Introduction to DHTML: cascading Style Sheet – Inline style – creating style sheets with style elements – conflicting styles – Linking external style sheets – Position elements – Backgrounds – element dimension – text flow – Box models.

UNIT-III

Introduction to Java Script – Operators – Arithmetic Operators – Precedence of Operators – Relational Operators – Control Structures – Assignment Operators – Increment and Decrement Operators – For loops – Switch – Do While – Break – Continues – Arrays – Functions .

UNIT- IV

Active Server Page (ASP): Introduction – How ASP work – five objects of ASP – Client Side Scripting Vs Server Side Scripting – Server Side ActiveX component – Session Tracking and Cookies – file system objects.

UNIT-V

Introduction to XML : Introduction – The Syntax of XML – XML Document Structure – Document Type Definitions – Namespaces – XML Schemas – Displaying Raw XML Documents – Displaying XML Documents with CSS – XML Processors.

Suggested Readings :

Text Books

- 1. H.M.Deitel and T.R.Nieto, (2000). *Internet and World Wide Web How to Program*. New Delhi, Pearson education.
- 2. Ivan Bayross, (2000). Web enables Commercial application development using HTML, DHTML, Java Script, Perl, CGI. New Delhi, BPB Publications.

Reference Books

- Robert .W.Sebesta, (2007). *Programming the World Wide Web* [3rd Edition]. New Delhi, Pearson Education.
- 2. Shelly Powers (2008) Dynamic Web Publishing. New Delhi, Techmedia.
- 3. Scot Johnson.(2010). *Using Active Server Pages*. New Delhi. Que Publications.
- J. Jaworski (2000). *Mastering JavaScript* [2nd Edition]. New Delhi, BPB Publications.
- Thomas Powell. (2002). *HTML Computer Reference*[4th Edition] New Delhi, Tata Mcgraw Hill

KARPAGAM ACADEMY OF HIGHER EDUCATION

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

<u>UNIT-I</u>

SYLLABUS

Introduction to Internet: Internet basics – World Wide Web – browser portability – Features of Internet Explorer. Introduction to HTML: Mark up languages – Common Tags – Headers – Text Styling – Linking – Images – Formatting Text – Special Characters – Horizontal Rules and Line Text – Unordered List – Nested Unordered lists – Basic HTML Tables – Intermediate HTML Tables.

1.1Introduction to Internet

• Internet can be defined as an electronic medium, which is connecting the world with the help of the computers and it brought the world in a roof called as the Global Village.

• Internet consists of large amount of data that can be accessed by the various users and because of this it is also referred to as the "Information Superhighway" of the world.

• With the help of the internet can easily be in touch with anyone in the whole world by sending electronic mail, by chatting, etc., travel booking can be made very easily, one can order books or buy anything online in simple terms it can be said that internet provides a very strong connection or network between computers globally, bringing people and their working close to each other.

• The network of networks – 'The Internet' is not a personal property of any one i.e. it is not owned by anyone, which allows individuals and the various organizations to get connected to any other server or any other user.

• Internet has become such an important and defining tool in today's competitive and market oriented environment that it helps a lot in getting business and making money.

Internet History

The history of the Internet begins with the development of electronic computers in the 1950s. Initial concepts of wide area networkingoriginated in several computer science laboratories in the United States, United Kingdom, and France. The U.S. Department of Defense awarded contracts as early as the 1960s, including for the development of the ARPANET project, directed by Robert Taylorand managed by Lawrence Roberts. The first message was sent over the ARPANET in 1969 from computer science Professor Leonard Kleinrock's laboratory

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

KARPAGAM ACADEMY OF HIGHER EDUCATIONCLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

at University of California, Los Angeles (UCLA) to the second network node at Stanford Research Institute (SRI).

Packet switching networks such as the NPL network, ARPANET, Tymnet, Merit Network, CYCLADES, and Telenet, were developed in the late 1960s and early 1970s using a variety of communications protocols. Donald Davies first demonstrated packet switching in 1967 at the National Physics Laboratory (NPL) in the UK, which became a testbed for UK research for almost two decades. The ARPANET project led to the development of protocols for internetworking, in which multiple separate networks could be joined into a network of networks.

The Internet protocol suite (TCP/IP) was developed by Robert E. Kahn and Vint Cerf in the 1970s and became the standard networking protocol on the ARPANET, incorporating concepts from the French CYCLADES project directed by Louis Pouzin. In the early 1980s the NSF funded the establishment for national supercomputing centers at several universities, and provided interconnectivity in 1986 with the NSFNET project, which also created network access to the supercomputer sites in the United States from research and education organizations. Commercial Internet service providers (ISPs) began to emerge in the very late 1980s. The ARPANET was decommissioned in 1990. Limited private connections to parts of the Internet by officially commercial entities emerged in several American cities by late 1989 and 1990, and the NSFNET was decommissioned in 1995, removing the last restrictions on the use of the Internet to carry commercial traffic.

In the 1980s, research at CERN in Switzerland by British computer scientist Tim Berners-Lee resulted in the World Wide Web, linking hypertext documents into an information system, accessible from any node on the network. Since the mid-1990s, the Internet has had a revolutionary impact on culture, commerce, and technology, including the rise of near-instant communication by electronic mail, instant messaging, voice over Internet Protocol (VoIP) telephone calls, two-way interactive video calls, and the World Wide Web with its discussion forums, blogs, social networking, and online shopping sites. The research and education community continues to develop and use advanced networks such as JANET in the United Kingdom and Internet2 in the United States. Increasing amounts of data are transmitted at higher and higher speeds over fiber optic networks operating at 1 Gbit/s, 10 Gbit/s, or more.

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

KARPAGAM ACADEMY OF HIGHER EDUCATION

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

The Internet's takeover of the global communication landscape was almost instant in historical terms: it only communicated 1% of the information flowing through two-way telecommunications networks in the year 1993, already 51% by 2000, and more than 97% of the telecommunicated information by 2007. Today the Internet continues to grow, driven by ever greater amounts of online information, commerce, entertainment, and social networking. However, the future of the global internet may be shaped by regional differences in the world.



Early research and development:

- 1965: NPL network planning starts
- 1966: Merit Networkfounded
- 1966: ARPANET planning starts
- 1967: NPL network packet switching pilot experiment
- 1969: ARPANET carries its first packets
- 1970: Network Information Center (NIC)

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

Page 3/54
CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

- 1971: Tymnet packet-switched network
- 1972: Merit Network's packet-switched network operational
- 1972: Internet Assigned Numbers Authority (IANA) established
- 1973: CYCLADES network demonstrated
- 1974: Telenet packet-switched network
- 1976: X.25 protocol approved
- 1978: Minitel introduced
- 1979: Internet Activities Board (IAB)
- 1980: USENET news using UUCP
- 1980: Ethernet standard introduced
- 1981: BITNET established

Merging the networks and creating the Internet:

- 1981: Computer Science Network (CSNET)
- 1982: TCP/IP protocol suite formalized
- 1982: Simple Mail Transfer Protocol (SMTP)
- 1983: Domain Name System(DNS)
- 1983: MILNET split off from ARPANET
- 1985: First .COM domain name registered
- 1986: NSFNET with 56 kbit/s links
- 1986: Internet Engineering Task Force (IETF)
- 1987: UUNET founded
- 1988: NSFNET upgraded to 1.5 Mbit/s (T1)
- 1988: OSI Reference Modelreleased
- 1988: Morris worm
- 1989: Border Gateway Protocol (BGP)
- 1989: PSINet founded, allows commercial traffic
- 1989: Federal Internet Exchanges (FIXes)
- 1990: GOSIP (without TCP/IP)
- 1990: ARPANETdecommissioned

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

- 1990: Advanced Network and Services (ANS)
- 1990: UUNET/Alternetallows commercial traffic
- 1990: Archie search engine
- 1991: Wide area information server (WAIS)
- 1991: Gopher
- 1991: Commercial Internet eXchange (CIX)
- 1991: ANS CO+RE allows commercial traffic
- 1991: World Wide Web(WWW)
- 1992: NSFNET upgraded to 45 Mbit/s (T3)
- 1992: Internet Society(ISOC) established
- 1993: Classless Inter-Domain Routing (CIDR)
- 1993: InterNIC established
- 1993: AOL added USENETaccess
- 1993: Mosaic web browserreleased
- 1994: Full text web search engines
- 1994: North American Network Operators' Group(NANOG) established

Commercialization, privatization, broader access leads to the modern Internet:

- 1995: New Internet architecture with commercial ISPs connected at NAPs
- 1995: NSFNETdecommissioned
- 1995: GOSIP updated to allow TCP/IP
- 1995: very high-speed Backbone Network Service(vBNS)
- 1995: IPv6 proposed
- 1996: AOL changes pricing model from hourly to monthly
- 1998: Internet Corporation for Assigned Names and Numbers (ICANN)
- 1999: IEEE 802.11bwireless networking
- 1999: Internet2/Abilene Network
- 1999: vBNS+ allows broader access
- 2000: Dot-com bubblebursts
- 2001: New top-level domain names activated

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

- 2001: Code Red I, Code Red II, and Nimda worms
- 2003: UN World Summit on the Information Society (WSIS) phase I
- 2003: National LambdaRailfounded
- 2004: UN Working Group on Internet Governance (WGIG)
- 2005: UN WSIS phase II
- 2006: First meeting of the Internet Governance Forum
- 2010: First internationalized country code top-level domains registered
- 2012: ICANN begins accepting applications for new generic top-level domain names
- 2013: Montevideo Statement on the Future of Internet Cooperation
- 2014: NetMundial international Internet governance proposal
- 2016: ICANN contract with U.S. Dept. of Commerce ends, IANA oversight passes to the global Internet community on October 1st

1.1.1: Internet Basics

Internet has been the most useful technology of the modern times which helps us not only in our daily lives, but also our personal and professional lives developments. The internet helps us achieve this in several different ways.

For the students and educational purposes the internet is widely used to gather information so as to do the research or add to the knowledge of various subjects. Even the business professionals and the professionals like doctors, access the internet to filter the necessary information for their use. The internet is therefore the largest encyclopedia for everyone, in all age categories. The internet has served to be more useful in maintaining contacts with friends and relatives who live abroad permanently.

The **Internet** is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/ IP) to serve billions of users worldwide. It is a *network of networks* that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic, wireless and optical networking technologies. The Internet carries a vast range of information resources and services, such as the inter- linked hypertext documents of the World Wide Web

KARPAGAM ACADEMY OF HIGHER EDUCATIONCLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

(WWW) and the infrastructure to support electronic mail.

Advantages of Internet:

• E-mail: Email is now an essential communication tools in business. With e-mail you can send and receive instant electronic messages, which works like writing letters. Your messages are delivered instantly to people anywhere in the world, unlike traditional mail that takes a lot of time. Email is free, fast and very cheap when compared to telephone, fax and postal services.

• 24 hours a day - 7 days a week: Internet is available, 24x7 days for usage.

• **Information:** Information is probably the biggest advantage internet is offering. There is a huge amount of information available on the internet for just about every subject, ranging from government law and services, trade fairs and conferences, market information, new ideas and technical support. You can almost find any type of data on almost any kind of subject that you are looking for by using search engines like google, yahoo, msn, etc.

• Online Chat: You can access many 'chat rooms' on the web that can be used to meet new people, make new friends, as well as to stay in touch with old friends. You can chat in MSN and yahoo websites.

• Services: Many services are provided on the internet like net banking, job searching, purchasing tickets, hotel reservations, guidance services on array of topics engulfing every aspect of life.

• **Communities:** Communities of all types have sprung up on the internet. Its a great way to meet up with people of similar interest and discuss common issues.

• E-commerce: Along with getting information on the Internet, you can also shop online. There are many online stores and sites that can be used to look for products as well as buy them using your credit card. You do not need to leave your house and can do all your shopping from the convenience of your home. It has got a real amazing and wide range of products from household needs, electronics to entertainment.

• Entertainment: Internet provides facility to access wide range of Audio/Video songs,

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

plays films. Many of which can be downloaded. One such popular website is YouTube.

• **Software Downloads:** You can freely download innumerable, softwares like utilities, games, music, videos, movies, etc from the Internet.

Limitations of Internet

• Theft of Personal information: Electronic messages sent over the Internet can be easily snooped and tracked, revealing who is talking to whom and what they are talking about. If you use the Internet, your personal information such as your name, address, credit card, bank details and other information can be accessed by unauthorized persons. If you use a credit card or internet banking for online shopping, then your details can also be 'stolen'.

Negative effects on family communication: It is generally observed that due to more time spent on Internet, there is a decrease in communication and feeling of togetherness among the family members.

• **Internet addiction:** There is some controversy over whether it is possible to actually be addicted to the Internet or not. Some researchers, claim that it is simply people trying to escape their problems in an online world.

• Children using the Internet has become a big concern. Most parents do not realize the dangers involved when their children log onto the Internet. When children talk to others online, they do not realize they could actually be talking to a harmful person. Moreover, pornography is also a very serious issue concerning the Internet, especially when it comes to young children. There are thousands of pornographic sites on the Internet that can be easily found and can be a detriment to letting children use the Internet.

• Virus threat: Today, not only are humans getting viruses, but computers are also. Computers are mainly getting these viruses from the Internet. Virus is is a program which disrupts the normal functioning of your computer systems. Computers attached to internet are more prone to virus attacks and they can end up into crashing your whole hard disk.

• **Spamming:** It is often viewed as the act of sending unsolicited email. This multiple or vast emailing is often compared to mass junk mailings. It needlessly obstruct the entire

KARPAGAM ACADEMY OF HIGHER EDUCATIONCLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

system. Most spam is commercial advertising, often for dubious products, get-rich-quick schemes, or quasi-legal services. Spam costs the sender very little to send — most of the costs are paid for by the recipient or the carriers rather than by the sender

1.1.2: World Wide Web

WWW stands for World Wide Web. A technical definition of the World Wide Web is : all the resources and users on the Internet that are using the Hypertext Transfer Protocol HTTPHTTP.

A broader definition comes from the organization that Web inventor Tim Berners-Lee helped found, the World Wide Web Consortium W3CW3C.

The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge.

In simple terms, The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.

Internet and Web is not the same thing: Web uses internet to pass over the information.

WWW Operation

WWW works on client- server approach. Following steps explains how the web works:

- 1. User enters the URL (say, http://www.tutorialspoint.com) of the web page in the address bar of web browser.
- 2. Then browser requests the Domain Name Server for the IP address corresponding to www.tutorialspoint.com.
- 3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
- 4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
- 5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

Protocols used in Internet

There are various protocols used in the internet for data transfer within web browser and web server. The protocols used in internet are as follows

- 1. TCP/IP Protocol
- 2. UDP Protocol
- 3. FTP Protocol
- 4. Telnet Protol
- 5. HTTP Protocol

1. Transmission Control Protocol TCP

TCP is a connection oriented protocol and offers end-to-end packet delivery. It acts as back bone for connection. It exhibits the following key features:

- Transmission Control Protocol TCPTCP corresponds to the Transport Layer of OSI Model.
- TCP is a reliable and connection oriented protocol.
- TCP offers:
 - Stream Data Transfer.
 - Reliability.
 - Efficient Flow Control
 - \circ Full-duplex operation.
 - Multiplexing.
- TCP offers connection oriented end-to-end packet delivery.
- TCP ensures reliability by sequencing bytes with a forwarding acknowledgement number that indicates to the destination the next byte the source expect to receive.
- It retransmits the bytes not acknowledged with in specified time period.

TCP Services

TCP offers following services to the processes at the application layer:

- Stream Delivery Service
- Sending and Receiving Buffers
- Bytes and Segments
- Full Duplex Service

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

- Connection Oriented Service
- Reliable Service

2. Internet Protocol IPIP

Internet Protocol is **connectionless** and **unreliable** protocol. It ensures no guarantee of successfully transmission of data.

In order to make it reliable, it must be paired with reliable protocol such as TCP at the transport layer.

Internet protocol transmits the data in form of a datagram as shown in the following diagram:

4	8	1	6	32
VER	HLEN	D.S. type of service	Total ler	ngth of 16 bits
	Identific	ration of 16 bits	Flags 3 bits	Fragmentation Offset (13 bits)
Time to live Protocol Header checksum (16 bi		ecksum (16 bits)		
	bit.	Source IP addres	5	
		Destination IP add	ess	
		Option + Paddir	g	

3.User Datagram Protocol UDP

Like IP, UDP is connectionless and unreliable protocol. It doesn't require making a connection with the host to exchange data. Since UDP is unreliable protocol, there is no mechanism for ensuring that data sent is received.

UDP transmits the data in form of a datagram. The UDP datagram consists of five parts as shown in the following diagram:

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

Source Port	Destination Port
Length	UDP checksum
Da	ata

- UDP is used by the application that typically transmit small amount of data at one time.
- UDP provides protocol port used i.e. UDP message contains both source and destination port number, that makes it possible for UDP software at the destination to deliver the message to correct application program.

4.File Transfer Protocol FTP

FTP is used to copy files from one host to another. FTP offers the mechanism for the same in following manner:

- FTP creates two processes such as Control Process and Data Transfer Process at both ends i.e. at client as well as at server.
- FTP establishes two different connections: one is for data transfer and other is for control information.
- Control connection is made between control processes while Data Connection is made between<="" b="">
- FTP uses **port 21** for the control connection and **Port 20** for the data connection.



KARPAGAM ACADEMY OF HIGHER EDUCATIONCLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

5.Telnet

Telnet is a protocol used to log in to remote computer on the internet. There are a number of Telnet clients having user friendly user interface. The following diagram shows a person is logged in to computer A, and from there, he remote logged into computer B.



6.Hyper Text Transfer Protocol HTTP

HTTP is a communication protocol. It defines mechanism for communication between browser and the web server. It is also called request and response protocol because the communication between browser and server takes place in request and response pairs.

HTTP Request

HTTP request comprises of lines which contains:

- Request line
- Header Fields
- Message body

Key Points

- The first line i.e. the **Request line** specifies the request method i.e. **Get** or **Post.**
- The second line specifies the header which indicates the domain name of the server from where index.htm is retrieved.

HTTP Response

Like HTTP request, HTTP response also has certain structure. HTTP response contains:

- Status line
- Headers
- Message body

1.1.3: Browser Portability

A web browser (commonly referred to as a browser) is a software application for accessing information on the World Wide Web. Each individual web page, image, and video is identified by a distinct URL, enabling browsers to retrieve and display them on the user's device.

A web browser is not the same thing as a search engine, though the two are often confused. For a user, a search engine is just a website, such as google.com, that stores searchable data about other websites. But to connect to and display websites on their device, a user needs to have a web browser installed.

The most popular web browsers are Chrome, Firefox, Safari, Internet Explorer, and Edge.

Ensuring a consistent look and feel on client-side browsers is one of the great challenges of developing web-based applications. Currently, a standard does not exist to which soft-ware developers must adhere when creating web browsers. Although browsers share a common set of features, each browser might render pages differently. Browsers are available in many versions and on many different platforms (Microsoft Windows, Apple Macintosh, Linux, UNIX, etc.). Vendors add features to each new version that sometimes result in cross-platform incompatibility issues. Clearly it is difficult to develop web pages that ren-der correctly on all versions of each browser. In this book we develop web applications that execute on both the Internet Explorer 7 and Firefox 2 browsers.

1.1.4: Features of Internet Explorer

Even though not many websites cover *Internet Explorer* extensively and write interesting articles and tutorials about it, this browser still manages to hold the title of the most used browser. One of the reasons for it is that it provides some interesting functionality that other browsers don't have or have not fully adopted. Here is a list of 5 great features *Internet Explorer* has and other browser do not.

• Visual Search Providers

One of the features in *Internet Explorer* is the ability to install and use visual search provides. They allow you not only to see suggestions for your searches but they are also accompanied by a visual preview that includes a picture and some text description.

KARPAGAM ACADEMY OF HIGHER EDUCATION **CLASS: III BCom CA** COURSE NAME: INTERNET AND WEB DESIGNING COURSE CODE: 16CCU601A UNIT: I **BATCH-2016-19** 😂 bill gates Wikipedia (en) Suggestions **Bill Gates** Villiam Henry "Bill" Gates III (born October 28, 1955) is an American business magnate, investor, philanthropist, and author. Bill & Melinda Gates Foundation The Bill & Melinda Gates Foundation (B&MGF or the Gates Foundation) is the largest transparently operated private foundation in Bill Gates' house Bill Gates' house is a large mansion in the side of a hill overlooking Lake Washington in Medina, Washington. Bill Gates (frontiersman) "Swiftwater" Bill Gates (7 Turn off suggestions (stop sending keystrokes to Wikipedia (en))

Very useful and looks great. This feature while doing the research for Google and Other 7 Search Providers for Internet Explorer 9. Unfortunately though, this feature is not used by many search providers and I would like to see more of it both in *Internet Explorer* and other browsers.

• Stats about the Performance Impact of Add-ons

W 😇 🔩 🌞 🛎

As highlighted in the article Internet Explorer 9 - How to Improve the Load & Navigation Speed, the browser monitors the performance impact of all add-ons and encourages users to disable those that slow things down.

Choose Add-ons			×
Speed up browsing by disal These add-ons increase the time it to or navigate to websites by an average disable add-ons in the Manage Add	bling the add-ons that ye takes to start the browser, open ge of 0,55 seconds. You can also I-ons dialog.	ou don't want a new tab e enable or	
Gathera.MyPlacesSearchBHO Not Available	0,27 seconds	Disable	^
IEVkbdBHO Class Kaspersky Lab	0,09 seconds	Disable	
LastPass Toolbar LastPass	0,06 seconds	Disable	
Office Document Cache Handler Microsoft Corporation	0,06 seconds	Disable	I
Java(tm) Plug-In 2 SSV Helper Sun Microsystems, Inc.	0,05 seconds	Disable	I
Snagit TechSmith Corporation	II 0,01 seconds	Disable	1
Windows Live ID Sign-in Helper			٣
Tell me when the delay caused by a	dd-ons exceeds: 0,20 seconds	•	
		Disable All Done	

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

More browsers would have a similar behavior and share in a transparent way how much each add-on contributes to slowing the browsing experience.

• Improved Security

If you follow the tech news on a regular basis, you definitely learned about *Internet Explorer* being declared the most secure browser or, at least an important contender to the title. The last debate, found here: Browser Study Sheds Light On Firefox's Insecurity (And Google Approves This Message), considers Google Chrome and *Internet Explorer* as being the top two contenders for the title.



Leaving this debate aside, during the tests we made for the Security for Everyone series, we noticed that *Internet Explorer* definitely provides better identification and reporting of malicious websites and downloads, when compared to other browsers and even some security suites. Another great feature is that *Internet Explorer* allows users to easily report malicious websites, directly from the main interface of the browser, thus helping increase security for all users.

We wish to see more improvements in this area from all major browsers, including *Internet Explorer*.

KARPAGAM ACADEMY OF HIGHER EDUCATIONCLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

• Power Saving Features

You might not be aware of this, but *Internet Explorer* allows you to squeeze a bit more juice from your battery, when running unplugged. You get less hardware acceleration while browsing, without making things terribly slow and a bit more battery time.

Power Options	?	×
Advanced settings		
Select the power plan that you want to customize then choose settings that reflect how you want y computer to manage power.	e, and our	
Change settings that are currently unavailable		
Balanced [Active]		
USB settings Power buttons and lid DCI Everges		*
 Processor power management Display 		
Multimedia settings		
 Internet Explorer JavaScript Timer Frequency Setting: Maximum Performance Maximum Power Savings Maximum Performance Restore plan defailed 	ults	Ŧ
OK Cancel	Ap	ply

Here is how it works: Save Battery Power While Browsing the Web in Internet Explorer 9.

• Pinning Websites to the Windows Taskbar or the Start Menu

This feature is not new and it has been adopted by some browsers, such as Google Chrome. However, I consider it useful and I think all browsers should offer such capabilities to their users.



CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

1.1.5: Introduction to HTML

- HTML is the standard markup language used for creating Web pages.
- HTML stands for Hyper Text Markup Language.
- HTML describes the structure of Web pages using markup.
- HTML elements are the building blocks of HTML pages.
- HTML elements are represented by tags.
- HTML programs are developed in word pad or note pad and the code is executed in any browsers.
- HTML programs are portable and are platform independent.
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on.
- Browsers do not display the HTML tags, but use them to render the content of the page.

Sample HTML Program

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>My First Heading</h1>

My first paragraph.

</body>

</html>

Program Explanation

<!____>: Comment line

<html> _____</html>: the element is the root element of an HTML page

<head>_____</head>: the element contains meta information about the document

<title>_____</title>: the element specifies a title for the document

<body> element contains the visible page content

The <h1> element defines a large heading

The element defines a paragraph

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

1.1.6: Markup languages

A markup language is a computer language that uses tags to define elements within a document. It is human-readable, meaning markup files contain standard words, rather than typical programming syntax. While several markup languages exist, the two most popular are HTML and XML.

HTML is a markup language used for creating webpages. The contents of each webpage are defined by HTML tags. Basic page tags, such as <head>, <body>, and <div> define sections of the page, while tags such as , <form>, <image>, and <a> define elements within the page. Most elements require a beginning and end tag, with the content placed between the tags. The beginning and end tags acts as an container for the content placed between the tags. For example,

TechTerms.com

Common Tags

Basic HTML tags

Tag	Description
	Defines the document type
<html></html>	Defines an HTML document
<head></head>	Defines information about the document
<title></title>	Defines a title for the document
<body></body>	Defines the document's body
<h1> to <h6></h6></h1>	Defines HTML headings
	Defines a paragraph
	Inserts a single line break
<hr/>	Defines a thematic change in the content
	Defines a comment

Formatting tags

Tag	Description
<acronym></acronym>	Not supported in HTML5. Use <abbr> instead. Defines an acronym</abbr>
<abbr></abbr>	Defines an abbreviation or an acronym
<address></address>	Defines contact information for the author/owner of a document/article
	Defines bold text
<bdi></bdi>	Isolates a part of text that might be formatted in a different direction from other text
	outside it
<bdo></bdo>	Overrides the current text direction
 big>	Not supported in HTML5. Use CSS instead. Defines big text
<blockquote></blockquote>	Defines a section that is quoted from another source
<center></center>	Not supported in HTML5. Use CSS instead. Defines centered text
<cite></cite>	Defines the title of a work
<code></code>	Defines a piece of computer code
	Defines text that has been deleted from a document

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

 Defines emphasized text Not supported in HTML5. Use CSS instead.Defines font, color, and size for text <i>> Defines a part of text in an alternate voice or mood <ins> Defines a text that has been inserted into a document <kbd> Defines keyboard input <mark> Defines marked/highlighted text <meter> Defines a scalar measurement within a known range (a gauge)</meter></mark></kbd></ins></i>	
 Not supported in HTML5. Use CSS instead.Defines font, color, and size for text <i>> Defines a part of text in an alternate voice or mood <ins> Defines a text that has been inserted into a document <kbd> Defines keyboard input <mark> Defines marked/highlighted text <meter> Defines a scalar measurement within a known range (a gauge)</meter></mark></kbd></ins></i>	
<i>> Defines a part of text in an alternate voice or mood <ins> Defines a text that has been inserted into a document <kbd> Defines keyboard input <mark> Defines marked/highlighted text <meter> Defines a scalar measurement within a known range (a gauge)</meter></mark></kbd></ins></i>	
<ins> Defines a text that has been inserted into a document <kbd> Defines keyboard input <mark> Defines marked/highlighted text <meter> Defines a scalar measurement within a known range (a gauge)</meter></mark></kbd></ins>	
<kbd> Defines keyboard input <mark> Defines marked/highlighted text <meter> Defines a scalar measurement within a known range (a gauge)</meter></mark></kbd>	
<mark> Defines marked/highlighted text <meter> Defines a scalar measurement within a known range (a gauge)</meter></mark>	
<meter> Defines a scalar measurement within a known range (a gauge)</meter>	
<pre> Defines preformatted text</pre>	
<progress> Represents the progress of a task</progress>	
<q> Defines a short quotation</q>	
<rp> Defines what to show in browsers that do not support ruby annotations</rp>	
<rt> Defines an explanation/pronunciation of characters (for East Asian typography)</rt>	
<ruby> Defines a ruby annotation (for East Asian typography)</ruby>	
<s> Defines text that is no longer correct</s>	
<samp> Defines sample output from a computer program</samp>	
<small> Defines smaller text</small>	
<strike> Not supported in HTML5. Use or <s> instead. Defines strikethrough text</s></strike>	
 Defines important text	
_{Defines subscripted text}	
^{Defines superscripted text}	
<template> Defines a template</template>	
<time> Defines a date/time</time>	
<tt> Not supported in HTML5. Use CSS instead.</tt>	
Defines teletype text	
<u> Defines text that should be stylistically different from normal text</u>	
<var> Defines a variable</var>	
<wbr/> Defines a possible line-break	

Forms and Input tags

Tag	Description
<form></form>	Defines an HTML form for user input
<input/>	Defines an input control
<textarea></textarea>	Defines a multiline input control (text area)
<button></button>	Defines a clickable button
<select></select>	Defines a drop-down list
<optgroup></optgroup>	Defines a group of related options in a drop-down list
<option></option>	Defines an option in a drop-down list
<label></label>	Defines a label for an <input/> element
<fieldset></fieldset>	Groups related elements in a form
<legend></legend>	Defines a caption for a <fieldset> element</fieldset>
<datalist></datalist>	Specifies a list of pre-defined options for input controls
<output></output>	Defines the result of a calculation

Frames tags

Tag	Description
<frame/>	Not supported in HTML5. Defines a window (a frame) in a frameset
<frameset></frameset>	Not supported in HTML5. Defines a set of frames

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

<noframes></noframes>	Not supported in HTML5. Defines an alternate content for users that do not support frames
<iframe></iframe>	Defines an inline frame

Images tags

Tag	Description
	Defines an image
<map></map>	Defines a client-side image-map
<area/>	Defines an area inside an image-map
<canvas></canvas>	Used to draw graphics, on the fly, via scripting (usually JavaScript)
<figcaption></figcaption>	Defines a caption for a <figure> element</figure>
<figure></figure>	Specifies self-contained content
<picture></picture>	Defines a container for multiple image resources
<svg></svg>	Defines a container for SVG graphics

Audio / Video tags

Tag	Description
<audio></audio>	Defines sound content
<source/>	Defines multiple media resources for media elements (<video>, <audio> and picture>)</audio></video>
<track/>	Defines text tracks for media elements (<video> and <audio>)</audio></video>
<video></video>	Defines a video or movie

Links tags

Tag	Description
<a>	Defines a hyperlink
<link/>	Defines the relationship between a document and an external resource (most used to
	link to style sheets)
<nav></nav>	Defines navigation links

Lists tags

Tag	Description
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dir></dir>	Not supported in HTML5. Use instead.
	Defines a directory list
<dl></dl>	Defines a description list
<dt></dt>	Defines a term/name in a description list
<dd></dd>	Defines a description of a term/name in a description list

Tables tags

Tag	Description
	Defines a table
<caption></caption>	Defines a table caption
	Defines a header cell in a table
(ui)	

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

Page 21/54

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

	Defines a row in a table
	Defines a cell in a table
<thead></thead>	Groups the header content in a table
	Groups the body content in a table
<tfoot></tfoot>	Groups the footer content in a table
<col/>	Specifies column properties for each column within a <colgroup> element</colgroup>
<colgroup></colgroup>	Specifies a group of one or more columns in a table for formatting

Styles and Semantics

Tag	Description
<style></style>	

Meta Info tags

Tag	Description
<head></head>	Defines information about the document
<meta/>	Defines metadata about an HTML document
<base/>	Specifies the base URL/target for all relative URLs in a document
<basefont/>	Not supported in HTML5. Use CSS instead.
	Specifies a default color, size, and font for all text in a document

Programming tags

Tag	Description	
<script></script>		

<Pre> Tag

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

The PRE element is used for *pre-formatted text*. What this means is that the text between the opening and closing tags is displayed exactly as shown in the HTML file. Normally, any line breaks etc. in the HTML file are ignored when the page is displayed: instead the browser breaks where it finds a
, starts a new paragraph at a <P>, etc. One main use for <PRE> is when you have a large chunk of text in a plain text file which you want to display without having to convert all the paragraphs and tables to HTML formatting. Example:

Consider the following code:

<P>There was an old man of Peru

Who dreamt he was eating his shoe

He woke up in the night

With a terrible fright

And found it was perfectly true</P>

 □ ptaghtmi
 ×

 ←
 →
 C
 ① File | file///E/Karpagam/ptag.html

 There was an old man of Peru Who dreamt he was eating his shoe He woke up in the night With a terrible fright And found it was perfectly true

📀 🛗 🍳 🕑 🧭 🖭 🥏 🖾 🥥

The spacing and line breaks in the HTML code are ignored. To get the right result we would normally use formatting tags:

<P>There was an old man of Peru
 Who dreamt he was eating his shoe
 He woke up in the night
 With a terrible fright
 And found it was perfectly true</P>

🖾 💋 🚱

\$

ptsp.html x - > C O File file://E/Karpagam/ptag.html ere was an old man of Peru	UNIT: I	BATCH-2016-19
C plag.html x + + + + + + + + + + + + + + + + + +		
→ → C O File file://E/Karpagam/ptag.html ere was an old man of Peru		- 0 - X -
ere was an old man of Peru		🖈 🖾 💋 🛙 🥥 🗄
ho dreamt he was eating his shoe woke up in the night th a terrible fright d found it was perfectly true		
	2	- Mar ■* (4) (2) - 840 AM (2)
owever, we can use the <pre> ele</pre>	ement:	
PRE>There was an old man of Per	u	
Tho dreamt he was eating his shoe		
He woke up in the night		
With a terrible fright		
nd found it was perfectly true <td>RE></td> <td></td>	RE>	
) ptag.html × +		
		🖈 🖾 💋 🗄

🚱 🚞 o 😉 🥝 🖭 🥠 🔕 🛋 🛷

1.1.7: Text Styling

The color property defines the text color for an HTML element:

Example

<h1 style="color:blue;">This is a heading</h1> This is a paragraph.

This is a heading

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

- ∰ ➡ ⊕ Ê <u>9:41 AM</u> 12/15/2018

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

This is a paragraph.

HTML Fonts

The font-family property defines the font to be used for an HTML element:

Example

<h1 style="font-family:verdana;">This is a heading</h1>This is a paragraph.

This is a heading

This is a paragraph.

HTML Text Size

The font-size property defines the text size for an HTML element:

Example

<h1 style="font-size:300%;">This is a heading</h1>This is a paragraph.

This is a heading

This is a paragraph.

HTML Text Alignment

The text-align property defines the horizontal text alignment for an HTML element:

Example

<h1 style="text-align:center;">Centered Heading</h1> Centered paragraph.

Centered Heading

Centered paragraph.

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

1.1.8: Linking

The HTML <link> tag is used for defining a link to an external document. It is placed in the <head> section of the document.

Example

<html>

<head>

<title>HTML link Tag</title>

k rel = "stylesheet" href = "stylenew.css">

</head>

<body>

<div id = "contentinfo">

Welcome to our website. We provide tutorials on various subjects.

</div>

</body>

</html>

Here is the css file stylenew.css

#contentinfo p {

line-height: 20px;

margin: 30px;

padding-bottom: 20px;

text-align: justify;

width: 140px;

color: red;

}

This will produce the following result -

Welcome to our website. We provide tutorials on various subjects.

Attributes

The HTML <link> tag also supports the following additional attributes -

Attribute	Value	Description
Charset	charset	Defines the character encoding of the linked document.

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

HreflanglanguageMediascreentty	Language code of the destination URL Specifies the device the document will be displayed on
Media screen tty	Specifies the device the document will be displayed on
tty	
tv	
projection	
handheld	
print	
braille	
aural	
all	
Rel alternate	Describes the relationship between the current
appendix	document and the destination URL.
bookmark	
chapter	
contents	
copyright	
glossary	
help	
home	
index	
next	
prev	
section	
start	
stylesheet	
subsection	
Rev alternate	Describes a reverse between the destination URI and
appendix	the current document.
bookmark	
chapter	

CLASS: III BCom CA COURSE CODE: 16CCU601A		COURSE NAME: INTERNET AND WEB DESIGNING	
		UNIT: I BATCH-2016-19	
	contents		
	copyright		
	glossary		
	help		
	home		
	index		
	next		
	prev		
	section		
	start		
	stylesheet		
	subsection		
sizes ጛ	HeightxWidth	Specifies the size of the linked resource.	
Target	blank	Specifies the target frame to load the page into.	
	_self		
	_top		
	_parent		
Type	mimetype	The MIMEtype of content at the link destination	

1.1.9: Images

Images can improve the design and the appearance of a web page.

Example

HTML Image

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19



HTML Images Syntax

In HTML, images are defined with the tag. The tag is empty, it contains attributes only, and does not have a closing tag. The src attribute specifies the URL (web address) of the image:

The alt Attribute

The alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

The value of the alt attribute should describe the image:

Example

Alternative text

The alt attribute should reflect the image content, so users who cannot see the image gets an understanding of what the image contains:

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19



If a browser cannot find an image, it will display the value of the alt attribute:

Image Size - Width and Height

You can use the style attribute to specify the width and height of an image.

Example

Alternatively, you can use the width and height attributes:

Image Size

Use the style attribute to specify the width and height of an image:



1.1.10: Formatting Text

If you use a word processor, you must be familiar with the ability to make text bold, italicized, or underlined; these are just three of the ten options available to indicate how text can appear in HTML and XHTML.

Bold Text

Anything that appears within ... element, is displayed in bold as shown below -

Example

<html>

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

<head>

```
<title>Bold Text Example</title>
</head>
<body>
The following word uses a <b>bold</b> typeface.
</body>
</html>
```

Italic Text

Anything that appears within <i>...</i> element is displayed in italicized as shown below -

Example

```
<html>
<head>
<title>Italic Text Example</title>
</head>
<body>
The following word uses an <i>italicized</i> typeface.
</body>
</html>
```

Underlined Text

Anything that appears within $\langle u \rangle$... $\langle u \rangle$ element, is displayed with underline as shown below –

Example

```
<html>
<head>
<title>Underlined Text Example</title>
</head>
<body>
The following word uses an <u>underlined</u> typeface.
</body>
</html>
```

Strike Text

Anything that appears within **<strike>...</strike>** element is displayed with strikethrough, which is a thin line through the text as shown below –

Example

<html>

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

<head>

<title>Strike Text Example</title> </head> <body> The following word uses a <strike>strikethrough</strike> typeface. </body> </html>

Superscript Text

The content of a **^{...}** element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

Example

<html> <head> <title>Superscript Text Example</title> </head> <body> The following word uses a ^{superscript} typeface. </body> </html>

Subscript Text

The content of a **_{...}** element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

Example

```
<html>
<head>
<title>Subscript Text Example</title>
</head>
<body>
The following word uses a <sub>subscript</sub> typeface.
</body>
</html>
```

Inserted Text

Anything that appears within **<ins>...</ins>** element is displayed as inserted text.

Example

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

<html>

```
<head>
<title>Inserted Text Example</title>
</head>
<body>
I want to drink <del>cola</del> <ins>wine</ins>
</body>
</html>
```

Deleted Text

Anything that appears within **...** element, is displayed as deleted text.

Example

```
<html>
<head>
<title>Deleted Text Example</title>
</head>
<body>
I want to drink <del>cola</del> <ins>wine</ins>
</body>
</html>
```

Larger Text

The content of the **<big>...</big>** element is displayed one font size larger than the rest of the text surrounding it as shown below –

Example

```
<html>
<head>
<title>Larger Text Example</title>
</head>
<body>
The following word uses a <big>big</big> typeface.
</body>
</html>
```

Smaller Text

The content of the **<small>...</small>** element is displayed one font size smaller than the rest of the text surrounding it as shown below –

Example

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

```
<html>
<head>
<title>Smaller Text Example</title>
</head>
<body>
The following word uses a <small>small</small> typeface.
</body>
```

</html>

1.1.11: Special Characters

Some characters are not available on a standard keyboard. In HTML they can be indicated by special character entities. The special character entities start with an ampersand &, and end with a semicolon;. Between these is a string of letters which usually indicates the character to be shown.

There is also an alternative system of referring to special characters by number but these are less easy to remember. On the other hand it can be more reliable if you are using the less common special characters.

The following is a list of some of the more common special character entities. Not all of them will display in all browsers. If you see, in the right-hand box, not a special character but just a repeat of the code in the left-hand box, your browser has failed to recognize the code.

Character entity	Special character
	Non-breaking space
&	& (ampersand)
<	<
>	>
©	© (copyright symbol)
®	® (Reg. trademark symbol)
≠	\neq (not-equals sign; a range of mathematical symbols are available)
±	± (plus-minus)
°	° (degree)
"	" (Quotation mark)
–	– (n-dash)
—	— (m-dash)
¶	¶ (paragraph symbol as used in word-processors)
£	£ (pound sterling symbol)
€	€ (Euro currency symbol)
¼	¹ / ₄ (fraction 1/4)
½	¹ / ₂ (fraction 1/2)
¾	³ / ₄ (fraction 3/4)
é	é
É	É
è	è
à	à

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

	-
ç	ç
ê	ê
â	â
û;	û
ä	ä
ö	ö
ü	ü
š	š
ñ	ñ
ß	β (the German character, not to be confused with beta (see below)
β	β (lower-case Beta; the whole Greek alphabet is accessible like this)

1.1.12: Horizontal Rules and Line Text

The HTML <hr> tag is used for creating a horizontal line. This is also called Horizontal Rule in HTML.

Definition and Usage

The <hr> tag defines a thematic break in an HTML page (e.g. a shift of topic).

The <hr> element is used to separate content (or define a change) in an HTML page.

Example

<h1>HTML</h1>

HTML is a language for describing web pages.....

<hr>

<h1>CSS</h1>

CSS defines how to display HTML elements.....

<u>Attributes</u>

Attribute	Value	Description
align	left center right	Not supported in HTML5. Specifies the alignment of a <hr/> element
<u>noshade</u>	noshade	Not supported in HTML5. Specifies that a <hr/> element should render in one solid color (noshaded), instead of a shaded color
size	pixels	Not supported in HTML5.

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

Page 36/54

CLASS: III BCom COURSE CODE: 1	CA 6CCU601A	COURSE NAME: II UNIT: I	NTERNET AND WEB DESIGNING BATCH-2016-19
		Specifies the heig	ht of a <hr/> element
width	pixels %	Not supported in Specifies the widt	HTML5. th of a <hr/> element

<html>

<head>

<style>

hr {

display: block;

margin-top: 0.5em;

margin-bottom: 0.5em;

margin-left: auto;

margin-right: auto;

border-style: inset;

border-width: 1px;

}

</style>

</head>

<body>

A hr element is displayed like this:

<hr>

Change the default CSS settings to see the effect.

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

Page 37/54

	COURSE NAME: INTERNET AND WEB DESIGNING				
COURSE CODE: 16CCU601A	UNIT: I	BAT	CH-2016-19		
file///D:/sample.html × +			- o ×		
(il file:///D:/sample.html		C Search	☆ 自 🔸 ⋒ 🛡 Ξ		
A hr element is displayed like this:					

1.1.13: Lists

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain –

- An unordered list. This will list items using plain bullets.
- An ordered list. This will use different schemes of numbers to list your items.
- <dl> A definition list. This arranges your items in the same way as they are arranged in a dictionary.

HTML Unordered Lists

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML tag. Each item in the list is marked with a bullet.

Example

<html>

KARPAGAM ACADEMY OF HIGHER EDUCATION CLASS: III BCom CA COURSE NAME: INTERNET AND WEB DESIGNING COURSE CODE: 16CCU601A UNIT: I BATCH-2016-19

<head>

<title>HTML Unordered List</title>

</head>

<body>

Beetroot

Ginger

Potato

Radish

</body>

</html>

This will produce the following result -

- Beetroot
- Ginger
- Potato
- Radish

The type Attribute

You can use **type** attribute for tag to specify the type of bullet you like. By default, it is

a disc. Following are the possible options -

Example

Following is an example where we used

<html>

<head>

<title>HTML Unordered List</title>

</head>
| CLASS: III BCom CA | COURSE NAME: INTERNET AND WEB DESIGNING |
|--|---|
| COURSE CODE: 10CC0001A | UNII:1 DATCH-2010-19 |
| <body></body> | |
| ul type = "square"> | |
| Beetroot | |
| Ginger | |
| Potato | |
| Radish | |
| | |
| | |
| | |
| This will produce the following res | sult – |
| Beetroot | |
| • Ginger | |
| Potato | |
| Radish | |
| Example | |
| Following is an example where we | used <ul type="disc"> - |
| <html></html> | |
| <head></head> | |
| <title>HTML Unordered List<</title> | /title> |
| | |
| <body></body> | |
| <ul type="disc"> | |
| Beetroot | |
| Ginger | |
| Potato | |
| Radish | |
| | |
| | |
| | |
| This will produce the following res | sult — |
| D | |

- Beetroot
- Ginger

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

Page 40/54

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

- Potato
- Radish

Example

Following is an example where we used $\langle ul type = "circle" \rangle -$

<html>

<head>

<title>HTML Unordered List</title>

</head>

<body>

Beetroot

Ginger

Potato

Radish

</body>

</html>

This will produce the following result -

- Beetroot
- Ginger
- o Potato
- o Radish

HTML Ordered Lists

If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with .

Example

<html>

<head>

<title>HTML Ordered List</title>

<head>

<body>

KARPAGAM ACADEMY OF HIGHER EDUCATIONCLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

<0l>

Beetroot

Ginger

Potato

Radish

</body>

</html>

This will produce the following result -

- 1. Beetroot
- 2. Ginger
- 3. Potato
- 4. Radish

The type Attribute

You can use **type** attribute for tag to specify the type of numbering you like. By default,

it is a number. Following are the possible options -

- Default-Case Numerals.

- Upper-Case Numerals.

 $\langle ol type = "i" \rangle$ - Lower-Case Numerals.

- Upper-Case Letters.

- Lower-Case Letters.

Example

Following is an example where we used $\langle ol type = "1" \rangle$

<html>

<head>

```
<title>HTML Ordered List</title>
```

</head>

<body>

```
Beetroot
```

Ginger

Potato

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

Radish

</body>

</html>

This will produce the following result -

- 1. Beetroot
- 2. Ginger
- 3. Potato
- 4. Radish

Example

Following is an example where we used

<html>

<head>

```
<title>HTML Ordered List</title>
```

</head>

<body>

```
Beetroot
```

Ginger

```
Potato
```

Radish

</body>

</html>

This will produce the following result -

- I. Beetroot
- II. Ginger
- III. Potato
- IV. Radish

Example

Following is an example where we used

<html>

KARPAGAM ACADEMY OF HIGHER EDUCATION COURSE NAME: INTERNET AND WEB DESIGNING **CLASS: III BCom CA** COURSE CODE: 16CCU601A UNIT: I BATCH-2016-19 <head> <title>HTML Ordered List</title> </head> <body> $\langle ol type = "i" \rangle$ Beetroot Ginger Potato Radish </body> </html> This will produce the following result i. Beetroot ii. Ginger iii. Potato iv. Radish Example Following is an example where we used $\langle ol type = "A" \rangle$ <html> <head> <title>HTML Ordered List</title> </head> <body> <ol type = "A">Beetroot Ginger Potato Radish </body>

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

</html>

KARPAGAM ACADEMY OF HIGHER EDUCATION CLASS: III BCom CA COURSE NAME: INTERNET AND WEB DESIGNING COURSE CODE: 16CCU601A UNIT: I BATCH-2016-19 This will produce the following result – A. Beetroot B. Ginger C. Potato D. Radish Example Following is an example where we used $\langle ol type = "a" \rangle$ <html> <head> <title>HTML Ordered List</title> </head> <body> $\langle ol type = "a" \rangle$

Ginger

Beetroot

Potato

Radish

</body>

</html>

This will produce the following result -

```
a. Beetroot
```

b. Ginger

```
c. Potato
```

d. Radish

The start Attribute

You can use **start** attribute for tag to specify the starting point of numbering you need.

Following are the possible options -

 $\langle ol type = "1" start = "4" \rangle$ - Numerals starts with 4.

- Numerals starts with IV.

 $\langle ol type = "i" start = "4" \rangle$ - Numerals starts with iv.

 $\langle ol type = "a" start = "4" \rangle$ - Letters starts with d.

CLASS: III BCom CA	COURSE NAME: INTERNET	AND WEB DESIGNING
COURSE CODE: 16CCU601A	UNIT: I	BATCH-2016-19

 $\langle ol type = "A" start = "4" \rangle$ - Letters starts with D.

Example

Following is an example where we used $\langle ol type = "i" start = "4" \rangle$

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

Beetroot

Ginger

Potato

```
Radish
```

```
</body>
```

</html>

This will produce the following result -

```
iv. Beetroot
```

- v. Ginger
- vi. Potato
- vii. Radish

HTML Definition Lists

HTML and XHTML supports a list style which is called **definition lists** where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

- <dl> Defines the start of the list
- < dt > -A term
- <dd>- Term definition
- </dl> Defines the end of the list

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

Example

<html>

<head>

<title>HTML Definition List</title>

</head>

<body>

< dl >

<dt>HTML</dt>

<dd>This stands for Hyper Text Markup Language</dd>

<dt>HTTP</dt>

<dd>This stands for Hyper Text Transfer Protocol</dd>

</dl>

```
</body>
```

</html>

This will produce the following result -

HTML

This stands for Hyper Text Markup Language

HTTP

This stands for Hyper Text Transfer Protocol

1.1.14: Basic HTML Tables

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the <table> tag in which the <tr> tag is used to create table rows and <td> tag is used to create data cells. The elements under <td> are regular and left aligned by default

```
Example
```

<html>

```
<head>
<title>HTML Tables</title>
</head>
```

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

```
<body>
Row 1, Column 1
Row 1, Column 2
Row 2, Column 1
Row 2, Column 1
```

</body> </html>

Here, the **border** is an attribute of $\langle table \rangle$ tag and it is used to put a border across all the cells. If you do not need a border, then you can use border = "0".

Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

Table Heading

Table heading can be defined using $\langle th \rangle$ tag. This tag will be put to replace $\langle td \rangle$ tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use $\langle th \rangle$ element in any row. Headings, which are defined in $\langle th \rangle$ tag are centered and bold by default.

Example

<html>

```
<head>
<title>HTML Table Header</title>
</head>
<body>
Name
Salary
```

KARPAGAM ACADEMY OF HIGHER EDUCATIONCLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

5000 Shabbir Hussein 7000

</body>

</html>

Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

Cellpadding and Cellspacing Attributes

There are two attributes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cells. The cellspacing attribute defines space between table cells, while cellpadding represents the distance between cell borders and the content within a cell.

Example

<html>

```
<head>
<title>HTML Table Cellpadding</title>
</head>
<body>
Name
  Salary
 Ramesh Raman
  5000
 Shabbir Hussein
  7000
```

KARPAGAM ACADEMY OF HIGHER EDUCATIONCLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IBATCH-2016-19

</body>

</html>

Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

Colspan and Rowspan Attributes

You will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way you will use **rowspan** if you want to merge two or more rows.

Example

```
<html>
```

```
<head>
<title>HTML Table Colspan/Rowspan</title>
</head>
<body>
Column 1
Column 2
```

```
Column 3
 Row 1 Cell 1
  Row 1 Cell 2
  Row 1 Cell 3
 Row 2 Cell 2
  Row 2 Cell 3
 Row 3 Cell 1
 </body>
```

</html>

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		·

Tables Backgrounds

You can set table background using one of the following two ways -

- **bgcolor** attribute You can set background color for whole table or just for one cell.
- **background** attribute You can set background image for whole table or just for one cell.

Example

<html>

```
<head>
<title>HTML Table Background</title>
</head>
<body>
Column 1
  Column 2
  Column 3
 Row 1 Cell 1
  Row 1 Cell 2
  Row 1 Cell 3
 Row 2 Cell 2
  Row 2 Cell 3
 Row 3 Cell 1
 </body>
```

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

</html>

Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

Table Height and Width

You can set a table width and height using **width** and **height** attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

Example

<html>

```
<head>
<title>HTML Table Width/Height</title>
</head>
<body>

Row 1, Column 1
Row 1, Column 1

Row 1, Column 2

Row 1, Column 1

Row 2, Column 2

Row 2, Column 2

Row 2, Column 2

Row 2, Column 2
```

</html> <u>Table Caption</u>

The **caption** tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML.

Example

<html>

<head> <title>HTML Table Caption</title>

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: I BATCH-2016-19

</head>

```
<body>
<caption>This is the caption</caption>
row 1, column 1row 1, columnn 2
row 2, column 1row 2, columnn 2
```

</body>

```
</html>

<u>Table Header, Body, and Footer</u>
```

Tables can be divided into three portions -a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are -

- **<thead>** to create a separate table header.
- to indicate the main body of the table.
- **<tfoot>** to create a separate table footer.

A table may contain several elements to indicate *different pages* or groups of data. But it is notable that <thead> and <tfoot> tags should appear before

Example

<html>

```
<head>
<title>HTML Table</title>
</head>
<body>
<thead>
This is the head of the table
```

KARPAGAM ACADEMY OF HIGHER EDUCATION COURSE NAME: INTERNET AND WEB DESIGNING **CLASS: III BCom CA** COURSE CODE: 16CCU601A UNIT: I BATCH-2016-19 <tfoot> This is the foot of the table </tfoot> Cell 1 Cell 2 Cell 3 Cell 4 </body> </html> This is the head of the table

This is the foot of the table

(Deemed University Established Under Section 3 of UGC Act 1956) Coimbatore - 641021. (For the candidates admitted from 2017 onwards) **Department of Commerce (CA)**

SUBJECT: INTERNET AND WEB DESIGNING SEMESTER : VI SUBJECT CODE: 16CCU601A CLASS : III B.COM CA

UNIT:I

S. N						
0	QUESTIONS	OPTION 1	OPTION 2	OPTION 3	OPTION 4	ANSWER
1	HTML stands for	Hypertext Markup Language	Hyperlink Markup Language	Hypertext Markedup Language	Hypertext Markup Language	Hypertext Markup Language
	are used to markup					
2	HTML elements.	HTML tags	HR tags	Head tags	SGML tags	HTML tags
	Aelement is used to define					
3	style information.	Title	Set	Style	Character	Style
	How many levels of heading elements					
4	are in HTML?	6	3	7	8	6
	tag used to break up					
5	sections of a document.	Line break	Horizontal line	Preserve	Heading	Horizontal line
	tag set a color for the back-					
6	ground of the page.	Bg color	Br color	Bg	Color	Bg color
	How many categories are in core					
7	attributes?	3	4	5	6	4

8	tag is used to create an					
9	Unordered list is used for the purpose of	Symbols and bullets	Alphabet and numeric	Numeric and symbols	Heading and symbols	Symbols and bullets
	<pre>content of content of conten</pre>			59110015	591110015	
10	definition list.	Style	Links	Paragraph	Heading	Heading
11	HTML supports a list style which is called as	Definition list	Nested list	Unordered list	Ordered list	Definition list
12	Each item in the list is marked with	Bullets	Numeric	Hash sign	Pound	Bullets
13	<p> tag is used for</p>	Font color	Alignment	Formatting	Links	Formatting
14	LIRL is abbreviated as	Uniform Resource	Uniform Resource locator	Unified Resource	Uniform Research	Uniform Resource
17	tag used to create a current	Location	Resource locator	Locator	Locator	
15	position in the HTML document.		<hr/>			<hr/>
16	HTML is known as language.	Platform dependent	Platform independent	Place dependent	Place Independent	Platform independent
17	The Internet was originally project agency.	NSF	NSA	APRA	ARPA	ARPA
18	HTML is used to create	Machine language program	High level program	Web page	Web server	Web page
19	Internet explorer falls under	Operating System	Compiler	Browser	IP address	Browser
20	WWW stands for	World Wide Web	World Wide Word	World Wide Wood	World Wide Width	World Wide Web
21	ALL HTML tags are enclosed in what ?	# and #	? and !	< and >	{ and }	< and >
22	To create HTML page, you need	web browser and text editor	web browser	text editor	Web server	web browser and text editor

23	$\langle a \rangle$ and $\langle a \rangle$ are the tags used for	Adding image	Aligning text	Audio-voiced text	Adding links to your page	Adding links to your page																	
24	To add a plain color background to your web page, use which of the following ?	<body bgcolor="<br">"36,24,35"></body>	<body color="<br">"Green"></body>	<body bgcolor= "Green"></body 	<body brcolor="<br">"Green"></body>	<body bgcolor= "Green"></body 																	
25	The BODY tag is usually used after	HTML tag	EM tag	TITLE tag	HEAD tag	HTML tag																	
26	Choose the correct HTML tag to make a text italic	<i></i>	<italic></italic>	<it></it>	<il></il>	<i></i>																	
27	What does the tag add to your webpage ?	Line break	pragraph break	long break	break	pragraph break																	
28	Adding a border to your image helps the visitor to recognize it	frame	link	nicture	tag	nicture																	
20	The first tag inside <table> tag is</table>																						
30	Which tag tells the browser where the page starts and stops ?	html	body	head	title	html																	
31	In HTML, tags that include both on and off tag are called	comment tag	document tag	container tag	command tag	container tag																	
32	Which tag will you add to specify a font for your whole page ?	<defaultfont></defaultfont>	<targetfont></targetfont>	<basefont/>		<defaultfont></defaultfont>																	
33	The tag used for creating hypertext and hypermedia links is	<hr/>	 	<pre></pre>	<a>	<a>																	
34	The main container for <tr>, <td> and <th> is</th></td><td><data></data></td><td><group></group></td><td><table></table></td><td><th></th></td><td><table></table></td></tr> <tr><td>35</td><td>What is the correct HTML for creating a hyperlink ?</td><td><a>https://w.w. w.gkseries.com </td><td> Gkseries.com</td><td> Gkseries.com< /a></a </td><td> Gkseries.com< /a></a </td><td> Gkseries.com< /a></a </td></tr> <tr><td>36</td><td>All normal webpages consists of</td><td>Top and bottom</td><td>Body and</td><td>Head and body</td><td>Body and html</td><td>Head and body</td></tr>	 and <th> is</th>	is	<data></data>	<group></group>	<table></table>	<th></th>		<table></table>	35	What is the correct HTML for creating a hyperlink ?	<a>https://w.w. w.gkseries.com 	 Gkseries.com	 Gkseries.com< /a></a 	 Gkseries.com< /a></a 	 Gkseries.com< /a></a 	36	All normal webpages consists of	Top and bottom	Body and	Head and body	Body and html	Head and body
 and <th> is</th>	is	<data></data>	<group></group>	<table></table>	<th></th>		<table></table>																
35	What is the correct HTML for creating a hyperlink ?	<a>https://w.w. w.gkseries.com 	 Gkseries.com	 Gkseries.com< /a></a 	 Gkseries.com< /a></a 	 Gkseries.com< /a></a 																	
36	All normal webpages consists of	Top and bottom	Body and	Head and body	Body and html	Head and body																	

			frameset			
27	4 T		TT 1 1 1.	Underlined		
37	Is a</td <td>Comment tag</td> <td>Underlined tag</td> <td>with italic tag</td> <td>common tag</td> <td>Comment tag</td>	Comment tag	Underlined tag	with italic tag	common tag	Comment tag
	Choose the correct HTML tag for the					
38	largest heading	<h1></h1>	<h6></h6>	<h3></h3>	<h2></h2>	<h1></h1>
39	To created a bulleted list, use			<dl></dl>		
	is a network in which the					
40	computers are connected directly.	LAN	WAN	NET	ARPANET	LAN
41	——————————————————————————————————————	WAN	LAN	WWW	None	WAN
	Which of the following is the					
42	"Backbone" of the network?	LAN	WAN	NET	Network	LAN
	LANs are connected by the special					
	purpose of computers are called					
43	·	Routers	Resources	Networks	None	Routers
				Client -		
44	——————————————————————————————————————	Clients	Servers	Servers	Internet	Clients
	In web language, a client program is					
45	called a	Browser	Address	Server	All the above	Browser
	To send and receive electronic mail,					
46	one should familiar with	mail client	client – server	internet	e-mail address	mail client
	Within the internet, each separate					
	computer is called					
47		Host	Node	Both 1 & 2	None	Host
	Programmers write their programs					
48	using	Protocols	Script	Both 1 & 2	None	Protocols
		Transmission		Transmission	Transmission	Transmission
		Control	Transiting	Control	Control	Control
49	TCP stands for	Protocol	Control Protocol	Process	Programs	Protocol
		Internet				Internet
50	IP stands for	Protocol	I – Protocol	Internet – P	None	Protocol

	will divide the message into a					
	number of					
51		TCP, Packets	IP, Packets	TCP, Host	IP, Host	TCP, Packets
	The data containing links to another					
	data is called					
52	·	Hypertext	ТСР	E-mail address	None	Hypertext
	The name Usenet is a contraction of					
53		User network	User's network	Network	None	User's network
	Which one of the following has main-					
54	menu and sub- menus?	Gopher	Usenet	Web	Telnet	Gopher
	The FTP service allows to					
55	from one computer to another.	Copy files	Send data	Connect	All the above	Copy files
		File Transfer	File Transaction	Fixed Transfer	File Transiting	File Transfer
56	FTP stands for .	Protocol	Protocol	Protocol	Protocol	Protocol
30		11010001	11000001			11000001
30	Anonymous "ftp" is one of the most					
30	Anonymous "ftp" is one of the most important in	Internet		Gopher	Usenet	Internet
57	Anonymous "ftp" is one of the most important in	Internet Services	Web Services	Gopher Services	Usenet Services	Internet Services
57	Anonymous "ftp" is one of the most important in The role of is to make the	Internet Services	Web Services	Gopher Services	Usenet Services	Internet Services
57 58	Anonymous "ftp" is one of the most important in The role of is to make the whole system manageable.	Internet Services Archie	Web Services Gopher	Gopher Services Website	Usenet Services Usenet	Internet Services Archie
57 58	Anonymous "ftp" is one of the most important in The role of is to make the whole system manageable. The service allowing us to login and	Internet Services Archie	Web Services Gopher	Gopher Services Website	Usenet Services Usenet	Internet Services Archie
<u>57</u> <u>58</u>	Anonymous "ftp" is one of the most important in The role of is to make the whole system manageable. The service allowing us to login and use a remote computer is called	Internet Services Archie	Web Services Gopher	Gopher Services Website	Usenet Services Usenet	Internet Services Archie
57 57 58 59	Anonymous "ftp" is one of the most important in The role of is to make the whole system manageable. The service allowing us to login and use a remote computer is called	Internet Services Archie Telnet	Web Services Gopher Usenet	Gopher Services Website	Usenet Services Usenet None	Internet Services Archie Telnet
57 57 58 59	Anonymous "ftp" is one of the most important in The role of is to make the whole system manageable. The service allowing us to login and use a remote computer is called Menu – Based information is a	Internet Services Archie Telnet	Web Services Gopher Usenet	Gopher Services Website IP	Usenet Services Usenet None	Internet Services Archie Telnet

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

<u>n toUNIT-II</u>

SYLLABUS

Formatting: Basic HTML forms – creating and using image maps – meta tags – frameset – nested frameset. **Introduction to DHTML:** cascading Style Sheet – Inline Style – Creating Style Sheet with style elements – conflicting styles – Linking External Style Sheets – Positioning Elements – Backgrounds – Element Dimension – text flow – box models.

2.1 Basic HTML Forms

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML <form> tag is used to create an HTML form and it has following syntax -

<form>

form elements

</form>

An HTML form contains **form elements**.

Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

The <input> Element

The <input> element is the most important form element.

The <input> element can be displayed in several ways, depending on the **type** attribute.

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

Here are some examples:

Туре	Description
<input type="text"/>	Defines a one-line text input field
<input type="radio"/>	Defines a radio button (for selecting one of many choices)
<input type="submit"/>	Defines a submit button (for submitting the form)

1. Text Input

<input type="text"> defines a one-line input field for text input:

<html>

<body>

<h2>Text Input</h2>

<form>

First name:

<input type="text" name="firstname">

Last name:

<input type="text" name="lastname">

</form>

Note that the form itself is not visible.

Also note that the default width of a text input field is 20 characters.

</body>

</html>

Produces the output as,

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II

BATCH-2016-19

Text Input

First name:

Last name:

Note that the form itself is not visible.

Also note that the default width of a text input field is 20 characters.

Sr.No	Attribute & Description
1	type
	Indicates the type of input control and for text input control it will be set to text .
2	name
	Used to give a name to the control which is sent to the server to be recognized and get the value.
3	value
	This can be used to provide an initial value inside the control.
4	size
	Allows to specify the width of the text-input control in terms of characters.
5	maxlength
	Allows to specify the maximum number of characters a user can enter into the text box.

2. Radio Button Input

<input type="radio"> defines a **radio button**.

Radio buttons let a user select ONE of a limited number of choices:

<html>

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

<body>

<h2>Radio Buttons</h2>

<form>

- <input type="radio" name="gender" value="male" checked> Male

- <input type="radio" name="gender" value="female"> Female

<input type="radio" name="gender" value="other"> Other

</form>

</body>

</html>

Produces the output as,

Radio Buttons

• Male

- Female
- Other

Sr.No	Attribute & Description
1	type
	Indicates the type of input control and for checkbox input control it will be set to
	radio.
2	name
	Used to give a name to the control which is sent to the server to be recognized and
	get the value.
3	value
	The value that will be used if the radio box is selected.
4	Checked
	Set to <i>checked</i> if you want to select it by default.

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

3. The Submit Button

<input type="submit"> defines a button for **submitting** the form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's **action** attribute:

<html>

<body>

<h2>HTML Forms</h2>

```
<form action="/action_page.php">
```

First name:


```
<input type="text" name="firstname" value="Mickey">
```


Last name:


```
<input type="text" name="lastname" value="Mouse">
```



```
<input type="submit" value="Submit">
```

</form>

If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".

</body>

</html>

Produces the output as,

HTML Forms

First name:

Mickey

Last name:

Mouse

<u>S</u>ubmit

If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

The Action Attribute

The action attribute defines the action to be performed when the form is submitted.

Normally, the form data is sent to a web page on the server when the user clicks on the submit button.

In the example above, the form data is sent to a page on the server called "/action_page.php".

This page contains a server-side script that handles the form data:

<form action="/action_page.php">

If the action attribute is omitted, the action is set to the current page.

The Target Attribute

The target attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.

The default value is "_self" which means the form will be submitted in the current window.

To make the form result open in a new browser tab, use the value "_blank":

<form action="/action_page.php" target="_blank">

Other legal values are "_parent", "_top", or a name representing the name of an iframe.

The Method Attribute

The method attribute specifies the HTTP method (**GET** or **POST**) to be used when submitting the form data:

<form action="/action_page.php" method="get">

The default method when submitting form data is GET.

However, when GET is used, the submitted form data will be visible in the page address field.

```
<form action="/action_page.php" method="post">
```

Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field.

4. <u>The <select> Element</u>

The <select> element defines a **drop-down list**: <html>

	COURSE NAME, INT	EN EDUCATION EDNIET AND MED DESIGNING
COURSE CODE: 16CCU601A	UNIT: II	BATCH-2016-19
<body></body>		
<h2>The select Element</h2>		
The select element defines a	drop-down list:	
<form action="/action_page.php'</td><td>'></td><td></td></tr><tr><td><select name=" cars"=""></form>		
<option value="volvo">Volvo</option>		
<option value="saab">Saab<td>option></td><td></td></option>	option>	
<option value="fiat">Fiat<td>tion></td><td></td></option>	tion>	
<option value="audi">Audi<td>option></td><td></td></option>	option>	
>		
<input type="submit"/>		
Produces the output as,		
<u>The select Element</u>		
The select element defines a drop	o-down list:	
-		

-

<u>S</u>ubmit

The <option> elements defines an option that can be selected.

By default, the first item in the drop-down list is selected.

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

Page 7/42

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

Allow Multiple Selections:

Use the multiple attribute to allow the user to select more than one value:

<html>

<body>

<h2>Allow Multiple Seletcions</h2>

Use the multiple attribute to allow the user to select more than one value.

<form action="/action_page.php">

<select name="cars" size="4" multiple>

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="fiat">Fiat</option>

```
<option value="audi">Audi</option>
```

</select>

>

<input type="submit">

</form>

Hold down the Ctrl (windows) / Command (Mac) button to select multiple options.

</body>

</html>

Produces the output as,

Allow Multiple Seletcions

Use the multiple attribute to allow the user to select more than one value.

Volvo Saab Fiat Audi

<u>S</u>ubmit

Hold down the Ctrl (windows) / Command (Mac) button to select multiple options.

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

The <textarea> Element

The <textarea> element defines a multi-line input field (a text area):

<html>

<body>

```
<h2>Textarea</h2>
```

The textarea element defines a multi-line input field.

```
<form action="/action_page.php">
```

```
<textarea name="message" rows="10" cols="30">The cat was playing in the
```

garden.</textarea>


```
<input type="submit">
```

</form>

</body>

</html>

Produces the output as,

<u>Textarea</u>

The textarea element defines a multi-line input field.

		Ţ
-		Þ

<u>S</u>ubmit

The rows attribute specifies the visible number of lines in a text area.

The cols attribute specifies the visible width of a text area.

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

2.2 Creating and using image maps

The <map> tag is used to define a client-side image-map. An image-map is an image with clickable areas.

The required name attribute of the <map> element is associated with the 's usemap attribute and creates a relationship between the image and the map.

The <map> element contains a number of <area> elements, that defines the clickable areas in the image map.

<html>

<body>

Click on the sun or on one of the planets to watch it closer:

```
<img src="planets.gif" width="145" height="126" alt="Planets" usemap="#planetmap">
```

<map name="planetmap">

```
<area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
```

<area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">

```
<area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
```

</map>

</body>

</html>

Produces the output as follows

Click on the sun or on one of the planets to watch it closer:



CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

Attribute	Value	Description
alt	text	Specifies an alternate text for the area. Required if
		the href attribute is present
coords	coordinates	Specifies the coordinates of the area
download	filename	Specifies that the target will be downloaded when
		a user clicks on the hyperlink
href	URL	Specifies the hyperlink target for the area
hreflang	language_code	Specifies the language of the target URL
media	media query	Specifies what media/device the target URL is
		optimized for
nohref	value	Not supported in HTML5.
		Specifies that an area has no associated link
rel	alternate	Specifies the relationship between the current
	author	document and the target URL
	bookmark	
	help	
	license	
	next	
	nofollow	
	noreferrer	
	prefetch	
	prev	
	search	
	tag	
shape	default	Specifies the shape of the area
	rect	
	circle	

CLASS: III BCom CA COURSE CODE: 16CCU601A UNIT: U			
COURSEC	ODE. 10CC0001A	UNII. II	DATC11-2010-19
	poly		
target	_blank	Specifies where t	o open the target URL
	_parent		
	_self		
	_top		
	framename		

2.3 Meta Tags

HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The **<meta>** tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.

Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing <meta> tags inside the header of the document which is represented by <head> and </head> tags. A meta tag can have following attributes in addition to core attributes –

Sr.No	Attribute & Description
1	Name
	Name for the property. Can be anything. Examples include, keywords, description,
	author, revised, generator etc.
2	content
	Specifies the property's value.

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

3	scheme						
	Specifies	a	scheme	to	interpret	the	property's
	value asdeclar	edinthe	contentattribut	easdecla	redinthecontentat	tribute.	
4	http-equiv						
	Used for http	respon	ise message he	eaders. F	For example, http	p-equiv ca	an be used to
	refresh the pa	ge or to	set a cookie.	Values in	nclude content-ty	pe, expire	es, refresh and
	set-cookie.						

Specifying Keywords

You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

Example

Following is an example, where we are adding HTML, Meta Tags, Metadata as important keywords about the document.

<html>

<head>

```
<meta charset="UTF-8">
```

<meta name="description" content="Free Web tutorials">

```
<meta name="keywords" content="HTML,CSS,XML,JavaScript">
```

```
<meta name="author" content="John Doe">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

</head>

<body>

```
All meta information goes in the head section...
```

</body>

</html>

This will produce the following result -

All meta information goes in the head section...

2.4 Frameset

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in

CLASS: III BCom CA	COURSE NAME: INTERNET	AND WEB DESIGNING
COURSE CODE: 16CCU601A	UNIT: II	BATCH-2016-19

the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

The <frameset> tag is not supported in HTML5.

The <frameset> tag defines a frameset.

The <frameset> element holds one or more <<u>frame></u> elements. Each <frame> element can hold a separate document.

The <frameset> element specifies HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.

Creating Frames

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The **rows** attribute of <frameset> tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Example

Following is the example to create three vertical frames -

<html>

<frameset cols="25%,*,25%">

<frame src="frame_a.html">

<frame src="frame_b.html">

<frame src="frame_c.html">

</frameset>

</html>

frame_a.html

<html>

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

<body bgcolor="pink">

<h1>Frame A</h1>

Note: The frameset, frame, and noframes elements are not supported in HTML5.

</body>

</html>

frame_b.html

<html>

<body bgcolor="blue">

<h1>Frame B</h1>

</body>

</html>

frame_c.html

<html>

<body bgcolor="red">

<h1>Frame c</h1>

</body>

</html>

This will produce the following result -

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

🗅 framesexample.html 🛛 🗙 🕂	A CONTRACTOR DURING	· · ··································	
\leftrightarrow \rightarrow C (i) File file:///E:/Karpagam/	framesexample.html		🖈 🔍 💋 🕴 😥
Frame A	Frame B		Frame c
Note: The frameset, frame, and noframes elements are not supported in HTML5.			

The <frameset> Tag Attributes

Following are important attributes of the <frameset> tag -

Sr.No	Attribute & Description
1	cols
	Specifies how many columns are contained in the frameset and the size of each column.
	You can specify the width of each column in one of the four ways –
	Absolute values in pixels. For example, to create three vertical frames, use $cols = "100, 500, 100"$
	A percentage of the browser window. For example, to create three vertical frames, use $cols = "10\% \ 80\% \ 10\%"$
	Using a wildcard symbol. For example, to create three vertical frames, use $cols = "10\%$, *, 10% ". In this case wildcard takes remainder of the window.
	As relative widths of the browser window. For example, to create three vertical frames, use $cols = "3^*, 2^*, 1^*$ ". This is an alternative to percentages. You can use relative
	widths of the browser window. Here the window is divided into sixths: the first column
	takes up half of the window, the second takes one third, and the third takes one sixth.
2	Rows
	This attribute works just like the cols attribute and takes the same values, but it is used
	to specify the rows in the frameset. For example, to create two horizontal frames,
	use $rows = "10\%$, 90%". You can specify the height of each row in the same way as
	explained above for columns.
3	Border
	This attribute specifies the width of the border of each frame in pixels. For example,
	border = "5". A value of zero means no border.
4	frameborder
CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 yesyes or 0 nono. For example frameborder = "0" specifies no border.
 framespacing This attribute specifies the amount of space between frames in a frameset. This can take

This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing = "10" means there should be 10 pixels spacing between each frames.

<u>The <frame> Tag Attributes</u>

Following are the important attributes of <frame> tag -

Sr.No	Attribute & Description
1	src
	This attribute is used to give the file name that should be loaded in the frame. Its value
	can be any URL. For example, src = "/html/top_frame.htm" will load an HTML file
	available in html directory.
2	name
	This attribute allows you to give a name to a frame. It is used to indicate which frame a
	document should be loaded into. This is especially important when you want to create
	links in one frame that load pages into an another frame, in which case the second frame
3	freeds a name to identify itself as the target of the link.
5	Traineportuer This attribute specifies whether or not the borders of that frame are shown; it overrides
	the value given in the frameborder attribute on the <frameset> tag if one is given, and</frameset>
	this can take values either 1 vesves or 0 nono
4	marginwidth
	This attribute allows you to specify the width of the space between the left and right of
	the frame's borders and the frame's content. The value is given in pixels. For example
	marginwidth = "10".
5	marginheight
	This attribute allows you to specify the height of the space between the top and bottom
	of the frame's borders and its contents. The value is given in pixels. For example
	marginheight = "10".
6	noresize
	By default, you can resize any frame by clicking and dragging on the borders of a frame.
	noresize = "noresize"
7	scrolling
-	This attribute controls the appearance of the scrollbars that appear on the frame. This
	takes values either "yes", "no" or "auto". For example scrolling = "no" means it should
	not have scroll bars.
8	longdesc
	This attribute allows you to provide a link to another page containing a long description
	of the contents of the frame. For example longdesc = "framedescription.htm"

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

2.5 Nested Frameset

When two different types of frameset like horizontal and vertical frames created for a same

webpage is said to a nested frameset.

Example

nestedframesetexample.htnl

<html>

<frameset rows="50%,50%">

<frame src="frame_a.html">

<frameset cols="25%,75%">

<frame src="frame_b.html">

```
<frame src="frame_c.html">
```

</frameset>

</frameset>

</html>

frame_a.html

<html>

<body bgcolor="pink">

<h1>Frame A</h1>

Note: The frameset, frame, and noframes elements are not supported in HTML5.

</body>

</html>

frame_b.html

<html>

<body bgcolor="blue">

CLASS: III BCom CA	COURSE NAME: INTER	RNET AND WEB DESIGNING
COURSE CODE: 16CCU601A	UNIT: II	BATCH-2016-19
<h1>Frame B</h1>		
frame_c.html		
<html></html>		
<body bgcolor="red"></body>		
<h1>Frame c</h1>		
Produces the output as follows:		
🗅 nestedframeset.html x 🕂	Marrie tor 1 E Marrie 1	
\leftrightarrow \rightarrow C (1) File file:///E:/Karpagam/nestedframeset.html		🖈 🖾 💋 🗄
Frame A		
Note: The frameset frame and poframes elements are not supported in HTMI 5		

Frame B	Frame c
3 🖹 D 🕑 🚺 6	- 🗂 Pr 🐠 🔐 1047 AM - 🗂 Pr 🐠 🔐 1047 AM

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

2.6 Introduction to DHTML

DHTML stands for Dynamic HTML, it is totally different from HTML. The browsers which support the dynamic HTML are some of the versions of Netscape Navigator and Internet Explorer of version higher than 4.0. The DHTML is based on the properties of the HTML, javascript, CSS, and DOM (Document Object Model which is used to access individual elements of a document) which helps in making dynamic content. It is the combination of HTML, CSS, JS, and DOM. The DHTML make use of Dynamic object model to make changes in settings and also in properties and methods. It also makes uses of Scripting and it is also part of earlier computing trends.

DHTML allows different scripting languages in a web page to change their variables, which enhance the effects, looks and many others functions after the whole page have been fully loaded or under a view process, or otherwise static HTML pages on the same. But in true ways, there is noting that as dynamic in DHTML, there is only the enclosing of different technologies like CSS, HTML, JS, DOM, and different sets of static languages which make it as dynamic.

DHTML is used to create interactive and animated web pages that are generated in real-time, also known as dynamic web pages so that when such a page is accessed, the code within the page is analyzed on the web server and the resulting HTML is sent to the client's web browser.

HTML: HTML stands for Hypertext Markup Language and it is a client-side markup It is block of language. used to build the web pages. Javascript: It is a Client-side Scripting language. Javascript is supported by most of the browser, also have cookies collection determine the needs. to user **CSS**: The abbreviation of CSS is Cascading Style Sheet. It helps in the styling of the web pages and helps in designing of the pages. The CSS rules for DHTML will be modified at different levels using JS with event handlers which adds a significant amount of dynamism little with verv code. **DOM**: It is known as a dynamic Object Model which act as the weakest links in it. The only defect in it is that most of the browser does not support DOM. It is a way to manipulate the static contents.

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

Key Features

Following are the some major key features of DHTML:

- Tags and their properties can be changed using DHTML.
- It is used for real-time positioning.
- Dynamic fonts can be generated using DHTML.
- It is also used for data binding.
- It makes a webpage dynamic and be used to create animations, games, applications along with providing new ways of navigating through websites.
- The functionality of a webpage is enhanced due to the usage of low-bandwidth effect by DHTML.
- DHTML also facilitates the use of methods, events, properties, and codes.

Usages of DHTML

DHTML makes a webpage dynamic but Javascript also does, the question arises that what different does DHTML do? So the answer is that DHTML has the ability to change a webpages look, content and style once the document has loaded on our demand without changing or deleting everything already existing on the browser's webpage. DHTML can change the content of a webpage on demand without the browser having to erase everything else, i.e. being able to alter changes on a webpage even after the document has completely loaded.

Advantages

- Size of the files are compact in compared to other interactional media like Flash or Shockwave, and it downloads faster.
- It is supported by big browser manufacturers like Microsoft and Netscape.
- Highly flexible and easy to make changes.
- Viewer requires no extra plug-ins for browsing through the webpage that uses DHTML, they do not need any extra requirements or special software to view it.
- User time is saved by sending less number of requests to the server. As it is possible to modify and replace elements even after a page is loaded, it is not required to create separate pages for changing styles which in turn saves time in building pages and also reduces the number of requests that are sent to the server.

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IIBATCH-2016-19

• It has more advanced functionality than a static HTML. it is capable of holding more content on the web page at the same time.

Disadvantages

- It is not supported by all the browsers. It is supported only by recent browsers such as Netscape 6, IE 5.5, and Opera 5 like browsers.
- Learning of DHTML requires a lot of pre-requisites languages such as HTML, CSS, JS, etc should be known to the designer before starting with DHTML which is a long and time-consuming in itself.
- Implementation of different browsers are different. So if it worked in one browser, it might not necessarily work the same way in another browser.
- Even after being great with functionality, DHTML requires a few tools and utilities that are some expensive. For example, the DHTML text editor, Dreamweaver. Along with it the improvement cost of transferring from HTML to DHTML makes cost rise much higher.

Difference between HTML and DHTML

- HTML is a markup language while DHTML is a collection of technologies.
- HTML is used to create static webpages while DHTML is capable of creating dynamic webpages.
- DHTML is used to create animations and dynamic menus but HTML not used.
- HTML sites are slow upon client-side technologies whereas DHTML sites are comparatively faster.
- Web pages created using HTML are rather simple and have no styling as it uses only one language whereas DHTML uses HTML, CSS, and Javascript which results in a much better and way more presentable webpage.
- HTML cannot be used as server side code but DHTML used as server side code.
- DHTML needs database connectivity but not in case of HTML.
- Files in HTML are stored using .htm or .html extension while DHTML uses .dhtm extension.
- HTML requires no processing from the browser but DHTML does.

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

2.7 Cascading Style Sheet

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

- CSS saves time You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- Pages load faster If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- Easy maintenance To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Superior styles to HTML CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- Multiple Device Compatibility Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IIBATCH-2016-19

CSS Syntax

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts –

- Selector A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or etc.
- **Property** A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
- Value Values are assigned to properties. For example, *color* property can have value either *red* or #*F1F1F1* etc.

You can put CSS Style Rule Syntax as follows -

selector { property: value }



Example

table{ border :1px solid #C00; }

Here table is a selector and border is a property and given value 1px solid #C00 is the value of that property.

The Type Selectors

This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings –

h1 {

color: #36CFFF;

}

The Universal Selectors

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type –

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

* {

color: #000000;

}

This rule renders the content of every element in our document in black.

The Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to element only when it lies inside <u> tag.

ul em {

color: #000000;

}

The Class Selectors

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

.black {

color: #000000;

}

This rule renders the content in black for every element with class attribute set to black in our document. You can make it a bit more particular. For example –

.black {

color: #000000;

}

This rule renders the content in black for only <h1> elements with class attribute set to black.

Consider the following example -

This para will be styled by the classes center and bold.

The ID Selectors

You can define style rules based on the id attribute of the elements. All the elements having that id will be formatted according to the defined rule.

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

#black {

color: #000000;

}

This rule renders the content in black for every element with id attribute set to black in our document. You can make it a bit more particular. For example –

h1#black {

color: #000000;

}

This rule renders the content in black for only <h1> elements with id attribute set to black. The true power of id selectors is when they are used as the foundation for descendant selectors, For example –

#black h2 {

color: #000000;

}

In this example all level 2 headings will be displayed in black color when those headings will lie with in tags having id attribute set to black.

The Child Selectors

You have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example –

body > **p** {

color: #000000;

}

This rule will render all the paragraphs in black if they are direct child of <body> element. Other paragraphs put inside other elements like <div> or would not have any effect of this rule.

The Attribute Selectors

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of text –

```
input[type = "text"] {
```

color: #000000;

}

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IIBATCH-2016-19

The advantage to this method is that the <input type = "submit" /> element is unaffected, and the color applied only to the desired text fields.

There are following rules applied to attribute selector.

- p[lang] Selects all paragraph elements with a lang attribute.
- p[lang="fr"] Selects all paragraph elements whose lang attribute has a value of exactly "fr".
- p[lang~="fr"] Selects all paragraph elements whose lang attribute contains the word "fr".
- p[lang|="en"] Selects all paragraph elements whose lang attribute contains values that are exactly "en", or begin with "en-".

Multiple Style Rules

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example –

h1 {

color: #36C; font-weight: normal; letter-spacing: .4em; margin-bottom: 1em; text-transform: lowercase;

Here all the property and value pairs are separated by a semicolon (;). You can keep them in a single line or multiple lines. For better readability, we keep them in separate lines.

For a while, don't bother about the properties mentioned in the above block. These properties will be explained in the coming chapters and you can find complete detail about properties in CSS References.

Grouping Selectors

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example –

h1, h2, h3 {

color: #36C;

[}]

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

font-weight: normal; letter-spacing: .4em; margin-bottom: 1em; text-transform: lowercase;

}

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

You can combine the various id selectors together as shown below -

#content, #footer, #supplement {

position: absolute;

left: 510px;

width: 200px;

}

Ways to insert style sheets in CSS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

- 1. Inline style (inside an HTML element)
- 2. External and internal style sheets (in the head section)
- 3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

2.8 Inline Style Sheet

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

The example below shows how to change the color and the left margin of a <1	h1> element:
The example below shows how to change the color and the left margin of a <	h1> element:
<html></html>	
<body></body>	
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>	
This is a paragraph.	
Output	
D horizon x +	
C → C 0 10 He He///E/Kapagam/Researcher/Robitione/ptagated C → C 0 00 He He//E/Kapagam/Researcher/Robitione/Ptagated C → C 0 00 He He//E/Kapagam/Robit	
This is a paragraph.	

😨 🗒 🧕 🥥 💽 🗋 🗷 🔗 🗻 - + + + + + + ***

2.9 Creating Style Sheets with Style Elements

An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

```
<html>
<head>
<style>
body {
background-color: linen;
}
h1 {
color: maroon;
```

CLASS: III BCom CA	COURSE NAME: INTERNET	AND WEB DESIGNING
COURSE CODE: 16CCU601A	UNIT: II	BATCH-2016-19
margin-left: 40px;		
}		
<body></body>		
<h1>This is a heading</h1>		
This is a paragraph.		
Output		
ptag.html x +	اط ۲ ۵ ۵	0 × 1
This is a heading		
This is a paragraph.		
🚱 📜 🔍 🕘 🥥 🗋 💷 🍠 🗻	- 21 - 0 1	513 PM 2/2/2019

2.10 Conflicting Styles

When more than one type of style sheet is used in a HTML program then it is said to conflicting styles.

Here in this example program you can see a HTML program with the combination of both external and internal style sheets.

In this example program style element is used along with the external css file.

<html>

<head>

```
k rel="stylesheet" type="text/css" href="mystyle.css">
```

<style>

KARPAGAM ACADEMY OF HIGHER EDUCATION COURSE NAME: INTERNET AND WEB DESIGNING **CLASS: III BCom CA** COURSE CODE: 16CCU601A UNIT: II BATCH-2016-19 h1 { color: orange; } </style> </head> <body> <h1>This is a heading</h1> The style of this document is a combination of an external stylesheet, and internal style </body> </html> mystyle.css

```
body {
```

```
background-color: lightblue;
```

```
}
```

```
h1 {
```

color: navy;

margin-left: 20px;

}

Output

<section-header>

2.11 Linking External Style Sheets

With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the link> element.

The k> element goes inside the <head> section.

Example Program

<html> <head> <link rel="stylesheet" type="text/css" href="mystyle.css"> </head> <body> <h1>This is a heading</h1> This is a paragraph. </body> </html> An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension. body { background-color: lightblue; }

h1 {

CLASS: III BCom CA	COURSE NAME: INTERNET AND WE	B DESIGNING
COURSE CODE: 16CCU601A	UNIT: II	BATCH-2016-19
color: navy;		
margin-left: 20px;		
}		
Output		
🗅 ptsg.html × +		x
\leftrightarrow \rightarrow C \odot File file:///E:/Karpagam/Referal/New%20folder/ptag.html	× 🛛 💋 🔞	1
This is a heading		
This is a paragraph.		

2.12 Position Elements

The position property specifies the type of positioning method used for an element (static, relative, absolute, fixed, or sticky).

Syntax

position: static/absolute/fixed/relative/sticky/initial/inherit;

Property Values

Value	Description
static	Default value. Elements render in order, as they appear in the document flow
absolute	The element is positioned relative to its first positioned (not static) ancestor element
fixed	The element is positioned relative to the browser window
relative	The element is positioned relative to its normal position, so "left:20px"
	adds 20 pixels to the element's LEFT position

CLASS: III BCom CA		COURSE NAME:	INTERNET AND WEB DESIGNING
COURSE CODE: 16CCU601A		UNIT: II	BATCH-2016-19
sticky	The element is posi	tioned based on the use	er's scroll position
	A sticky element to	ggles between relative	and fixed, depending on the
	scroll position. It is	positioned relative unt	il a given offset position is met
	in the viewport - the	en it "sticks" in place ()	like position:fixed).
initial	Sets this property to) its default value.	
inherit	Inherits this propert	ty from its parent eleme	ent.

Example Program

```
<html>
<head>
<style>
h2.pos_left {
 position: relative;
 left: -30px;
}
h2.pos_right {
 position: relative;
 left: 50px;
}
</style>
</head>
<body>
<h1>The position Property</h1>
Relative positioning moves an element RELATIVE to its original position.
The style "left: -30px;" subtracts 30 pixels from the element's original left position.
The style "left: 50px;" adds 50 pixels to the element's original left position.
<h2 class="pos_left">This heading is moved left according to its normal position</h2>
<h2 class="pos_right">This heading is moved right according to its normal position</h2>
</body>
</html>
```

KARPAGAM ACADEMY OF HIGHER EDUCATION COURSE NAME: INTERNET AND WEB DESIGNING **CLASS: III BCom CA** COURSE CODE: 16CCU601A **UNIT: II** BATCH-2016-19 Output ptag.htm × + ← → C ③ File | file:///E:/Karpaga 🖈 🖾 💋 🔞 E The position Property elative positioning moves an element RELATIVE to its original p The style "left: -30px;" subtracts 30 pixels from the element's original left po The style "left: 50px;" adds 50 pixels to the element's original left positi is heading is moved left according to its normal position This heading is moved right according to its normal position

🚳 🛗 🔍 曼 🥝 💽 🔟 💷 🥥

2.13 Backgrounds

The background property is a shorthand property for:

- background-color
- background-image
- background-position
- background-size
- background-repeat
- background-origin
- background-clip
- background-attachment

It does not matter if one of the values above are missing, e.g. background:#ff0000 url(smiley.gif); is allowed.

Syntax

background: *bg-color bg-image position/bg-size bg-repeat bg-origin bg-clip bg-attachment* initial|inherit;

If one of the properties in the shorthand declaration is the bg-size property, you must use a / (slash) to separate it from the bg-position property, e.g. background:url(smiley.gif) 10px 20px/50px 50px; will result in a background image, positioned 10 pixels from the left, 20 pixels from the top, and the size of the image will be 50 pixels wide and 50 pixels high.

If using multiple background-image sources but also want a background-color, the background-color parameter needs to be last in the list.

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

Property Values

Property Values

Value	Description
background-color	Specifies the background color to be used
background-image	Specifies ONE or MORE background images to be used
background-	Specifies the position of the background images
<u>position</u>	
background-size	Specifies the size of the background images
background-repeat	Specifies how to repeat the background images
background-origin	Specifies the positioning area of the background images
background-clip	Specifies the painting area of the background images
background-	Specifies whether the background images are fixed or scrolls with
<u>attachment</u>	the rest of the page
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

The background-image property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

Example

<html> <head> <style> body { background-image: url("gradient_bg.png"); background-repeat: repeat-x;

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

} </style> </head> <body> <h1>Hello World!</h1> Here, a background image is repeated only horizontally! </body> </html>

Output

Hello World!

Here, a background image is repeated only horizontally!

2.14 Element Dimension

The height and width properties are used to set the height and width of an element.

The height and width can be set to auto (this is default. Means that the browser calculates the height and width), or be specified in length values, like px, cm, etc., or in percent (%) of the containing block.

Setting max-width

The max-width property is used to set the maximum width of an element. The max-width can be specified in length values, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).

The problem with the $\langle div \rangle$ above occurs when the browser window is smaller than the width of the element (500px). The browser then adds a horizontal scrollbar to the page.

KARPAGAM ACADEMY OF HIGHER EDUCATION CLASS: III BCom CA COURSE NAME: INTERNET AND WEB DESIGNING COURSE CODE: 16CCU601A UNIT: II BATCH-2016-19 Using max-width instead, in this situation, will improve the browser's handling of small windows. **Example Program** <html> <head> <style> div { max-width: 500px; height: 100px; background-color: powderblue; } </style> </head> <body> <h2>Set the max-width of an element</h2> This div element has a height of 100px and a max-width of 500px: <div></div> Resize the browser window to see the effect. </body> </html> Output × + x 🖾 🤰 🎯 Set the max-width of an element This div element has a height of 100px and a max-width of 500ps Resize the browser window to see the effect. 🔞 🚞 🧕 🕘 🧔 🚺 🖳 🚽 💋 🚿

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

Page 38/42

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

2.15 Text Flow

The text-overflow property specifies how overflowed content that is not displayed should be signaled to the user. It can be clipped, display an ellipsis (...), or display a custom string.

Both of the following properties are required for text-overflow:

- white-space: nowrap;
- overflow: hidden;

Syntax

text-overflow: clip/ellipsis/string/initial/inherit;

Property Values

Value	Description
clip	Default value. The text is clipped and not accessible
ellipsis	Render an ellipsis ("") to represent the clipped text
string	Render the given string to represent the clipped text
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. <u>Read</u>
	<u>about inherit</u>

Example Program

<html> <head> <style> div.a { white-space: nowrap; width: 100px; overflow: hidden; text-overflow:ellipsis; border: 1px solid #000000;

}

KARPAGA	M ACADEMY OF HIGHER	EDUCATION
CLASS: III BCom CA	COURSE NAME: INTER	NET AND WEB DESIGNING BATCH-2016-19
COURSE CODE. INCCOURT	UNII.II	DATCII-2010-17
div.a:hover {		
overflow: visible;		
}		
<body></body>		
<h1>The text-overflow Property</h1>		
Hover over the div below, to	see the entire text.	
<div class="a">This is some lon</div>	g text that will not fit in the b	oox.
Output		
🗅 ptag.html 🗙 🕂		- 1 ×
← → C ③ File file:///E:/Karpagam/Referal/New%20folder/p	tag.html	🖈 🖻 💋 E
I ne text-overflow Property		
This is some		

2.16 Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

- 10:03 PM 2/2/2019

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: II BATCH-2016-19

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



Explanation of the different parts:

Content - The content of the box, where text and images appear

Padding - Clears an area around the content. The padding is transparent

Border - A border that goes around the padding and content

Margin - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

Example Program

<html></html>
<head></head>
<style></style>

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING **UNIT: II** BATCH-2016-19

border: 25px solid green; padding: 25px; margin: 25px; }

</style>

</head>

<body>

<h2>Demonstrating the Box Model</h2>

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

<div>This text is the actual content of the box. We have added a 25px padding, 25px margin and a 25px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div>

</body>

</html>

Output



(Deemed University Established Under Section 3 of UGC Act 1956) Coimbatore - 641021. (For the candidates admitted from 2017 onwards) **Department of Commerce (CA)**

SUBJECT: INTERNET AND WEB DESIGNING SEMESTER : VI SUBJECT CODE: 16CCU601A CLASS : III B.COM CA

UNIT:II

S.	QUESTIONS	OPTION 1	OPTION 2	OPTION 3	OPTIO	ANSWER
Ν					N 4	
0						
	HTML attribute is used to define					
1	inline styles.	Class	Styles	Style	Font	Style
	The CSS property controls the text				font-	
2	size.	text-style	font-size	text-size	style	font-size
	The CSS Property is used to give		border-		border-	
3	border around an image.	border-color	decoration	border-style	line	border-style
	The property is used to display			noncontent-		
4	border around a cell without any content.	empty-cell	blank-cell	cell	void-cell	empty-cell
	If we want to wrap a block of text around an					
5	image, which css property will we use?	wrap	push	float	align	float
	value is used to show an arrow as a					
6	cursor.	Pointer	Default	Arrow	Arr	Default
	The CSS property is used to not					
7	allow floating point to DIV.	margin	clear	float	padding	clear

	Suppose we want to arrange five nos of divs so that div4 is placed above div1. Now which					
	property of css will we use to control the order					
8	of stack?	d-index	s-index	x-index	z-index	z-index
					Dynamic	
					Hyper	
					Text	Dynamic
		Domain Hyper	Dual Hyper		Markup	Hyper Text
		Text Markup	Text Markup		Languag	Markup
9	DHTML stands for	Language	Language		e	Language
	The CSS selector is used to define					
10	style for an unique element.	id	text	name	class	id
11	The <map> tag defines a</map>	image map	map	map image	images	map
		Background				Background
	Which one of the following is difficult to	color, text		improper	all the	color, text
12	read?	color	text color	statements	above.	color
	are used to select different kinds of					
13	user input.	HTML forms	HTML table	HTML lists	None	HTML forms
	In HTML, Comments lines are denoted					
14	by	&	- & -	! & ->	!&!	&
					<mail< th=""><th></th></mail<>	
				<mail>XXX</mail>	href	
			<a href="</th"><th>@YYY<th>="XXX</th><th></th></th>	@YYY <th>="XXX</th> <th></th>	="XXX	
		<u><a href="mailto:</u"></u>	"XXX@YYY"	1	@YYY"	<u><a href="mailto:</u"></u>
15	How can you make an e-mail link?	XXX@YYY>	>	>	>	XXX@YYY>
1.0	Images are embedded within a web document	•	T		1	T
16	page with the use of	input	Img tag	text file	http	Img tag
17	Does the image tag has an end tag?	random	yes	no	optional	no
	The html element defines the of the			_		
18	document.	optional	Whole	Part	Body	Whole
	What is the correct HTML for making a text		<input type="</td"/> <td><input type="</td"/><td><text< td=""><td><input type="</td"/></td></text<></td></td>	<input type="</td"/> <td><text< td=""><td><input type="</td"/></td></text<></td>	<text< td=""><td><input type="</td"/></td></text<>	<input type="</td"/>
21	area?	input	"text area">	"text box">	area>	"text area">

22	How many different layers are in HTML?	6	2	3	4	2
	The event that captures one or more event and				Release	
23	routes to object is	event	Capture event	Route event	event	Capture event
	For every web page we create, it must					
24	have	<body></body>	<html></html>	<head></head>	<title></title>	<html></html>
	Which of these functions will encloses the		<html><td><body><td></td><td><html></html></td></body></td></html>	<body><td></td><td><html></html></td></body>		<html></html>
25	entire HTML document?		l>	ody>	<hr/>	ml>
	When the mouse is over the image, the					
26	function refers to	high	change-high	change-low	change	change-high
	When the mouse leaves the image area, the					
27	function refers to	high	change-high	change-low	change	change-low
	Html Attributes are always specified in the					
					Commen	
28	tag.	Start	End	Paragraph	t line	Start
			Not case	Additional		
29	Attribute values are	Case sensitive	sensitive	information	optional	Case sensitive
	The tag defines how to divide the					
30	window into frames.	Frameset	Table	Frame	Form	Frameset
	The tag is used to get response form					
31	the user at runtime.	Frameset	Table	Frame	Form	Form
	To avoid empty cells we specify			Null		
32		()	Hyphen	character	0	()
					<text< td=""><td></td></text<>	
33	A form is defined with the	<html></html>	<form></form>	<input/>	area>	<form></form>
					no close	
34	The image tag ends with the				tag	no close tag
		<body< td=""><td><background></background></td><td><body< td=""><td></td><td><body< td=""></body<></td></body<></td></body<>	<background></background>	<body< td=""><td></td><td><body< td=""></body<></td></body<>		<body< td=""></body<>
	What is the correct html for adding a	color="vellow"	vellow <td>bgcolor="vell</td> <td></td> <td>bgcolor="vello</td>	bgcolor="vell		bgcolor="vello
35	background color?	>	ound>	ow">	<image/>	w">
	are used to select different kinds of					
36	user input.	HTML forms	HTML table	HTML lists	None	HTML forms
37	is not the type of input in html forms.	text area	button	list	images	images

	is tag is used to give details about the					
38	webpage.	meta	image	title	frame	meta
	The frames tag should have atleast					
39	divisions.	1	2	3	4	2
	The frame tag is used to divide the webpage	rows and	header and			rows and
40	into	columns	footer	tables	forms	columns
	The page which is to be printed in the frame is					
41	given in parameter.	src	img	href	ref	src
	What is the correct tag for inserting a line					
42	break?		<lb></lb>	<break></break>	<none></none>	
43	element will follow the head tag.	Body	Title	hr	br	title
44	Table values are displayed in a format.	Tabular	Frames	rows	columns	Tabular
45	For a $\langle p \rangle$ tag the end tag is required or not?	Required	Not Required	Optional	must	Optional
	Choose the correct HTML tag to make a text		-			
46	italic?	<ii></ii>	<i></i>	<italic></italic>	<it></it>	<i></i>
	Choose the correct HTML tag to make a text				<underli< td=""><td></td></underli<>	
47	underline?		<u></u>	<und></und>	ne>	<u></u>
	The base element specifies an absolute					
48		URL address	IP address	TCP address	href	URL address
	How can you make a list that lists the items					
49	with bullets?	<list></list>		<01>	<db></db>	
	Passing a Boolean value the sound					
50	playing.	Start	begins	ends	pause	begins
	Which of these attributes will specify				code	
51	"location of an objects implementation?	code base	class id	data	type	class id
	Which of the following options are not object				code	
52	tag attributes?	code base	class id	code base	type	code base
	which of the following is not a direct		Structured			
53	animation controls?	code base	graphics	Path	Sprite	code base
	Each IRC conversation is carried on a					
54	·	Channel	Wires	Cables	Circuits	Channel

	The Description of web search engines is		Receives	Gathering	All the	
55		Search the web	messages	information	above	Search the web
56	Menu – Based information is a	Gopher	Archie	Usenet	Mail	Gopher
					Using of	
57	Multi – linked information is a	The web	Mailing lists	Talk facility	e-mail	The web
58		Modem	Cable	Wires	None	Modem
	is the process of converting from a					
59	digital to Analog format.	Modulation	Demodulation	Modem	Data	Modulation
	is the process of converting from					
60	Analog to digital format.	Modulation	Demodulation	Modem	Data	Demodulation

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

<u>UNIT-III</u>

SYLLABUS

Introduction to Java Script: Operators – Arithmetic Operators – Precedence of Operators – Relational Operators – Control Structures – Assignment Operators – Increment and Decrement Operators – For Loops – Switch – Do While – Break – Continue – Arrays – Functions.

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript**, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

Java Scrip Syntax

JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script> HTML tags in a web page.

You can place the **<script>** tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the **<head>** tags.

The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

<script...> JavaScript code </script>

The script tag takes two important attributes -

- Language This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
- **Type** This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IIIBATCH-2016-19

So your JavaScript segment will look like -

<script language="javascript" type="text/javascript"> JavaScript code </script>

Let us take a sample example to print out "Hello World". We added an optional HTML comment that surrounds our JavaScript code. This is to save our code from a browser that does not support JavaScript. The comment ends with a "//-->". Here "//" signifies a comment in JavaScript, so we add that to prevent a browser from reading the end of the HTML comment as a piece of JavaScript code. Next, we call a function **document.write** which writes a string into our HTML document.

This function can be used to write text, HTML, or both. Take a look at the following code.

<html>

<body>

```
<script language="javascript" type="text/javascript">
```

```
<!---
Document.write("Hello World!")
```

```
//-->
</script>
</body>
</html>
```

This code will produce the following result –

Hello World!

Whitespace and Line Breaks

JavaScript ignores spaces, tabs, and newlines that appear in JavaScript programs. You can use spaces, tabs, and newlines freely in your program and you are free to format and indent your programs in a neat and consistent way that makes the code easy to read and understand.

Semicolons are Optional

Simple statements in JavaScript are generally followed by a semicolon character, just as they are in C, C++, and Java. JavaScript, however, allows you to omit this semicolon if each of your statements are placed on a separate line. For example, the following code could be written without semicolons.

```
<script language="javascript" type="text/javascript">
<!---
var1=10
```

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

var2=20

//--> </script>

Case Sensitivity

JavaScript is a case-sensitive language. This means that the language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

So the identifiers **Time** and **TIME** will convey different meanings in JavaScript.

Comments in JavaScript

JavaScript supports both C-style and C++-style comments, Thus -

- Any text between a // and the end of a line is treated as a comment and is ignored by JavaScript.
- Any text between the characters /* and */ is treated as a comment. This may span multiple lines.
- JavaScript also recognizes the HTML comment opening sequence <!--. JavaScript treats this as a single-line comment, just as it does the // comment.
- The HTML comment closing sequence --> is not recognized by JavaScript so it should be written as //-->.

JavaScript Datatypes

One of the most fundamental characteristics of a programming language is the set of data types it supports. These are the type of values that can be represented and manipulated in a programming language.

JavaScript allows you to work with three primitive data types -

- Numbers, eg. 123, 120.50 etc.
- **Strings** of text e.g. "This text string" etc.
- **Boolean** e.g. true or false.

JavaScript also defines two trivial data types, **null** and **undefined**, each of which defines only a single value. In addition to these primitive data types, JavaScript supports a composite data type known as **object**. We will cover objects in detail in a separate chapter.

CLASS: III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGCOURSE CODE: 16CCU601AUNIT: IIIBATCH-2016-19

JavaScript does not make a distinction between integer values and floating-point values. All numbers in JavaScript are represented as floating-point values. JavaScript represents numbers using the 64-bit floating-point format defined by the IEEE 754 standard.

JavaScript Variables

Like many other programming languages, JavaScript has variables. Variables can be thought of as named containers. You can place data into these containers and then refer to the data simply by naming the container.

Before you use a variable in a JavaScript program, you must declare it. Variables are declared with the **var** keyword as follows.

```
<script type="text/javascript">
<!---
var money, name;
var name;
```

```
//-->
</script>
```

Storing a value in a variable is called **variable initialization**. You can do variable initialization at the time of variable creation or at a later point in time when you need that variable.

For instance, you might create a variable named **money** and assign the value 2000.50 to it later. For another variable, you can assign a value at the time of initialization as follows.

```
<script type="text/javascript">
<!---
```

```
var name="Ali";
var money;
money=2000.50;
```

//--> </script>

3.1 Operators

Let us take a simple expression **4** + **5** is equal to **9**. Here 4 and 5 are called **operands** and '+' is called the **operator**. JavaScript supports the following types of operators.

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

- Assignment Operators
- Increment/Decrement Operators

3.2 Arithmetic Operators and Increment/Decrement Operators

JavaScript supports the following arithmetic operators -

Assume variable A holds 10 and variable B holds 20, then -

Sr.No.	Operator & Description
1	+ (Addition)
	Adds two operands
	Ex: A + B will give 30
2	- (Subtraction)
	Subtracts the second operand from the first
	Ex: A - B will give -10
3	* (Multiplication)
	Multiply both operands
	Ex: A * B will give 200
4	/ (Division)
	Divide the numerator by the denominator
	Ex: B / A will give 2
5	% (Modulus)
	Outputs the remainder of an integer division
	Ex: B % A will give 0
6	++ (Increment)
	Increases an integer value by one
	Ex: A++ will give 11
7	(Decrement)
	Decreases an integer value by one
	Ex: A will give 9

Example

The following code shows how to use arithmetic operators in JavaScript.

<html>

<body>

```
<script type="text/javascript">
<!---
```

```
var a =33;
```

var b =10;
```
CLASS: III BCom CA
COURSE CODE: 16CCU601A
```

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

var c="Test"; var linebreak="
"; var result; document.write("a+b="); result =a+b; document.write(result); doucument.write(linebreak); document.write("a/b = "); result=a/b; document.write(result); document.write(linebreak); document.write("a%b ="); result =a%b; document.write(result); document.write(linebreak); document.write("a+b+c ="); result=a+b+c; document.write(result); document.write(linebreak); a=++a; document.write("++a = "); result=++a; document.write(result); document.write(linebreak); b=--b; document.write(result); document.write(linebreak); //--> </script> </body>

```
</html>
```

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

Output

a+b=4 a-b=23 a/b=3.3 a%b=3 a+b+c=43Test ++a=35 --b=8

3.3 Precedence of Operators

Precedence of operators gives the order of evaluating an expression with n number of operators and the order in which the evaluation of an expression. The following table define the precedence of the operators in JavaScript,

Value	Operator	Description	Example
20	()	Expression grouping	(3+4)
	·		
19		Member	person.name
19	[]	Member	person["name"]
19	0	Function call	myFunction()
19	new	Create	new Date()
17	++	Postfix Increment	i++
17	-	Postfix Decrement	i
16	++	Prefix Increment	++i
16		Prefix Decrement	i
16	!	Logical not	!(x==y)
16	typeof	Туре	typeof x
15	**	Exponentiation (ES7)	10 ** 2
14	*	Multiplication	10 * 5
14	/	Division	10 / 5
14	%	Division Remainder	10 % 5
13	+	Addition	10 + 5
13	-	Subtraction	10 - 5
12	<<	Shift left	x << 2

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

Page 7/12

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

12	>>	Shift right	x >> 2
12	>>>	Shift right (unsigned)	x >>> 2
11	<	Less than	x < y
11	<=	Less than or equal	x <= y
11	>	Greater than	x > y
11	>=	Greater than or equal	x >= y
11	in	Property in Object	"PI" in Math
11	instanceof	Instance of Object	instanceof Array
10	==	Equal	x == y
10	===	Strict equal	x === y
10	!=	Unequal	x != y
10	!==	Strict unequal	x !== y
9	&	Bitwise AND	х & у
8	^	Bitwise XOR	x ^ y
7		Bitwise OR	x y
6	&&	Logical AND	x && y
5		Logical OR	x y
4	?:	Condition	? "Yes" : "No"
3	+=	Assignment	x += y
3	+=	Assignment	x += y
3	-=	Assignment	x -= y
3	*=	Assignment	x *= y
3	%=	Assignment	x %= y
3	<<=	Assignment	x <<= y
3	>>=	Assignment	x >>= y
3	>>>=	Assignment	x >>>= y
3	&=	Assignment	x &= y
3	^ <u>=</u>	Assignment	x ^= y
3	=	Assignment	$x \models y$
2	yield	Pause Function	yield x
1	,	Comma	5,6
P			

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

3.4 Relational Operators

JavaScript supports the following relational operators -

Assume variable A holds 10 and variable B holds 20, then -

Sr.No.	Operator & Description			
1	&& (Logical AND)			
	If both the operands are non-zero, then the condition becomes true.			
	Ex: (A && B) is true.			
2	(Logical OR)			
	If any of the two operands are non-zero, then the condition becomes true.			
	Ex: $(A \parallel B)$ is true.			
3	! (Logical NOT)			
	Reverses the logical state of its operand. If a condition is true, then the Logical			
	NOT operator will make it false.			
	Ex: ! (A && B) is false.			
Example				

Try the following code to learn how to implement Logical Operators in JavaScript.

<html> <body>

```
<script type="text/javascript">
       <!--
       var a=true;
       var b=false;
       var linebreak="<br/>br/>";
       document.write("(a && b) =>");
       result=(a \&\& b);
       document.write(result);
       document.write(linebreak);
       document.write("(a \parallel b) =>");
       result=(a \parallel b);
       document.write(result);
       document.write(linebreak);
       document.write("!(a && b) =>");
       result=(!(a && b));
       document.write(result);
       document.write(linebreak);
       //-->
</script>
</body>
</html>
```

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

Output

(a && b)=>false (a || b)=>true !(a && b)=>true

3.5 Control Structures

3.6 Assignment Operators

JavaScript supports the following assignment operators -

Sr.No.	Operator & Description
1	= (Simple Assignment)
	Assigns values from the right side operand to the left side operand
	Ex: $C = A + B$ will assign the value of $A + B$ into C
2	+= (Add and Assignment)
	It adds the right operand to the left operand and assigns the result to the left
	operand.
	Ex: $C += A$ is equivalent to $C = C + A$
3	-= (Subtract and Assignment)
	It subtracts the right operand from the left operand and assigns the result to the
	left operand.
	Ex: C \rightarrow = A is equivalent to C = C - A
4	*= (Multiply and Assignment)
	It multiplies the right operand with the left operand and assigns the result to the
	left operand.
	Ex: $C = A$ is equivalent to $C = C A$
5	/= (Divide and Assignment)
	It divides the left operand with the right operand and assigns the result to the left
	operand.
	Ex: C \neq A is equivalent to C = C \neq A
6	%= (Modules and Assignment)
	It takes modulus using two operands and assigns the result to the left operand.
	Ex: C $\%$ = A is equivalent to C = C $\%$ A

Example

Try the following code to implement assignment operator in JavaScript.

<html>

<body>

CLASS: III BCom CA COURSE CODE: 16CCU601A

UNIT: III

COURSE NAME: INTERNET AND WEB DESIGNING BATCH-2016-19

<script type="text/javascript"> <!--var a=3; var b=10; var linebreak="
br/>"; document.write("Value of a=> (a=b)=>"); result = (a=b); document.write(result); document.write(linebreak); document.write("Value of a=> (a+=b)=>"); result = (a+=b); document.write(result); document.write(linebreak); document.write("Value of a => (a -= b) =>"); result = (a -= b); document.write(result); document.write(linebreak); document.write("Value of $a \Rightarrow (a = b) \Rightarrow$ "): result = (a *= b);document.write(result): document.write(linebreak); document.write("Value of a=> (a /= b)=>"); result = $(a \neq b)$; document.write(result); document.write(linebreak); document.write("Value of a=> (a %= b)=>"); result = (a % = b);document.write(result); document.write(linebreak); //--> </script>

</body> </html>

Output

Value of a => (a = b) => 10Value of $a \Rightarrow (a += b) \Rightarrow 20$ Value of a => (a -= b) => 10Value of a => (a *= b) => 100Value of $a => (a \neq b) => 10$ Value of a => (a % = b) => 0

3.8 For Loop

The 'for' loop is the most compact form of looping. It includes the following three important parts -

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

- The **loop initialization** where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.
- The **test statement** which will test if a given condition is true or not. If the condition is true, then the code given inside the loop will be executed, otherwise the control will come out of the loop.
- The iteration statement where you can increase or decrease your counter.

You can put all the three parts in a single line separated by semicolons.

Flow Chart

The flow chart of a for loop in JavaScript would be as follows -



{

Statement(s) to be executed if test condition is true }

Parameter Values

Parameter	Description
initialization	Optional. Executed before the loop (the code block) starts. Normally
	this statement is used to initialize a counter variable. To initiate
	multiple values, separate each value with a comma.
	Note: This parameter can be omitted. However, do not omit the

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

	semicolon ";"			
test condition	Optional. Defines the condition for running the loop (the code block)Normally this statement is used to evaluate the condition of the countvariable. If it returns true, the loop will start over again, if it returnsfalse, the loop will end.Note: This parameter can be omitted. However, do not omit thesemicolon ";". Also, if you omit this parameter, you must provide a			
	break inside the loop. Otherwise the loop will never end, which will crash your browser			
iteration statement	Optional. Executed each time after the loop (the code block) has been executed. Normally this statement is used to increment or decrement the counter variable. Note: This parameter can be omitted (e.g. to increase/decrease values inside the loop)			

Example program

<html>

<body>

```
<script type = "text/javascript">
```

var count;

```
document.write("Starting Loop" + "<br />");
```

for(count = 0; count < 10; count++)</pre>

{

```
document.write("Current Count : " + count );
document.write("<br />");
```

}

```
document.write("Loop stopped!");
```

</script>

</body>

</html>

Output

Starting Loop Current Count : 0 Current Count : 1 Current Count : 2 Current Count : 3 Current Count : 4 Current Count : 5

Prepared by R.Bhuvaneswari, Asst Prof, Department of Commerce, KAHE

Page 13/12

KARPAGAM ACADEMY OF HIGHER EDUCATION **CLASS: III BCom CA** COURSE NAME: INTERNET AND WEB DESIGNING

COURSE CODE: 16CCU601A

BATCH-2016-19 UNIT: III

Current Count : 6 Current Count : 7 Current Count: 8 Current Count : 9 Loop stopped!

3.9 Switch Statement

The most basic loop in JavaScript is the while loop which would be discussed in this chapter. The purpose of a while loop is to execute a statement or code block repeatedly as long as an **expression** is true. Once the expression becomes **false**, the loop terminates.

Flow Chart

The flow chart of while loop looks as follows



CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III

BATCH-2016-19

Example Program

```
<html>
```

<body> <script type = "text/javascript"> var count = 0; document.write("Starting Loop ");

```
while (count < 10) {
         document.write("Current Count : " + count + "<br />");
         count++;
       }
      document.write("Loop stopped!");
   </script>
   </body>
</html>
```

Output

Starting Loop

- Current Count : 0
- Current Count: 1
- Current Count : 2
- Current Count : 3
- Current Count : 4
- Current Count : 5
- Current Count : 6
- Current Count : 7
- Current Count: 8
- Current Count : 9
- Loop stopped!

3.10 Do While

The do...while loop is similar to the while loop except that the condition check happens at the end of the loop. This means that the loop will always be executed at least once, even if the condition is **false**.



CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING **UNIT: III**

BATCH-2016-19

while (count < 5); document.write ("Loop stopped!"); </script> </body> </html>

Output:

Starting Loop Current Count : 0 Current Count : 1 Current Count: 2 Current Count: 3 Current Count: 4 Loop Stopped!

3.11 Break and Continue statement

Break and Continue statement are also known as unconditional branching statements.

The break statement "jumps out" of a loop.

The continue statement "jumps over" one iteration in the loop.

The Break Statement

You have already seen the break statement used in an earlier chapter of this tutorial. It was used to "jump out" of a switch() statement.

- The break statement can also be used to jump out of a loop.
- The break statement breaks the loop and continues executing the code after the loop.

Example Program

```
<html>
<body>
A loop with a break.
<script>
var text = "";
var i:
for (i = 0; i < 10; i++)
```

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III

BATCH-2016-19

```
if (i === 3) { break; }
 text += "The number is " + i + " <br>";
}
</script>
</body>
</html>
```

Output

A loop with a break.

The number is 0 The number is 1 The number is 2

The Continue Statement

The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

Example Program:

This example skips the value of 3:

```
<html>
<body>
<p>A loop which will skip the step where i = 3.</p>
<script>
var text = "";
var i:
for (i = 0; i < 10; i++) {
 if (i === 3) { continue; }
 text += "The number is " + i + " < br>";
}
</script>
</body>
</html>
```

Output:

A loop which will skip the step where i = 3.

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

The number is 0 The number is 1 The number is 2 The number is 4 The number is 5 The number is 6 The number is 7 The number is 8

The number is 9 JavaScript arrays are used to store multiple values in a single variable.

3.12 Arrays

JavaScript arrays are used to store multiple values in a single variable.

Creating an Array

Using an array literal is the easiest way to create a JavaScript Array.

Syntax:

```
var array_name = [item1, item2, ...];
```

An array is a special variable, which can hold more than one value at a time.

Example:

<html>

<body>

```
<h2>JavaScript Arrays</h2>
```

<script>

```
var cars = ["Saab", "Volvo", "BMW"];
```

document.getElementById("demo").innerHTML = cars;

</script>

</body>

</html>

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

Output

JavaScript Arrays

Saab, Volvo, BMW

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

var car1 = "Saab";

var car2 = "Volvo";

var car3 = "BMW";

An array can hold many values under a single name, and you can access the values by referring to an index number.

You access an array element by referring to the index number.

This statement accesses the value of the first element in cars:

cars[1] = "Saab";

cars[2] = "Volvo";

cars[3]= "BMW";

Array Properties

Here is a list of the properties of the Array object along with their description.

0.	Property & Description
<u>constructor</u>	
Returns a ref	ference to the array function that created the object.
index	
The property	y represents the zero-based index of the match in the string
input	
This propert	y is only present in arrays created by regular expression matches.
length	
Reflects the	number of elements in an array.
prototype	
The prototyp	be property allows you to add properties and methods to an object.
Reflects the prototype The prototype	number of elements in an array.

CLASS: III BCom CACOURSECOURSE CODE: 16CCU601AUNIT: III

COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

In the following sections, we will have a few examples to illustrate the usage of Array properties.

Array Methods

Here is a list of the methods of the Array object along with their description.

Sr.No.	Method & Description			
1	concat			
	Returns a new array comprised of this array joined with other arrayss and/or valuess.			
2	every			
	Deturns true if every element in this encourage tisfies the provided testing function			
3	Returns true if every element in this array satisfies the provided testing function.			
5				
	Creates a new array with all of the elements of this array for which the provided			
	filtering function returns true.			
4	forEach			
	Calls a function for each element in the array.			
5	indexOf			
	Deturns the first least least in day of an element within the arrow equal to the specified			
	Returns the first least least index of an element within the array equal to the specified value, or -1 if none is found			
6	ioin			
	Joins all elements of an array into a string.			
7	lastIndexOf			
	Returns the last greatest greatest index of an element within the array equal to the			
8	man			
0	map			
	Creates a new array with the results of calling a provided function on every element			
	in this array.			
9	pop			
10	Removes the last element from an array and returns that element.			
10	push			
	Adds one or more elements to the end of an array and returns the new length of the			
	array.			
11	reduce			
	Apply a function simultaneously against two values of the			

CLASS COURS	III BCom CACOURSE NAME: INTERNET AND WEB DESIGNINGE CODE: 16CCU601AUNIT: IIIBATCH-2016-19			
	array fromleft-to-rightfromleft-to-right as to reduce it to a single value.			
12	reduceRight			
12				
	A galar a function simultaneously assignt two values of the			
	Apply a function simultaneously against two values of the			
10	array from fight-to-left from fight-to-left as to reduce it to a single value.			
13	reverse			
	Reverses the order of the elements of an array the first becomes the last, and the			
	last becomes the first.			
14	shift			
	Removes the first element from an array and returns that element.			
15	slice			
	Extracts a section of an array and returns a new array			
16	Extracts a section of an array and returns a new array.			
10	some			
	Deturns true if at least one clament in this way of for the specific least			
	function			
15	function.			
17	toSource			
	Represents the source code of an object			
18	sort			
	Sorts the elements of an array			
19	splice			
	Adds and/or removes elements from an array			
20	toString			
20	tosting			
21	Returns a sung representing the array and its elements.			
21	unsnitt			
	Adds one or more elements to the front of an array and returns the new length of the			
	array.			

3.13 Functions

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it).

JavaScript Function Syntax

function name(parameter1, parameter2, parameter3) {

CLASS: III BCom CA COURSE CODE: 16CCU601A

COURSE NAME: INTERNET AND WEB DESIGNINGUNIT: IIIBATCH-2016-19

 $\ensuremath{\textit{//}}\xspace$ code to be executed

}

- A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ().
- Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).
- The parentheses may include parameter names separated by commas: (parameter1, parameter2, ...)
- The code to be executed, by the function, is placed inside curly brackets: {}
- Function parameters are listed inside the parentheses () in the function definition.
- Function arguments are the values received by the function when it is invoked.
- Inside the function, the arguments (the parameters) behave as local variables.
- A Function is much the same as a Procedure or a Subroutine, in other programming languages.

Function Invocation

The code inside the function will execute when "something" invokes (calls) the function, as

- 1. When an event occurs (when a user clicks a button)
- 2. When it is invoked (called) from JavaScript code
- 3. Automatically (self invoked)

Function Return

When JavaScript reaches a return statement, the function will stop executing.

If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.

Functions often compute a **return value**. The return value is "returned" back to the "caller":

```
<html>
<body>
<h2>JavaScript Functions</h2>
This example calls a function which performs a calculation and returns the result:
<script>
var x = myFunction(4, 3);
document.getElementById("demo").innerHTML = x;
function myFunction(a, b) {
return a * b;
}
```

CLASS: III BCom CA COURSE CODE: 16CCU601A COURSE NAME: INTERNET AND WEB DESIGNING UNIT: III BATCH-2016-19

</script> </body> </html>

Output JavaScript Functions

This example calls a function which performs a calculation and returns the result:

12



(Deemed University Established Under Section 3 of UGC Act 1956) Coimbatore - 641021. (For the candidates admitted from 2017 onwards) **Department of Commerce (CA)**

SUBJECT: INTERNET AND WEB DES	IGNING		
SEMESTER : VI			
SUBJECT CODE: 16CCU601A	CLASS	: III	
B.COM CA			UNIT:III

S.N		OPTION	OPTION	OPTION	OPTION	
0	QUESTIONS	1	2	3	4	ANSWER
		Applicati		Programm	compilin	
1	JavaScript is a language.	on	Scripting	ing	g	Scripting
2	JavaScript is a side scripting language.	server	client	browser	ISP	browser
			add		execute	add
			interactivit		query	interactivit
		style	y to	perform	related to	y to
		HTML	HTML	server side	DB on	HTML
3	JavaScript is designed to	page	page	scripting	server	page
		directly			included	directly
		into		directly on	separatel	into
		HTML	directly	the server	y to	HTML
4	JavaScript can be written	page	into JS file	script	HTML	page
		<scriptin< th=""><th><javascript< th=""><th></th><th></th><th></th></javascript<></th></scriptin<>	<javascript< th=""><th></th><th></th><th></th></javascript<>			
5	JavaScript code is placed inside HTML element.	g>	>	<script></script>		

			n()	myFunctio		
				n()		
				if i==5	if i=5	
7	How to write an IF statement in JavaScript?	if(i==5)	if i=5	then	then	if(i==5)
					for $(i = 0;$	
		for $i = 1$	for $(i = 0; i)$	for (i <=	i <= 5;	for $(i = 0; i)$
8	How does a FOR loop start?	to 5	<= 5)	5; i++)	i++)	<= 5; i++)
		Math.		Math.max(Math.max(
9	How do you find the number with the highest value of x and y?	Ceil(x, y)	top(x, y)	x, y)	ceil(x, y)	x, y)
10	Which of the following is not an assignment operator?	==	+=	-=	*=	==
	Executable single line of script in JavaScript is known as				Code	
11		Statement	Breakpoint	Line		Statement
12	JavaScript statements are executed by	Server	Compiler	JVM	Browser	JVM
13	Java Statement terminated by	Slash	Comma	Semicolon	Full Stop	Semicolon
	statement is used to declare variable in	Assignme	Declaratio	Executabl	Condition	Declaratio
14	JavaScript.	nt	n	e	al	n
15	Variable can hold value at a time.	Double	Triple	Single	Multiple	Single
	Integer Variable is declared using syntax in		Integer		integer	
16	JavaScript.	var num	num;	int num;	num;	var num
	The method is used to return the value of the last					
17	element inserts in an array	last()	get()	pop()	push()	pop()
	The statement is used to print a line using	window.	document.	navigator.	window.	document.
18	JavaScript.	write()	write()	write()	write()	write()
	• • • • • • • • • • • • • • • • • • •		fontcolor(
19	The function is used to print a string in italic.	fixed())	fontsize()	italic ()	italic ()
	The function of array object is used to reduce two		reduceRig			
20	array values into a single array value.	reduce()	ht()	pop()	push()	reduce()
	The method is used to insert an element into an					
21	array.	last()	get()	pop()	push()	push()
22	The equal comparison operator in JavaScript is	<	!=	"_"	^ <u>_</u>	==

	To add number to sum, you write (Note: JavaScript is	number	number =	sum=num	sum +=	sum +=
22	case_sensitive).	+= sum;	sum +	ber	number;	number;
23	Which of the following expression results in a value 12	2.0/ 1	number;	00/ 0	27.0/ 6	27.0/ 6
24	Which of the Declear surressions below is in correct?	2 % 1	13 % 4	9%9	37%0	37%0
	which of the Boolean expressions below is incorrect?	(true)	!(X > 0)	!(X > 1)	(x = 0)	(true) & &
25		$\alpha \alpha (5 =>$	$\alpha \alpha (x > 0)$	$\alpha \alpha (x > 0)$	$\ (\mathbf{x} = 0)$	(3 = > 4)
23	Which of the following is the correct expression that evaluates	$\frac{4}{1 < \mathbf{v} < \mathbf{v}}$	((x < 100))	$\frac{1}{1} \leq \mathbf{x} \leq \mathbf{x}$	$(1 > \mathbf{v} >$	((x < 100))
	to true if the number x is between 1 and 100 or the number is	1 < X < 100 & &	((X < 100))	1 < X < 100 & x x	$(1 > x > 100) \parallel (x > 100) \parallel $	((X < 100))
	negative?	x < 0	1)) (x < 1)	< 10	< 0	$1)) \parallel (\mathbf{x} \leq \mathbf{x})$
26	inguitte.	A VO	$\begin{pmatrix} 1 \end{pmatrix} \parallel \begin{pmatrix} n \\ 0 \end{pmatrix}$			0
	The "less than or equal to" comparison operator in JavaScript	<	<=	>	<<	<=
27	is					
	Suppose x=10 and y=10 what is x after evaluating the	9	10	11	12	11
28	expression $(y > 10) \& (x ++ > 10).$					
	What is the representation of the third element in an array	a[2]	a(2)	a{2}	a(3)	a[2]
29	called a?					
30	the length of a string by calling the method	strlen()	len()	length()	none	length()
31	Unary operators requireOperand	One	Two	Three	Four	One
32	Binary Operator requiresOperand	One	Two	Three	Four	Two
		It			It	It
		seperates			combines	combines
		the first			the first	the first
		operand			operand	operand
		and the	It reduces	It increase	and the	and the
22		second	the data	the data	second	second
33	In case of strings '+' is used to	operand	size	size	operand	operand
	The "<<" is known as	insertion	extraction	lesser	greater	insertion
34		operator	operator			operator
	Test is performed at theof the for loop.	top	middle	end	program	top
35					terminates	
36	The statements of javascript has multiple statements that	script tag	title tag	hr tag	font tag	script tag

	should be enclosed within					
37	The parameters of script tag are and	language and type	language and range	type and char	language and char	language and type
38	The value of ++I is	i=i+1	i=1+i	i=1+1	i=1-i	i=1+i
39	The value of I is	i=i-1	1=1-i	i=1-1	i=i-i	i=i-1
40	Which is not a increment decrement operator?	i++	i—	i	1+i	1+i
41	Which loop executes the statement and then checks on the condition?	while	for	do while	for next	do while
42	In number of time the statement executed by do while is	equal to while	1+ while	1-while	2+while	1+ while
43	is not a unconditional branching statement	break	continue	goto	switch	switch
44	Find the odd man out?	Start	>		<hr/>	start
45	The amount of space between the cells in the table refers to	cell spacing	cell padding	table	cells	cell spacing
	Which of the following options are not object tag attributes?	cell	Quote	Single	none	cell
46		spacing		quote		spacing
47	which of the following is not a direct animation controls?	cell spacing	quote	Single quote	none	cell spacing
48	is an example of method call	mul();	mul;	mul	mul()	mul();
49	is type of varibles can be given as arrays.	char	numbers	string	all types of data	all types of data
50	Which type of data type is not declared as var?	String	array	numbers	character s	String
51	Data and time can be declared as data type	String	array	var	date and time	date and time
52	Functions are known as	sub	main	arrays	datas	sub
52	Functions should have	declaratio	call of	call of	call of	Declaratio
53		n	function	values	data	n
54	Array are stored in memory.	random	same	in sequence	as per user data	same

	The method is used to return the value of the first					
55	element inserts in an array.	last()	get()	pop()	push()	last()
56	The method is used to get input values.	last()	get()	pop()	push()	get()
57	The alternate language of javascript is	jscript	j2ee	jstring	jswing	jscript
	JavaScript was developed by	Sun Microsyst	Microsoft	IBM	Netscape Navigator	Sun Microsyste
58		em				m
59	JavaScript is language	scripting	markup	extended	standard	scripting
60	JavaScript is used to design a	Web page	Image	Frame	Flex	Web page

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE

Class: III BCOM (CA) Course Code: 16CCU601A **Course Name: INTERNET AND WEB DESIGN**

Unit -IV Semester: VI

Year: 2016-2019 Batch

<u>UNIT-IV- Active Server Page (ASP)</u> <u>SYLLABUS</u>

Introduction – How ASP work – five objects of ASP – Client Side Scripting Vs Server Side Scripting – Server Side ActiveX component – Session Tracking and Cookies – file system objects.

Introduction:

Active Server Pages were introduced by Microsoft in 1996 as a downloadable feature of Internet Information Server(IIS). An Active Server Page allows code written in the JavaScript or VBScript languages to be embedded within the HTML tags of a Web page and executed on the Web server. There are great advantages to this, not the least of which is security. Since your code is executed on the Web server, only HTML tags are sent to the browser. The result is that the ASP code is invisible to the end user.

The server-side script concept is that it allows things like database connections to be made from the Web server rather than from the client. Therefore, any special configurations that might need to be set up, like ODBC data sources, only have to exist on the server. Of course, before you can create an Active Server Page (ASP), you'll need to look at the software requirements.

There are products to allow ASPs to be used on just about any Web server out there. This fact makes using ASPs that much more attractive because you aren't limited in the choice of hardware, operating system, or Web server to host your Web pages. As you can see, there are even ASP-compatibility products for the iSeries.

Active Server Page support to non-Microsoft Web servers. It has a more complete list of ASP compatibility products and the operating systems and Web servers they run on. IIS or Personal Web Server, no additional software is required to support Active Server Pages. To allow a user to access an ASP, the ability to do so must be enabled on the IIS server. This is done by

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 1/32



Page 2/32



ASP Compatibility

- ASP is a Microsoft Technology
- To run IIS you must have Windows NT 4.0 or later
- To run PWS you must have Windows 95 or later
- ChiliASP is a technology that runs ASP without Windows OS
- InstantASP is another technology that runs ASP without Windows

What is an ASP File?

- An ASP file is just the same as an HTML file
- An ASP file can contain text, HTML, XML, and scripts
- Scripts in an ASP file are executed on the server
- An ASP file has the file extension ".asp"

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 3/32

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE

Class: III BCOM (CA)Course Name: INTERNET AND WEB DESIGNCourse Code: 16CCU601AUnit -IVSemester: VIYear: 2016-2019 Batch

Advantages of Using Asp:

• Minimizes network traffic by limiting the need for the browser and server to talk to each

other.

- Makes for quicker loading time since HTML pages are only downloaded.
- Allows running programs in languages that are not supported by the browser.
- Can provide the client with data that does not reside on the client's machine.
- Provides improved security measures since the script cannot be viewed by the browser.

How Does ASP Differ from HTML?

- When a browser requests an HTML file, the server returns the file
- When a browser requests an ASP file, IIS passes the request to the ASP engine. The ASP engine reads the ASP file, line by line, and executes the scripts in the file. Finally, the ASP file is returned to the browser as plain HTML.

ASP	HTML
ASP is a server-side language. This means that the code that is written gets sent to the server, and it returns some code depending on what it was asked to do.	HTML is a client-side language. Basically it has to do with the user interface, with which the user interacts. This interface, is most often, the browser on the user's machine.
ASP can use any scripting language, so as to embed programming and a server side directives into an HTML web page.	HTML allows web browsers to interpret display content written between tags. It allows images and objects to be embedded in the webpage.
ASP is used to design user-interactive	HTML is basically used to

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 4/32

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE					
Class: III BCOM (CA)	Co	ourse Name: INTERNE	Γ AND W	EB DESIGN	
Course Code: 16CCU601A	Unit -IV	Semester: VI	Year: 2	2016-2019 Batch	
or dynamic web pages.		create static web pages.			
ASP is case sensitive.		HTML is not case sensit	ive.		
ASP or ASP.NET pages can	connect				
to the database so as to deriv	e its	HTML cannot connect to	o a		
content.		database.			
When the browser requests a	n ASP				
file, it passes the request to the	he ASP				
engine. The ASP engine read	ls the file,				
line by line, and then execute	es the	When a browser request	s an		
script line by line. Finally, the	e ASP file	HTML file, the server re	eturns		
returns to the browser as plai	n HTML.	the file.			
A sample ASP coding:					
<html></html>					
<head></head>					
<title>Sample ASP Clock<</title>					
<body bgcolor="BLAC</td><td>CK"></body>					
	>				
<script language="VE</td><td>Script" run<="" td=""><td>IAT="SERVER"></td><td></td><td></td></tr><tr><td>RESPONSE.WRITE NOW()</td><td></td><td></td><td></td><td></td></tr><tr><td></script>					
Declaring an ASP script wit	hin a webpaş	ge			
An ASP script begins with <9	∕₀ and ends w	ith %>. It can be placed a	nywhere	within a webpag	
Syntax:					
<% script contents go here %?	>				

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 5/32

Class: III BCOM (CA)	Cour	se Name: INTERNE	T AND WEB DESIGN
Course Code: 16CCU601A	Unit -IV	Semester: VI	Year: 2016-2019 Batch
Example:			
<html></html>			
<head></head>			
<title>ASP script</title>			
<body></body>			
<% Response.Write "ASP is co	ool!"; %>		
Output:			
ASP is cool			
Printing text on a webpage			
Printing text on a webpage with	ASP is simple	e. To do so use the Re	sponse.Write() statement.
Syntax:			
Response.Write("textToPrint").			
Example:			
<% Response.Write("ASP stand	ds for Active S	erver Pages") %>	
Output:			
ASP stands for Active Server P	ages		
Alternatively, you could just us	e an equal sign	instead of the Respon	nse.Write() statement like this:
<%="ASP stands for Active Set	rver Pages"%>		
Output:			
ASP stands for Active Server P	ages		
How ASP work:			

For example, a client types in a URL into your browser. The browser requests the ASP page from the web server. The server proceeds the file with ".asp" extension to ASP Engine in which Objects

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 6/32

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE				
Class: III BCOM (CA)	Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN			
Course Code: 16CCU601A	Unit -IV	Semester: VI	Year: 2016-2019 Batch	

or ActiveX Components can be used to extend the web server with application-specific functionality. In addition, ASP will use ADO to connect to a database (SQL, Access, Oracle, etc.) to pull out the relevant data, such as the current weather in a specific area.

Thus, a different page is generated according to the area specified and time that the page is accessed. Then, the server generates HTML tags before sending it back to the client. Therefore, when you view the source of an ASP file, you will not see any difference from a standard HTML file.

A page created using ASP typically contains a mixture of HTML, scripts, and other components written in any programming language. When a client requests an ASP file, the scripts in the file are processed on the server.

The scripts can reference components running on either the local server or any other accessible server, and can perform actions such as accessing a database, sending e -mail, or processing information in another fashion. The result is then returned by the server to the client as a standard HTML file and displayed in the usual way.

For example, when requested, the following ASP file will return the current time and browser type to the requesting client:

<HTML>
<HEAD>
<TITLE>Sample Web Page</TITLE>
</HEAD>
<BODY>
The time right now is <% = now %>
Your browser type is
<% =Request.ServerVariables("http_user_agent")%>
</BODY>
</HTML>



Two Request methods:

- GET Gets (retrieves) information from the server Retrieve HTML document or image
- \bullet POST Posts (sends) data to the server Send info from HTML form »Client-entered data »

Info to search Internet »Query for a database »Authentication info.



Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 8/32

	Cour	se Name: INTERNF	T AND WEB DESIGN
Course Code: 16CCU601A	Unit -IV	Semester: VI	Year: 2016-2019 Batch
Five objects of ASP:			
Δ SP includes five build-in obje	ects:		
Asi menudes rive build-in obje			
 Request – to get infor 	rmation from th	e user that is passed a	long with an HTTP request
 Request – to get infor Response – to output 	rmation from th data to the requ	e user that is passed a sting client	along with an HTTP request
 Request – to get infor Response – to output Server – to control th 	rmation from th data to the requ e Internet Infor	e user that is passed a sesting client mation Server	along with an HTTP request
 Request – to get infor Response – to output Server – to control th Session – to store var 	rmation from th data to the requ e Internet Infor iables associate	te user that is passed a testing client mation Server and with a given user s	along with an HTTP request
 Request – to get infor Response – to output Server – to control th Session – to store var Application – to store 	rmation from the data to the request e Internet Infor iables associate information the	the user that is passed a desting client mation Server and with a given user s that remains active for	along with an HTTP request ession the lifetime of an application,
 Request – to get infor Response – to output Server – to control th Session – to store var Application – to store 	rmation from the data to the requ e Internet Infor iables associate information th	the user that is passed a cuesting client mation Server and with a given user s that remains active for	along with an HTTP request ession the lifetime of an application,
 Request – to get infor Response – to output Server – to control th Session – to store var Application – to store such as a page counter. 	rmation from the data to the requ e Internet Infor iables associate information th	te user that is passed a desting client mation Server ed with a given user s hat remains active for	along with an HTTP request ession the lifetime of an application,
 Request – to get infor Response – to output Server – to control th Session – to store var Application – to store such as a page counter. Request Object It allows data to be read that 	rmation from the data to the requ e Internet Infor iables associate information the was sent by t	he user that is passed a clesting client mation Server ed with a given user s hat remains active for he client browser: F	along with an HTTP request ession the lifetime of an application, ⁵ orm, Querystring,
 Request – to get infor Response – to output Server – to control th Session – to store var Application – to store such as a page counter. Request Object It allows data to be read that and HTTP Cookie. It also pr 	rmation from the data to the requ e Internet Infor tables associate information the was sent by the rovides information	the user that is passed a desting client mation Server and with a given user s that remains active for the client browser: F mation on the server,	along with an HTTP request ession the lifetime of an application, ⁵ orm, Querystring, the client browser, and
 Request – to get infor Response – to output Server – to control th Session – to store var Application – to store such as a page counter. Request Object It allows data to be read that and HTTP Cookie. It also pr 	rmation from the data to the requ e Internet Infor iables associate information the was sent by the rovides information	te user that is passed a desting client mation Server ed with a given user s hat remains active for he client browser: F dation on the server, r's machine. Can re	along with an HTTP request ession the lifetime of an application, ⁵ orm, Querystring, the client browser, and trieve data from a form usin

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 9/32

Course Code: 16CCU601A Unit -IV Semester: VI Year: 2016-2019 Batel It allows access to both the header and body of the HTTP request. Syntax: Request.QueryString ("parameter") • • Form: used today with method POST to provide access to the pair - fieldname = value. Syntax: Request.Form ("fieldname") • Query String: used in the past with method GET to pass parameters. • Server Variables: displays the content of the server environment variables, including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") • • Cookies: allows cookie processing. Syntax: Request.Cookies () • ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.QueryString reads data sent by GET. ~% Response.Write "Welcome" "!" %> ************************************	Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIG					
It allows access to both the header and body of the HTTP request. Syntax: Request.QueryString ("parameter") • Form: used today with method POST to provide access to the pair - fieldname = value. Syntax: Request.Form ("fieldname") • Query String: used in the past with method GET to pass parameters. • Server Variables: displays the content of the server environment variables, including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") • Cookies: allows cookie processing. Syntax: Request.Cookies () • ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.QueryString reads data sent by POST. Request.QueryString reads data sent by GET. <% Response.Write "Welcome" "!" %>	Course Code: 16CCU601A	Unit -IV	Semester: VI	Year: 2016-2019 Batch		
It allows access to both the header and body of the HTTP request. Syntax: Request.QueryString ("parameter") • Form: used today with method POST to provide access to the pair - fieldname = value. Syntax: Request.Form ("fieldname") • Query String: used in the past with method GET to pass parameters. • Server Variables: displays the content of the server environment variables, including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") • Cookies: allows cookie processing. Syntax: Request.Cookies () • ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. <% Response.Write "Welcome" "!" %>						
It allows access to both the header and body of the HTTP request. Syntax: Request.QueryString ("parameter") • Form: used today with method POST to provide access to the pair - fieldname = value. Syntax: Request.Form ("fieldname") • Query String: used in the past with method GET to pass parameters. • Server Variables: displays the content of the server environment variables, including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") • Cookies: allows cookie processing. Syntax: Request.Cookies () • ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. <% Response.Write "Welcome" "!" %>						
Syntax: Request.QueryString ("parameter") • Form: used today with method POST to provide access to the pair - fieldname = value. Syntax: Request.Form ("fieldname") • Query String: used in the past with method GET to pass parameters. • Server Variables: displays the content of the server environment variables, including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") • Cookies: allows cookie processing. Syntax: Request.Cookies () • ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. « Response.Write "Welcome" "!" %>	It allows access to both the h	eader and body of	f the HTTP request.			
 Request.QueryString ("parameter") Form: used today with method POST to provide access to the pair - fieldname = value. Syntax: Request.Form ("fieldname") Query String: used in the past with method GET to pass parameters. Server Variables: displays the content of the server environment variables, including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") Cookies: allows cookie processing. Syntax: Request.Cookies () ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. %>	Syntax:					
 Form: used today with method POST to provide access to the pair - fieldname = value. Syntax: Request.Form ("fieldname") Query String: used in the past with method GET to pass parameters. Server Variables: displays the content of the server environment variables, including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") Cookies: allows cookie processing. Syntax: Request.Cookies () ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. %	Request.QueryString ("parar	neter")				
 Syntax: Request.Form ("fieldname") Query String: used in the past with method GET to pass parameters. Server Variables: displays the content of the server environment variables, including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") Cookies: allows cookie processing. Syntax: Request.Cookies () ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. «%	• Form: used today wit	h method POST t	o provide access to th	ne nair - fieldname = value		
 Query String: used in the past with method GET to pass parameters. Server Variables: displays the content of the server environment variables, including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") Cookies: allows cookie processing. Syntax: Request.Cookies () ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. %>	Svntax: Request.Form ("field	lname")		ie pair neronanie valuer		
 Server Variables: displays the content of the server environment variables, including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") Cookies: allows cookie processing. Syntax: Request.Cookies () ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. « Response.Write "Welcome" "!"	Ouery String: used in	the past with me	thod GET to pass par	ameters.		
 including REQUEST_METHOD which allows the testing of a method (GET or POST). Syntax: Request.ServerVariables ("environment-varname") Cookies: allows cookie processing. Syntax: Request.Cookies () ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. %>	• Server Variables: dis	plays the content	of the server environ	ment variables,		
 Syntax: Request.ServerVariables ("environment-varname") Cookies: allows cookie processing. Syntax: Request.Cookies () ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. %>	including REQUEST_METH	HOD which allow	s the testing of a met	hod (GET or POST).		
 Cookies: allows cookie processing. Syntax: Request.Cookies () ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. %>	Syntax: Request.ServerVaria	bles ("environme	nt-varname")			
 ClientCertificate: used in secure transactions. Syntax: Request.ClientCertificate () Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. <% Response.Write "Welcome" "!" 	Cookies: allows cookie processing. Syntax: Request.Cookies ()					
Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. <% Response.Write "Welcome" "!" %>	ClientCertificate: use	d in secure transa	ctions. Syntax: Requ	est.ClientCertificate ()		
Example: Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. <% Response.Write "Welcome" "!" %>				P		
Request.Form reads data sent by POST. Request.QueryString reads data sent by GET. <% Response.Write "Welcome" "!" %>	Example:					
Request.QueryString reads data sent by GET. <% Response.Write "Welcome" "!" %>	Request.Form reads data s	ent by POST.				
<pre><% Response.Write "Welcome" "!" %></pre>	Request OueryString read	s data sent by G	ET			
Response.Write "Welcome" "!" %>	<%					
	Response.Write "Welcome" %>					



• Buffer: controls if the response is sent as the page and script are created, or if are kept in a buffer and sent when processing is completed.

Syntax: Response.Buffer = True (False is the default)

• CacheControl: allows caching by a Proxy server, but the Proxy server needs to be setup to cache pages.

Syntax: Response.CacheControl = "setting", where setting is Public or Private(default)

ContentType: allows setting the page content type. The default id text/html.

Example:

0

- <% Response.ContentType = "text/HTML" %> <% Response.ContentType = "image/GIF" %> <% Response.ContentType = "image/JPEG" %> <% Response.ContentType = "text/plain" %>
- <% Response.ContentType = "image/JPEG" %>

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 11/32



control features related to starting and stopping the application, as well as store

information that should be accessed by the application as a whole. It stores global variables.

Example:

<%

Application("Ali") = "My ASP Application"

Response.Write "Welcome to " & Server.HTMLEncode(Application("Ali")) & "!"

%>


Server Object:

The server object, this is used to contain and send information about the web server itself,

perform cleanup of user information no longer needed, and create and terminate input/output

connection to user files and other applications like database records.

Example:

Dim oAdoCon, oAdoRec, oAdoStm, oCdoCon, oCdoMsg, oSciDic, oSciFsm, oMswAdr

Set oAdoCon = Server.CreateObject("ADODB.Connection")

Set oAdoRec = Server.CreateObject("ADODB.Recordset")

Set oAdoStm = Server.CreateObject("ADODB.Stream")

Set oCdoCon = Server.CreateObject("CDO.Configuration")

Set oCdoMsg = Server.CreateObject("CDO.Message")



Session object:

This object is used to store information with a scope to that user session. The information

stored is maintained even when the user moves through various pages in the web application.

The session object has two properties.

- SessionID Created by the web application and sent as a cookie to the client.
- TimeOut To set a Session timeout period.

Example:

<%

If Len(Request.QueryString("name")) > 0 Then

Session("name") = Request.QueryString("name")

End If

Response.Write "Welcome " & Server.HTMLEncode(Session("name")) & "!"

%>



The Session object —This is used to store data and create a "work area" for each user who visits and navigates around the Web site. It provides the continuity ("state") that is lost in static HTML pages when a user moves from one HTML page to another and can initiate special routines or display special pages when a user first arrives at the site and exits from the site.

Client Side Scripting Vs Server Side Scripting:

Client-side Environment:

The client-side environment used to run scripts is usually a browser. The processing takes place on the end users computer. The source code is transferred from the web server to the user's computer over the internet and run directly in the browser.

The scripting language needs to be enabled on the client computer. Sometimes if a user is conscious of security risks they may switch the scripting facility off. When this is the case a message usually pops up to alert the user when the script is attempting to run.

The scripts can be written in two forms, at the server end (back end) or at the client end (server end). The main difference between server-side scripting and client-side scripting is that the server side scripting involves server for its processing. On the other hand, client-side scripting requires browsers to run the scripts on the client machine but does not interact with the server while processing the client-side scripts.

A script is generally a series of program or instruction, which has to be executed on other program or application. As we know that the web works in a client-server environment. The client-

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 15/32

KARPAGAM ACA	DEMY OF H	IGHER EDUCATIO	DN, COIMBATORE	
Class: III BCOM (CA)	Cours	se Name: INTERNE	T AND WEB DESIGN	
Course Code: 16CCU601A	Unit -IV	Semester: VI	Year: 2016-2019 Batch	

side script executes the code to the client side which is visible to the users while a server-side script is executed in the server end which users cannot see.

Server-side Environment

The server-side environment that runs a scripting language is a web server. A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser. It is usually used to provide interactive web sites that interface to databases or other data stores on the server.

This is different from client-side scripting where scripts are run by the viewing web browser, usually in JavaScript. The primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements, access rights, or queries into data stores. Server-side scripting is a technique of programming for producing the code which can run software on the server side, in simple words any scripting or programming that can run on the web server is known as server-side scripting. The operations like customization of a website, dynamic change in the website content, response generation to the user's queries, accessing the database, and so on are performed at the server end.

The server-side scripting constructs a communication link between a server and a client (user). Earlier the server side scripting is implemented by the CGI (Common Gateway Interface) scripts. The CGI was devised to execute the scripts from programming languages such as C++ or Perl on the websites.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 16/32



KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN Course Code: 16CCU601A Unit -IV Semester: VI Year: 2016-2019 Batch

Server Side ActiveX component:

An ActiveX control is a component program object that can be re-used by many application programs within a computer or among computers in a network. The technology for creating ActiveX controls is part of Microsoft's overall ActiveX set of technologies, chief of which is the Component Object Model (COM).

Types of Server side ActiveX Components:

Ad Rotator

The Ad Rotator streamlines the process of setting up a delivery system for your banner ads. In a separate file, you store information regarding the banner. The component then delivers a randomly selected banner every time the page is loaded.

The Browser component lets you determine what browser a user is using and what features are supported by that browser Collaboration Data Objects (CDO).

Tied in with the IIS SMTP server, CDO lets you send and receive email. With CDO, for example, you can process a form without relying upon a Perl script and CGI.

Content Linking

This is a handy object for creating a linear or sequential pathway through your site or a subsection of the site. You maintain a simple text file that lists the files in the proper sequence. Simple next and previous links then can be added to the page, and a table of contents can be easily generated.

Content Rotator

If you have a need for rotating content, this will be a favorite component. It is easy to use and allows you to add dynamic content to any page without using a database. In a separate text file, you store chunks of HTML code that you want alternately dropped into a space on the page. The Content Rotator will display one of the chunks each time the page is reloaded.

Database Access

Using this component, you can hook into a database to write contents to the browser screen and to create or update existing database files.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 18/32

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE

Course Name: INTERNET AND WEB DESIGN

Course Code: 16CCU601A

Class: III BCOM (CA)

Unit -IV Semester: VI

Year: 2016-2019 Batch

MSWC.BrowserType	ActiveX component for gathering information about the
	client's browser (e.g., type, version).
MSWC.AdRotator	ActiveX component for rotating advertisements on a
	Web page.
MSWC.NextLink	ActiveX component for linking Web pages together.
MSWC.ContentRotator	ActiveX component for rotating HTML content on a
	Web page.
MSWC.PageCounter	ActiveX component for storing the number of times a
	Web page has been requested.
MSWC.Counters	ActiveX components that provide general-purpose
	persistent counters.
MSWC.MyInfo	ActiveX component that provides information about a
	Web site (e.g., owner name, owner address).
Scripting.FileSystemObject	ActiveX component that provides an object library for
	accessing files on the server or on the server's network.
ActiveX Data Objects (ADO)	ActiveX components that provide an object library
Data Access Components	for accessing databases.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 19/32



Session Tracking and Cookies:

Session object stores information about, or change settings for a user session.



The Session object

The Session object stores information about, or change settings for a user session. The Variables stored in a Session object hold information about one single user, and are available to all pages in one application. Common information stored in session variables are name, id, and preferences. The server creates a new Session object for each new user, and destroys the Session object when the session expires.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 20/32

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE **Course Name: INTERNET AND WEB DESIGN** Class: III BCOM (CA) Course Code: 16CCU601A **Unit -IV** Semester: VI Year: 2016-2019 Batch **Session Start:** A session starts when: A new user requests an ASP file, and the Global.asa file includes a Session_OnStart procedure A value is stored in a Session variable A user requests an ASP file, and the Global.asa file uses the <object> tag to instantiate an object with session scope Session End: A session ends if a user has not requested or refreshed a page in the application for a specified period. By default, this is 20 minutes. If you want to set a timeout interval that is shorter or longer than the default, use the Timeout property. **Example:** <% Session.Timeout=5 %> Store and Retrieve Session Variables <% Session("username")="Donald Duck" Session("age")=50 %> **Remove Session Variables** The Contents collection contains all session variables. It is possible to remove a session variable with the Remove method. <% Session.Contents.RemoveAll() %>

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATOREClass: III BCOM (CA)Course Name: INTERNET AND WEB DESIGNCourse Code: 16CCU601AUnit -IVSemester: VIYear: 2016-2019 Batch

Cookies:

The Cookies collection is used to set or get cookie values. If the cookie does not exist, it will be created, and take the value that is specified. The Response. Cookies command must appear before the <html> tag.

Example:



Syntax:

Response.Cookies(name)[(key)|. attribute] =value variablename=Request.Cookies(name)[(key)|. attribute]

cookies message read and set:



Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 22/32

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE

Course Name: INTERNET AND WEB DESIGN

Course Code: 16CCU601A

Class: III BCOM (CA)

Unit -IV Semester: VI

Year: 2016-2019 Batch

Parameter	Description
Name	Required. The name of the cookie valueRequired
	for the Response.Cookies command. The value of the cookie.

Attribute	Optional. Specifies information about the
	cookie. Can be one of the following
	parameters:
1. Domain	Write only. The cookie is sent only to
	requests to
	this domain
2. Expires	Write only. The date when the cookie
	expires. If no date is specified, the cookie
	will expire when
	the session ends
3. HasKeys	Read only. Specifies whether the cookie
	has keys
	(This is the only attribute that can be used
	with the Request.Cookies command)
4. Path	Write only. If set, the cookie is sent only to
	requests to this path. If not set, the
	application path is used
Secure	Write only. Indicates if the cookie is
	secure
key	Optional. Specifies the key to where
	the value is assigned

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 23/32

Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN				
Course C	ode: 16CCU601A	Unit -IV	Semester: VI	Year: 2016-2019 Batch
\triangleright	A name-value	pair containin	g the actual data	
\triangleright	An expiry date	e after which i	t is no longer valid	d
\triangleright	The domain a	nd path of the	server it should be	e sent to
Syntax :				
<%				
Response.	Cookies("firstname")	="Alex"		
%>				
Example	:			
DOCTYI	PE html>			
<html></html>				
<body></body>				
<%				
dim x,y				
for each x	in Request.Cookies			
response	.write("")			
if Reques	st.Cookies(x).HasKey	ys then		
for each	n y in Request.Cookie	es(x)		
respon	nse.write(x & ":" & y	& "=" & Reque	st.Cookies(x)(y))	
respon	nse.write(" ")			
next				
else				
Respon	se.Write(x & "=" & F	Request.Cookies	(x) & " ")	
end if				
response	.write ""			
next				
%>				

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 24/32



Creating a Cookie

Cookies are created using response. Cookies. Cookies are sent in the headers of the response so you should always set cookies before sending the body of the response.

Example Cookie Creation

```
<%
Response.Cookies("username") = "earthskater"
Response.Cookies("username"). Expires = Date + 365
%>
```

In the example above a cookie named user was created with a value of "earthskater". Then the expiry date was set using the expires property. The cookie was set to expire 1 year from the day's date. When the cookie expires it is deleted from the user's Web computer.

Retrieving Cookies

To retrieve cookies we use request.cookies.

Example Cookie Retrieval

```
<%
username = Request.Cookies("username")
```

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 25/32

		IGHER EDUCATIO	ON, COIMBATORE
Class: III BCOM (CA)	Cours	se Name: INTERNE	T AND WEB DESIGN
Course Code: 16CCU601A	Unit -IV	Semester: VI	Year: 2016-2019 Batch
response.write("Username: " & %>	username)		
This code retrieves the value of cookies can be retrieved anywh anything is written to the body o	the cookie nan ere in the code, of the response.	ned "user name and p unlike creating them	rints it in the browser. Note t which has to be done before
Cookies with Multiple Values			
Cookies can also contain multip	ole values, using	g something called ke	e ys . Take a look at this exam
<%			
Response.Cookies("user")("real	lname") = "Joh	n Doe"	
Response Cookies("user")("user Response Cookies("user")("age	rname") = "joh: ") = "55"	nny55"	
%>) = 33		
This is fairly straightforward to	understand. Re	trieving the cookies i	s very simple too
		the cookies i	is very simple too.
<%		the cookies i	s very simple too.
<% realname = Response.Cookies("	'user")("realnar	ne")	s very simple too.
<% realname = Response.Cookies(" username = Response.Cookies('user")("realnar "user")("userna	ne") me")	s very simple too.
<% realname = Response.Cookies(' username = Response.Cookies(' age = Response.Cookies(''user'')	'user")("realnar "user")("userna)("age")	ne") ume")	s very simple too.
<% realname = Response.Cookies(" username = Response.Cookies(" age = Response.Cookies("user") %>	'user")("realnar "user")("userna)("age")	ne") ume")	s very simple too.
<% realname = Response.Cookies(" username = Response.Cookies(age = Response.Cookies("user") %> Difference between cookies an	'user")("realnar "user")("userna)("age") nd sessions:	me") me")	s very simple too.
<% realname = Response.Cookies(" username = Response.Cookies(age = Response.Cookies("user") %> Difference between cookies an	'user")("realnar "user")("userna)("age") nd sessions:	ne") ime")	s very simple too.
<% realname = Response.Cookies(' username = Response.Cookies(age = Response.Cookies("user") %> Difference between cookies an 1. Cookies can store	'user")("realnar "user")("userna)("age") nd sessions:	ne") ime") 1.Session can	store any data
<% realname = Response.Cookies(" username = Response.Cookies(age = Response.Cookies("user" %> Difference between cookies an 1. Cookies can store "string" data type	'user")("realnar "user")("userna)("age") nd sessions:	ne") me") 1.Session can	store any data
<% realname = Response.Cookies(" username = Response.Cookies(age = Response.Cookies("user" %> Difference between cookies an 1. Cookies can store "string" data type 2. They are stored a	'user")("realnar "user")("userna)("age") nd sessions: e only at Client	ne") me") 1.Session can type 2. These are s	store any data tored at Server
<% realname = Response.Cookies(' username = Response.Cookies() age = Response.Cookies("user") %> Difference between cookies an 1. Cookies can store "string" data type 2. They are stored a side	'user")("realnar "user")("userna)("age") nd sessions: e only at Client	ne") me") 1.Session can type 2. These are s side	store any data tored at Server
<% realname = Response.Cookies(' username = Response.Cookies() age = Response.Cookies("user" %> Difference between cookies an 1. Cookies can store "string" data type 2. They are stored a side 3. Cookie is non-sec	<pre>'user")("realnar "user")("userna)("age") nd sessions: e only at Client cure since</pre>	ne") me") 1.Session can type 2. These are s side 3. Session are	store any data tored at Server secure because
<% realname = Response.Cookies(' username = Response.Cookies() age = Response.Cookies("user") %> Difference between cookies an 1. Cookies can store "string" data type 2. They are stored a side 3. Cookie is non-sec stored in text format	 'user")("realnar "user")("userna)("age") ad sessions: ad sessions: ad sessions: ad client cure since cure since 	ne") me") 1.Session can type 2. These are s side 3. Session are it is stored in b	store any data tored at Server secure because inary format
<% realname = Response.Cookies(' username = Response.Cookies() age = Response.Cookies("user") %> Difference between cookies an 1. Cookies can store "string" data type 2. They are stored a side 3. Cookie is non-sec stored in text format side	 'user")("realnar "user")("userna)("age") ad sessions: ad only at Client cure since cure since cat client 	ne") me") 1.Session can type 2. These are s side 3. Session are it is stored in b	store any data tored at Server secure because inary format
<% realname = Response.Cookies(' username = Response.Cookies() age = Response.Cookies("user" %> Difference between cookies an 1. Cookies can store "string" data type 2. They are stored a side 3. Cookie is non-sec stored in text format side 4. Only in few situati	 'user")("realnar "user")("userna)("age") ad sessions: ad sessions: ad only at Client cure since cure since cat client ions we 	ne") me") 1.Session can type 2. These are s side 3. Session are it is stored in b 4. For all cond	store any data tored at Server secure because inary format ition /situations
<% realname = Response.Cookies('username = Response.Cookies(age = Response.Cookies("user")%> Difference between cookies and 1. Cookies can store "string" data type 2. They are stored a side 3. Cookie is non-secondaria side 4. Only in few situation can use cookies bed	 'user")("realnar "user")("userna)("age") ad sessions: ad sessions: ad only at Client cure since cure since cat client ions we cause 	ne") me") 1.Session can type 2. These are s side 3. Session are it is stored in b 4. For all cond we can use se	store any data tored at Server secure because inary format ition /situations ssions

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 26/32

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE

Unit -IV

Class: III BCOM (CA)

Course Name: INTERNET AND WEB DESIGN

Course Code: 16CCU601A

Semester: VI Year:

Year: 2016-2019 Batch

Cookies Collection:

Property	Description	Access
Domain	Tells the browser only to send the cookie to the specified domain.	Write
	y i	only
Evnires	Tells the browser to delete the cookie at the specified date. If no date is	Write
Explics	specified, the cookie expires when the browser is closed.	only
Hasl	Keys Determines whether the cookie has any keys.	
		Read
		only
Path	Tells the browser only to send the cookie to the specified path.	Write
		only
Secure	If set to true, then the cookie will only be sent to the server when using a secure	Write
becute	connection. This defaults to false.	only

File system objects:

The File System Object is used to access the file system on a server. The File System Object object is used to access the file system on a server. This object can manipulate files, folders, and directory paths. It is also possible to retrieve file system information with this object.



Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 27/32

KARPAGAM ACADEMY OF HIGHER EDUCATION, COIMBATORE

Class: III BCOM (CA)Course Name: INTERNET AND WEB DESIGNCourse Code: 16CCU601AUnit -IVSemester: VIYear: 2016-2019 Batch

Syntax

FileSystemObject.CreateTextFile(filename[,overwrite[,unicode]])

Scripting.FileSystemObject

- The CreateObject function returns the FileSystemObject (fs)
- The CreateTextFile method creates the file as a TextStream object (a).
- The WriteLine method writes a line of text to the created text file.
- The Close method flushes the buffer and closes the file.

Example:

<%

dim fs,fname

```
set fs=Server.CreateObject("Scripting.FileSystemObject")
```

set fname=fs.CreateTextFile("c:\test.txt",true)

fname.WriteLine("Hello World!")

fname.Close

set fname=nothing

set fs=nothing

%>

Property	Description
AtEndOfLine	Returns true if the file pointer is positioned immediately before the end-of-line marker in a TextStream file, and false if not.
AtEndOfStream	Returns true if the file pointer is at the end of a TextStream file, and false if not.
Column	Returns the column number of the current character position in an input stream.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 28/32

	s: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN			
Course Code: 160	CCU601A	Unit -IV	Semester: VI	Year: 2016-2019 Batch
Property	Descript	ion		
Line	Returns t	he current line	number in a TextStre	eam file.
Method	Descripti	on		
Close	Closes ar	n open TextStre	am file.	
Read	Reads a s returns th	Reads a specified number of characters from a TextStream file and returns the result.		
ReadAll	Reads an	Reads an entire TextStream file and returns the result.		
ReadLine	Reads one line from a TextStream file and returns the result.			
Skip	Skips a s file.	Skips a specified number of characters when reading a TextStream file.		
SkipLine	Skips the	Skips the next line when reading a TextStream file.		
Write	Writes a specified text to a TextStream file.			
WriteBlankLines	Writes a	Writes a specified number of new-line characters to a TextStream file.		
WriteLine	Writes a	specified text a	nd a new-line charac	ter to a TextStream file.

The File System Object properties and methods:

Property	Description
Drives	Returns a collection of all Drive objects on the computer

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 29/32

CL UL DCOM (CA)			
Course Code: 16CCU601A	Unit -IV Semester	: VI Year: 2016-2019 Batch	
Methods			
Method	Description		
BuildPath	Appends a name to an exist	ing path	
CopyFile	Copies one or more files from	om one location to another	
CopyFolder	Copies one or more folders	from one location to another	
CreateFolder	Creates a new folder		
CreateTextFile	Creates a text file and return	ns a TextStream object that can be use	
	read from, or write to the fil	le	
DeleteFile	Deletes one or more specific	ed files	
DeleteFolder	Deletes one or more specific	ed folders	
DriveExists	Checks if a specified drive of	exists	
FileExists	Checks if a specified file ex	ists	
FolderExists	Checks if a specified folder	exists	
GetAbsolutePathName	Returns the complete path f	from the root of the drive for the specif	
	path		
GetBaseName	Returns the base name of a	specified file or folder	
GetDrive	Returns a Drive object corre	esponding to the drive in a specified p	
GetDriveName	Returns the drive name of a	specified path	
GetExtensionName	Returns the file extension n	ame for the last component in a specif	

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 30/32

Course Code: 16CCU601A Unit -IV Semester: VI Year: 2016-2019 Batch path path getFile Returns a File object for a specified path Semester: <% dim fs,f set fs=Server.CreateObject("Scripting.FileSystemObject") set fs=ScreateTextFile("c:\test.txt",true) f.WriteLine("Hello World!") f.Close set fs=nothing set fs=nothing %> %> % f.Close %> %> %	Class: III BCOM (CA) Course Name: INTERNET AND WEB DESIGN			
path GetFile Returns a File object for a specified path Example: <% dim fs,f set fs=Server.CreateObject("Scripting.FileSystemObject") set f=fs.CreateTextFile("c:\test.txt",true) f.Close set f=nothing set fs=nothing %>	Course Code: 16CCU601A	Unit -IV	Semester: VI	Year: 2016-2019 Batch
GetFile Returns a File object for a specified path		path		
Example: <% dim fs,f set fs=Server.CreateObject("Scripting.FileSystemObject") set f=fs.CreateTextFile("c:\test.txt",true) f.WriteLine("Hello World!") f.Close set f=nothing set fs=nothing %>	GetFile	Returns a File	e object for a specifie	d path
<% dim fs,f set fs=Server.CreateObject("Scripting.FileSystemObject") set f=fs.CreateTextFile("c:\test.txt",true) f.WriteLine("Hello World!") f.Close set f=nothing set fs=nothing %>	Example:			
dim fs,f set fs=Server.CreateObject("Scripting.FileSystemObject") set f=fs.CreateTextFile("c:\test.txt",true) f.WriteLine("Hello World!") f.Close set fs=nothing set fs=nothing %>	<%			
<pre>set fs=Server.CreateObject("Scripting.FileSystemObject") set f=fs.CreateTextFile("c:\test.txt",true) f.WriteLine("Hello World!") f.Close set f=nothing set fs=nothing %></pre>	dim fs,f			
set f=fs.CreateTextFile("c:\test.txt",true) f.WriteLine("Hello World!") f.Close set f=nothing set fs=nothing %>	set fs=Server.CreateObject("S	cripting.FileSys	temObject")	
f.WriteLine("Hello World!") f.Close set f=nothing %>	set f=fs.CreateTextFile("c:\tes	t.txt",true)		
f.Close set f=nothing %>	f.WriteLine("Hello World!")			
set f=nothing %>	f.Close			
set fs=nothing %>	set f=nothing			
%>	set fs=nothing			
	%>			

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 31/32

Class: III BCOM (CA)	Cours	se Name: INTERNE	ET AND WEB DESIGN
Course Code: 16CCU601A	Unit -IV	Semester: VI	Year: 2016-2019 Batch
1	POSSIBLE OU	JESTIONS –UNIT-J	IV
]	PART A (1 Ma	urk)	
(Online Examin	ations)	
P	ART B (2 Mar	·ks)	
1. What is ASP?			
2. What is a Scripting Lan	guage?		
3. List the five objects in A	ASP.		
4. Write the syntax for Res	sponse Object.		
5. What are the Methods in	n Session Objec	et?	
6. Compare and contrast the	ne Post & Get M	Iethod.	
7. Differentiate the Client-	side Script and	Server-side Script	
8. Define a cookie.			
9. How to create session tr	acking in ASP?	,	
10. What are the types of A	ctivex server co	omponents?	P
11. What is Ad Rotator con	ponent?		
12. Write the syntax for file	system object.		
	P	art-C (6 Marks)	
. Explain the Active server pa	age with an exa	mple.	
2. Explain the working princip	ole of ASP.		
3. Describe the various types of	of object in ASI	2.	

- 4. Discuss the benefits of Server-side scripts.
- 5. How does the cookie create and retrieve with an example?
- 6. List the Activex Server Components and explain with anexample.
- 7. Elucidate the session start and end with an example?
- 8. Expound the different methods and properties of the file system object.

Prepared by Shamile B, Assistant Professor, Department of Commerce(Computer Application), KAHE. Page 32/32



KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University) (Established Under Section 3 of UGC Act 1956) Pollachi Main Road, Eachanari Post, Coimbatore - 641021 (For the candidates admitted from 2016 onwards)

COURSE NAME: INTERNET AND WEB DESIGN COURSE CODE :16CCU601A

UNIT: IV (ACTIVE SERVER PAGE)

S.NO	QUESTIONS	OPTION 1	OPTION 2	OPTION 3	OPTION 4	ANSWER
1	The ASP stands for	Active Server Pages	Activex Server Pages	Active Source Pages	Active Script Pager	Active Server Pages
2	The ASP is a program that runs inside an	ISI	SII	IIS	ISS	IIS
3	The ASP property is used to identify an	Cookies	The Application object	The Server object	ASP Name	Cookies
4	The ASP come with astandard component.	AdRotator	Advertise	Rotator Ads	Advertisement	AdRotator
5	The command is used to retrieve the information in ASP.	Response.Cookies	Request.SetCookies	Request.SetCookies	Request.Cookies	Request.Cookies
6	The represent a hierarchical organization chart.	Object model	Website	Dynamic object model	Web page	Object model
7	A buffer in ASP can be cleared by method.	Clear	Clean	Buffer	Buffer clear	Clear
8	A common object model is a standard communication mechanism between the	Components	Protocols	Scripts	Cookies	Components
9	The text-based files are combination of and	Html tags and ASP Scripts	Html tags and Dhtml	XML and DHTML	XML and ASP	Html tags and ASP Scripts
10	The application logic processed by the server and the component is used	Variables	Constants	Operation	Keywords	Variables
11	The base directory of ASP application is referred to as a	Virtual root	Physical root	Root node	Directory	Virtual root
12	The ASP programming is the processing of script.	Sever-side	Client-side	Server –client-side	Object	Sever-side
13	An ASP script the delimiter is	<% %>	!	<@ @>	<\$ \$>	<% %>
14	A is used to track and manage individual user session in an asp application.	Session object	Application object	Request object	Response object	Session object
15	The Session object hasevents.	2	3	4	1	2
16	The fundamental Active Server Page consists of objects.	6	5	4	2	6
17	A is used to manage all the information in ASP application.	Application object	Session object	Server Object	Response object	Application object
18	A control transaction processing using the MTS is	Context Object	Server object	Request object	Response object	Context Object

19	The property is used to name a web control.	Control name	Designation	Id	Name	Id
20	Theis responsible for sending output from the server to the requesting client.	Request object	Response object	Server object	Client object	Response object
21	The most common response object method is	Write	Retrieve	Request	Scripts	Write
22	The object enrich the value on the client's browser is	Cookies	Variables	Scripts	Functions	Cookies
23	In ASP object have properties, methods, events and	Collections	Containers	Counters	Classes	Collections
24	Auser action will not generate a server-side event.	Mouse move	Text change	Button click	Button	Mouse move
25	Theis used to retrieve information from a client browser.	Request object	Response object	Object context object	Session object	Request object
26	Afile extension is used for an ASP.	asn	asx	asp	aspx	asp
27	The page is refreshed in AdRotator component display a	New banner image	Old banner image	Constant	Current banner image	New banner image
28	Thecommand is used to create cookies.	Response.Cookies	Request.SetCookies	Request.Cookies	Response.SetCookies	Response.Cookies
29	The destroy all the stored object in ASP.	Session Object	Cookies	Script	Activex	Session Object
30	Theis a method of server object.	MapPath	AddHeader	BinaryWrite	Clear	MapPath
31	An ASP can access a database through theobject.	Active data	Data Active	Database	Data	Active data
32	The script executed by the browser.	Server-side	Client-side script	Web server	Server page	Client-side
33	Thescript executed by the web server.	Web server	Web page	client-side	server-side	server-side
34	Theis a Asp Scripting Object.	Server	Text Stream	Client	Application	Text Stream
35	Themethod generate a form collection.	Put	Get	Post	Set	Post
36	Thecommand is used to retrieve cookies value.	Response.Cookies	Response.SetCookies	Request.Cookies	Request.setCookies	Request.Cookies
37		File system	Application	Session	Cookies	File system
	Theobject is used to access the file on a server.					
38	Themethod append a name to an existing path in a file system.	Copyfile	Createfolder	Driveexists	Buildpath	Buildpath
39	Themethod return a string that contains the drive name of the specified path.	Getdrivename	Getname	Getextensionname	Getfile	Getdrivename

40	Ais a conversation between the server and the client.	File system	Session tracking	Web server	Web browser	Session tracking
41	The Session Tracking information is stored in	Application object	Server	Scripts	Cookies	Cookies
42	The EOF stands for	End of file	End of format	Entering of file	Exit of file	End of file
43	The IIS stands for	Internet interchange services	Information internet services	Internet information services	Interchange information services	Internet information services
44	Theproperty is a unique identifier that is generated by the server.	Session	Cookies	Activex component	Objects	Session
45	The major object use with ADO is	Connection	Component	Session	Object	Record set
46	Thecomponent is the collection of object that enable ASP developer to connect a database.	Record set	ADO	DAO	Data	ADO
47	An error collection belongs toobject.	Help file	String	Source	Connection	Connection
48	Themethod send information by appending it to the URL.	Post	Set	Put	Get	Get
49	Amethod is used to send HTTP output to the browser.	Write	Redirect	Response	Request	Write
50	Acomponent display and rotate different advertisement on the web pages.	Ad Rotator	Advertisement	Activex	DAO	Ad Rotator
51	The property is used to set the permission for a connection.	Mode	Connection string	Connection Time out	Connection method	Mode
52	Atext file that manages a frequency of the displayed banner.	Properties	Rotation schedule	Redirection	Rotator	Redirection
53	The component is used to create a record in ASP.	ADO	DAO	RDO	DOA	ADO
54	In Ad Rotator Component required files are	Rotation Schedule	Redirection	Rotation Schedule and Redirection	Rotator	Rotation Schedule and Redirection
55	In AdRotator Component the page is refreshed it displays a	New banner image	Old banner image	Constant	Banner image	New banner image
56	Amethod is used to controls the content.	Response. Write	Response. Redirect	Response.ContentType	Response. Cookies	Response.ContentType
57	Amethod allows accessing the content stored on the web server.	Text stream	File stream	Object stream	Image stream	Text stream
58	Atag is used to submit information back to the server.	From	Form	Text	Label	Form
59	The GET method has a data limit.	3kb	4kb	6kb	2kb	2kb
60	Acontain all the variables established for a session without using the tag.	Static collection	Content collection	Dynamic collection	Static collection and Content collection	Content collection



KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University) (Established Under Section 3 of UGC Act 1956) Pollachi Main Road, Eachanari Post, Coimbatore - 641021 (For the candidates admitted from 2016 onwards)

COURSE NAME: INTERNET AND WEB DESIGN

COURSE CODE :16CCU601A UNIT: V (XML) **OPTION 1 OPTION 4** S.NO QUESTIONS **OPTION 2 OPTION 3** The XML stands for Extended Extensible X-Mark 1 Extended Meaningful Markup Markup Language Language language Language 2 XSL XML The XML files end with the extension Txt Bmp 3 The XML tabs are____ Case sensitive Case insensitive Easy Difficult block can be to embed a CSS style sheet within an XML <Bk:Book> <htps://www.endowenstrates.com/states/states.com/states <XML:STYL An <Element Type> 4 document. 5 The XML document must contain root element. Exactly one Infinite number Optionally one Two Which of the following is an error in XML? Nesting of elements Not declaring the version of XML files not ending Creating more 6 XML with.xml root element Extension 7 The XML is a _____ ___language. Programming Formatting Meta Processing 8 Output Write Display Print The method of the Xml Document. 9 XML-Parser DTD file Schema XSL The ______is required to parse the XML Document. 10 What is an XML namespace? Pertaining to a particular Specific XML Specific spaces Specific XML vocabulary vocabulary within an XML document. CSS _is used to manipulate the contents of the XML. XSL DTD DOM 11 A Schema allows_ _model. Top-down approach Bottom-up approach Closed model Procedure mo 12 In XPath the XML document is conceptually viewed as a _____. Text file Method Event Tree 13 14 EBCDIC ASCII The XML document may contain____character. Unicode BCD

SEMES	TER: VI
	ANSWERS
	Extensible Markup Language
	XML
	Case sensitive
E>	<html:style></html:style>
	Exactly one
than one	Creating more than one root element
	Meta
	Write
	XML-Parser
. parser	Specific XML vocabulary
	DOM
del	Bottom-up approach
	Tree
	Unicode

15	An XPath is abased language.	Object	Event	Structure	String	String
16	An XLink element that specifies the linking information is called	Hyperlinks	Linking elements	Inline links	X-links	Linking elements
17	An XLink attributeis to specify the resource should be retrieved.	Show	View	Retrieve	Actuate	Actuate
18	The XLink attribute is assigned a value locator forresources.	Local	Extended	Simple	Remote	Remote
19	The XML Element is enclosed within	<>	()	{ }	[]	<>
20	Thecontains a starting tag and an ending tag.	Element	Attribute	Entity	Character	Element
21	Ais used to reference fragments of a document.	Xlink	Xpath	Xpointer	Xbase	Xpointer
22	The cascading style sheet can bemethod.	External, Inline	Internal, External	Internal, External, Inline	External only	Internal, External, Inline
23	Ais used to check XML for syntax errors.	XML Validator	XML parser	XML Browser	XML Webpage	XML Validator
24	Theis a term used about text data in the XML parser.	CDATA	PCDATA	EDATA	CPDATA	PCDATA
25	A comment in XML lies within	<>		/* */		
26	An XML Document can be viewed in	IE 3.0	IE 2.0	IE 6.0	IE X.0	IE 6.0
27	A naming collision can be resolved by using	Markup text	CDATA Section	Built -in entities	Namespaces	Namespaces
28	TheKeyword is used to create namespace prefixes.	XML	Msxml	Xmlns	Xns	Xmlns
29	The XSL object uses to hold the content of the body is	List-block	List-item-body	L ist-item	List-item-label	List-item-body
30						
31	The ASL object which formats the data is The is a security technology in XML.	Public Key Cryptography	Encryption	Hashing Technology	XML Signature	XML Signature
32	Anis a basic building block of an XML document.	Element	Attribute	Tag	Text	Element
33	In XMLmethod is used to store data.	Verify	Transport	Design	Process	Transport

34	In XMLis a syntax errors.	Schema	Validator	Browser	Parser	Validator
35	An entity in XML is a	Class	Logical information	Simple information	The flexible information storage unit	The flexible information storage unit
36	In XML schema has single precision offloating point.	16 bit	32 bit	8 bit	4 bit	8 bit
37	An attribute value must be quoted with	Double quotes	Single quotes	Double quotes and Single quotes	Name of attributes	Double quotes
38	An XML Schemas consist of	Properties and methods	Elements and attributes	Structure and data	Tables and relationships	Elements and attributes
39	An XML component that defines the structure of a document is known as a	HTML Stylesheet	DOCTYPE	DTD	#PCDATA	DTD
40		Type-invalid	Type-valid	Not-type-valid	HTML document	Type-valid
	An XML data instance conforms to the DTD, the document is said to be					
41	The XML Schemas can be eliminate	Intersection table	Global elements	Normalized table	Parameters	Global elements
42	A character is used to declare	Elements	Parameter entity	Attribute defaults	Enumeration	Parameter entity
43	The conditional sections of DTDs are often used with	Elements	Entities	Attributes	Parameters	Entities
44	The materialize XML documents is to use	DTD	XSLT	HTML	SOAP	XSLT
45	In XML schema Boolean type holds	True, False	1,0	True ,false and 1,0	True,False and any number except 0	True ,false and 1,0
46	The DTD is defined outside the XML document it is called	Public file	External subset	Internal subset	System file	External subset
47	In Document type definitionkeyword used in XML.	External	Doctype	System	Subset	System
48	An XML is astate method.	Static	Dynamic	Static and Dynamic	Partial static and Dynamic	Dynamic
49	The is a subset of SGML.	XML	HTML	XSL	SGML	XML

50	In XML empty tags are close withslash.	Backslash	Forward slash	Backslash and Forward slash	Black slash	Forward slash
51	In method is used to the XML document into the memory.	Get()	Post()	Load()	Parse()	Load()
52	The XSL stands for	Extensible standard Language	Extensible Stylesheet Language	Extended Simple Language	Extended Standard Language	Extensible Stylesheet Language
53	An XML document begins at a node.	Root	Context	Location path	Expanded name	Context
54	The element frequency is specified by using the	Element type indicator	Occurrence indicator	Generic identifier	Attributes elements	Occurrence indicator
55	Thetag is used in XML.	Predefined	Custom	Defined	Predefined and custom	Custom
56	A schema document use syntax.	EBNF	XML	Java	DTD	XML
57	A incorporate the element of CSS.	XSL	XML	DSL	Xlink	XSL
58	Amethod is to replace the structure.	CSS	XSL	DOM	Schemas	Schemas
59	Ais the root element of every XML Schema documents.	<xml></xml>	<root></root>	<schema></schema>	<xsl></xsl>	<schema></schema>
60	In XML Schemadefines an element.	Element	ElementType	Element	Xml:Element	ElementType

Register No.:

[16CCU601A]

KARPAGAM ACADEMY OF HIGHER EDUCATION (Deemed to be University) (Established Under Sec. 3 of UGC Act, 1956) COIMBATORE – 641 021 (For the Candidates admitted from 2016 onwards)

FIRST INTERNAL EXAMINATION – DECEMBER 2018

III B. Com. (CA)

SIXTH SEMESTER

INTERNET AND WEB DESIGNING

Time Date a	: 2 Hours and Session : 17.12.201	18 and FN PART – A (2 ANSWER AI	20X 1 = 20 M L THE OUF	ARKS)	Max. Mark: 50
1. HTI	ML stands for	·			
	a) Hypertext Markup	Language	b) Hy	perlink Marku	p Language
	c) Hypertext Markedu	up Language	d) Hy	pertext Marku	p Language
2	are used to ma	arkup HTML el	ements.		
3. A_	a) HTML tags element is u	b) HR tags sed to define st	c) Head tags yle informatic	d) SC	GML tags
	a) Title	b) Set	c) Sty	le	d) Character
4. Ho	w many levels of heading	ng elements are	in HTML?		
5	a) 6 tag used to brea	b) 3 Ik up sections o	c) 7 f a document.		d) 8
6	a) Line break tag set a color fo	b) Hori r the back-grou	zontal line nd of the pag	c) Preserve e.	d) Heading
7. Ho	a) Bg color w many categories are i	b) Colo n core attribute	or c) Br s?	color	d) text
8	a) 3 tag is used to	b) 4 o create an unor	c) 5 dered list.		d) 6
	a) 	b) <ol< td=""><td><i>`></i></td><td>c) </td><td>d) <dt></dt></td></ol<>	<i>`></i>	c) 	d) <dt></dt>

9. Unordered list is used for the purp	pose of	<u> </u> .		
a) Symbols and bullets		b) Al	lphabet and	numeric
c) Numeric and symbols		d) He	eading and	symbols
10. <dt> tag is used for</dt>	in definition	list.		
a) Style b) Lin	ıks	c) Heading		d) Paragraph
11. HTML supports a list style which	h is called as		<u> </u> .	
a) Definition list	b) Nested list	c) Ur	nordered lis	at d) Ordered list
12. Each item in the list is marked w	/ith	·		
a) Bullets	b) Numeric	c) Ha	ash sign	d) Pound
13. <pre> tag is used for</pre>				
a) Font color	b) Alignment	c) Fo	ormatting	d) Links
14. URL is abbreviated as				
a) Uniform Resource Location	on	b) Uı	niform Reso	ource locator
c) Unified Resource Locator		d) Ui	niform Rese	earch Locator
15tag used to create a curr	ent position in	the HTML do	ocument.	
a) b) <hr< td=""><td>></td><td>c) </td><td></td><td>d) <pre></pre></td></hr<>	>	c) 		d) <pre></pre>
16. HTML is known as				
a) Platform dependent		b) Platform	n independe	ent
c) Place dependent		d) Place In	Idependent	
17. The Internet was originally	project a	gency.		
a) NSF b) NSA	4	c) APRA		d) ARPA
18. HTML is used to create	<u> </u> .			
a) Machine language program	m b) High leve	el program	c) Web pag	ge d) Web server
19. Internet explorer falls under	•			
a) Operating System	b) Compiler	c) Br	owser	d) IP address
20. XML stands for				
a) World Wide Web		b) World Wi	ide Word	
c) World Wide Wood		d) World Wi	ide Width	

PART - B (3 X 2 = 6 MARKS) (ANSWER ALL THE QUESTIONS)

21. List the features of internet explorer.

22. What is HTML? Write the format of HTML program.

23. What is the use of <form> tag?

PART - C (3 X 8 = 24 MARKS) (ANSWER ALL THE QUESTIONS)

24a.What is HTML tags? Describe the various HTML tags in detail.

(**OR**)

b. How Special characters are represented in HTML?

25a. Design a homepage for your college website using HTML.

(**OR**)

b. Describe the various types of intermediate HTML tables.

26a. Explain in detail about the Meta tag in HTML.

(**OR**)

b. What is frameset? Explain the nested frameset with an example.

* * * * *

Reg..... [16CCU601A]

KARPAGAM ACADEMY OF HIGHER EDUCATION (Deemed to be University) (Established Under Section 3 of UGC Act, 1956) Coimbatore-641021 (For candidates admitted from 2016 onwards) CONTINUOUS INTERNAL ASSESSMENT II – FEBRUARY, 2019 BACHELOR OF COMMERCE (COMPUTER APPLICATION) SIXTH SEMESTER INTERNET AND WEB DESIGNING

Time: 2 Hours Date: 04.02.2019 Maximum: 50 Marks Session : FN

PART – A (20X 1 = 20 MARKS) ANSWER ALL THE QUESTIONS

- DHTML stands for _____.
 a. Domain Hyper Text Markup Language
 - b. Dual Hyper Text Markup Language
 - c. Dynamic Hyper Text Markup Language
 - d. Direct Hyper Text Markup Language
- 2. The _____ CSS selector is used to define style for an unique element. a.id b. text c. name d. class
- 3. The _____ CSS property is used to not allow floating point to DIV. a. margin b. clear c. float d. padding
- 4. Suppose we want to arrange five numbers of divs so that div4 is placed above div1. Now, which property of css will we use to control the order of stack?
 a. d-index
 b. s-index
 c. x-index
 d. z-index
- 5. If we want to wrap a block of text around an image, which css property will we use? a. wrap b. push c. float d. align
- 6. ______ value is used to show an arrow as a cursor.a. Pointerb. Defaultc. Arrowd. Arr
- 7. The _____ CSS Property is used to give border around an image.
 a. border-color
 b. border-decoration
 c. border-style
 d. border-line
- 8. The _____ property is used to display border around a cell without any content. a. empty-cell b. blank-cell c. noncontent-cell d. void-cell
- 9.HTML attribute is used to define inline styles.a. Classb. CSSc. Styled. Font

10. The C	SS property controls	the text size.	
a. text-style	b. font-size	c. text-size	d. font-style
11 JavaScript is a	language		
a. Scripting	b. Application	c. Programming	d. compiling
1 0			I U
12. JavaScript is a _	side scriptin	ng language.	1 100
a. server	b. client	c. browser	d. ISP
13. JavaScript is des	signed to		
a. style HTML j	page	b. add interactivity	to HTML page
c. perform serve	r side scripting	d. execute query re	elated to DB on server
14. JavaScript can b	e written .		
a. directly into H	ITML page	b. directly on the s	erver script
c. directly into J	S file	d. included separat	tely to HTML
15. JavaScript code	is placed inside	HTML element.	
a. <scripting></scripting>	b. <js></js>	c. <script></script>	

(ANSWER ALL THE QUESTIONS)

24. a. Demonstrate the steps in creating an Inline Style Sheet with an example.
(Or) b. Enumerate Linking External Style Sheets with an example.

25. a. Explain with sample code i) Text flow ii) Box Model

(Or)

b. Explain various operators in JavaScript with an example.

26. a. Classify various control statements in JavaScript with an example. (Or)
b. Elucidate the usage of i) Functions with sample code

ii) Arrays with sample code

* * * * *

Reg..... 16CCU601A

KARPAGAM ACADEMY OF HIGHER EDUCATION (Deemed to be University) (Established Under Section 3 of UGC Act, 1956) Coimbatore-641021 (For candidates admitted from 2017 onwards) CONTINUOUS INTERNAL ASSESSMENT II BACHELOR OF COMMERCE (COMPUTER APPLICATION) SIXTH SEMESTER INTERNET AND WEB DESIGNING					
Time: 2 Hours		Maximum: 50 Marks			
Date: 11.03.2019 Session : FN PART – A (20 X 1 = 20 Marks) ANSWER ALL THE QUESTIONS					
 The XML stands for a) Extended Markup language c) Extended Meaningful Language 		b) Extensible Markup Language d) X-Mark Language			
2. The XML files end with the extension					
a) xml	b) xsl	c) txt		d) bookmark	
 3. A block can be a) <bk:book></bk:book> c) <html:style></html:style> 4. The XML document must c a) infinite number c) exponent number 	to embed a CSS ontain 1	style sheet within a b) <element type<br="">d) <xml:style root element. b) Exactly one d) Two</xml:style </element>	an XML d >> >	ocument.	
5. The ASP stands fora) Active Server Pagesc) Active Source Page		b) ActiveX Server d) Active Script P	: Page age		
6. The ASP is a program that a) ISS	runs inside an b) ISI	c) SII		d) ISA	
7. The ASP property is used to a) cookie	o identify a b) application	c) server		d) browser	
8. The ASP come with a a) AdRotator	b) Advertise	ard component. c) Rotator		d) AdsRotator	
9. The command is used to retrieve the information in ASP.a) Response.Cookiesb) Request.SetCookiesc) Request.SetCookiesd) Request.Cookies					
10. The represent a hierarchical organization chart.					
a) Object model	b) Webs	ite c) bro	owser	d) client	

11. A buffer in ASP can be cleared bymethoa) clearb) browser clean	d. c) clean d) del	ete			
12. A common object model is a standard commun a) Components b) Protocols	nication mechanism be c) Scripts	tween the d) Cookies			
 13. Which of the following is an error in XML? a) Nesting of elements c) XML files not ending with.xml 14. The XML is a language. 	b) Not declaring the version of XMLd) Creating more than one root element				
a) Programming b) scripting	c) formatting	d) meta			
a) write b) output	c) display	d) print			
16 is used to manipulate the contents of a) CSS b) DOC	of the XML. c) DOM	d) FTP			
17. The text-based files are combination of a) Html tags and ASP Scripts c) XML and DHTMLb) Html tags and Dhtml d) XML and ASP					
18. The application logic processed by the server aa) Variableb) Opponent	and the component is us c) constant	sed as a d) keyword			
19. The base directory of ASP application is referrance a) virtual rootb) physical root	red to as a c) root directory	d) node			
20. The ASP programming is the processing of a) Server b) Client	c) client-server	d) browser			
PART – B (3 X 2 = 6 Marks) ANSWER ALL THE QUESTIONS					
21. Write the syntax of a XML document?					
22. What is meant by Cookies?					
23. What is meant by session tracking?					
PART – C (3 X 8 = 24 Marks) ANSWER ALL THE QUESTIONS					
24. a. Describe the functioning of ASP in developing a website.					
(OR) b. Write a program to develop a website for an international school.					
25. a. Give detailed note on Document type definition in XML. (OR)					
b. Explicate about an XML namespace?					
26. a. Describe file system objects with sample program.					

(OR)

b. Describe about Client Side Scripting and Server Side Scripting.