CCU403B **HTML PROGRAMMING** Semester – IV

Hours/week: L: 2 T: 0 P: 0 2H – 2C

Marks: Internal: 40 External: 60 Total: 100

End Semester Exam: 3 Hours

COURSE OBJECTIVES:

To make the students

1. To understand the concept of HTML
2. To familiarize on the HTML CSS.
3. To comprehend on the JAVASCRIPTS operators and functions
4. To understand the application of HTML Forms
5. To familiarize on the creation of HTML image Maps

COURSE OUTCOMES:

Learners should be able to

1. Familiarize on the HTML sample documents and Its platformand apply the learning for lifelong.
2. Use the HTML CSS FilesOperators, arrays and functions
3. Understand usage of HTML forms and Create HTML image maps
4. Critically analyse the need and create the HTML functions required for the situation.
5. Write the program and present orally and in written form.

UNIT I: INTRODUCTION TO HTML

Basics – Sample HTML document - Elements – Attributes – Headings – Paragraphs – Styles – Formatting – Quotations – Comments – Colors.

UNIT II : HTML CSS

Format of a CSS file – Usage of CSS Files – Links – Images – Tables – Lists – Blocks – Classes – HTML ID – Frames.

UNIT III : JAVASCRIPT:

Introduction - 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 4: HTML FORMS

Form Elements – Input Type – Text Area – Button – List box- Check Box - Input Attributes.

UNIT 5: HTML IMAGE MAPS

Creating and using image maps.HTML Graphics – Canvas – SVG. HTML Media – Video – Audio – Plug-ins.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University) (Established under section 3 of UGC Act 1956)
Coimbatore-641021

DEPARTMENT OF COMMERCE (CA)

Name: **C.Jothish & R.Lydia Priyadharsini**
Department: **Commerce(CA)**
Subject Code: **18CCU403 B** Semester: **IV**
Subject: **HTML PROGRAMMING** Year: **2018-21Batch**

UNIT -1

S.NO	LECTURE DURATION	TOPICS	SUPPORT MATERIALS
1.	1	Introduction to HTML,Basics,Sample HTML documents ,Elements,Attributes	T2
2.	1	Headinds,Paragraghs,Styles	T2
3.	1	Formatting,Quotations,Comments,Colors	T2
4.	1	Recapitulation and Discussion of Important Questions	
		Total Number of Hours planned for Unit I	**4**

UNIT -2

S.NO	LECTURE DURATION	TOPICS	SUPPORT MATERIALS
1.	1	Format of a CSS file,Usage of CSS file,Links,Images,Tables	T5
2.	1	Lists,Blocls,Classes,ID,Frames	T5
3.	1	Recapitulation and Discussion of Important Questions	
		Total Number of Hours planned for Unit I	**3**

UNIT -3

S.NO	LECTURE DURATION	TOPICS	SUPPORT MATERIALS
1.	1	JavaScript :Introduction,Operators,Arithmetic & Relational Operators,Precedence of Operators	T2
2.	1	Control structures,Assignment,Increment,Decrement Operators	W1
3.	1	Forloops,Switch,Do While,Break	W1
4.	1	Continue,Array,Functions	W1
5.	1	Recapitulation and Discussion of Important Questions	
		Total Number of Hours planned for Unit I	**5**

UNIT -4

S.NO	LECTURE DURATION	TOPICS	SUPPORT MATERIALS
1.	1	HTML Forms :Form elements,Input Type –Text area,Button,List Box	W1
2.	1	Check Box ,Input attributes	W1
3.	1	Recapitulation and Discussion of Important Questions	
		Total Number of Hours planned for Unit I	**3**

Unit - 5

S.NO	LECTURE DURATION	TOPICS	SUPPORT MATERIALS
1.	1	HTML Image Maps,HTML Graphics,Canvas,SVG	W1
2.	1	HTML Media,Video,Audio,Plug Ins	W2
3.	1	Recapitulation and Discussion of Important Questions	
4.	2	Discussion Of Previous year ESE Question Papers	
		Total Number of Hours planned for Unit I	**5**

UNIT 1

INTRODUCTION TO HTML:

HTML or **Hyper Text Markup language** is the standard markup language of web used to build a website or web application. **HTML** along with CSS and JavaScript are the core parts of **web technologies**. **HTML** is used to create & display content of a website. This content can be text, headings, media, list , tables etc on a web browser.

HTML is a **markup language**, as it contains **markup tags** or **tags**. HTML Tags are used to build structure of a website including paragraphs, headings, tables, lists, buttons, forms, images, audio, video etc.

HTML build the **structure of web pages**. It is also known as **Building Block of a website**. Browsers understand **html tags** and render content. **HTML is Browser Interpreted** language, thus load fast as compilation is not required.

To learn web designing, one should start with HTML first. HTML includes
140+ tags and attributes to build webpage layout. The latest version of html is HTML5, which included HTML, CSS and JavaScript. HTML and CSS are static, while JavaScript is dynamic.

HISTORY OF HTML

HTML was invented by a CERN scientist **Tim Berners-Lee** in 1989. The primary purpose to invent **HTML** was to share information on web for Research Scientists and Engineers so that other professionals in same domain can have access to your research work. Writing Books and Journals are only limited to some people. To run html, a web browser *WorldWideWeb* was developed, but later on it was renamed to *Nexus*.

HTML Versions

HTML was first formed in 1991. Till now, HTML receive many updates. Here is a list of HTML versions with release date.

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

HTML Editors

HTML Editor is the software used to write html code. A web browser is used to view webpage. Some Popular code editors are notepad++, brackets, sublime text , atom, and Visual Studio Code etc. You can use any one of them.

How to create a website using html

To **build a webpage using html**, use following steps. These steps are based on HTML5 web standards. Just follow these simple steps, and your **first html page** is readyOpen any code editor.

- Create the **doctype of webpage**. e.g. <!doctype html>
- Create Parent **html tag** e.g. <html> </html>
- Create **head tag** inside **html tag**, e.g. <head> </head>
- Create **body tag** after head tag closing, e.g. <body> </body>
- Add **title tag** inside head, e.g. <title> </title>
- Add **meta tag** inside head, e.g. <meta charset="utf-8">
- Save page as **index.html** file on your system.
- Double Click the file and your webpage is live on browser.
- To edit webpage, right click on html file and select open in editor.

Example :

A Simple HTML Document

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Example Explained

The <!DOCTYPE html> declaration defines this document to be HTML5

The <html> element is the root element of an HTML page

The <head> element contains meta information about the document

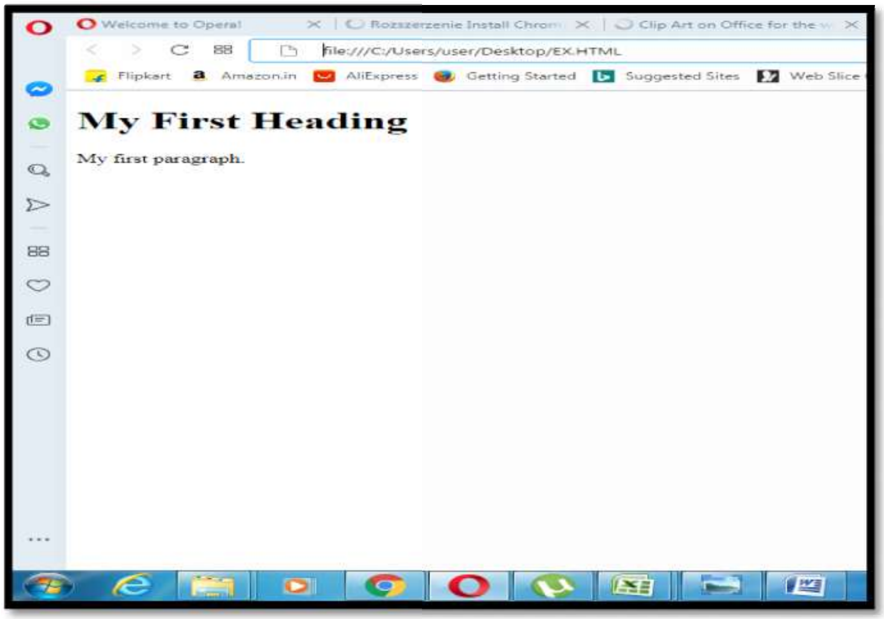
The <title> element specifies a title for the document

The <body> element contains the visible page content

The <h1> element defines a large heading

The <p> element defines a paragraph

OUTPUT :


Welcome to Opera!
Rozszerzenie Install Chrom
Clip Art on Office for the
file:///C:/Users/user/Desktop/EX.HTML
Flipkart
Amazon.in
AliExpress
Getting Started
Suggested Sites
Web Slice

My First Heading

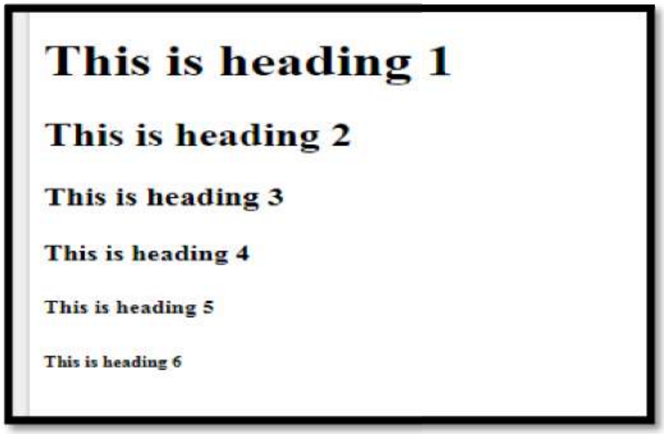
My first paragraph.

HTML Headings

HTML headings are defined with the <h1> to <h6> tags.
<h1> defines the most important heading. <h6> defines the least important heading:
Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Output :



This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

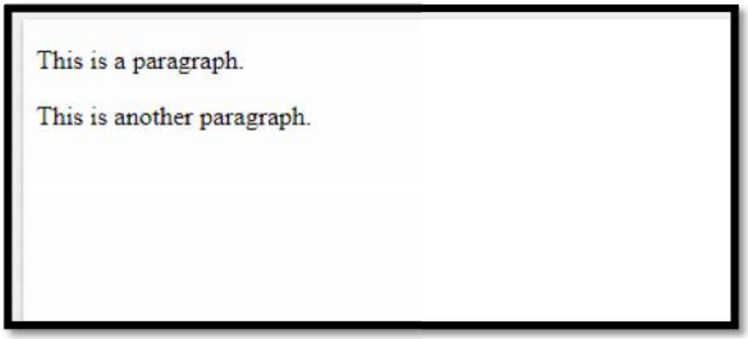
HTML Paragraphs

HTML paragraphs are defined with the <p> tag:

Example

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

Output :



This is a paragraph.

This is another paragraph.

HTML Links

HTML links are defined with the <a> tag:

Example

```
<a href="https://www.w3schools.com">This is a link</a>
```

The link's destination is specified in the href attribute.

Attributes are used to provide additional information about HTML elements.

You will learn more about attributes in a later chapter.

HTML Images

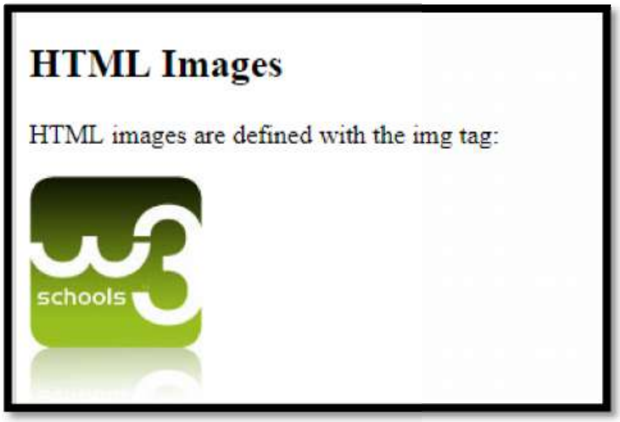
HTML images are defined with the <img> tag.

The source file (src), alternative text (alt), width, and height are provided as attributes:

Example

```
<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">
```

Output:



HTML Images

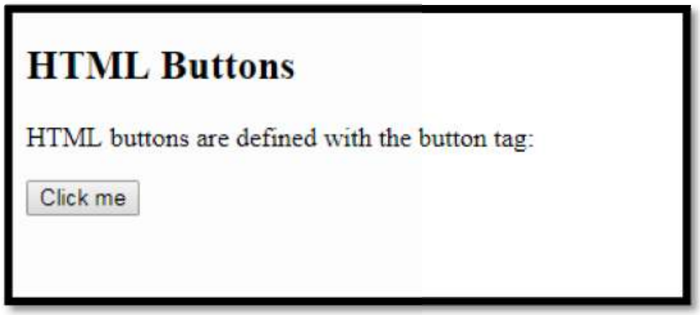
HTML images are defined with the img tag:

HTML Buttons

HTML buttons are defined with the <button> tag:

Example

```
<button>Click me</button>
```



HTML Buttons

HTML buttons are defined with the button tag:

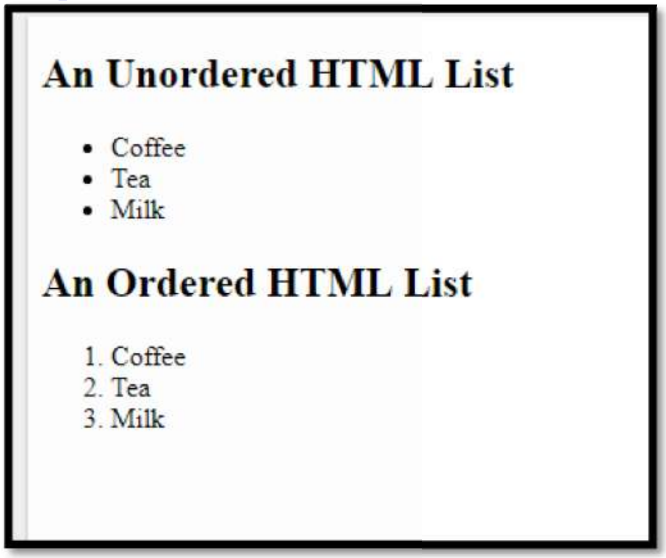
HTML Lists

HTML lists are defined with the <ul> (unordered/bullet list) or the <ol> (ordered/numbered list) tag, followed by <li> tags (list items):

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Output :



An Unordered HTML List

- Coffee
- Tea
- Milk

An Ordered HTML List

1. Coffee
2. Tea
3. Milk

HTML Elements

An HTML element usually consists of a **start** tag and an **end** tag, with the content inserted in between:

```
<tagname>Content goes here...</tagname>
```

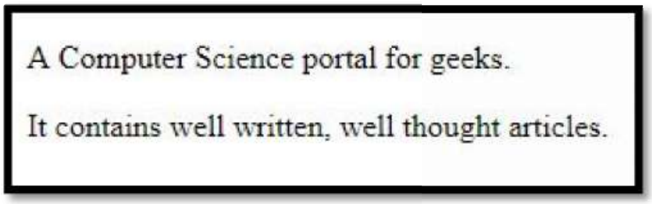
The HTML **element** is everything from the start tag to the end tag:

```
<p>My first paragraph.</p>
```

<p> tag:
The **<p>** tag in HTML defines a paragraph. These have both opening and closing tag. So anything mentioned within **<p>** and **</p>** is treated as a paragraph. Most browsers read a line as a paragraph even if we don't use the closing tag i.e, </p>, but this may raise unexpected results. So, it is both a good convention and we **must** use the closing tag.

Syntax:

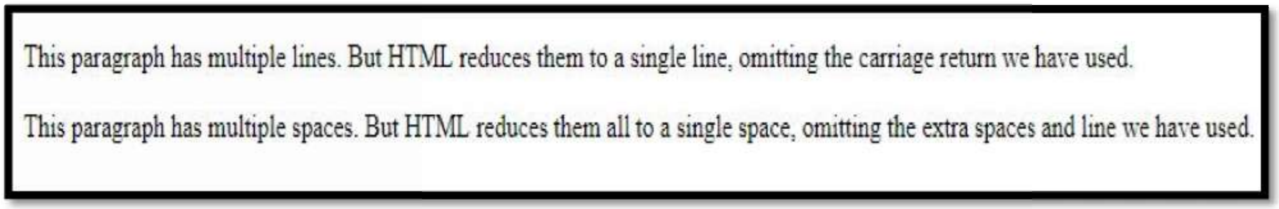
```
 <p> Content </p>
<!DOCTYPE html>
<html>
<head>
        <title>Paragraph</title>
</head>
<body>
        <p>A Computer Science portal for geeks.</p>
        <p>It contains well written, well thought articles.</p>
</body>
</html>
```



A Computer Science portal for geeks.

It contains well written, well thought articles.

```
<!DOCTYPE html>
<html>
<head>
        <title>Display_Paragraph</title>
</head>
<body>
        <p>
        This paragraph has multiple
        lines. But HTML reduces them
        to a single line, omitting
        the carriage return we have used.
        </p>
        <p>
        This paragraph            has multiple
        spaces.  But HTML reduces them
        all to a  single space, omitting
        the extra spaces and line we have used.
        </p>
</body>
</html>
```



This paragraph has multiple lines. But HTML reduces them to a single line, omitting the carriage return we have used.

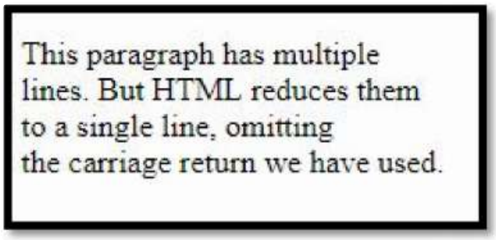
This paragraph has multiple spaces. But HTML reduces them all to a single space, omitting the extra spaces and line we have used.

<br> tag:

There is a way to let the HTML know where does the browser need to change the lines by the use of **<br>** tag. These tags do not have any closing tag. So, just a single opening tag will change the line.

Syntax:

```
 <br>
<!DOCTYPE html>
<html>
<head>
        <title>Display_Paragraph</title>
</head>
<body>
        <p>
        This paragraph has multiple<br>lines.
        But HTML reduces them<br>to a single
        line, omitting<br>the carriage return
        we have used.
        </p>
</body>
</html>
```



This paragraph has multiple
lines. But HTML reduces them
to a single line, omitting
the carriage return we have used.

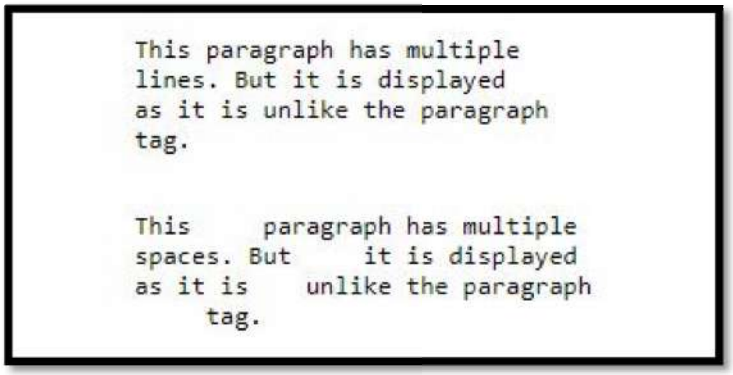
The <pre> element:

We have seen how the paragraph tag ignores all the change of lines and extra spaces within a paragraph, but there is a way to preserve this by the use of **<pre>** tag. It also contains an opening and a closing tag. It displays a text within a fixed height and width and preserves the extra lines and spaces we use.

Syntax:

```
 <pre> Content </pre>
<!DOCTYPE html>
<html>
<head>
        <title>Display_Paragraph</title>
</head>
<body>
```

```
        <pre>
        This paragraph has multiple
        lines. But it is displayed
        as it is unlike the paragraph
        tag.
        </pre>

        <pre>
        This     paragraph has multiple
        spaces. But      it is displayed
        as it is unlike the paragraph
                tag.
        </pre>
</body>
</html>
```



```
This paragraph has multiple
lines. But it is displayed
as it is unlike the paragraph
tag.


This     paragraph has multiple
spaces. But     it is displayed
as it is    unlike the paragraph
     tag.
```

Nested HTML Elements

HTML elements can be nested (elements can contain elements).

All HTML documents consist of nested HTML elements.

This example contains four HTML elements:

Example

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Example Explained

The <html> element defines the **whole document**.

It has a **start** tag <html> and an **end** tag </html>.

Inside the <html> element is the <body> element.

The <body> element defines the **document body**.

It has a **start** tag <body> and an **end** tag </body>.
Inside the <body> element is two other HTML elements: <h1> and <p>.

```
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
```

The <h1> element defines a **heading**.
It has a **start** tag <h1> and an **end** tag </h1>.
The element **content** is: My First Heading.

```
<h1>My First Heading</h1>
```

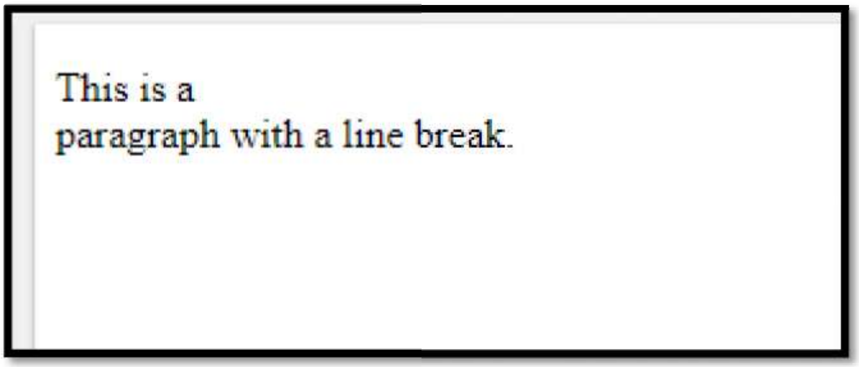
The <p> element defines a **paragraph**.
It has a **start** tag <p> and an **end** tag </p>.
The element **content** is: My first paragraph.

```
<p>My first paragraph.</p>
```

Empty HTML Elements

HTML elements with no content are called empty elements.
<br> is an empty element without a closing tag (the <br> tag defines a line break):
Example

```
<p>This is a <br> paragraph with a line break.</p>
```



This is a
paragraph with a line break.

Empty elements can be "closed" in the opening tag like this: <br />.

HTML5 does not require empty elements to be closed. But if you want stricter validation, or if you need to make your document readable by XML parsers, you must close all HTML elements properly.

HTML Is Not Case Sensitive
HTML tags are not case sensitive: <P> means the same as <p>.
The HTML5 standard does not require lowercase tags, but W3C recommends lowercase in HTML, and demands lowercase for stricter document types like XHTML.

HTML Attributes

All HTML elements can have attributes
Attributes provide additional information about an element
Attributes are always specified in the start tag
Attributes usually come in name/value pairs like: name="value"
HTML Attributes
Below is an alphabetical list of some attributes often used in HTML,

Attribute	Description
alt	Specifies an alternative text for an image, when the image cannot be displayed
disabled	Specifies that an input element should be disabled
href	Specifies the URL (web address) for a link
id	Specifies a unique id for an element
src	Specifies the URL (web address) for an image
style	Specifies an inline CSS style for an element
title	Specifies extra information about an element (displayed as a tool tip)

The href Attribute

HTML links are defined with the <a> tag. The link address is specified in the href attribute:

```
<a href="https://www.w3schools.com">This is a link</a>
```

The src Attribute

HTML images are defined with the <img> tag.
The filename of the image source is specified in the src attribute:

```
<img src="img_girl.jpg">
```

The width and height Attributes

HTML images also have width and height attributes, which specifies the width and height of the image:

```
<img src="img_girl.jpg" width="500" height="600">
```

The width and height are specified in pixels by default; so width="500" means 500 pixels wide.

The alt Attribute
The alt attribute specifies an alternative text to be used, if an image cannot be displayed.

The value of the alt attribute can be read by screen readers. This way, someone "listening" to the webpage, e.g. a vision impaired person, can "hear" the element.

```
<img src="img_book.jpg" alt="HTML Book">
```

The alt attribute is also useful if the image cannot be displayed (e.g. if it does not exist):

The style Attribute

The style attribute is used to specify the styling of an element, like color, font, size etc.

```
<p style="color:red">This is a paragraph.</p>
```

The lang Attribute

The language of the document can be declared in the <html> tag.
The language is declared with the lang attribute.
Declaring a language is important for accessibility applications (screen readers) and search engines:

```
<!DOCTYPE html>
<html lang="en-US">
<body>

...

</body>
</html>
```

The first two letters specify the language (en). If there is a dialect, add two more letters (US).

The title Attribute

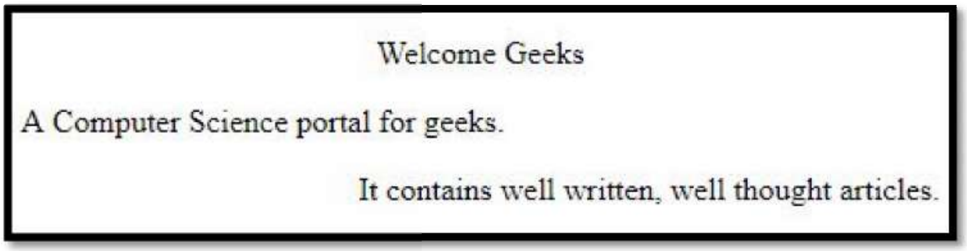
Here, a title attribute is added to the <p> element. The value of the title attribute will be displayed as a tooltip when you mouse over the paragraph:
Example

```
<p title="I'm a tooltip">
This is a paragraph.
</p>
```

The align attribute:
The <p> tag specifically supports the alignment attribute and allows us to align our paragraphs in left, right or center alignment.
Syntax:

```
<p align="value">
<!DOCTYPE html>
<html>
<head>
```

```
        <title>Paragraph</title>
</head>
<body>
        <p align="center">Welcome Geeks</p>
        <p align="left">A Computer Science
                                portal for geeks.</p>
        <p align="right">It contains well
        written, well thought articles.</p>
</body>
</html>
```



Welcome Geeks

A Computer Science portal for geeks.

It contains well written, well thought articles.

HTML Styles

Styles in HTML are basically rules that describe how a document will be presented in a browser. Style information can be either attached as a separate document or embedded in the HTML document.

There are 3 ways of implementing style in HTML :

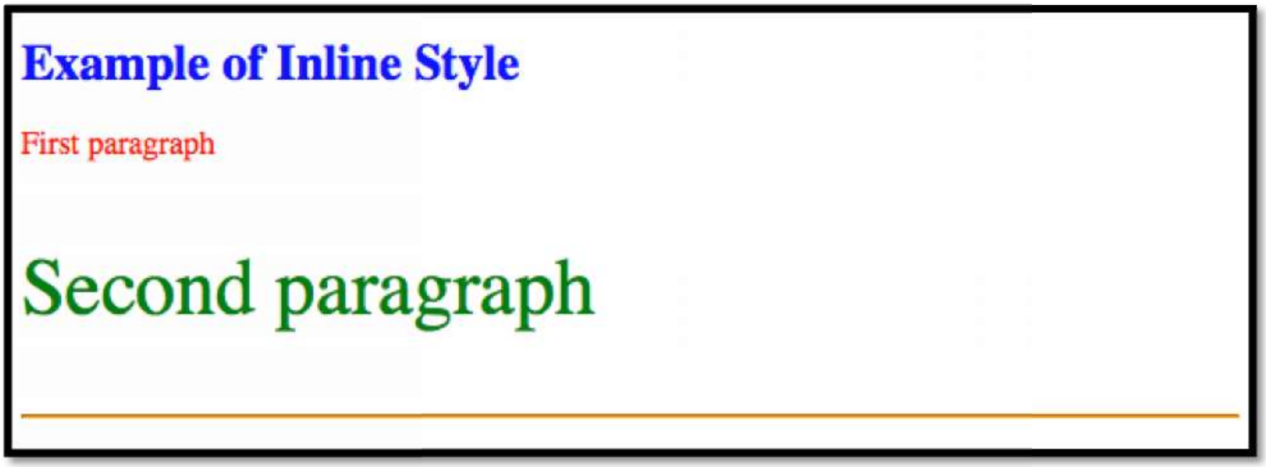
Inline Style : In this method, the style attribute is used inside the HTML start tag.

Embedded Style : In this method, the style element is used inside the <head> element of the document.

External Style Sheet : In this method the <link> element is used to point to an external CSS file.

Inline Style : In Inline styling, the CSS rules are directly written inside the starting tag using the style attribute. The style attribute includes a series of CSS property and value pairs. Each ‘ property : value ‘ pair is separated by a semicolon (;).

```
<!DOCTYPE html>
<html>
<head>

</head>
<body>

        <h1 style="color:Blue;font-size:25px;">
                Example of Inline Style
        </h1>

        <p style="color:red;">First paragraph</p>

        <p style="color:green;font-size:40px;">
                Second paragraph
        </p>
        <hr style="border-color:orange;">

</body>

</html>
```



Example of Inline Style

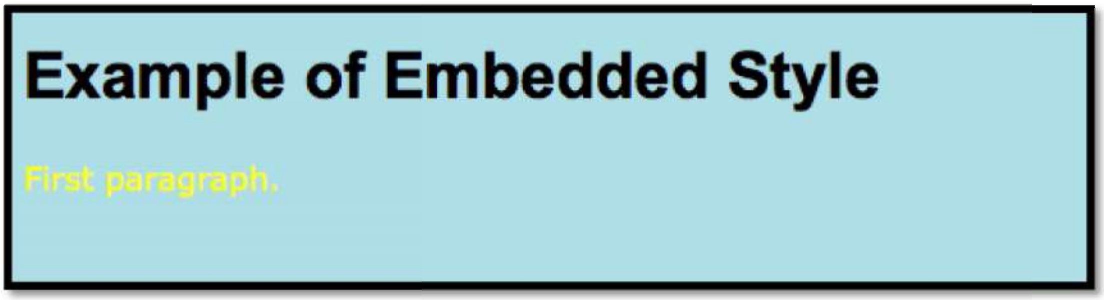
First paragraph

Second paragraph

Embedded Style : Embedded or internal style sheets only affect the document they are embedded in. Embedded style sheets are defined in the <head> section of an HTML document using the <style> tag.

```
<!DOCTYPE html>
<html lang="en">

<head>
        <style type="text/css">
                body {
                        background-color: powderblue;
                }

                h1 {
                        color: black;
                        font-family: arial;
                }

                p {
                        color: yellow;
                        font-family: verdana;
                }
        </style>
</head>

<body>
        <h1>Example of Embedded Style</h1>
        <p>First paragraph.</p>
</body>
```

```
</html>
```



Example of Embedded Style

First paragraph.

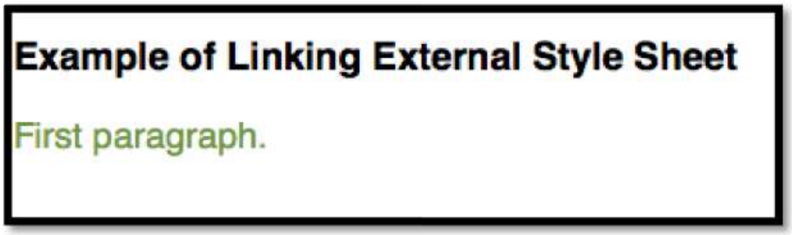
Use the style attribute for styling HTML elements
Use background-color for background color
Use color for text colors
Use font-family for text fonts
Use font-size for text sizes
Use text-align for text alignment

External Style Sheet: External Style Sheets method can be useful when the CSS has to be applied to various web pages. An external style sheet holds all the style rules in a separate document that you can link from an HTML file on your site.

There are two ways of attaching external style sheets –
Linking External Style Sheets
Importing External Style Sheets

Linking External Style Sheets :
In this method, an external style sheet is linked to an HTML document using the <link> tag.

```
<!DOCTYPE html>
<html>
<head>
        <link rel="stylesheet" type="text/css"
                        href="/html/css/externalstyle.css">
</head>
<body>
        <h3>Example of Linking External Style Sheet</h3>
        <p>First paragraph.</p>
</body>
</html>
```



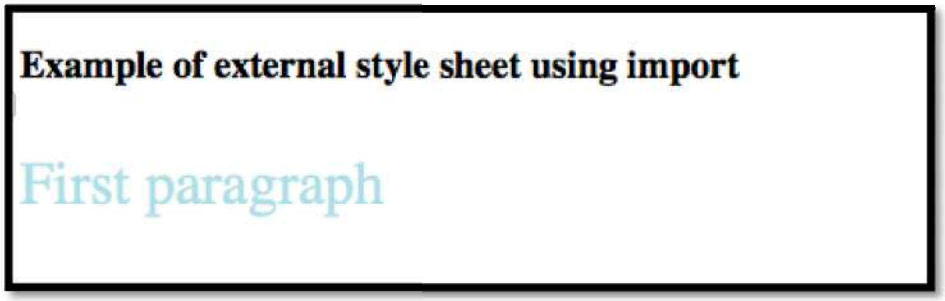
Example of Linking External Style Sheet

First paragraph.

Importing External Style Sheets :
External style sheets can be loaded into an HTML document using “@import”. The “@import” statement instructs the browser to load the CSS file. Other CSS rules can also be included using the <style> element.

```
<!DOCTYPE html>
```

```
<html>
<head>
<style type = "text/css">
        @import url("/html/css/importstyle.css");
p{color:powderblue; font - size : 30px;}
</style>
</head>
<body>
        <h3>Example of external style sheet using import</h3>
        <p>First paragraph</p>
</body>
</html>
```



Example of external style sheet using import

First paragraph

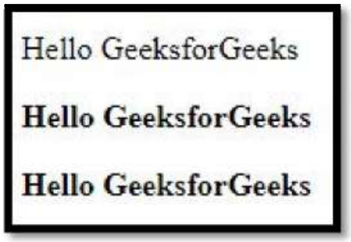
HTML | Text Formatting

HTML provides us with the ability for formatting text just like we do it in MS Word or any text editing software. In this article, we would go through few such options.

Making text Bold or Strong: We can make the text **bold** using the **<b>** tag. The tag uses both opening and closing tag. The text that needs to be made bold must be within **<b>** and **</b>** tag. We can also use the **<strong>** tag to make the text strong, with added semantic importance. It also opens with <strong> and ends with </strong> tag.

Example:

```
<!DOCTYPE html>
<html>
<head>
        <title>Bold</title>
</head>
<body>
        <!--Normal text-->
        <p>Hello GeeksforGeeks</p>
        <!--Text in Bold-->
        <p><b>Hello GeeksforGeeks</b></p>
        <!--Text in Strong-->
        <p><strong>Hello GeeksforGeeks</strong></p>
</body>
</html>
```



Hello GeeksforGeeks

Hello GeeksforGeeks

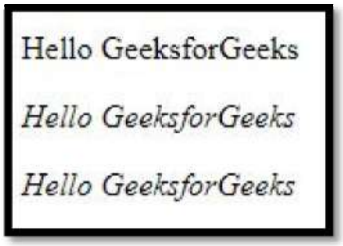
Hello GeeksforGeeks

2. Making text *Italic* or *emphasize*: The **<i>** tag is used to *italicise* the text. It opens with <i> and ends with </i> tag.

The **<em>** tag is used to *emphasize* the text, with added semantic importance. It opens with <em> and ends with </em> tag.

Example:

```
<!DOCTYPE html>
<html>
<head>
        <title>Italic</title>
</head>
<body>
        <!--Normal text-->
        <p>Hello GeeksforGeeks</p>
        <!--Text in Italics-->
        <p><i>Hello GeeksforGeeks</i></p>
        <!--Text in Emphasize-->
        <p><em>Hello GeeksforGeeks</em></p>
</body>
</html>
```



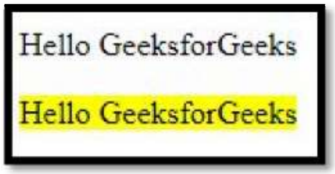
Hello GeeksforGeeks

Hello GeeksforGeeks

Hello GeeksforGeeks

3.Highlighting a text: It is also possible to highlight a text in HTML using the **<mark>** tag. It has a opening tag <mark> and a closing tag </mark>.

```
<!DOCTYPE html>
<html>
<head>
        <title>Highlight</title>
</head>
<body>
        <!--Text in Normal-->
        <p>Hello GeeksforGeeks</p>
        <!--Text in Highlight-->
        <p><mark>Hello GeeksforGeeks</mark></p>
</body>
</html>
```



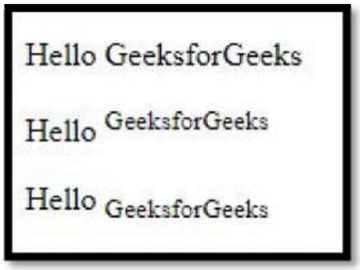
Hello GeeksforGeeks

Hello GeeksforGeeks

4.Making a text Subscript or Superscript: The **<sup>** element is used to superscript a text and <sub> element is used to subscript a text. They both have opening and a closing tag.

Example:

```
<!DOCTYPE html>
<html>
<head>
        <title>Superscript and Subscript</title>
</head>
<body>
        <!--Text in Normal-->
        <p>Hello GeeksforGeeks</p>
        <!--Text in Superscript-->
        <p>Hello <sup>GeeksforGeeks</sup></p>
        <!--Text in Subcript-->
        <p>Hello <sub>GeeksforGeeks</sub></p>
</body>
</html>
```



Hello GeeksforGeeks

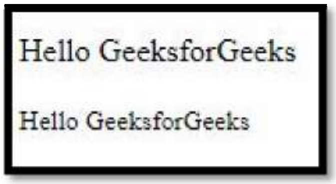
Hello $^{\text{GeeksforGeeks}}$

Hello $_{\text{GeeksforGeeks}}$

5. **Making text smaller:** The **<small>** element is used to make the text smaller. The text that needs to be displayed smaller should be written inside <small> and </small> tag.

Example:

```
<!DOCTYPE html>
<html>
<head>
        <title>Small</title>
</head>
<body>
        <!--Text in Normal-->
        <p>Hello GeeksforGeeks</p>
        <!--Text in small-->
        <p><small>Hello GeeksforGeeks</small></p>
</body>
</html>
```



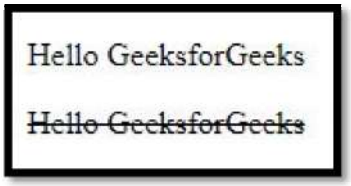
Hello GeeksforGeeks

Hello GeeksforGeeks

6. **Striking through the text:** The **<del>** element is used to strike through the text marking the part as deleted. It also has an opening and a closing tag.

Example:

```
<!DOCTYPE html>
<html>
<head>
        <title>Delete</title>
</head>
<body>
        <!--Text in Normal-->
        <p>Hello GeeksforGeeks</p>
        <!--Text in Delete-->
        <p><del>Hello GeeksforGeeks</del></p>
</body>
</html>
```



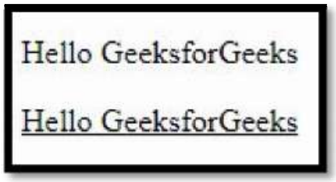
Hello GeeksforGeeks

~~Hello GeeksforGeeks~~

7.Adding a text: The **<ins>** element is used to underline a text marking the part as inserted or added. It also has an opening and a closing tag.

Example:

```
<!DOCTYPE html>
<html>
<head>
        <title>Inserting</title>
</head>
<body>
        <!--Text in Normal-->
        <p>Hello GeeksforGeeks</p>
        <!--Text in Insert-->
        <p><ins>Hello GeeksforGeeks</ins></p>
</body>
</html>
```



Hello GeeksforGeeks

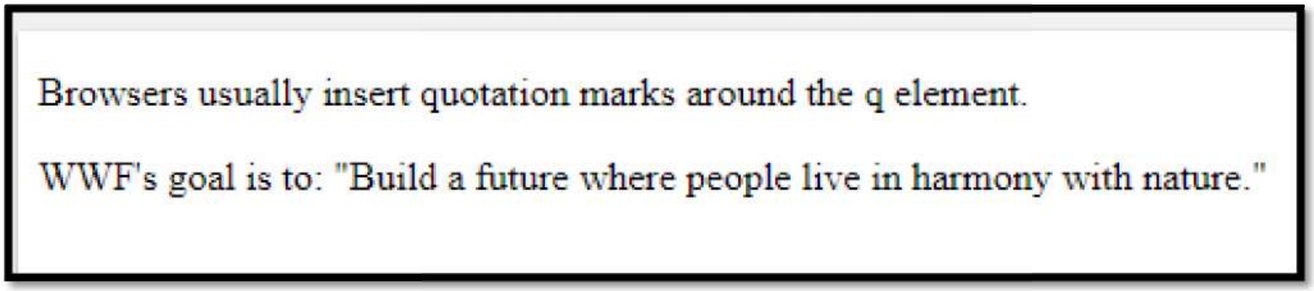
Hello GeeksforGeeks

HTML Quotation and Citation Elements

Tag	Description
<abbr>	Defines an abbreviation or acronym
<address>	Defines contact information for the author/owner of a document
<bdo>	Defines the text direction
<blockquote>	Defines a section that is quoted from another source
<cite>	Defines the title of a work
<q>	Defines a short inline quotation

The HTML <q> element defines a short quotation.
Browsers usually insert quotation marks around the <q> element.
Example

```
<p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>
```

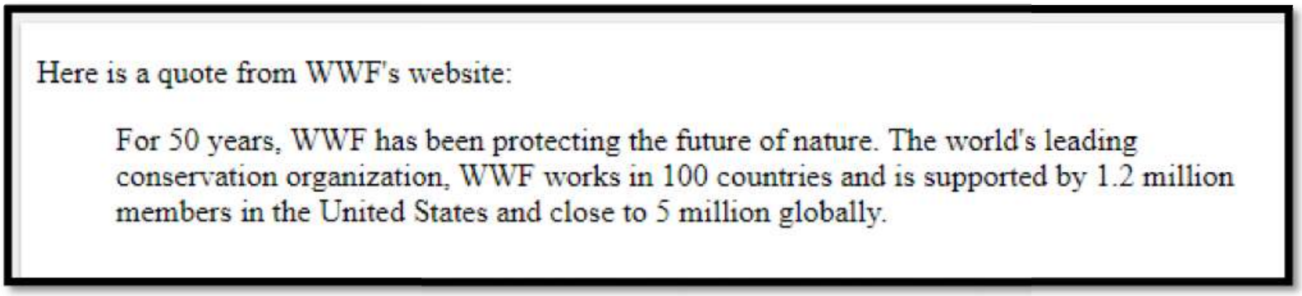


```
Browsers usually insert quotation marks around the q element.

WWF's goal is to: "Build a future where people live in harmony with nature."
```

HTML <blockquote> for Quotations
The HTML <blockquote> element defines a section that is quoted from another source.
Browsers usually indent <blockquote> elements.
Example

```
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature.
```

```
The world's leading conservation organization,
WWF works in 100 countries and is supported by
1.2 million members in the United States and
close to 5 million globally.
</blockquote>
```



Here is a quote from WWF's website:

> For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

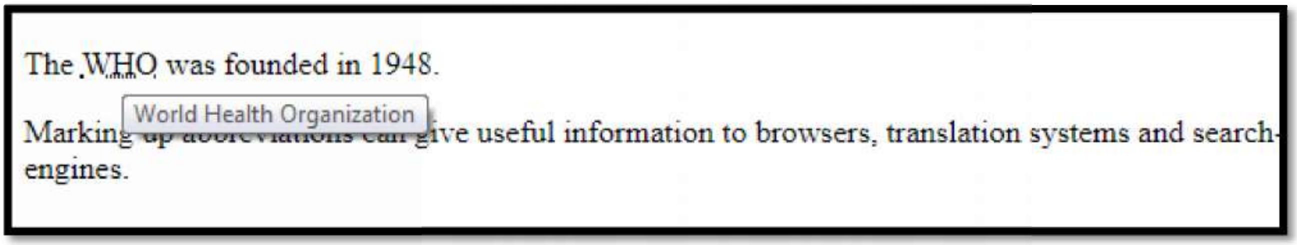
HTML <abbr> for Abbreviations

The HTML <abbr> element defines an abbreviation or an acronym.

Marking abbreviations can give useful information to browsers, translation systems and search-engines.

Example

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```



The WHO was founded in 1948.

Marking up abbreviations can give useful information to browsers, translation systems and search-engines.

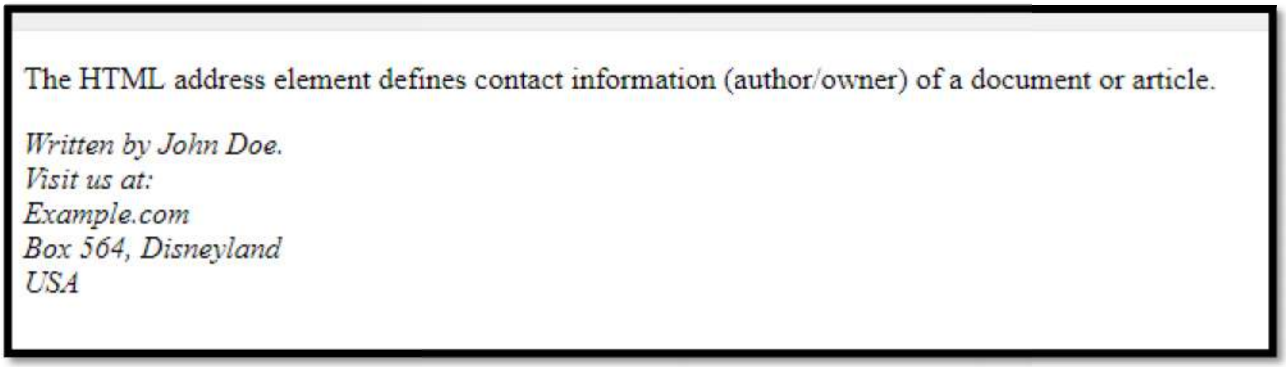
HTML <address> for Contact Information

The HTML <address> element defines contact information (author/owner) of a document or an article.

The <address> element is usually displayed in italic. Most browsers will add a line break before and after the element.

Example

```
<address>
Written by John Doe.<br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>
```



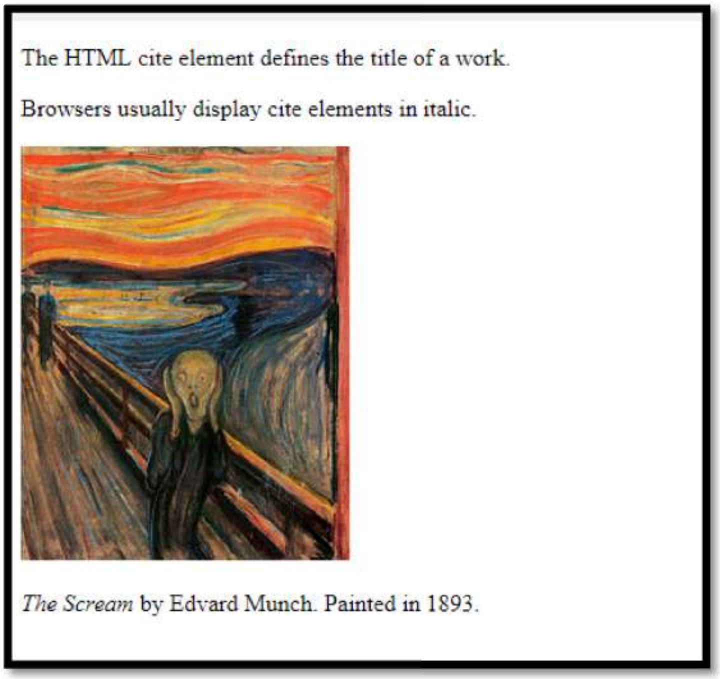
The HTML address element defines contact information (author/owner) of a document or article.

Written by John Doe.
Visit us at:
Example.com
Box 564, Disneyland
USA

HTML <cite> for Work Title

The HTML <cite> element defines the title of a work.
Browsers usually display <cite> elements in italic.
Example

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>
```



The HTML cite element defines the title of a work.

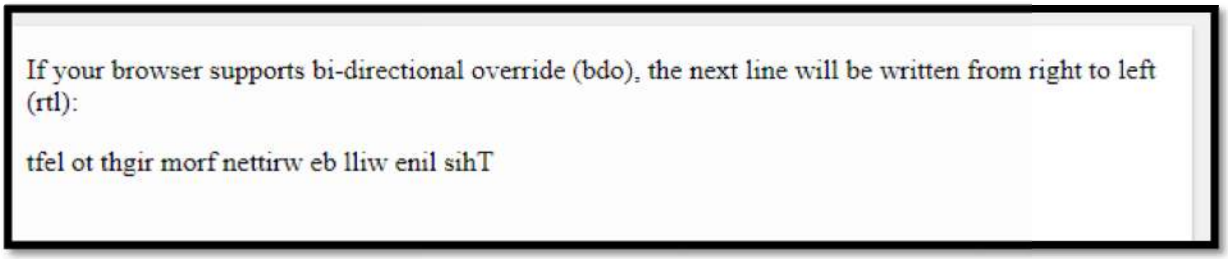
Browsers usually display cite elements in italic.

The Scream by Edvard Munch. Painted in 1893.

HTML <bdo> for Bi-Directional Override

The HTML <bdo> element defines bi-directional override.
The <bdo> element is used to override the current text direction:
Example

```
<bdo dir="rtl">This text will be written from right to left</bdo>
```



If your browser supports bi-directional override (bdo), the next line will be written from right to left (rtl):

tfel ot thgir morf nettirw eb lliw enil sihT

HTML Comments

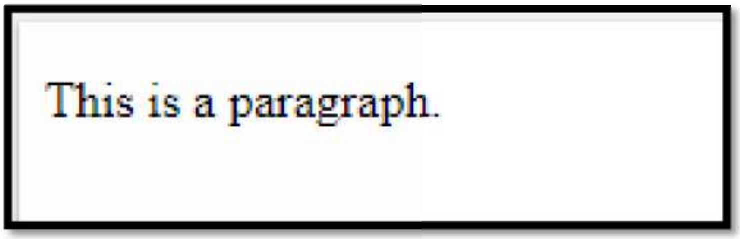
Comment tags are used to insert comments in the HTML source code.

HTML Comment Tags
You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here -->
```

Notice that there is an exclamation point (!) in the opening tag, but not in the closing tag.
Note: Comments are not displayed by the browser, but they can help document your HTML source code.
With comments you can place notifications and reminders in your HTML:
Example

```
<!-- This is a comment -->

<p>This is a paragraph.</p>

<!-- Remember to add more information here -->
```



This is a paragraph.

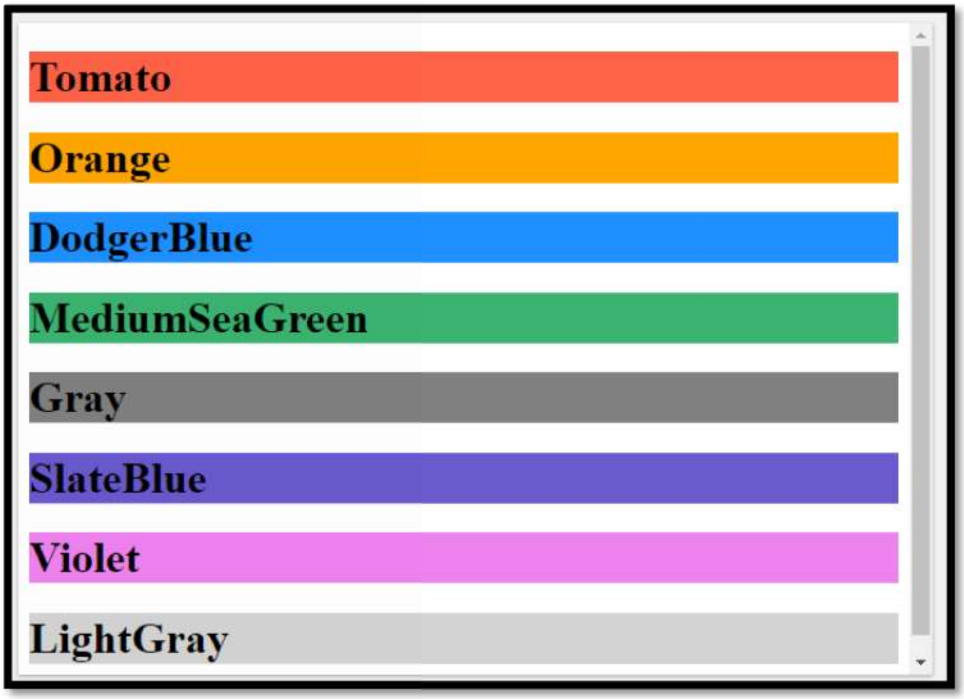
HTML Colors

HTML colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

Color Names

In HTML, a color can be specified by using a color name:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:Tomato;"> Tomato </h1>
<h1 style="background-color:Orange;"> Orange </h1>
<h1 style="background-color:DodgerBlue;"> DodgerBlue </h1>
<h1 style="background-color:MediumSeaGreen;"> MediumSeaGreen </h1>
<h1 style="background-color:Gray;"> Gray </h1>
<h1 style="background-color:SlateBlue;"> SlateBlue </h1>
<h1 style="background-color:Violet;"> Violet </h1>
<h1 style="background-color:LightGray;"> LightGray </h1>

</body>
</html>
```



Tomato

Orange

DodgerBlue

MediumSeaGreen

Gray

SlateBlue

Violet

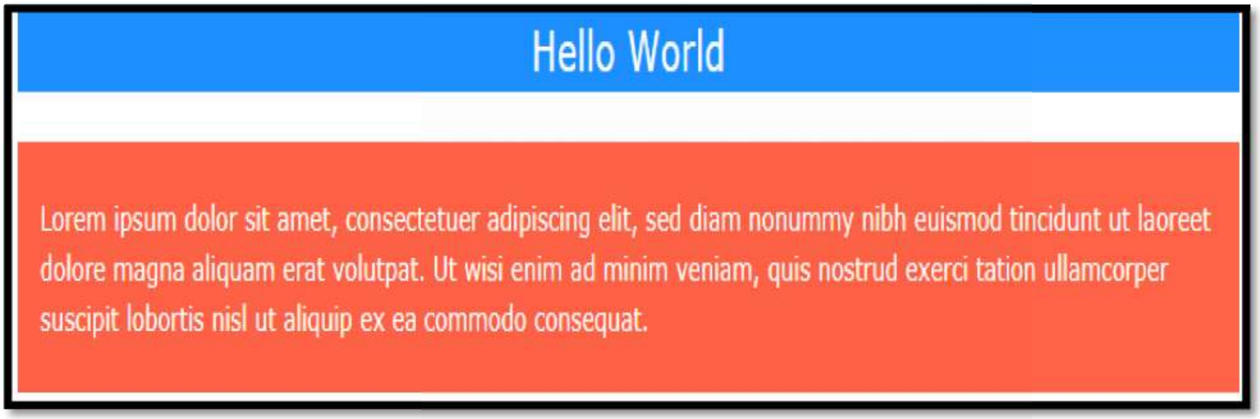
LightGray

Background Color

You can set the background color for HTML elements:

Example

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```



Hello World

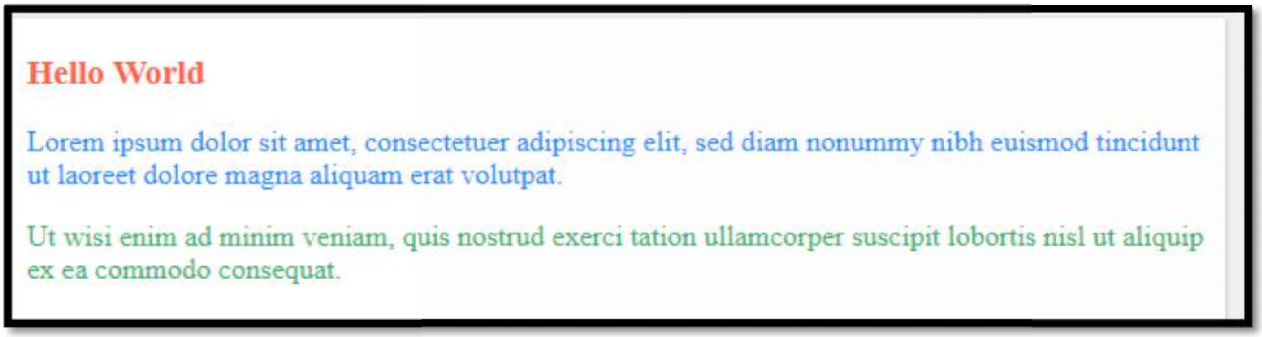
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Text Color

You can set the color of text:

```
<!DOCTYPE html>
<html>
<body>

<h3 style="color:Tomato;">Hello World</h3>
```

```
<p style="color:DodgerBlue;">Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>

<p style="color:MediumSeaGreen;">Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>

</body>
</html>
```



Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

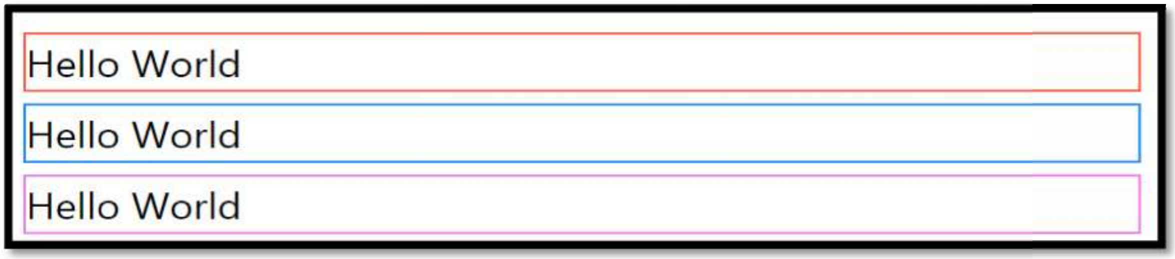
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Border Color

You can set the color of borders:

Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```



Hello World

Hello World

Hello World

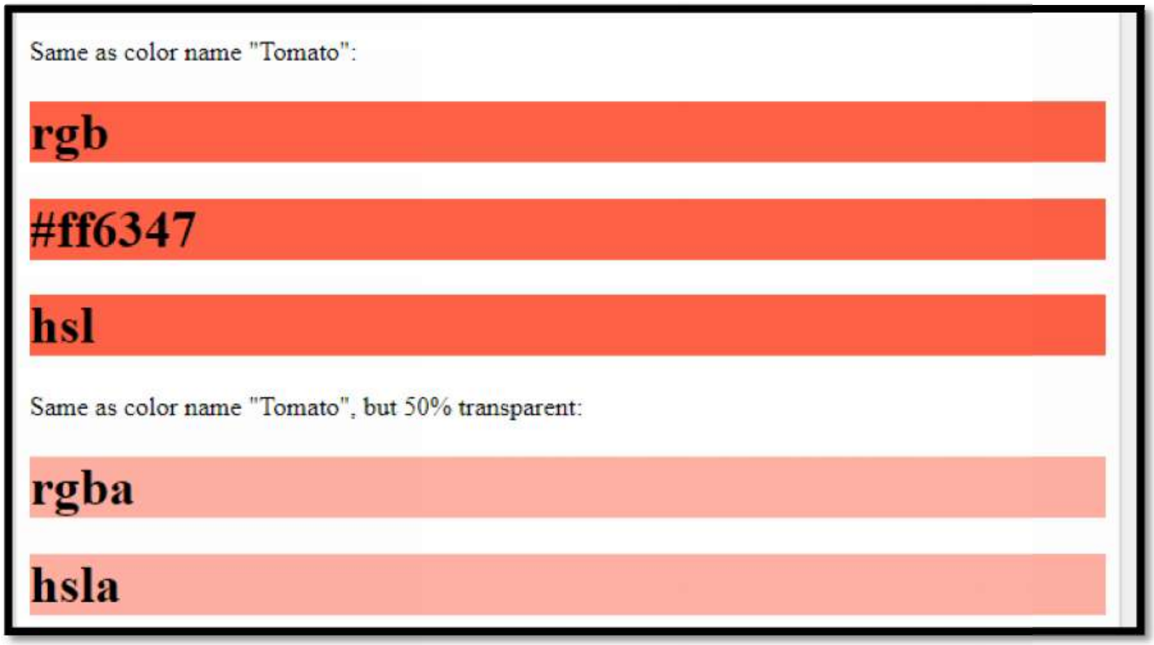
Color Values

In HTML, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

Same as color name "Tomato":

```
<!DOCTYPE html>
<html>
<body>

<p>Same as color name "Tomato":</p>

<h1 style="background-color:rgb(255, 99, 71);">rgb</h1>
<h1 style="background-color:#ff6347;">#ff6347</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">hsl</h1>
```

```
<p>Same as color name "Tomato", but 50% transparent:</p>
<h1 style="background-color:rgba(255, 99, 71, 0.5);">rgba</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">hsla</h1>

</body>
</html>
```



Same as color name "Tomato":

rgb

#ff6347

hsl

Same as color name "Tomato", but 50% transparent:

rgba

hsla

			UNIT 1			
S.no	**QUESTIONS**	**OPTION 1**	**OPTION 2**	**OPTION 3**	**OPTION 4**	**ANSWER**
1	HTML Stands for ______________.	Hyper Text Makeup Language	Hyper Tech Markup Language	None of these	Hyper Text Markup Language	Hyper Text Markup Language
2	HTML is considered as _________ language.	Markup Language	Programming Language	OOP Language	Higher Level Language	Markup Language
3	HTML tags are used to describe document ____________.	tags	Language	Content	Definition	Content
4	HTML Document can contain _______________.	Plain Text	All of these	Tags	Attributes	All of these
5	Page Designed in HTML is called as ________.	Front Page	Web Page	Yellow Page	Server Page	Web Page
6	HTML program is saved using ____________ extension.	.htnl	.html	.hml	.htl	.html
7	HTML program can be read and rendered by _________.	Compiler	Server	Interpreter	Web Browser	Web Browser
8	Who was the primary author of HTML (Hyper Text Markup Language). ?	SabeerBhatiya	Tim Berners-Lee	Brendan Eich	Google Inc.	Tim Berners-Lee
9	HTML tags are surrounded by __________ brackets.	Angle	Squart	Curly	Round	Angle

10	All normal webpages consists of_______________ .	Body and frameset	CSS	Top and bottom	Head and body	Head and body
11	Which attribute specifies a unique alphanumeric identifier to be associated with an element?	class	id	article	html	id
12	The ______________ attribute specifies an inline style associated with an element, which determines the rendering of the affected element.	dir	style	class	article	style
13	HTML documents may contains a ____________ element, which is used to set the header section of a document.	header	footer	section	drive	header
14	_____________ contains the navigation menu, or other navigation functionality for the page.	section	header	nav	aside	nav
15	Which of the following element is used for highlighting content similarly to how a	em	strong	mark	bold	mark

	highlighter pen might be used on important text in a book?					
16	Which tag is used if you want to highlight something that is important to the reader?	<b>	<strong>	<em>	<i>	<b>
17	Which tag is not used if you want to indicate the importance of the phrase?	<i>	<h1>	<em>	<strong>	<i>
18	Which of the following statement is true?	Attribute names must be in uppercase	Attribute values must be quoted	Attribute minimization is mandatory	Attribute values should not be quoted	Attribute values should not be quoted
19	In which part of the HTML metadata is contained?	body tag	html tag	head tag	title tag	head tag
20	What is the role of charset attribute? It specifies	a scheme to be used to interpret the value of the content attribute	a name for the metadata	the character encoding for the HTML document	the character decoding for the HTML document	the character encoding for the HTML document
21	Type of tag used for inserting an image in web document is	<imp>	<img>	<image>	<src>	<img>
22	Scrolling piece of text displayed either horizontally or vertically is created by	<floating text> tag	<scrolling text>	<marquee> tag	None of them	<marquee> tag
23	A tag which is used for overriding current text	<bdodir="rtl"></bdo>	<bdodir="ltr"></bdo>	<bdo = "right to left"></bdo>	None	<bdodir="rtl"></bdo>

	direction e.g. Right to left side					
24	which of the following tag is used to mark a begining of paragraph ?	<TD>	<br>	<P>	<TR>	<P>
25	Correct HTML tag for the largest heading is	<head>	<h6>	<heading>	<h1>	<h1>
26	Web pages starts with which ofthe following tag?	<Body>	<Title>	<HTML>	<Form>	<HTML>
27	Tags and text that are not directly displayed on the page are written in _____ section.	<html>	<head>	<title>	<body>	<head>
28	Which of the following CSS Color Feature can be used as a macro for whatever the current color is?	CurrentColor keyword	HSLa Color	HSL Color	RGB Color	CurrentColor keyword
29	Which one from following is a block level element in HTML?	<span>	<div>	<class>	Both A and B	<div>
30	For uniquely identifying an elements name which attribute is used?	id	class	header	none	id
31	If you want to merge two or more rows in a table which attribute you can use?	Rowmerge	Rowspan	Colmerge	Colspan	Rowspan

32	Text inside a <pre> element tag is displayed	In table	In one line	In fixed width font	In half paragraph	In fixed width font
33	Which language is used to create web documents?	PHP	JavaScript	HTML	J-Query	HTML
34	Tags used to tell browser that how to display text enclosed (e.g. <b>, <big>) are called as	Physical tags	Logical tags	User defined tags	None	Physical tags
35	Title element defines title of document at	Header	Web browser	Middle of the Document	Footer	Web browser
36	"href " attribute in HTML document is used to define	Paragraph	Heading	Table	Link Destination	Link Destination
37	An image-map is an image with clickable areas, so which tag we can use for defining an image-map?	<image map>	<map image>	<map>	<image>	<map>
38	Anchor tag in HTML is denoted as	<a>	<anchor>	<ach>	None	<a>
39	_blank target attribute opens a linked document in	new window	new tab	same window	Both A and B	Both A and B
40	To define a style for a special type of elements, we can add a	Class	Margin	Border	Table	Class
41	A webpage displa ys a picture. What??? ??tag ?????was	picture	image	img	src	img

	used to display that picture?					
42	<b> tag makes the enclosed text bol What is other tag to make text bold?	<strong>	<dar>	<black>	<emp>	<strong>
43	Tags and text that are not directly displayed on the page are written in ______ section.	<html>	<head>	<title>	<body>	<head>
44	Which tag inserts a line horizontally on your web page?	<hr>	<line>	<line direction="horizontal">	<tr>	<hr>
45	What should be the first tag in any HTML document?	<head>	<title>	<html>	<document>	<head>
46	Which tag allows you to add a row in a table?	<td> and </td>	<cr> and </cr>	<th> and </th>	<tr> and </tr>	<tr> and </tr>
47	How can you make a bulleted list?	<list>	<nl>	<ul>	<ol>	<ul>
48	How can you make a numbered list?	<dl>	<ol>	<list>	<ul>	<ol>
49	How can you make an e-mail link?	<a href=”xxx@yyy ”>	<mail href=”xxx@yyy ”>	<mail>xxx@yyy </mail>	<a href=”mailto:xxx@yyy ”>	<a href=”xxx@yyy ”>
50	What is the correct HTML for making a hyperlink?	<a href=”http:// mcqsets.com“ >ICT Trends Quiz</a>	<a name=”https://mcqsets.com“ >ICT Trends Quiz</a>	<https://mcqsets.com</a>	url=”https://mcqsets.com“ >ICT Trends Quiz	<a href=”http:// mcqsets.com“ >ICT Trends Quiz</a>
51	Choose the correct HTML tag to make a text italic	<ii>	<italics>	<italic>	<i>	<a href=”http:// mcqsets.com“ >ICT Trends Quiz</a>
52	What is the correct HTML for adding a background color?	<body color=”yellow”>	<body bgcolor=”yellow”>	<background>yellow</background>	<body background=”yellow”>	<body bgcolor=”yellow”>

53	Choose the correct HTML tag for the smallest size heading?	<heading>	<h6>	<h1>	<head>	<h1>
54	What is the correct HTML tag for inserting a line break?	<br>	<lb>	<break>	<newline>	<br>
55	What doesvlink attribute mean?	visited link	virtual link	very good link	active link	visited link
56	Which of the following is not a pair tag?	<p>	< u >	<i>	<img>	<img>
57	To create HTML document you require a	web page editing software	High powered computer	Just a notepad can be used	None of above	Just a notepad can be used
58	In HTML document the tags	Should be written in upper case	should be written in lower case	should be written in propercase	can be written in both uppercase or lowercase	can be written in both uppercase or lowercase
59	Marquee is a tag in HTML to	mark the list of items to maintaininqueue	Mark the text so that it is hidden in browser	Display text with scrolling effect	None of above	Display text with scrolling effect
60	There are ____ different of heading tags in HTML	4	5	6	7	6

UNIT -2

Styling HTML with CSS :

CSS stands for **C**ascading **S**tyle **S**heets.

CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**.

CSS **saves a lot of work**. It can control the layout of multiple web pages all at once.

CSS can be added to HTML elements in 3 ways:

Inline - by using the style attribute in HTML elements

Internal - by using a <style> element in the <head> section

External - by using an external CSS file

The most common way to add CSS, is to keep the styles in separate CSS files. However, here we will use inline and internal styling, because this is easier to demonstrate, and easier

Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the style attribute of an HTML element.

This example sets the text color of the <h1> element to blue:

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

This is a Blue Heading

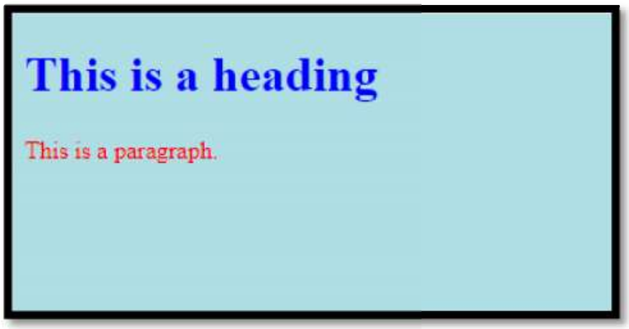
Internal CSS or Embedded CSS

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the <head> section of an HTML page, within a <style> element:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

```
</body>
</html>
```



This is a heading

This is a paragraph.

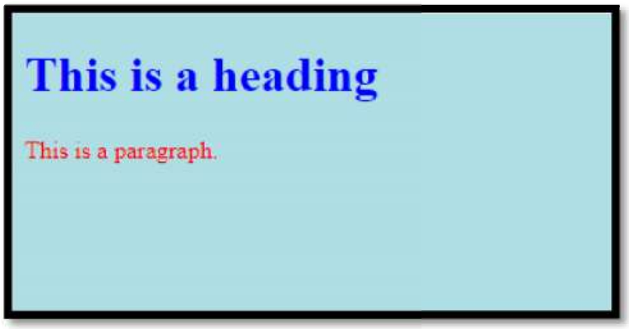
External CSS

An external style sheet is used to define the style for many HTML pages.

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

To use an external style sheet, add a link to it in the <head> section of the HTML page:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



This is a heading

This is a paragraph.

An external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

Here is how the "styles.css" looks:

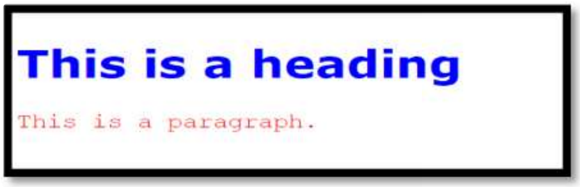
```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

CSS Fonts

The CSS color property defines the text color to be used.
The CSS font-family property defines the font to be used.
The CSS font-size property defines the text size to be used.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p  {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

```
</body>
</html>
```



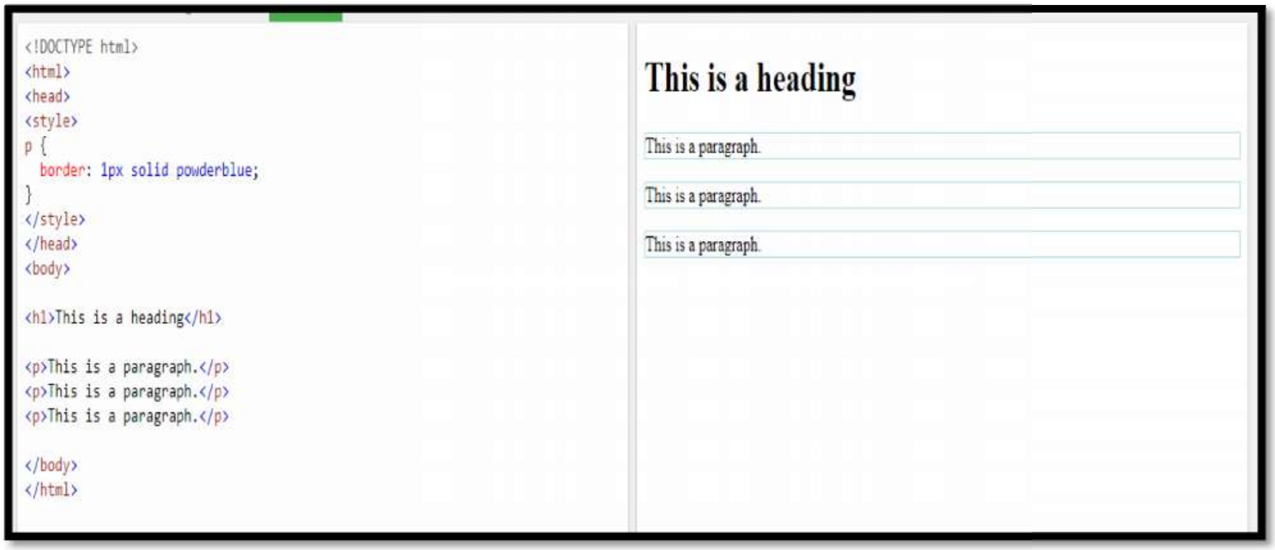
This is a heading

This is a paragraph.

CSS Border

The CSS border property defines a border around an HTML element:

```
p {
  border: 1px solid powderblue;
}
```



```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 1px solid powderblue;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>

</body>
</html>
```

This is a heading

This is a paragraph.

This is a paragraph.

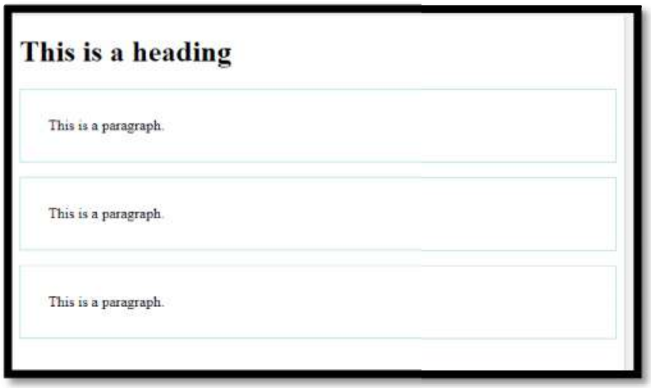
This is a paragraph.

CSS Padding

The CSS padding property defines a padding (space) between the text and the border:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 1px solid powderblue;
  padding: 30px;
}
</style>
</head>
<body>
```

```
<h1>This is a heading</h1>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>

</body>
</html>
```



This is a heading

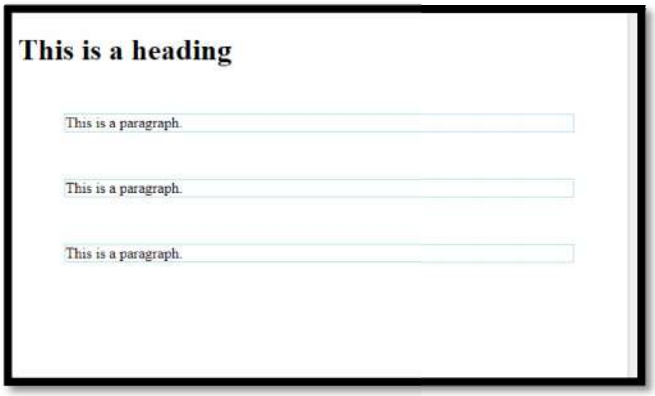
This is a paragraph.

This is a paragraph.

This is a paragraph.

CSS Margin

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 1px solid powderblue;
  margin: 50px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>

</body>
</html>
```



This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

The id Attribute

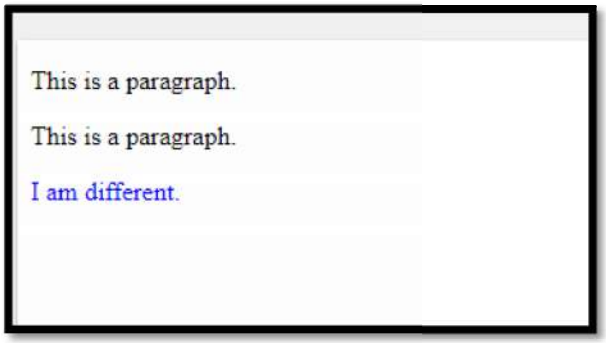
To define a specific style for one special element, add an id attribute to the element:

```
<p id="p01">I am different</p>
```

then define a style for the element with the specific id:

```
<!DOCTYPE html>
<html>
<head>
<style>
#p01 {
  color: blue;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p id="p01">I am different.</p>

</body>
</html>
```

Note: The id of an element should be unique within a page, so the id selector is used to select one unique element!



This is a paragraph.

This is a paragraph.

I am different.

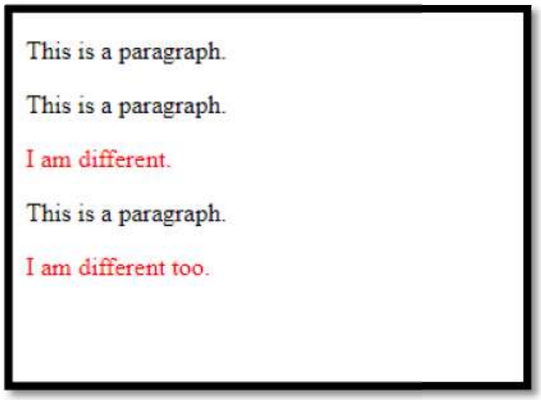
The class Attribute

To define a style for special types of elements, add a class attribute to the element:

```
<p class="error">I am different</p>
```

then define a style for the elements with the specific class:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.error {
  color: red;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p class="error">I am different.</p>
<p>This is a paragraph.</p>
<p class="error">I am different too.</p>

</body>
</html>
```



This is a paragraph.

This is a paragraph.

I am different.

This is a paragraph.

I am different too.

HTML LINKS

Links are found in nearly all web pages. Links allow users to click their way from page to page.

HTML Links - Hyperlinks
HTML links are hyperlinks.
You can click on a link and jump to another document.
When you move the mouse over a link, the mouse arrow will turn into a little hand.
Note: A link does not have to be text. It can be an image or any other HTML element.
HTML Links - Syntax
Hyperlinks are defined with the HTML <a> tag:

```
<a href="url">link text</a>
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

The href attribute specifies the destination address (https://www.w3schools.com/html/) of the link.
The **link text** is the visible part (Visit our HTML tutorial).
Clicking on the link text will send you to the specified address.
Note: Without a forward slash at the end of subfolder addresses, you might generate two requests to the server. Many servers will automatically add a forward slash to the end of the address, and then create a new request.

Local Links

The example above used an absolute URL (a full web address).
A local link (link to the same web site) is specified with a relative URL (without https://www....).

```
<a href="html_images.asp">HTML Images</a>
```

HTML Links - The target Attribute
The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

_blank - Opens the linked document in a new window or tab
_self - Opens the linked document in the same window/tab as it was clicked (this is default)
_parent - Opens the linked document in the parent frame
_top - Opens the linked document in the full body of the window
framename - Opens the linked document in a named frame
This example will open the linked document in a new browser window/tab:

Example

```
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

Tip: If your webpage is locked in a frame, you can use target="_top" to break out of the frame:

Example

```
<a href="https://www.w3schools.com/html/" target="_top">HTML5 tutorial!</a>
```

HTML Links - Image as Link

It is common to use images as links:

Example

```
<!DOCTYPE html>
<html>
<body>

<h2>Image Links</h2>

<p>The image is a link. You can click on it.</p>

<a href="default.asp">
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;border:0">
</a>

<p>We have added "border:0" to prevent IE9 (and earlier) from displaying a border around the
image.</p>

</body>
</html>
```

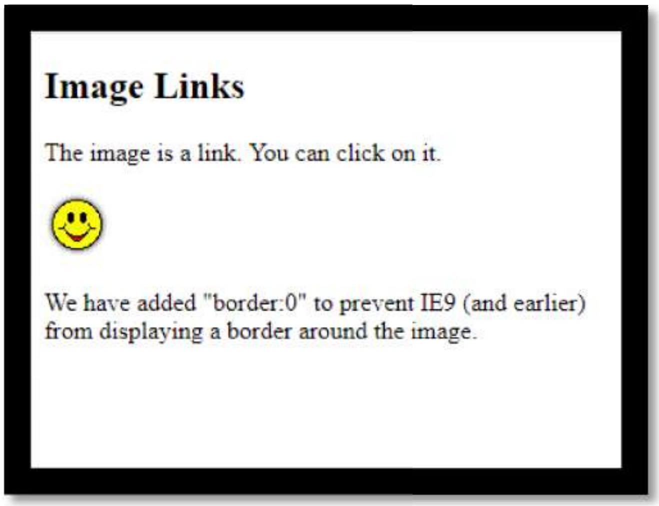


Image Links

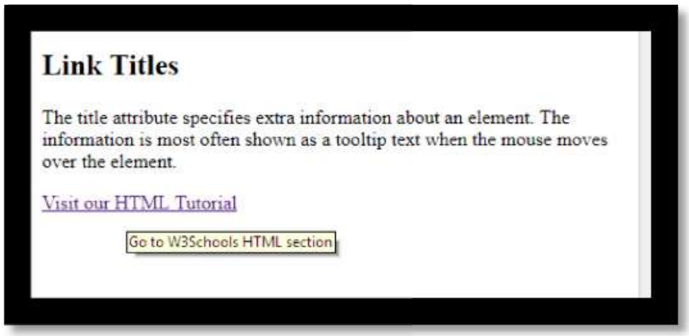
The image is a link. You can click on it.

We have added "border:0" to prevent IE9 (and earlier) from displaying a border around the image.

Link Titles

The title attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

```
<!DOCTYPE html>
<html lang="en-US">
<body>

<h2>Link Titles</h2>
<p>The title attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.</p>
<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML section">Visit our HTML Tutorial</a>

</body>
</html>
```



Link Titles

The title attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

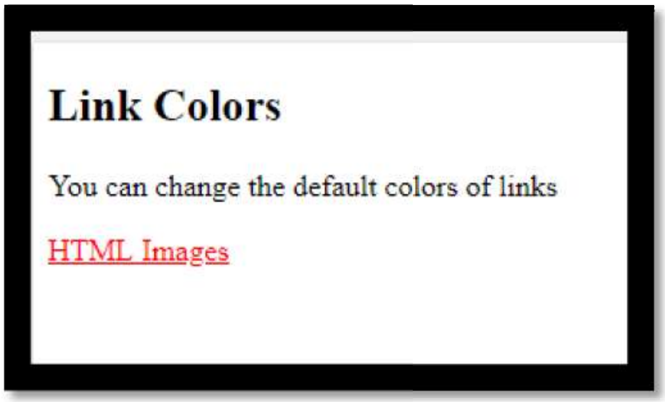
Visit our HTML Tutorial

HTML Link Colors

By default, a link will appear like this (in all browsers):

An unvisited link is underlined and blue
A visited link is underlined and purple
An active link is underlined and red
You can change the default colors, by using CSS:

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link {
  color: green;
  background-color: transparent;
  text-decoration: none;
}
a:visited {
  color: pink;
  background-color: transparent;
  text-decoration: none;
}
a:hover {
  color: red;
  background-color: transparent;
  text-decoration: underline;
}
a:active {
  color: yellow;
  background-color: transparent;
  text-decoration: underline;
}
</style>
</head>
<body>

<h2>Link Colors</h2>

<p>You can change the default colors of links</p>

<a href="html_images.asp" target="_blank">HTML Images</a>

</body>
</html>
```



Link Colors

You can change the default colors of links

HTML Images

HTML IMAGES :

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

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Image</h2>
<img src="pic_trulli.jpg" alt="Trulli" width="500" height="333">

</body>
</html>
```

HTML Images Syntax

In HTML, images are defined with the <img> tag.

The <img> 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

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.

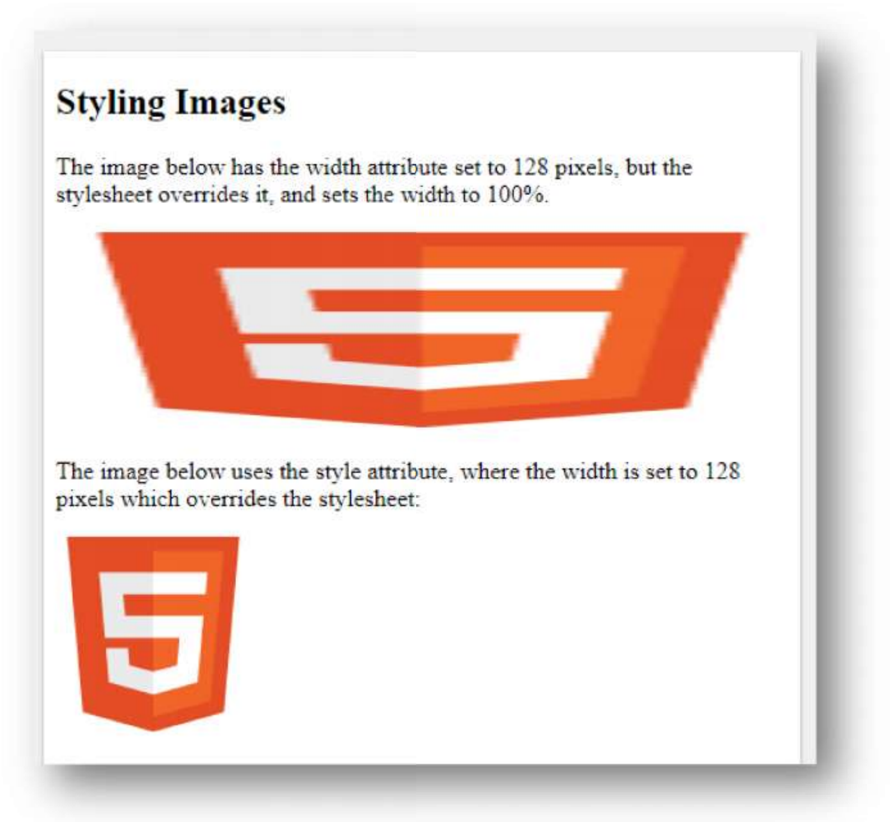
```
<img src="pic_trulli.jpg" alt="Trulli" style="width:500px;height:600px;">
```

Width and Height, or Style?

The width, height, and style attributes are valid in HTML.

However, we suggest using the style attribute. It prevents styles sheets from changing the size of images

```
<!DOCTYPE html>
<html>
<head>
<style>
/* This stylesheet sets the width of all images to 100%: */
img {
  width: 100%;
}
</style>
</head>
<body>

<h2>Styling Images</h2>
<p>The image below has the width attribute set to 128 pixels, but the stylesheet overrides it, and sets the width to 100%.</p>
<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

<p>The image below uses the style attribute, where the width is set to 128 pixels which overrides the stylesheet:</p>
<img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">

</body>
</html>
```



Styling Images

The image below has the width attribute set to 128 pixels, but the stylesheet overrides it, and sets the width to 100%.

The image below uses the style attribute, where the width is set to 128 pixels which overrides the stylesheet:

HTML Tables

An HTML table is defined with the <table> tag.

Each table row is defined with the <tr> tag. A table header is defined with the <th> tag. By default, table headings are bold and centered. A table data/cell is defined with the <td> tag.

HTML Table - Adding a Border

If you do not specify a border for the table, it will be displayed without borders.

A border is set using the CSS border property

```
table, th, td {
  border: 1px solid black;
}
```

HTML Table - Collapsed Borders

If you want the borders to collapse into one border, add the CSS border-collapse property:

Example

```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
```

HTML Table - Adding Cell Padding

Cell padding specifies the space between the cell content and its borders.

If you do not specify a padding, the table cells will be displayed without padding.

To set the padding, use the CSS padding property:

Example

```
th, td {
  padding: 15px;
}
```

HTML Table - Left-align Headings

By default, table headings are bold and centered.

To left-align the table headings, use the CSS text-align property:

Example

```
th {
  text-align: left;
}
```

HTML Table - Adding Border Spacing

Border spacing specifies the space between the cells.

To set the border spacing for a table, use the CSS border-spacing property:

Example

```
table {
  border-spacing: 5px;
}
```

Note: If the table has collapsed borders, border-spacing has no effect.

HTML Table - Cells that Span Many Columns

To make a cell span more than one column, use the colspan attribute:

Example

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```

HTML Table - Cells that Span Many Rows

To make a cell span more than one row, use the rowspan attribute:

Example

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

HTML Table - Adding a Caption

To add a caption to a table, use the <caption> tag:

Example

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

Note: The <caption> tag must be inserted immediately after the <table> tag.

A Special Style for One Table
To define a special style for a special table, add an id attribute to the table:

Example

```
<table id="t01">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Now you can define a special style for this table:

```
table#t01 {
  width: 100%;
  background-color: #f1f1c1;
}
```

And add more styles:

```
table#t01 tr:nth-child(even) {
  background-color: #eee;
}
table#t01 tr:nth-child(odd) {
  background-color: #fff;
}
table#t01 th {
  color: white;
  background-color: black;
}
```

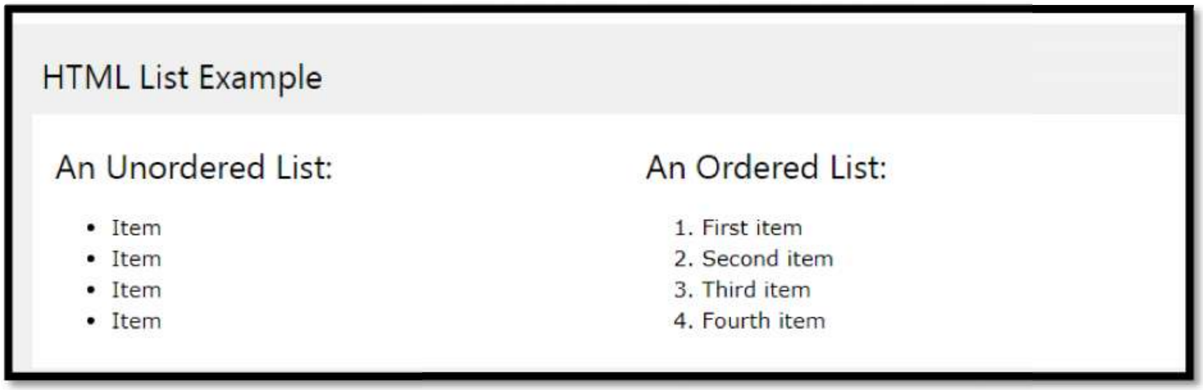
Chapter Summary

Use the HTML <table> element to define a table
Use the HTML <tr> element to define a table row
Use the HTML <td> element to define a table data
Use the HTML <th> element to define a table heading
Use the HTML <caption> element to define a table caption

Use the CSS border property to define a border
Use the CSS border-collapse property to collapse cell borders
Use the CSS padding property to add padding to cells
Use the CSS text-align property to align cell text
Use the CSS border-spacing property to set the spacing between cells
Use the colspan attribute to make a cell span many columns
Use the rowspan attribute to make a cell span many rows
Use the id attribute to uniquely define one table
HTML Table Tags

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

HTML LISTS



HTML List Example

An Unordered List:

- Item
- Item
- Item
- Item

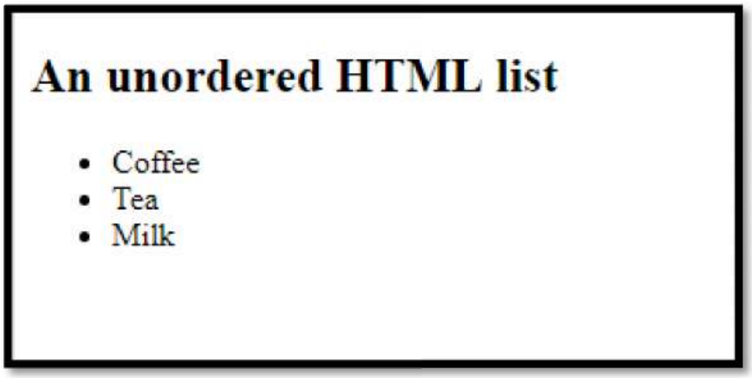
An Ordered List:

1. First item
2. Second item
3. Third item
4. Fourth item

Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.
The list items will be marked with bullets (small black circles) by default:

```
<!DOCTYPE html>
<html>
<body>

<h2>An unordered HTML list</h2>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>
```



An unordered HTML list

- Coffee
- Tea
- Milk

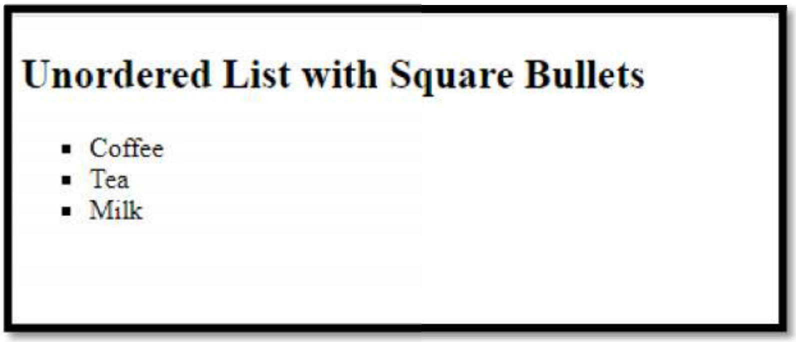
Unordered HTML List - Choose List Item Marker

The CSS list-style-type property is used to define the style of the list item marker:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Example :

```
<!DOCTYPE html>
<html>
<body>

<h2>Unordered List with Square Bullets</h2>

<ul style="list-style-type:square;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>
```



Unordered List with Square Bullets

- Coffee
- Tea
- Milk

Ordered HTML List

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default:

Ordered HTML List - The Type Attribute

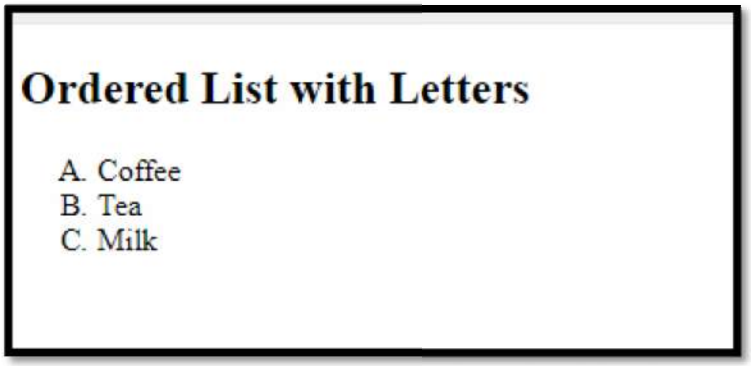
The type attribute of the <ol> tag, defines the type of the list item marker:

Type	**Description**
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Example :

```
<!DOCTYPE html>
<html>
<body>

<h2>Ordered List with Letters</h2>

<ol type="A">
```

```
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```



Ordered List with Letters

A. Coffee
B. Tea
C. Milk

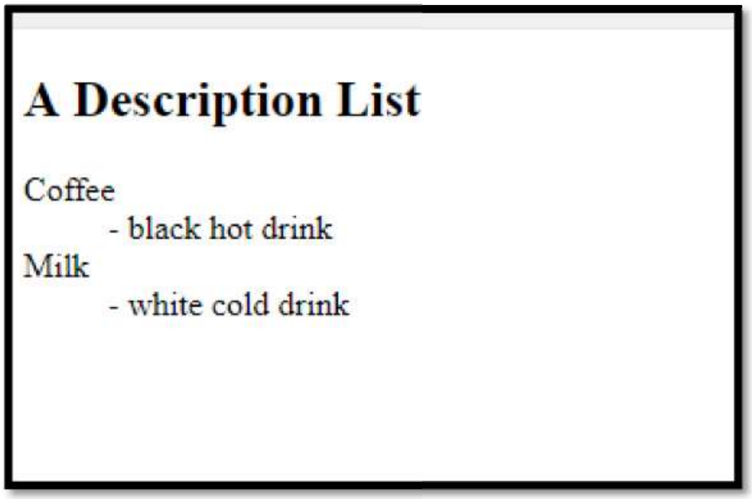
HTML Description Lists

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

```
<!DOCTYPE html>
<html>
<body>

<h2>A Description List</h2>

<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>

</body>
</html>
```



A Description List

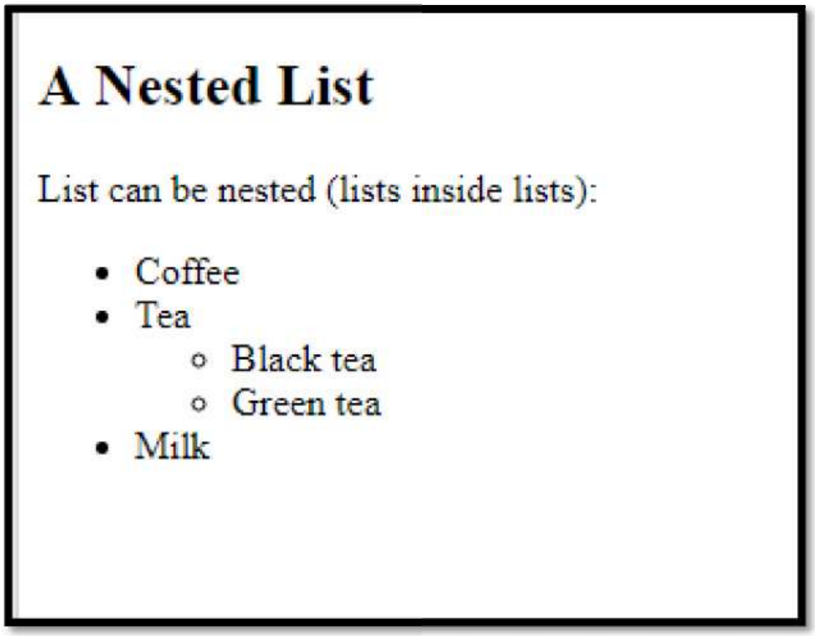
Coffee
- black hot drink

Milk
- white cold drink

Nested HTML Lists

List can be nested (lists inside lists):

```
<!DOCTYPE html>
<html>
<body>

<h2>A Nested List</h2>
<p>List can be nested (lists inside lists):</p>

<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>

</body>
</html>
```



A Nested List

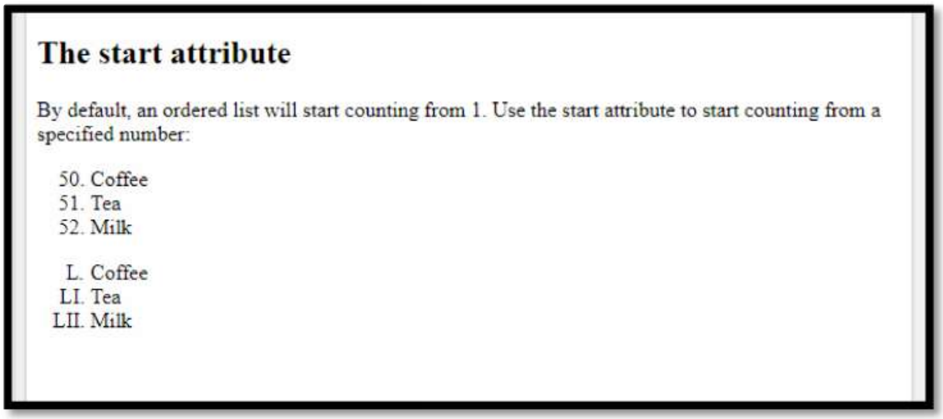
List can be nested (lists inside lists):

- Coffee
- Tea
  - Black tea
  - Green tea
- Milk

Control List Counting

By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the start attribute:

```
<!DOCTYPE html>
<html>
<body>

<h2>The start attribute</h2>
<p>By default, an ordered list will start counting from 1. Use the start attribute to start counting from a specified number:</p>

<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<ol type="I" start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```



The start attribute

By default, an ordered list will start counting from 1. Use the start attribute to start counting from a specified number:

```
 50. Coffee
 51. Tea
 52. Milk

   L. Coffee
  LI. Tea
 LII. Milk
```

HTML BLOCKS

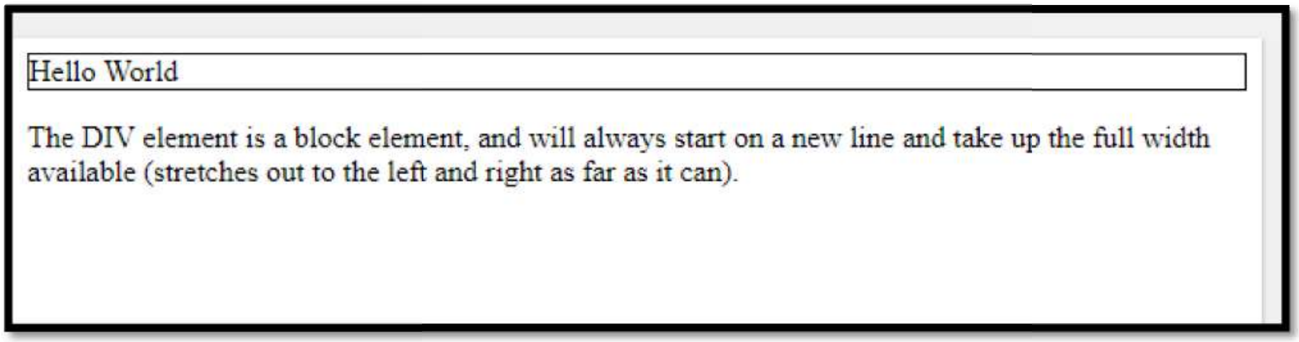
Every HTML element has a default display value depending on what type of element it is.

The two display values are: block and inline.

Block-level Elements

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

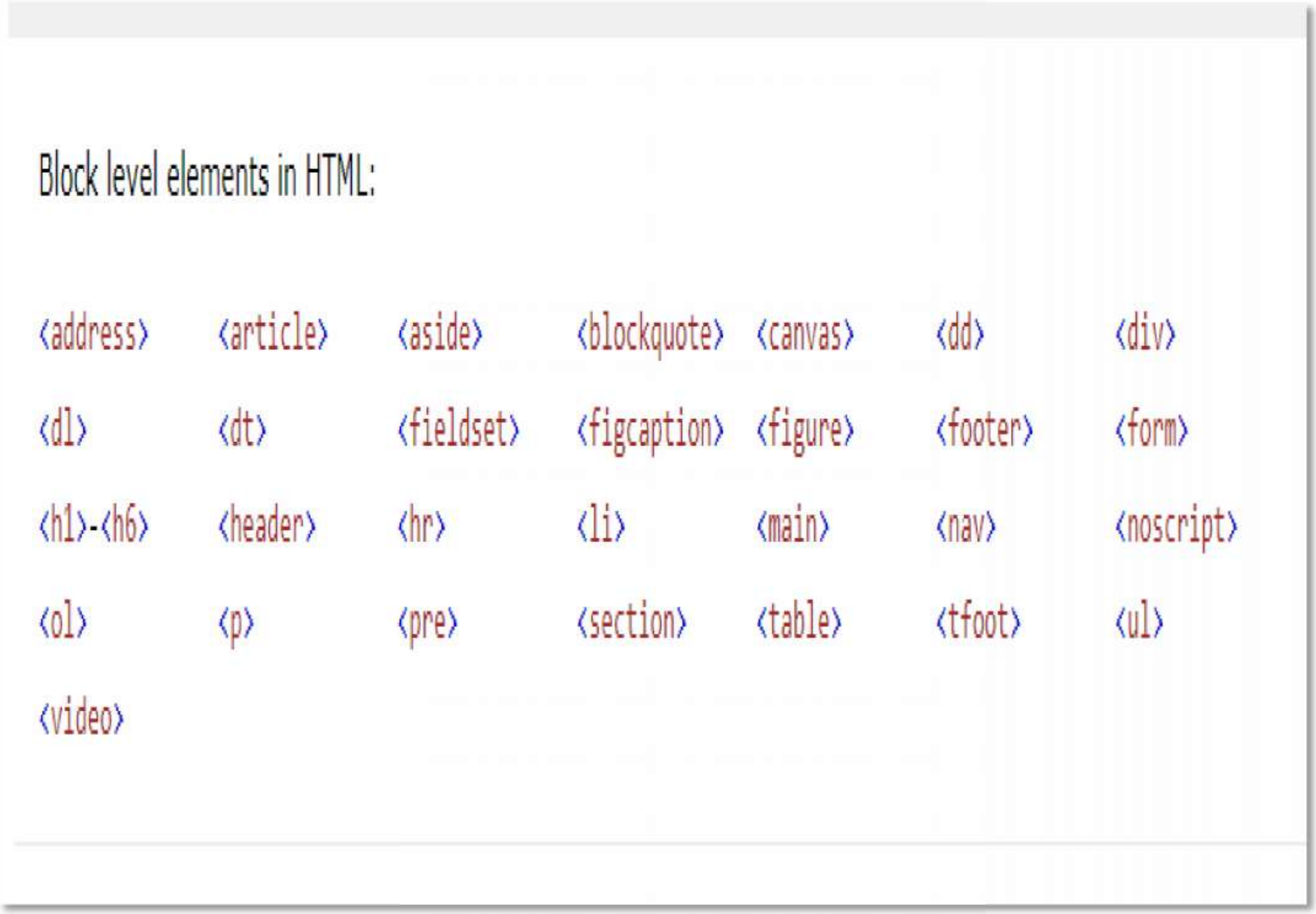
```
<!DOCTYPE html>
<html>
<body>

<div style="border: 1px solid black">Hello World</div>

<p>The DIV element is a block element, and will always start on a new line and take up the full width available (stretches out to the left and right as far as it can).</p>

</body>
</html>
```



Hello World

The DIV element is a block element, and will always start on a new line and take up the full width available (stretches out to the left and right as far as it can).

Block level elements in HTML:



Block level elements in HTML:

<address> <article> <aside> <blockquote> <canvas> <dd> <div>

<dl> <dt> <fieldset> <figcaption> <figure> <footer> <form>

<h1>-<h6> <header> <hr> <li> <main> <nav> <noscript>

<ol> <p> <pre> <section> <table> <tfoot> <ul>

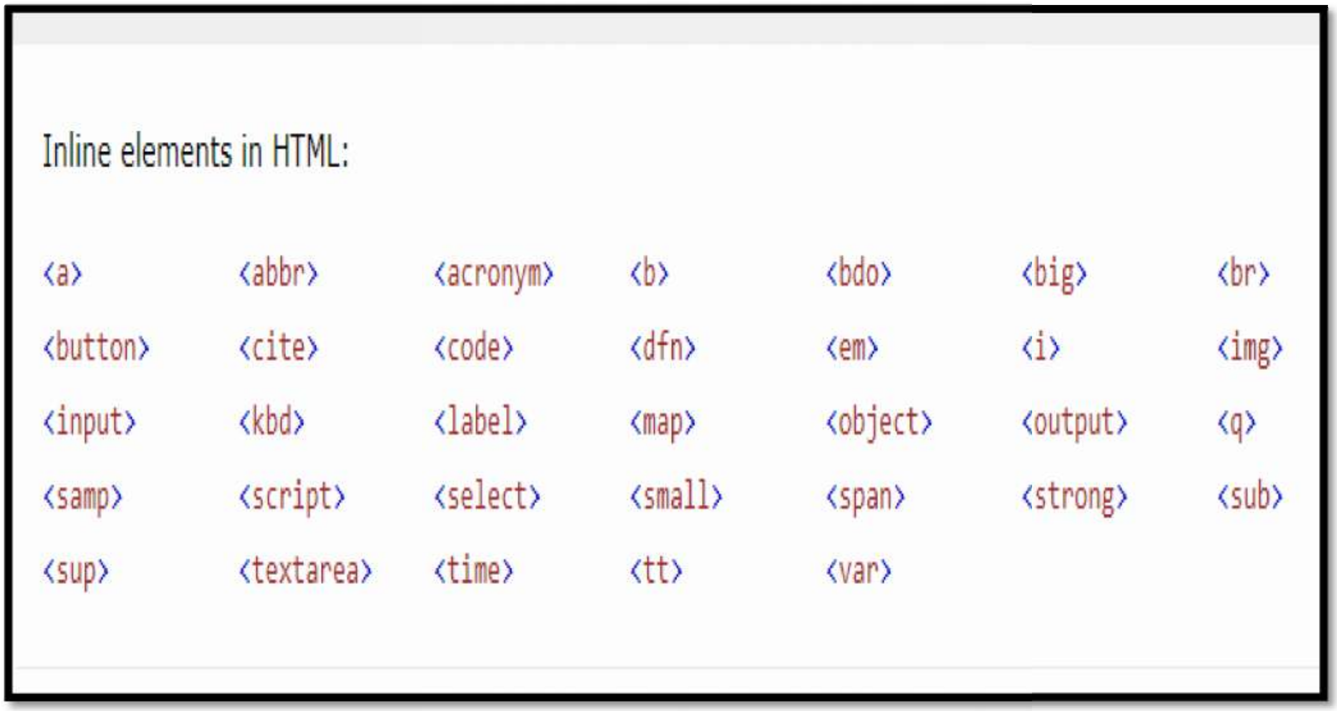
<video>

Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline <span> element inside a paragraph.

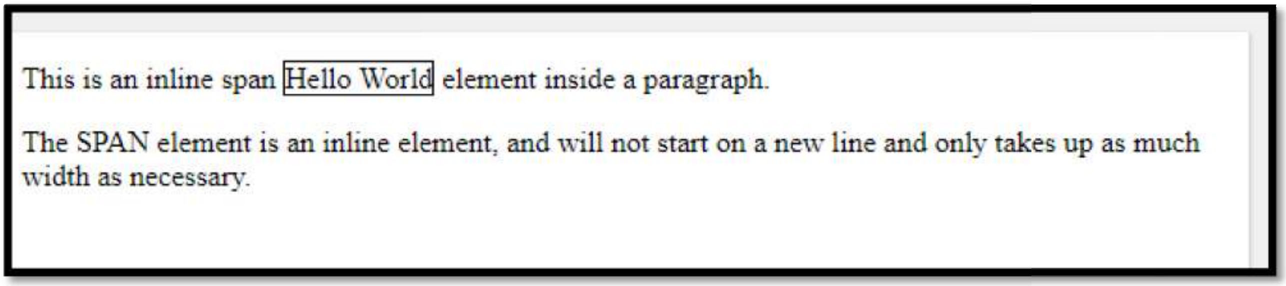
```
<span>Hello World</span>
```



Inline elements in HTML:

<a>	<abbr>	<acronym>	<b>	<bdo>	<big>	<br>
<button>	<cite>	<code>	<dfn>	<em>	<i>	<img>
<input>	<kbd>	<label>	<map>	<object>	<output>	<q>
<samp>	<script>	<select>	<small>	<span>	<strong>	<sub>
<sup>	<textarea>	<time>	<tt>	<var>		

```
<!DOCTYPE html>
<html>
<body>

<p>This is an inline span <span style="border: 1px solid black">Hello World</span> element
inside a paragraph.</p>

<p>The SPAN element is an inline element, and will not start on a new line and only takes up as
much width as necessary.</p>

</body>
</html>
```



This is an inline span Hello World element inside a paragraph.

The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.

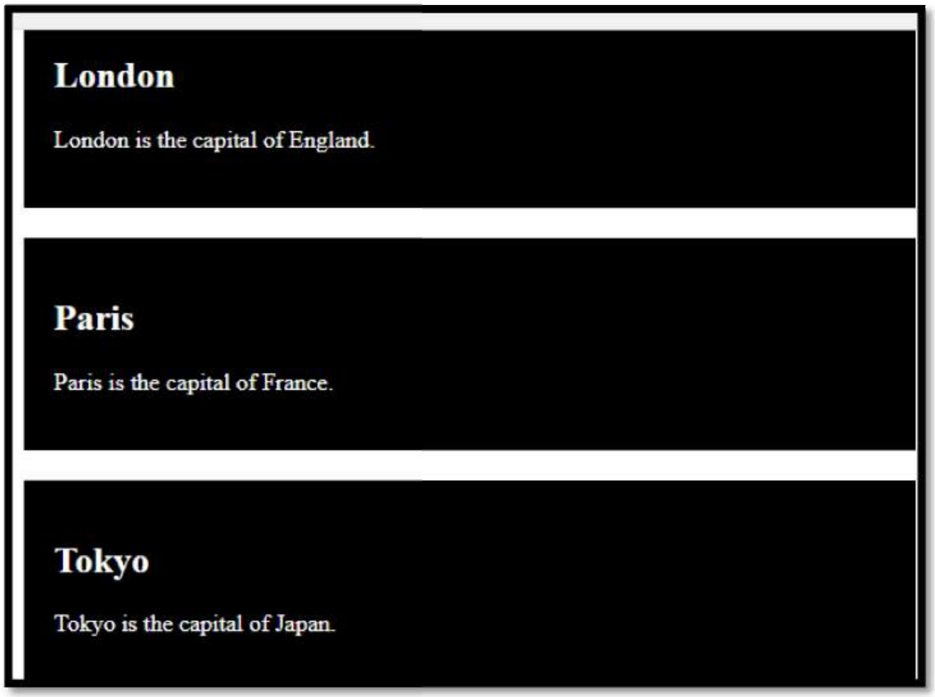
Using The class Attribute

The HTML class attribute is used to define equal styles for elements with the same class name. So, all HTML elements with the same class attribute will get the same style.

Here we have three <div> elements that point to the same class name:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.cities {
  background-color: black;
  color: white;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="cities">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>

<div class="cities">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="cities">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```



London

London is the capital of England.

Paris

Paris is the capital of France.

Tokyo

Tokyo is the capital of Japan.

HTML The id Attribute

Using The id Attribute

The id attribute specifies a unique id for an HTML element (the value must be unique within the HTML document).

The id value can be used by CSS and JavaScript to perform certain tasks for the element with the specific id value.

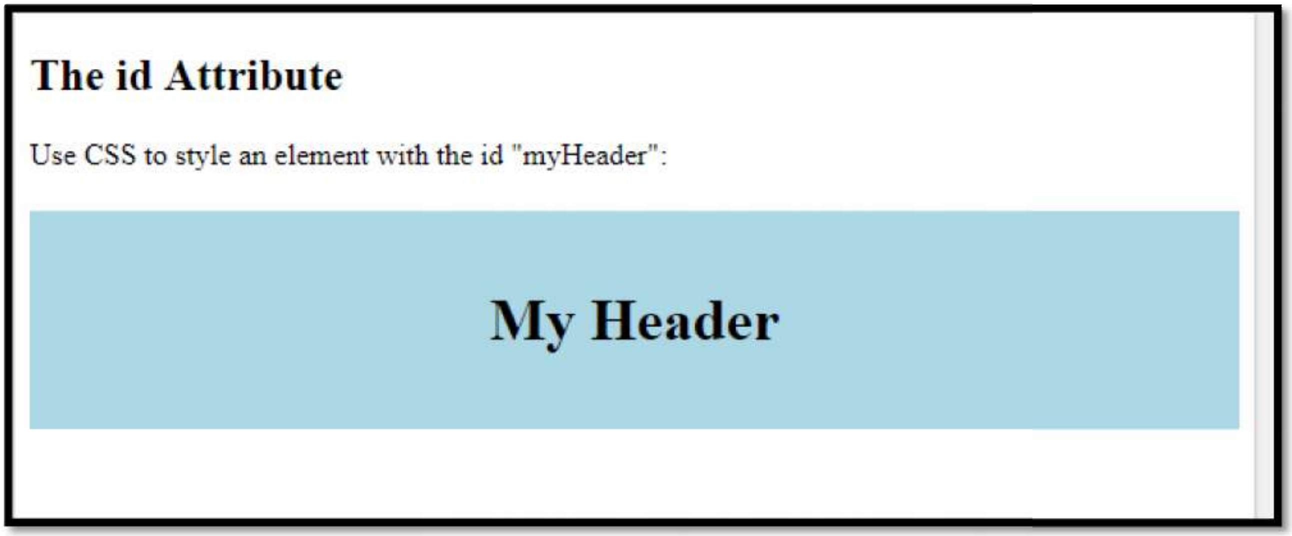
In CSS, to select an element with a specific id, write a hash (#) character, followed by the id of the element:

Example

Use CSS to style an element with the id "myHeader":

```
<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
```

```
</head>
<body>

<h2>The id Attribute</h2>
<p>Use CSS to style an element with the id "myHeader":</p>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```



The id Attribute

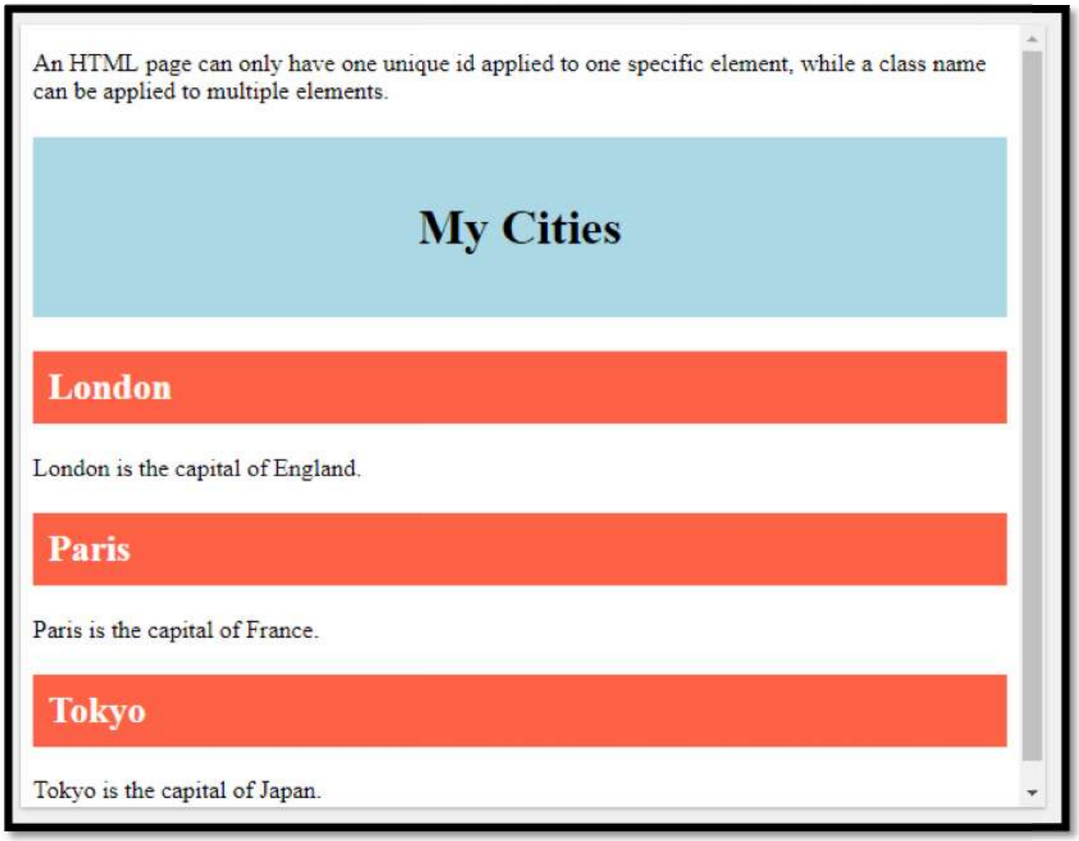
Use CSS to style an element with the id "myHeader":

My Header

Difference Between Class and ID

An HTML element can only have one unique id that belongs to that single element, while a class name can be used by multiple elements:

```
<!DOCTYPE html>
<html>
<head>
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<p>An HTML page can only have one unique id applied to one specific element, while a class name can be applied to multiple elements.</p>

<!-- A unique element -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple similar elements -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>

</body>
</html>
```



An HTML page can only have one unique id applied to one specific element, while a class name can be applied to multiple elements.

My Cities

London

London is the capital of England.

Paris

Paris is the capital of France.

Tokyo

Tokyo is the capital of Japan.

HTML FRAMES :

Iframe Syntax

An HTML iframe is defined with the <iframe> tag:

<iframe src="*URL*"></iframe>

The src attribute specifies the URL (web address) of the inline frame page.

Iframe - Set Height and Width

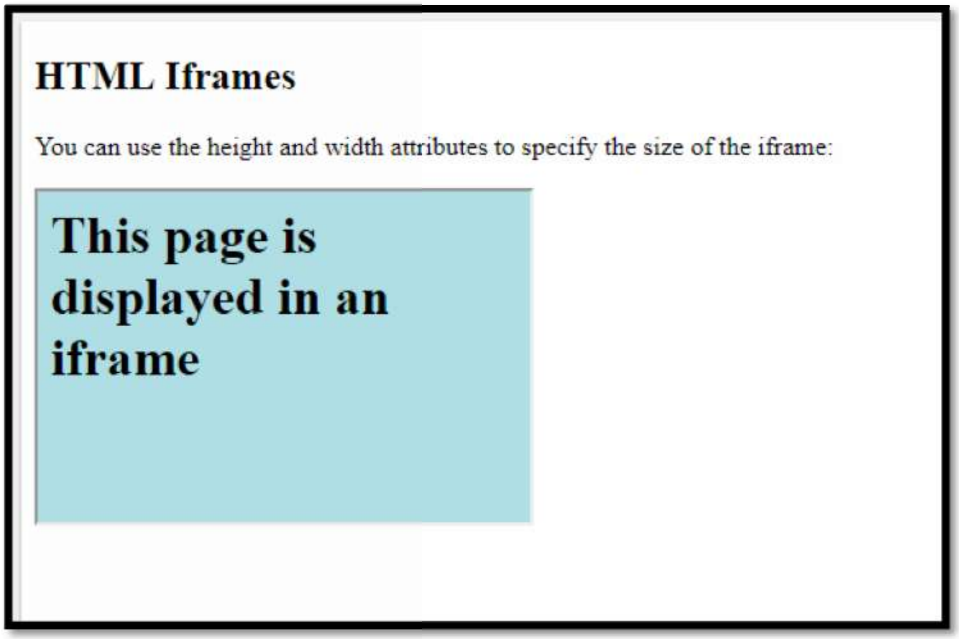
Use the height and width attributes to specify the size of the iframe.

The height and width are specified in pixels by default:

EXAMPLE

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Iframes</h2>
<p>You can use the height and width attributes to specify the size of the iframe:</p>

<iframe src="demo_iframe.htm" height="200" width="300"></iframe>

</body>
```

```
</html>
```



HTML Iframes

You can use the height and width attributes to specify the size of the iframe:

This page is displayed in an iframe

Or you can use CSS to set the height and width of the iframe:

Example

```
<iframe src="demo_iframe.htm" style="height:200px;width:300px;"></iframe>
```

Iframe - Remove the Border

By default, an iframe has a border around it.

To remove the border, add the style attribute and use the CSS border property:

Example

```
<iframe src="demo_iframe.htm" style="border:none;"></iframe>
```

With CSS, you can also change the size, style and color of the iframe's border:

Example

```
<iframe src="demo_iframe.htm" style="border:2px solid red;"></iframe>
```

			UNIT 2			
S.no	**QUESTIONS**	**OPTION 1**	**OPTION 2**	**OPTION 3**	**OPTION 4**	**ANSWER**
1	Which of the following property specifies whether an element is an accelerator indicator or not.	move	@keyframes	accelerator	pointer	accelerator
2	CSS stands for ___________	Color Style Sheets	Cascade Sheets Style	Cascade Style Sheet	Cascading Style Sheets	Cascading Style Sheets
3	In CSS, h1 can be called as ________	Selector	Attribute	Value	Tag	Selector
4	In css, “color:red” can be called as ______________	Selector	Rule	Declaration	Value	Declaration
5	Which selector is used to specify a rule to bind a particular unique element?	id	class	tag	both class and tag	id
6	In CSS, “font-size” can be called as _________	Selector	Rule	Property	Property-Name	Property-Name
7	__________ selectors are used to specify a group of elements.	id	class	tag	both class and tag	class
8	Which of the following tag is used to embed css in html page?	<script>	<style>	<css>	<!DOCTYPE html>	<style>
9	___________ has introduced text, list, box, margin, border, color, and background properties.	css	html	ajax	php	css
10	Which of the following property sets the size of the font?	font-size	font-variant	font-style	font-weight	font-size
11	Which of the following property converts text to initial capitals, all uppercase, or all lowercase?	text-transform	text	text-decoration	text-uppercase	text-transform
12	Which of the following property applies one or more shadows to text?	text-shadow	shadowed	shadow	word-shadow	text-shadow
13	Which of the following property adds padding to the top of an element?	height	padding-height	top	padding-top	padding-height
14	Each cell of the table can be represented by using ___________	<tr>	<td>	<th>	<thead>	<td>
15	For heading we can use _____________	<td>	<tr>	<thead>	<th>	<th>
16	Headings of table lies inside ____________	<thead>	<tfoot>	<th>	<tbody>	<thead>

17	Which of the following is not the element associated with HTML table layout?	size	spanning	alignment	color	color
18	Which of the following element is not associated with a class attribute?	Row	<thead>	Column cell	Rows	<thead>
19	For adding caption to the table we use _____________	<caption>	<thead>	<th>	<tr>	<caption>
20	border-spacing is given in ______________	pixels	cm	mm	inch	pixels
21	Borders can’t be applied on _________________	<th>	<td>	<tr>	<thead>	<tr>
22	Which attribute defines numbers of columns in a group?	width= multi-length[CN].	span=number[CN].	scope=scope-name[CN].	headers=idrefs[CS].	span=number[CN].
23	Which of the following does not specify a column width?	Fixed	Percentage	Proportional	Pixels	Pixels
24	Which of the following is not the value for align attribute?	justify	char	middle	left	middle
25	valign attribute does not take the value ___________________	justify	middle	baseline	bottom	justify
26	Which tag is used for List items?	<li>	<ol>	<ul>	<dl>	<li>
27	Which element contains definition?	<dl>	<dd>	<dt>	<ul>	<dd>
28	Which of the following can’t be the value of list-style-type?	square	circle	ellipse	disc	ellipse
29	For displaying a list horizontally, we can use _______________	<dd>	display:inline	<dt>	type	display:inline
30	Which attribute is only used with <ol>?	value	type	compact	start	start
31	Which element is restricted to inline content?	<dt>	<dd>	<ul>	<dl>	<dt>
32	Which element was designed for creating multicolumn directory lists?	menu	dir	ul	ol	dir
33	The content property does not include ___________	Strings	URIs	Normal	Color	Color
34	Which character is used to differentiating between choices to list items?	‘|’	‘>’	‘#’	‘$’	‘|’
35	In CSS, overline can be called as _______	Selector	Attribute	Value	Tag	Value
36	In css, “body” can be called as ______________	Selector	Rule	Declaration	Value	Selector

37	In CSS, “text-decoration” can be called as _________	Selector	Rule	Property	Property-Name	Property-Name
38	___________ has introduced text properties.	css	html	ajax	php	css
39	Which of the following property sets the color of the font?	font	font-variant	font-color	font-rgb	font
40	Which of the following property places an underline for a given text?	text-transform	text	text-decoration	text-uppercase	text-decoration
41	Which of the following property adds padding to the left of an element?	height	padding-height	top	padding-left	padding-left
42	Each cell of the table can be represented as heading by using ___________	<tr>	<td>	<th>	<thead>	<th>
43	For grouping cells as heading we can use _____________	<td>	<tr>	<thead>	<th>	<thead>
44	Footings of table lies inside ____________	<thead>	<tfoot>	<th>	<tbody>	<tfoot>
45	For adding a row to the table we use _____________	<caption>	<thead>	<th>	<tr>	<th>
46	Which of the following is used to merge columns in a table?	colspan	rowspan	colmerge	rowmerge	colspan
47	valign attribute does not take the value to ____________________ text appearance	justify	middle	baseline	bottom	justify
48	Which tag is used for items that are not numbered or bulletted?	<li>	<ol>	<ul>	<dl>	<dl>
49	Which element contains the term?	<dl>	<dd>	<dt>	<ul>	<dt>
50	Which of the following can’t be the value of list-style-type?	arrow	discus	circle	triangle	circle
51	Which attribute can be used with <ol>?	value	type	compact	start	start
52	DHTML stands for _______	Domain Hyper Text Markup Language	Dual Hyper Text Markup Language	Dynamic Hyper Text Markup Language	None of these	Dynamic Hyper Text Markup Language
53	Which one of the following is difficult to read?	Background color, text color	text color	improper statements	all the above	Background color, text color
54	Images are embedded within a web document page with the use of _______ .	Input	Img tag	text file	http	Img tag
55	For every web page we create, it must have ________ .	<body>	<html>	<title>	</head>	<html>

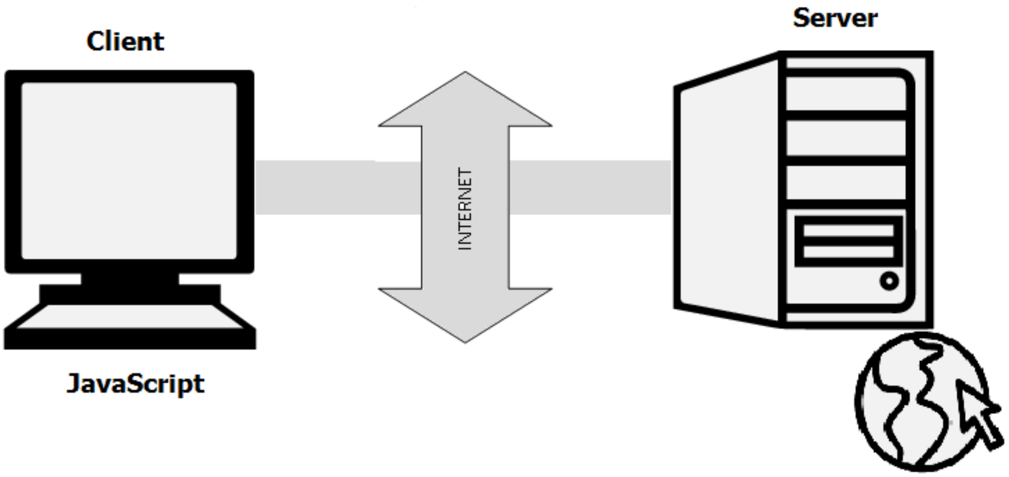
56	Attribute values are ___________ .	Case sensitive	Not case sensitive	Additional information	Optional	Case sensitive
57	___________ are used to select different kinds of user input.	HTML forms	HTML table	HTML lists	None	HTML forms
58	Which of the following options are not object tags attributes?	code base	class id	code type	code class	code base
59	The frames tag should have atleast __________ divisions.	1	2	3	4	2
60	_____ is not the type of input in html forms	text area	button	list	images	images

Unit 3

JAVASCRIPT

What is JavaScript?

JavaScript is a very powerful **client-side scripting language**. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.



JavaScript was developed by Brendan Eich in 1995, which appeared in Netscape, a popular browser of that time.

JavaScript and Java are very much unrelated. Java is a very complex programming language whereas JavaScript is only a scripting language. The syntax of JavaScript is mostly influenced by the programming language C.

How to Run JavaScript?

Being a scripting language, **JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code**. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it. The main advantage of JavaScript is that **all modern web browsers support** JavaScript.

JavaScript will be supported. Also, JavaScript **runs on any operating system** including Windows, Linux or Mac.

```
<html>
  <body>
    <script>

        document.write("Hello World!")
    </script>
```

```
   </body>
</html>
```

JavaScript Variable: Declare, Assign a Value with Example

Variables are used to **store values** (name = "John") **or expressions** (sum = x + y).

Declare Variables in JavaScript

Before using a variable, you first need to declare it. You have to use the keyword **var** to declare a variable like this:

```
var name;
```

Assign a Value to the Variable

You can assign a value to the variable either while declaring the variable or after declaring the variable.

```
var name = "John";
```

OR

```
var name;

name = "John";
```

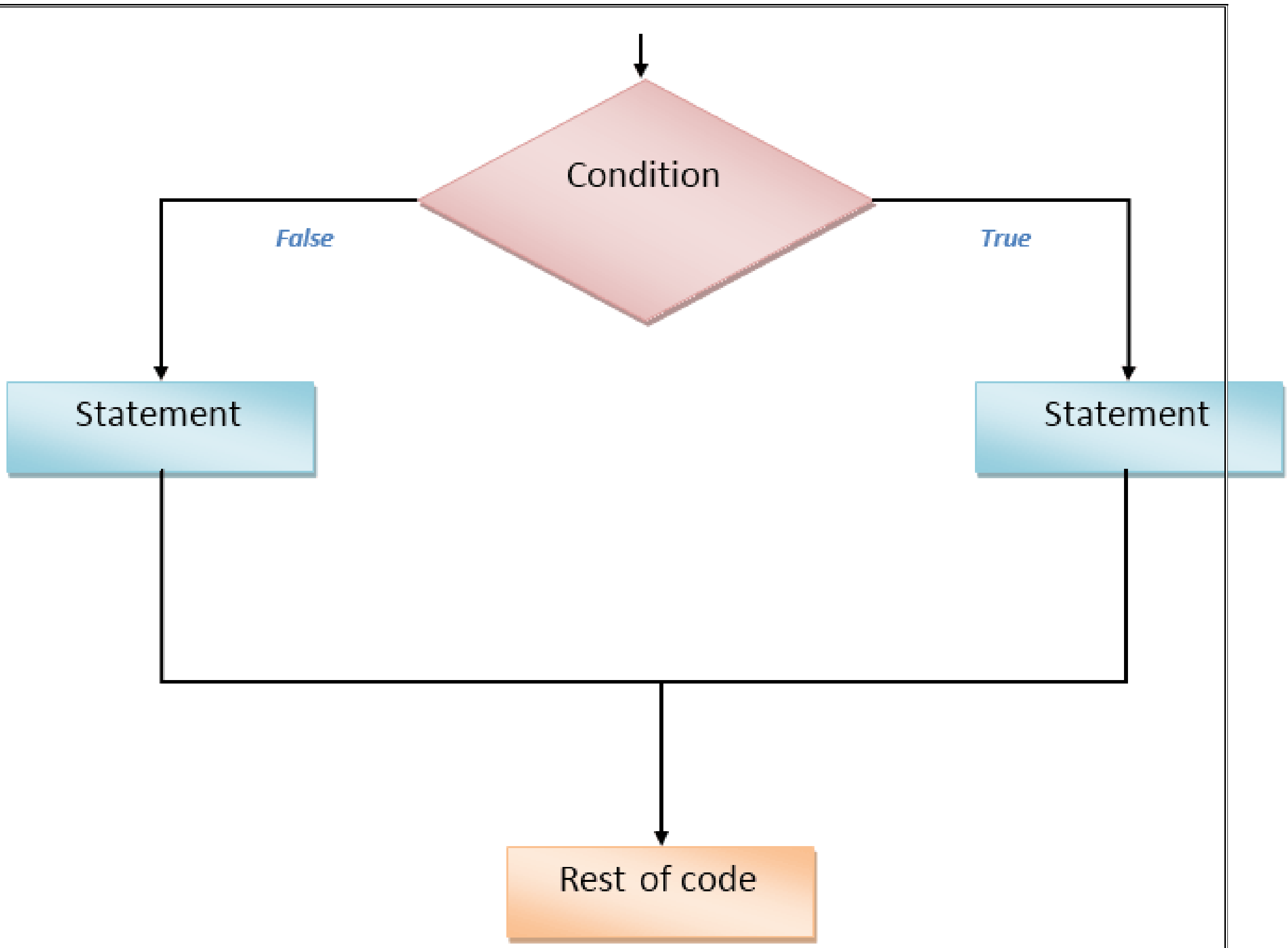
JavaScript Conditional Statements: IF, Else, Else IF (Example)

In this tutorial, you will learn-

- How to use Conditional Statements
- Different Types of Conditional Statements
- If statement
- If…Else statement
- If…Else If…Else statement

How to use Conditional Statements

Conditional statements are used to decide the flow of execution based on different conditions. If a condition is true, you can perform one action and if the condition is false, you can perform another action.



Different Types of Conditional Statements

There are mainly three types of conditional statements in JavaScript.

1. If statement
2. If…Else statement
3. If…Else If…Else statement

If statement

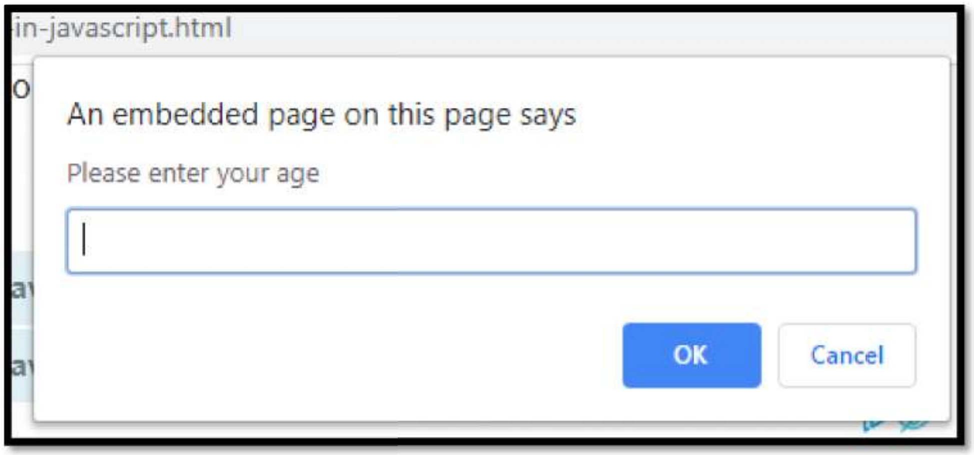
Syntax:

```
if (condition)

{

lines of code to be executed if condition is true

}
```

You can use If statement if you want to check only a specific condition.

<html>

```
<head>
        <title>IF Statments!!!</title>
        <script type="text/javascript">
                var age = prompt("Please enter your age");
                if(age>=18)
                document.write("You are an adult <br />");
                if(age<18)
                document.write("You are NOT an adult <br />");
        </script>
</head>
<body>
</body>
</html>
```


An embedded page on this page says

Please enter your age

If…Else statement

Syntax:

```
if (condition)
{
lines of code to be executed if the condition is true

}
else
{
lines of code to be executed if the condition is false
}
```

You can use If….Else statement if you have to check two conditions and execute a different set of codes.

```
<html>
<head>
        <title>If...Else Statments!!!</title>
        <script type="text/javascript">
                // Get the current hours
                var hours = new Date().getHours();
                if(hours<12)
                document.write("Good Morning!!!<br />");
                else
                document.write("Good Afternoon!!!<br />");
        </script>
</head>
<body>
</body>
```

```
</html>
```

Output :

```
Good Afternoon!!!
```

JAVASCRIPT FUNCTIONS :

What is Function in JavaScript?

Functions are very important and useful in any programming language as they make the code reusable A function is a block of code which will be executed only if it is called. If you have a few lines of code that needs to be used several times, you can create a function including the repeating lines of code and then call the function wherever you want.

How to Create a Function in JavaScript

1. Use the keyword **function** followed by the name of the function.
2. After the function name, open and close parentheses.
3. After parenthesis, open and close curly braces.
4. Within curly braces, write your lines of code.

Syntax:

```
function functionname()

{

  lines of code to be executed

}
```

```
<html>
<head>
        <title>Functions!!!</title>
        <script type="text/javascript">
    function myFunction()
    {
        document.write("This is a simple function.<br />");
    }
                myFunction();
        </script>
</head>
<body>
</body>
</html>
```

Output :

```
This is a simple function.
```

Function with Arguments

You can create functions with arguments as well. Arguments should be specified within parenthesis

Syntax:

```
function functionname(arg1, arg2)

{

  lines of code to be executed

}
```

OPERATORS :

What is an Operator?

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
- Assignment Operators
- Conditional (or ternary) Operators

Lets have a look on all operators one by one.

Arithmetic 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

Note − Addition operator (+) works for Numeric as well as Strings. e.g. "a" + 10 will give "a10".

Example

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

```
<html>
  <body>

    <script type = "text/javascript">
      <!--
        var a = 33;
        var b = 10;
        var c = "Test";
        var linebreak = "<br />";

        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 = ");
        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("--b = ");
        result = --b;
        document.write(result);
        document.write(linebreak);
      //-->
    </script>

    Set the variables to different values and then try...
  </body>
</html>
```

Output

```
a + b = 43
a - b = 23
a / b = 3.3
a % b = 3
a + b + c = 43Test
++a = 35
--b = 8
Set the variables to different values and then try...
```

Comparison Operators

JavaScript supports the following comparison operators −

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

Sr.No.	Operator & Description
1	**= = (Equal)** Checks if the value of two operands are equal or not, if yes, then the condition becomes true. **Ex:** (A == B) is not true.
2	**!= (Not Equal)** Checks if the value of two operands are equal or not, if the values are not equal, then the condition becomes true. **Ex:** (A != B) is true.
3	**> (Greater than)** Checks if the value of the left operand is greater than the value of the right operand, if yes, then the condition becomes true. **Ex:** (A > B) is not true.
4	**< (Less than)** Checks if the value of the left operand is less than the value of the right operand, if yes, then the condition becomes true. **Ex:** (A < B) is true.
5	**>= (Greater than or Equal to)** Checks if the value of the left operand is greater than or equal to the value of the right operand, if yes, then the condition becomes true. **Ex:** (A >= B) is not true.
6	**<= (Less than or Equal to)** Checks if the value of the left operand is less than or equal to the value of the right operand, if yes, then the condition becomes true. **Ex:** (A <= B) is true.

Example

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

```
<html>
  <body>
    <script type = "text/javascript">
      <!--
        var a = 10;
        var b = 20;
        var linebreak = "<br />";

        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) => ");
        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) => ");
        result = (a >= b);
        document.write(result);
        document.write(linebreak);

        document.write("(a <= b) => ");
        result = (a <= b);
        document.write(result);
        document.write(linebreak);
      //-->
    </script>
    Set the variables to different values and different operators and then try...
  </body>
</html>
```

Output

```
(a == b) => false
(a < b) => true
(a > b) => false
(a != b) => true
(a >= b) => false
a <= b) => true
```

Set the variables to different values and different operators and then try...

Logical Operators

JavaScript supports the following logical 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 || 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 />";

        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) => ");
         result = (!(a && b));
         document.write(result);
         document.write(linebreak);
       //-->
     </script>
     <p>Set the variables to different values and different operators and then try...</p>
   </body>
</html>
```

Output

```
(a && b) => false
(a || b) => true
!(a && b) => true
Set the variables to different values and different operators and then try...
```

Bitwise Operators

JavaScript supports the following bitwise operators −

Assume variable A holds 2 and variable B holds 3, then −

Sr.No.	Operator & Description
1	**& (Bitwise AND)** It performs a Boolean AND operation on each bit of its integer arguments. **Ex:** (A & B) is 2.
2	**| (BitWise OR)** It performs a Boolean OR operation on each bit of its integer arguments. **Ex:** (A | B) is 3.
3	**^ (Bitwise XOR)** It performs a Boolean exclusive OR operation on each bit of its integer arguments. Exclusive OR means that either operand one is true or operand two is true, but not both. **Ex:** (A ^ B) is 1.
4	**~ (Bitwise Not)** It is a unary operator and operates by reversing all the bits in the operand. **Ex:** (~B) is -4.

5	**<< (Left Shift)** It moves all the bits in its first operand to the left by the number of places specified in the second operand. New bits are filled with zeros. Shifting a value left by one position is equivalent to multiplying it by 2, shifting two positions is equivalent to multiplying by 4, and so on. **Ex:** (A << 1) is 4.
6	**>> (Right Shift)** Binary Right Shift Operator. The left operand's value is moved right by the number of bits specified by the right operand. **Ex:** (A >> 1) is 1.
7	**>>> (Right shift with Zero)** This operator is just like the >> operator, except that the bits shifted in on the left are always zero. **Ex:** (A >>> 1) is 1.

Example

Try the following code to implement Bitwise operator in JavaScript.

```
<html>
  <body>
    <script type = "text/javascript">
      <!--
        var a = 2; // Bit presentation 10
        var b = 3; // Bit presentation 11
        var linebreak = "<br />";

        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) => ");
        result = (a ^ b);
        document.write(result);
        document.write(linebreak);

        document.write("(~b) => ");
        result = (~b);
        document.write(result);
        document.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);
      //-->
    </script>
    <p>Set the variables to different values and different operators and then try...</p>
  </body>
</html>
```

```
(a & b) => 2
(a | b) => 3
(a ^ b) => 1
(~b) => -4
(a << b) => 16
(a >> b) => 0
Set the variables to different values and different operators and then try...
```

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 -= 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 /= A is equivalent to C = C / 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

Note − Same logic applies to Bitwise operators so they will become like <<=, >>=, >>=, &=, |= and ^=.

Example

Try the following code to implement assignment operator in JavaScript.

```
<html>
  <body>
    <script type = "text/javascript">
      <!--
        var a = 33;
        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 => (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);
      //-->
    </script>
    <p>Set the variables to different values and different operators and then try...</p>
  </body>
</html>
```

Output

```
Value of a => (a = b) => 10
Value of a => (a += b) => 20
Value of a => (a -= b) => 10
Value of a => (a *= b) => 100
Value of a => (a /= b) => 10
Value of a => (a %= b) => 0
Set the variables to different values and different operators and then try...
```

Miscellaneous Operator

We will discuss two operators here that are quite useful in JavaScript: the **conditional operator** (? :) and the **typeof operator**.

Conditional Operator (? :)

The conditional operator first evaluates an expression for a true or false value and then executes one of the two given statements depending upon the result of the evaluation.

Sr.No.	**Operator and Description**
1	**? : (Conditional)** If Condition is true? Then value X : Otherwise value Y

Example

Try the following code to understand how the Conditional Operator works in JavaScript.

```
<html>
  <body>
    <script type = "text/javascript">
      <!--
        var a = 10;
        var b = 20;
        var linebreak = "<br />";

        document.write ("((a > b) ? 100 : 200) => ");
        result = (a > b) ? 100 : 200;
        document.write(result);
        document.write(linebreak);

        document.write ("((a < b) ? 100 : 200) => ");
        result = (a < b) ? 100 : 200;
        document.write(result);
        document.write(linebreak);
      //-->
    </script>
    <p>Set the variables to different values and different operators and then try...</p>
  </body>
</html>
```

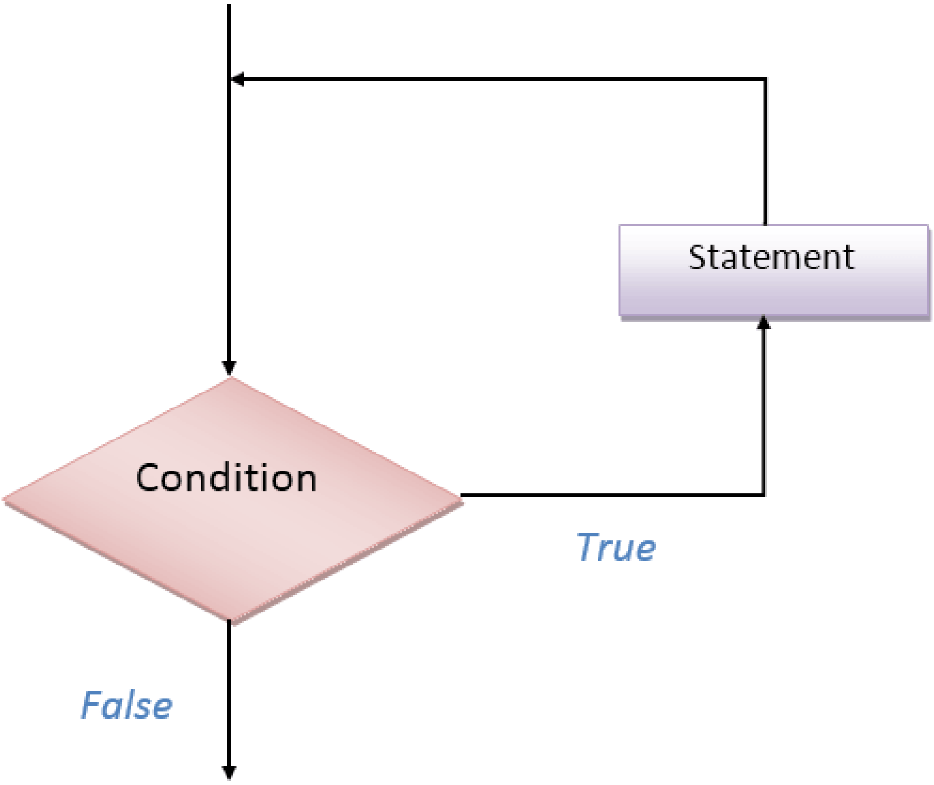
Output

```
((a > b) ? 100 : 200) => 200
((a < b) ? 100 : 200) => 100
Set the variables to different values and different operators and then try...
```

For, While and Do While LOOP in JavaScript (with Example)

How to use Loop?

Loops are useful when you have to execute the same lines of code repeatedly, for a specific number of times or as long as a specific condition is true. Suppose you want to type a ‘Hello’ message 100 times in your webpage. Of course, you will have to copy and paste the same line 100 times. Instead, if you use loops, you can complete this task in just 3 or 4 lines.



Different Types of Loops

There are mainly four types of loops in JavaScript.

1. for loop
2. for/in a loop (explained later)
3. while loop
4. do…while loop

for loop

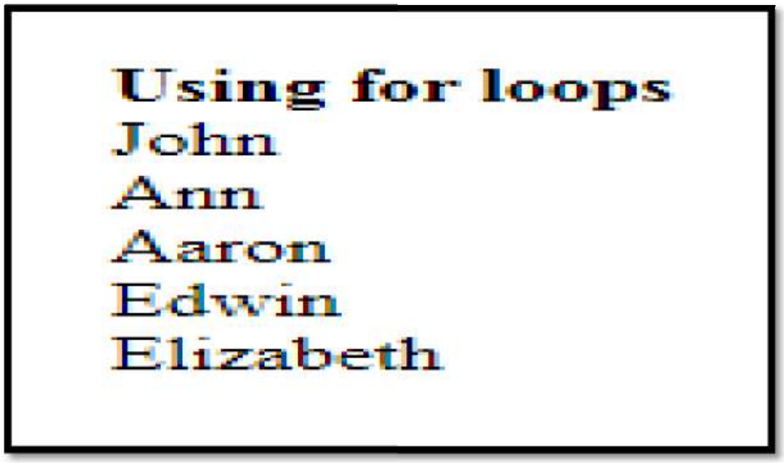
Syntax:

```
for(statement1; statement2; statment3)

{

lines of code to be executed

}
```

1. The statement1 is executed first even before executing the looping code. So, this statement is normally used to assign values to variables that will be used inside the loop.
2. The statement2 is the condition to execute the loop.
3. The statement3 is executed every time after the looping code is executed.

Try this yourself:

```
<html>
<head>
    <script type="text/javascript">
        var students = new Array("John", "Ann", "Aaron", "Edwin", "Elizabeth");
        document.write("<b>Using for loops </b><br />");
        for (i=0;i<students.length;i++)
        {
        document.write(students[i] + "<br />");
        }
    </script>
</head>
<body>
</body>
</html>
```



```
Using for loops
John
Ann
Aaron
Edwin
Elizabeth
```

while loop
Syntax:

```
while(condition)

{

lines of code to be executed

}
```

The “while loop” is executed as long as the specified condition is true. Inside the while loop, you should include the statement that will end the loop at some point of time. Otherwise, your loop will never end and your browser may crash.

```
<html>
<head>
    <script type="text/javascript">
        document.write("<b>Using while loops </b><br />");
        var i = 0, j = 1, k;
        document.write("Fibonacci series less than 40<br />");
        while(i<40)
        {
            document.write(i + "<br />");
            k = i+j;
            i = j;
            j = k;
        }
    </script>
</head>
<body>
</body>
</html>
```

Output:

```
Using while loops
Fibonacci series less than 40
0
1
1
2
3
5
8
13
21
34
```

do…while loop

Syntax:

```
do

{

block of code to be executed

} while (condition)
```

The do…while loop is very similar to while loop. The only difference is that in do…while loop, the block of code gets executed once even before checking the condition.

```
<html>
<head>
    <script type="text/javascript">
        document.write("<b>Using do...while loops </b><br />");
        var i = 2;
        document.write("Even numbers less than 20<br />");
        do
        {
            document.write(i + "<br />");
            i = i + 2;
        }while(i<20)
    </script>
```

```
</head>
<body>
</body>
</html>
```

Using do...while loops

```
Even numbers less than 20
2
4
6
8
10
12
14
16
18
```

JavaScript Arrays

What is an Array?

An array is an object that can store a **collection of items**. Arrays become really useful when you need to store large amounts of data of the same type. Suppose you want to store details of 500 employees. If you are using variables, you will have to create 500 variables whereas you can do the same with a single array. You can access the items in an array by referring to its **indexnumber** and the index of the first element of an array is zero.

JavaScript Create Array

You can create an array in JavaScript as given below.

```
var students = ["John", "Ann", "Kevin"];
```

Here, you are initializing your array as and when it is created with values "John", "Ann" and "Kevin".The index of "John", "Ann" and "Kevin" is 0, 1 and 2 respectively. If you want to add more elements to the students array, you can do it like this:

```
students[3] = "Emma";
students[4] = "Rose";
```

You can also create an array using Array constructor like this:

```
var students = new Array("John", "Ann", "Kevin");
```

OR

```
var students = new Array();
```

```
students[0] = "John";

students[1] = "Ann";

students[2] = "Kevin";
```

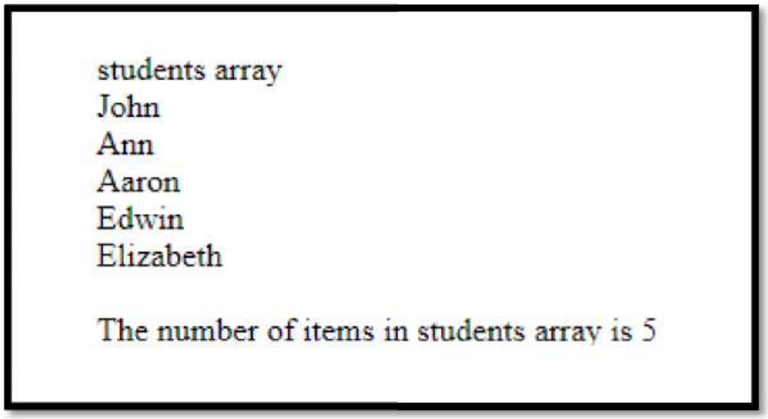
JavaScript Array Methods

The Array object has many properties and methods which help developers to handle arrays easily and efficiently. You can get the value of a property by specifying arrayname.property and the output of a method by specifying arrayname.method().

1. **length property** --> If you want to know the number of elements in an array, you can use the length property.
2. **prototype property** --> If you want to add new properties and methods, you can use the prototype property.
3. **reverse method** --> You can reverse the order of items in an array using a reverse method.
4. **sort method -->** You can sort the items in an array using sort method.
5. **pop method** --> You can remove the last item of an array using a pop method.
6. **shift method** --> You can remove the first item of an array using shift method.
7. **push method** --> You can add a value as the last item of the array.

```
<html>
<head>
    <title>Arrays!!!</title>
    <script type="text/javascript">
        var students = new Array("John", "Ann", "Aaron", "Edwin", "Elizabeth");
        Array.prototype.displayItems=function(){
            for (i=0;i<this.length;i++){
                document.write(this[i] + "<br />");
            }
        }
        document.write("students array<br />");
        students.displayItems();
        document.write("<br />The number of items in students array is " +
students.length + "<br />");
        document.write("<br />The SORTED students array<br />");
        students.sort();
        students.displayItems();
        document.write("<br />The REVERSED students array<br />");
        students.reverse();
        students.displayItems();
        document.write("<br />THE students array after REMOVING the LAST
item<br />");
        students.pop();
        students.displayItems();
   document.write("<br />THE students array after PUSH<br />");
   students.push("New Stuff");
        students.displayItems();
    </script>
</head>
```

```
<body>
</body>
</html>
```



```
students array
John
Ann
Aaron
Edwin
Elizabeth

The number of items in students array is 5
```

			UNIT 3			
S. no	QUESTIONS	OPTION 1	OPTION 2	OPTION 3	OPTION 4	ANSWER
1	javaScript Code can be called by using ___________	RMI	Triggering Event	Preprocessor	Function/Method	Function/Method
2	The type of a variable that is volatile is _______________	Volatile variable	Mutable variable	Immutable variable	Dynamic variable	Mutable variable
3	JavaScript _________ when there is an indefinite or an infinite value during an arithmetic computation.	Prints an exception error	Prints an overflow error	Displays "Infinity"	Prints the value as such	Displays "Infinity"
4	Which of the following is not considered as an error in JavaScript?	Syntax error	Missing of semicolons	Division by zero	Missing of Bracket	Division by zero
5	The escape sequence '\f' stands for _________	Floating numbers	Representation of functions that returns a value	\f is not present in JavaScript	Form feed	Form feed
6	The statement a===b refers to _________	Both a and b are equal in value, type and reference address	Both a and b are equal in value	Both a and b are equal in value and type	There is no such statement	Both a and b are equal in value and type
7	A function definition expression can be called as __________	Function prototype	Function literal	Function calling	Function declaration	Function literal
8	The property of a primary expression is ____________	stand-alone expressions	basic expressions containing all necessary functions	contains variable references alone	contains only keywords	stand-alone expressions
9	Which of the operator is used to test if a particular property exists or not?	in	exist	within	exists	in

10	JavaScript is a ________________ language.	Object-Oriented	High-level	Assembly-language	Object-Based	Object-Based
11	A conditional expression is also called a _________________	Alternative to if-else	Immediate if	If-then-else statement	Switch statement	Immediate if
12	What is a block statement in JavaScript?	conditional block	block that contains a single statement	both conditional block and a single statement	block that combines multiple statements into a single compound statement	block that combines multiple statements into a single compound statement
13	When an empty statement is encountered, a JavaScript interpreter ___________	Ignores the statement	Prompts to complete the statement	Throws an error	Shows a warning	Ignores the statement
14	The “var” and “function” are ___________	Keywords	Declaration statements	Data types	Prototypes	Declaration statements
15	The enumeration order becomes implementation dependent and non-interoperable if ____________	If the object inherits enumerable properties	The object does not have the properties present in the integer array indices	The delete keyword is never used	Object.defineProperty() is not used	If the object inherits enumerable properties
16	What are the three important manipulations done in a for loop on a loop variable?	Updation, Incrementation, Initialization	Initialization,Testing, Updation	Testing, Updation, Testing	Initialization,Testing, Incrementation	Initialization,Testing, Updation
17	One of the special features of an interpreter in reference with the for loop is that ____________	Before each iteration, the interpreter evaluates the variable expression and assigns the name of the property	The iterations can be infinite when an interpreter is used	The body of the loop is executed only once	the iteration is finite when an interpreter is used	Before each iteration, the interpreter evaluates the variable expression and assigns the name of the property
18	What will happen if the body of a for/in loop deletes	The property will be stored in a cache	The loop will not run	That property	The property will be enumerated	That property will not be enumerated

	a property that has not yet been enumerated?			will not be enumerated		
19	Among the keywords below, which one is not a statement?	debugger	with	if	use strict	use strict
20	What will happen if reverse() and join() methods are used simultaneously?	Reverses and stores in the same array	Reverses and concatenates the elements of the array	Reverses	Stores the elements of an array in normal order	Reverses and stores in the same array
21	The primary purpose of the array map() function is that it __________	maps the elements of another array into itself	passes each element of the array and returns the necessary mapped elements	passes each element of the array on which it is invoked to the function you specify, and returns an array containing the values returned by that function	pass the elements of the array into another array	passes each element of the array on which it is invoked to the function you specify, and returns an array containing the values returned by that function
22	What are the three important manipulations done in a for loop on a loop variable?	Updation, Incrementation, Initialization	Initialization,Testing, Updation	Testing, Updation, Testing	Initialization,Testing, Incrementation	Initialization,Testing, Updation
23	The ……………. operator is known as the equality operator, which checks whether its two operators are “equal” using a more relaxed definition of sameness that allows type conversion.	=	= =	= = =	All of the above	= =

24	The ……………………. operator is known as strict equality operator, and it checks whether two operands are “identical” using a strict definition of sameness.	=	= =	= = =	All of the above	= = =
25	The …………………. returns false if two values are equal to each other according to = = and returns true otherwise.	!=	!	!= =	All of the above	!=
26	JavaScript is a ________ side scripting language.	Server	Client	Browser	ISP	Client
27	Array are stored in _____________ memory.	Random	same	in sequence	as per user data	same
28	Functions should have ____________.	declaration	call of function	call of values	call of data	declaration
29	Functions are known as _________ .	sub program	main program	arrays	datas	sub program
30	Data and time can be declared as ___________ data type .	String	Array	Var	date and time	date and time
31	Which type of data type is not declared as var?	String	Arrays	Numbers	Characters	String
32	The amount of space between the cells in the table refers to __________.	cell spacing	cell padding	table	cell	cell spacing
33	______ is not a unconditional branching statement.	Break	Continue	Goto	Switch	Switch
34	Which loop executes the statement and then checks on the condition?	While	Do while	For	Switch	Do while

35	Which is not a increment decrement operator?	1+i	i++	i- -	--i	1+i
36	The value of I—is	i=i-1	1=1-i	i=1-1	i=i-i	i=i-1
37	The value of ++I is	i=i+1	i=1+i	i=1+1	i=1-i	i=1+i
38	Test is performed at the of the for loop.	Top	Middle	End	Program terminates	Top
39	The “<<” is known as ______ .	insertion operator	extraction operator	lesser	greater	insertion operator
40	In case of strings '+' is used to _______	separate the first operand and the second operand	reduces the data size	increase the data size	combines the first operand and the second operand	combines the first operand and the second operand
41	What is the representation of the third element in an array called a?	a[2]	a(2)	a{2}	a(3)	a[2]
42	Suppose x=10 and y=10 what is x after evaluating the expression (y > 10) & (x++ > 10).	9	10	11	12	11
43	The "less than or equal to" comparison operator in JavaScript is ______.	<	<=	>	<<	<=
44	Which of the following expression results in a value 1?	2 % 1	15 % 4	9%9	37 % 6	37 % 6
45	The equal comparison operator in JavaScript is _________ .	<>	!=	“-“	= =	= =
46	The function is used to print a string in italic.	fixed( )	fontcolor( )	fontsize( )	italic ()	italic ()
47	The ___________ statement is used to print a line using JavaScript.	window. write( )	document. write( )	navigator. write( )	task. write( )	document. write( )
48	Integer Variable is declared using syntax in JavaScript.	var num	Integer num;	int num;	number num;	var num

49	Variable can hold value at a time.	Double	Triple	Single	Multiple	Single
50	___________ statement is used to declare variable in JavaScript.	Assignment	Declaration	Executable	Conditional	Declaration
51	Java Statement terminated by ________.	Slash	Semicolon	Comma	Full stop	Semicolon
52	JavaScript statements are executed by ______.	Server	Compiler	JVM	Browser	JVM
53	Executable single line of script in JavaScript is known as ____________ ..	Statement	Breakpoint	Line	Code	Statement
54	How to write an IF statement in JavaScript?	if(i==5)	if i=5	if i==5 then	if i=5 then	if(i==5)
55	JavaScript code is placed inside __________ HTML element.	<scripting>	<javascript>	<script>	<js>	<script>
56	JavaScript can be written _____________ .	directly into HTML page	directly into JS file	directly on the server script	included separatel y to HTML	directly into HTML page
57	How does a FOR loop start?	for i = 1 to 5	for (i = 0; i <= 5)	for (i <= 5; i++)	for (i = 0; i <= 5; i++)	for (i = 0; i <= 5; i++)
58	JavaScript is designed to ____________ .	style HTML page	add interactivity to HTML page	perform server side scripting	execute query related to DB on server	add interactivity to HTML page
59	The statements of JavaScript has multiple statements that should be enclosed within ___.	script tag	title tag	hr tag	font tag	script tag
60	The parameters of script tag are __________ and _________.	language and type	language and range	type and char	language and char	language and type

			UNIT 4			
S.no	**QUESTIONS**	**OPTION 1**	**OPTION 2**	**OPTION 3**	**OPTION 4**	**ANSWER**
1	What is the default type of ‘type’ attribute of <input> element?	Text	Password	Numerals	Special Characters	Text
2	Which of the following is a new input attribute introduce by HTML5?	text	checkbox controls	submit buttons	date	date
3	How does the color attribute work?	Changes color of the text	Changes background color	The color picker is defined by it	Changes color of the text as well as background	The color picker is defined by it
4	Which attribute is used for activation of JavaScript?	button	checkbox	url	submit	button
5	Which attribute defines the file-select field?	file	checkbox	button	text	file
6	How image attribute works?	Sets an image background	Set an image as submit button	Set an image anywhere on the page	Bring default image to the page	Set an image as submit button
7	month attribute defines ____________	the only month	month and year	date	date and time	month and year
8	week attribute defines _____________	week	year	week and year	week, month and year	week and year
9	tel attribute is supported by the __________ browser.	Chrome	Safari	Opera	Internet Explorer	Safari
10	Which attribute is not used on new forms?	size	text	name	maxlength	size
11	Which of the following is not used with password attribute?	name	size	maxlength	min	min
12	Which element is used to create multi-line text input?	text	textarea	submit	radio button	textarea
13	Which attribute is not used for the radio type?	name	value	checked	selected	selected
14	Which attribute is used with <select> element?	multiple	selected	name	value	multiple
15	What can be used to verify a displayed graphics?	Property checkpoints	Region checkpoints	On-Screen Action	Local Storage	Region checkpoints

16	Which element has been removed from HTML5 (W3 specification?	<figure>	<nav>	<hgroup>	<aside>	<hgroup>
17	Which of the following is used for plug-in content?	<embed>	<progress>	<meter>	<source>	<embed>
18	Which of the following does not lie under the attribute of <input> and is not supported by Opera?	url	time	tel	option	option
19	Which attribute does not lie inside <a> and <area> elements?	Media	Ping	Hreflang	Preload	Preload
20	Which element does not support autofocus attribute?	<input>	<select>	<textarea>	<base>	<base>
21	Which element does not support form attribute?	<input>	<output>	<meta>	<button>	<meta>
22	Which attribute of <input> element can be used both with <datalist> and <select> elements?	List	Pattern	Multiple	Max	List
23	Which is not a Boolean attribute?	Multiple	Novalidate	Formvalidate	Formtarget	Formtarget
24	What is the work of async attribute?	It influences script loading	It gives a label to the menu	It validates form data	Enables a set of extra restrictions on any content	It influences script loading
25	Which is not <iframe> attribute?	Seamless	Srcdoc	Sizes	Sandbox	Sizes
26	Which is not a new global attribute in HTML5?	Contenteditable	Contextmenu	Title	Draggable	Title
27	To specify number intervals for numeric type input which attribute is used?	Interval	Step	Limit	None	Step
28	SGML stands for	Standardized Global Markup Language	Standard Generalized Markup Language	Social General Markup Language	None of them	Standard Generalized Markup Language
29	HTML tags are?	Case sensitive	Not case sensitive	lowercase stricter	Uppercase stricter	Not case sensitive
30	Which of the following is not a type of attribute for input tag?	day	week	month	time	day

31	The new ___________ element is supposed to represent some form of extra details, such as a tooltip or revealed region that may be shown to a user.	progress	meter	details	menu	details
32	The ___________ attribute effectively renders the iframe as an inline include, which allows the parent document’s CSS to affect the contents of the iframe.	allow-forms	seamless	embed	allow-scripts	seamless
33	Which of the following allows the sandboxed iframe to run scripts from the same domain?	allow-same-origin	allow-forms	allow-scripts	allow-pointer-lock	allow-scripts
34	___________ allows the iframe to pull in content from elsewhere in the same domain.	allow-same-origin	allow-forms	allow-scripts	allow-pointer-lock	allow-same-origin
35	Which of the following is not a HTML5 added form element?	<datalist>	<keygen>	<output>	<password>	<password>
36	Which element specifies a list of pre-defined options for an input element?	<datalist>	<keygen>	<output>	<password>	<datalist>
37	Which of the following defines a group of related options in a drop-down list?	<form>	<optgroup>	<output>	<option>	<optgroup>
38	In HTML form <input type="text"> is used for	One line text	Block of text	One paragraph	None	One line text
39	HTML classes that is already defined and allow us to apply styles on it are called as	Pseudo classes	Css classes	Javascript classes	None	Pseudo classes
40	Which of following is not an inline element?	<span>	<a>	<img>	<div>	<div>
41	Which is the new attribute of <input> element that is used to change the appearance of checkbox?	Indeterminate	Media	EI	Target	Media
42	Which attribute does not lie inside <a> and <area> elements?	Media	Ping	Hreflang	Preload	Preload
43	Which of the following gives a text description of the image if it is not available?	alt	title	src	height	alt

44	Which of the following is not the form type for adding text?	Text input	Text area	Password input	Submit button	Submit button
45	In the processing of information, the server does not use the language _____	C#	JAVA	C++	VB.net	C++
46	For creating single line text box for searching queries, we use the type ___________	placeholder	search	url	hidden	search
47	Form validation traditionally was performed by ___________	PHP	HTML	JavaScript	jQuery	JavaScript
48	For grouping form controls we can use ___________	<legend>	<fieldset>	<label>	<select>	<fieldset>
49	Which of the following is a new input type in HTMl 5?	Button	Text	Address	Date	Date
50	For checking a regular expression in input fields of HTML forms which attribute is used?	<checker>	<valid input>	<pattern>	None of them	<pattern>
51	Which HTML element is used to define a multi-line input field?	<text>	<textarea>	<blocktext>	<textfields>	<textarea>
52	Which variable is used to collect form data sent with both the GET and POST methods?	$BOTH	$_BOTH	$REQUEST	$_REQUEST	$_REQUEST
53	When you use the $_POST variable to collect data, the data is visible to..	none	only you	everyone	selected few	only you
54	<fieldset> element in HTML forms is used to	Group related data	Group irrelevant data	Text field	None	Group related data
55	The HTML5 attribute, …………………. is used in <textarea> element should be set to a string that corresponds to the id of the form element that an interactive control such as a button is associated with.	form	pattern	placeholder	control	form
56	…………………. is the HTML5 attribute that specifies a regular expression against which the field should be validated.	form	pattern	validate	control	pattern

57	…………………… is the HTML5 attribute that specifies a short bit of text that is used to help the user figure out what type of information to fill in for a form control.	form	pattern	placeholder	control	placeholder
58	<textarea> element cannot be descendant of a/an ………………………… element.	form	div	button	article	button
59	………………….. element is used to group the rows within the footer of a table so that common alignment and style defaults can easily be set for numerous cells.	<tgroup>	<tfoot>	<grows>	<frows>	<tfoot>
60	The textarea element traditionally lacks a ………………………………. attribute, which causes a more obvious security risk.	length	maxlength	poster	autoplay	maxlength

			UNIT 5			
S.no	**QUESTIONS**	**OPTION 1**	**OPTION 2**	**OPTION 3**	**OPTION 4**	**ANSWER**
1	Which side of the image map can be created using JavaScript?	Server side	Client side	Both Server and Client side	User side	Client side
2	Which is the attribute used to enable the Client-side image map?	map	area	use map	areamap	use map
3	Which are the special tags used for image mapping?	map and area	map and usemap	only map	only usemap	map and area
4	Which is the element that follows the use of "img"?	area	usemap	map	any element can follow the use of "img"	map
5	What is the purpose of the area element?	Area of the text	Shape and coordinates of the hotspot	Shape and area of the hotspot	Coordinates and area	Shape and coordinates of the hotspot
6	Which of the following is not a navigator property?	platform[]	plugin[]	userAgent[]	browser[]	browser[]
7	What is the purpose of the platform[] property in a navigator?	Platform of the script	Platform where the image map was designed	Platform where the browser was compiled	Platform where the plugin was designed	Platform where the browser was compiled
8	What is the purpose of the preference method in a navigator?	Set Browser preference	Set Netscape preference	Both Set Browser & Netscape preference	Sets user preference	Set Netscape preference
9	Which protocol is supported by Android browsers?	HTTPS	HLS	RTMP	FTP	HLS
10	___________ is a JavaScript library that implements the most common user interface elements and interactions like	JavaScript	JQuery UI	VTS	JCL	JQuery UI

	sliders, accordions, tabs, and so on.				
11	In HTML Audio/Video DOM, __________ sets or returns whether the audio/video should be loaded when the page loads.	preload	autoplay	buffered	controller	preload
12	What will happen if height and width of video are not set while video loads?	page flickers	page does not load	page crash	page closes	page flickers
13	Which of the following HTML Video – Media Type is not supported in IE?	WebM	MP4	Ogg	MP4 FLAC	Ogg
14	In HTML Audio/Video DOM, __________ sets or returns the default speed of the audio/video playback.	currentTime	duration	defaultPlaybackRate	playbackRate	defaultPlaybackRate
15	In HTML Audio/Video DOM, __________ sets or returns the CORS settings of the audio/video.	currentTime	duration	defaultPlaybackRate	crossOrigin	crossOrigin
16	In HTML Audio/Video DOM, __________ returns a TimeRanges object representing the buffered parts of the audio/video.	preload	networkState	buffered	controller	buffered
17	What does ‘On-Screen Action’ means in the testing of HTML5 applications?	Using Drag action	Using new input type	Using play and pause action of audio and video elements	Verifying SVG and Canvas elements	Using Drag action
18	What can be used to verify a displayed graphics?	Property checkpoints	Region checkpoints	On-Screen Action	Local Storage	Region checkpoints

19	Which element has been removed from HTML5 (W3 specification?	<figure>	<nav>	<hgroup>	<aside>	<hgroup>
20	Which of the following is used for plug-in content?	<embed>	<progress>	<meter>	<source>	<embed>
21	Which of the following plug-in can't be embedded by <object> element?	Java applets	Flash players	PDF readers	Map	Map
22	Which protocol is supported by Android browsers?	HTTPS	HLS	RTMP	FTP	HLS
23	___________ is a JavaScript library that implements the most common user interface elements and interactions like sliders, accordions, tabs, and so on.	JavaScript	JQuery UI	VTS	JCL	JQuery UI
24	In HTML Audio/Video DOM, ___________ sets or returns whether the audio/video should be loaded when the page loads.	preload	autoplay	buffered	controller	preload
25	Which of the following is not a HTML5 tag?	<video>	<source>	<track>	<slider>	<slider>
26	What will happen if height and width of video are not set while video loads?	page flickers	page does not load	page crash	page closes	page flickers
27	Which of the following MP3 player has not been written in Flash?	musicplayer.sourceforge.net	www.wimpyplayer.com	flash-mp3-player.net	soundcloud.com	soundcloud.com
28	Which of the following statement is not true?	SVG stands for Scalable Vector Graphics	SVG is used to define graphics for the Web	SVG is a W3C recommendation	SVG doesn't support event handlers	SVG doesn't support event handlers

29	To draw on the canvas, authors must first obtain a reference to a context using the _______________ method of the canvas interface element.	getImageData	toDataURL	getContext	restore	getContext
30	What can be used to verify a displayed graphics?	Property checkpoints	Region checkpoints	On-Screen Action	Local Storage	Region checkpoints
31	Which of the following is not the attribute for <audio> element?	controls	src	preload	width	width
32	What is the work of controls?	specify the path to an audio file	indicates if the player displaying controls	audio starts playing automatically	play again after finishing the audio	indicates if the player displaying controls
33	Which element is used for inserting more than one audio file?	<source>	<src>	<command>	<ins>	<source>
34	What is the work of src?	audio starts playing automatically	play again after finishing the audio	specify the path to an audio file	insert more than one audio	specify the path to an audio file
35	Which of the following file extension is not used for audio MIME?	.ogv	.aac	.wav	.webm	.ogv
36	Which format is a restricted version of Matroska?	WebM	Ogg Opus	Ogg Flac	MP4 Flac	WebM
37	Which format is not supported in Internet Explorer?	WebM	Ogg Theora Vorbis	MP4 H.264	MP3	Ogg Theora Vorbis
38	MP3 format is not supported by _____________	Firefox	Chrome	Safari	Opera	Opera
39	Which of the following is not a Boolean attribute for <audio> element?	autoplay	loop	muted	buffered	buffered

40	What can not be done by controls attribute?	playback	resume playback	pause playback	looping	looping
41	Which of the following is not the property of SVG images?	SVG images are scalable	SVG images are zoomable	SVG is an open standard	SVG images are resolution dependent	SVG images are resolution dependent
42	________ drawings can be dynamic and interactive.	Canvas based	SVG	CSS3	JavaScript	SVG
43	Which of the following is the predefined shape elements that can be used by developers?	Path <path>	Rectangle <rect>	Circle <circle>	Line <lin>	Line <lin>
44	Which element must reference a resource that can provide an image for the cursor graphic?	hover	i	cursor	fill	cursor
45	Which element serves as a container for atomic filter operations?	filter	feimage	feblend	tref	filter
46	The ___________ element must reference either an ‘altGlyphDef’ element or a ‘glyph’ element.	href	src	animate	altGlyph	altGlyph
47	The ____________ element must reference a ‘linearGradient’ or ‘radialGradient’ element.	script	radialGradient	animateTransform	clip-Path	radialGradient
48	Which property of SVG restricts the region to which paint can be applied?	animateTransform	clip-Path	linearGradient	radialGradient	clip-Path
49	Which SVG element produces the same effect as if the nodes were deeply cloned into a non-exposed DOM?	tref	use	stroke	bin	use

50	___________ can be used to advise the browser to download media content in the background to improve playback.	poster	autobuffer	buffer	data-X	autobuffer
51	The __________ element is used to render simple graphics such as line art, graphs, and other custom graphical elements on the client side.	metadata	css	canvas	art	canvas
52	Which plugins can provide the fallback support for old browsers?	Flash	Quicktime	Both Flash and Quicktime	Fireback and Quickertime	Both Flash and Quicktime
53	Which of the following is not the promises of the open web platform?	Security and Privacy	Performance and Tuning	Media and Real-Time Communications	Device Interconnection	Device Interconnection
54	Which of the following elements in HTML5 defines video or movie content?	<media>	<video>	<movie>	<audio>	<video>
55	For exporting movie into SWF format which element is used?	<object>	<video>	<datalist>	<dd>	<object>
56	which extension flash file is to be saved?	.fla	.swf	.jpg	.gif	.fla
57	Which plugin is used to view Flash?	Firebug	Flash Player	Widget	Zotero	Flash Player
58	Which one of the following is not used inside the <script> tag in Flash videos?	replace	location	version	id	id
59	Which one of the following is not the online video format available?	Ogg Theora	VHS	WebM	H264	VHS

60	For playing Flash Video you need to convert video into the ____________ format.	FLV	BlueRay	WebM	MPEG	FLV

HTML PROGRAMMING
(18CCU403B)

UNIT - V

Images

Most of the material about imagees is covered in the tutorial

at http://catcode.com/immgguide/. Please read that tutorial before continuing.

Making Images Clickable

To make an image clickable, just enclose it in an <a> element. Here is an exampple of the icon for the W3C linking to the W3C home page.

And here is the markup:

```
<a href="http://w3.orgg/"><img src="screenshots/w3book.png" alt="book with
        W3C on cover" width="49" height="611" /></a>
```

You can also make a small “thummbnail” picture link to a larger version. (Click the picture; you will see a larger version.)

And here is the markup:

```
<a href="screenshots/mmarco.jpg"><img
src="screenshots/marco_th.jpg"
    alt="white kitten liicking its paw"
    width="199" height=""150"/></a>
```

ImageMaps

Note: This discussion of ImageMMaps is not in the book, and the assignment does not require it, but it is interesting information.

Sometimes you to use an image for navigation and have different parts of the image go to different locations. Move your mouse around the image below and look in the status bar at the bottom of the browser window to see where a click will lead you:
This is called an **imagemap**, because it maps sections of your image to different URLS. Setting up an imagemap by hand is a multi-step process.

Create a <map> element

All of the information about the imagemap will be contained within a <map> element. Put the

tag into your document (it can go anywhere; usually imagemaps go at the bottom of the
document), and think up a name for your imagemap. This name will be the value of the
map’s name attribute. In this example, we'll call our imagemap buttonmap. In XHTML, you

must also have an id attribute, and it must have the same value as the name attribute.

```
<map name="buttonmap" id="buttonmap">
        <!-- imagemap info here -->
</map>
```

Identify the coordinates of the active sections of the image

Now figure out which areas of the image will be clickable, and where they are located within the image. If you open the image in a program like Adobe Photoshop on Windows or the GNU Image Manipulation Program on Linux, you can move a pointer around the image and see the cursor’s **coordinates**. Coordinates are a pair of numbers; the first number tells how many pixels from the left the cursor is, and the second tells how many pixels down from the top the cursor is. You can see the coordinates of the arrow in the lower left of this picture:

Make a table of the upper left and lower right coordinates of the active areas of your image, and where they should go when clicked:

Coordinates		
Upper left	**Lower right**	**Destination**
7, 7	103, 83	http://www.linux.org/
139, 7	223, 83	http://www.openoffice.org/
62, 88	145, 150	http://slashdot.org/
207, 88	297, 150	http://www.gphoto.org/

Create <area> elements

Each of these table entries will become a rectangular area. You use the <area> element with an attribute of shape="rect" and a coords="..." attribute to describe the area. You may also have an alt="..." attribute to make the information accessible to blind users. You may also use a title="..." attribute to create a tooltip.

Here is what the imagemap will look like once you have entered all the information. When you enter the coordinates, you may not put blanks after the commmas.

```
<map name="buttonmap" id="buttonmap">
<area shape="rect" coords="7,8,103,83"
href="http://www.linux.org/"
        alt="Linux information" title="Linux information" /> <area
    shape="rect" coords="139,8,223,83"
href="http://www.openoffice.org/"
        alt="Free office suite" title="Free office suite" /> <area
    shape="rect" coords="62,88,145,150"
href="http://slashdot.org/"
        alt="News for nerds - stuff that matters" title="News for
        nerds - stuff that matters" />

    <area shape="rect" coords="207,88,297,150"
```

```
href="http://www.gphoto.org/"
          alt="Connect your digital camera to Linux" title="Connect
          your digital camera to Linux" />

</map>
```

The shape attribute tells what shape the area has; rect is obviously the easiest. There are other possibilities.

Connect your image to the imagemap

Add a usemap attribute to your <img> element; its value will be the URL of the imagemap.

Since the imagemap is in the same file as the image, you will use a # in the URL:

```
<img src="screenshots/imagemap.png" usemap="#buttonmap"
        alt="Image map with four buttons"
        width="311" height="157"      border="0" />
```

Imagemap shapes

Here is a summary of all the values for the shape attribute inside of an <area>:

shape	coords value example
shape="rect"	Coordinates of upper left and lower right of a rectangular coords="7,7,25,82" area, separated by commas
shape="circle"	Coordinates of the center of a circular area, followed by the radius of the area. The example coords="60,40,20" shows a 20-pixel circular area centered at coordinates (60, 40). An arbitrary polygon; you list the coordinates of each vertex of the polygon. The example hape="poly" coords="10,10,30,30,50,10" shows a triangular area ith vertices at (10,10), (30,30), and (50,10)
shape="default"	No coordinates needed; this describes a click anywhere in the image that is not occupied N/A by other <area> specifications.

GRAPHICS SVG

SVG stands for **S**calable **V**ector **G**raphics and it is a language for describing 2D-graphics and graphical applications in XML and the XML is then rendered by an SVG viewer.

SVG is mostly useful for vector type diagrams like Pie charts, Two-dimensional graphs in an X, Y coordinate system etc.

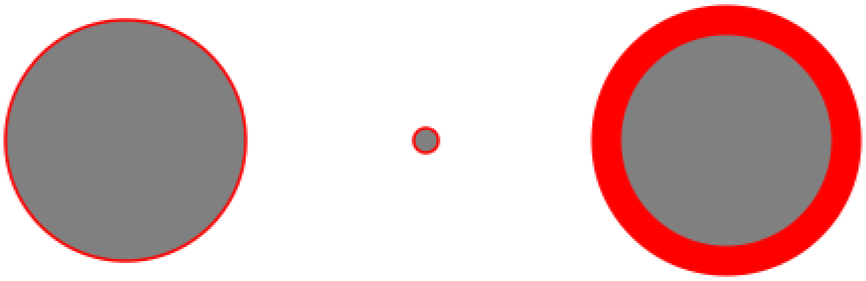
HTML5 allows embedding SVG directly using **<svg>...</svg>** tag which has following simple syntax :

```
<svg xmlns = "http://www.w3.org/2000/svg">
   ...
</svg
```

HTML5 − SVG Circle

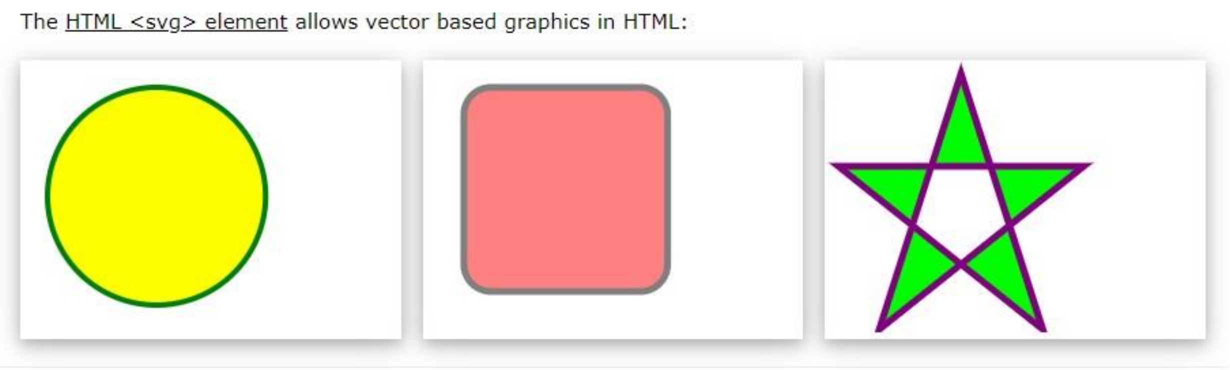
Following is the HTML5 version of an SVG example which would draw a circle using <circle> tag

```
<svg viewBox="0 0 300 100" xmlns="http://www.w3.org/2000/svg" stroke="red"
fill="grey">
  <circle cx="50" cy="50" r="40" />
  <circle cx="150" cy="50" r="4" />

  <svg viewBox="0 0 10 10" x="200" width="100">
    <circle cx="5" cy="5" r="4" />
  </svg>
</svg>
```



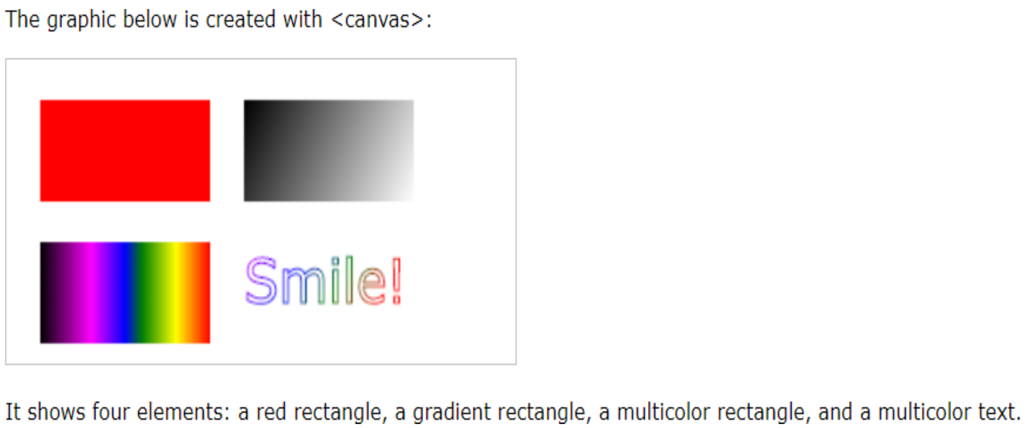
HTML Canvas

The HTML <canvas> element can be used to draw graphics on a web page:



The HTML <svg> element allows vector based graphics in HTML:

The graphic below is created with <canvas>:
It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.



The graphic below is created with <canvas>:

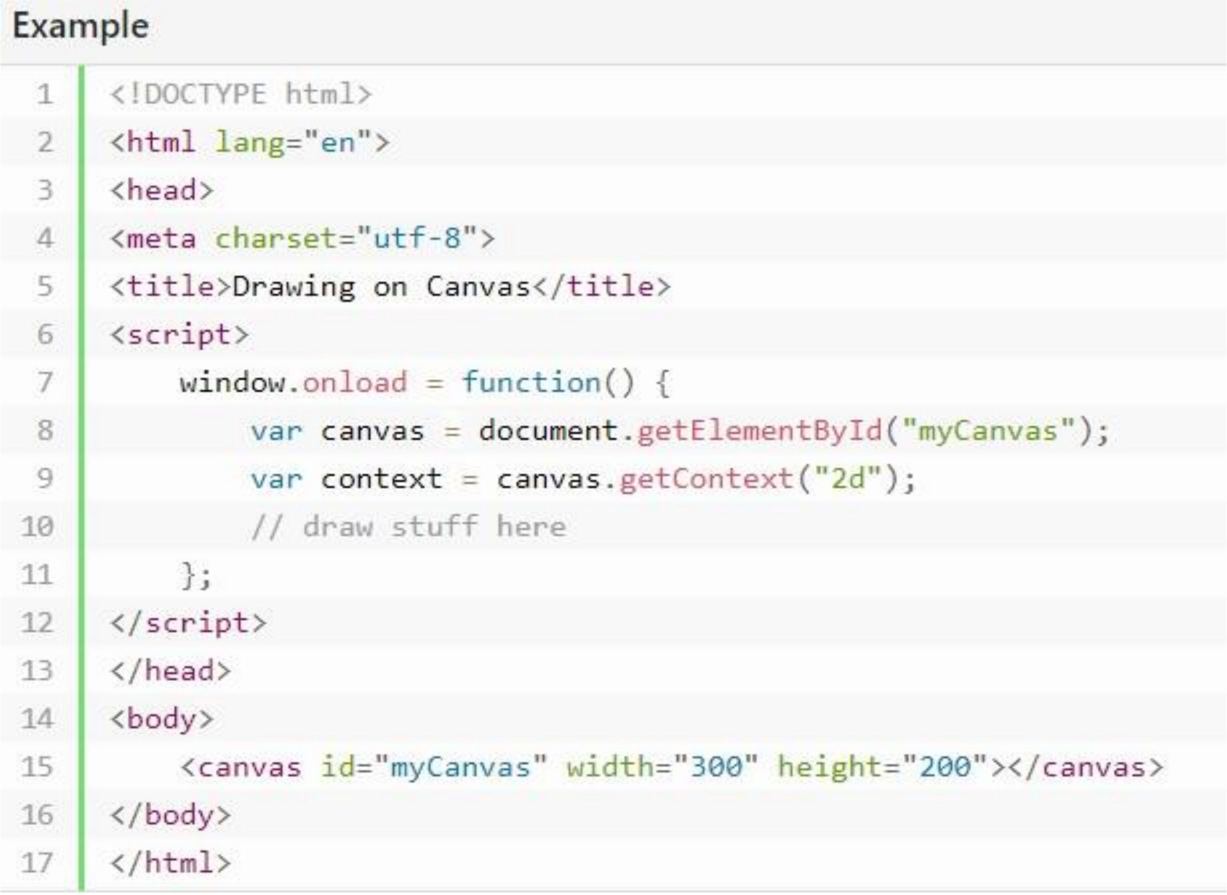
It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

The HTML5 canvas element can be used to draw graphics on the webpage via JavaScript. The canvas was originally introduced by Apple for the Mac OS dashboard widgets and to power graphics in the Safari web browser. Later it was adopted by the Firefox, Google Chrome and Opera. Now the canvas is a part of the new HTML5 specification for next generation web technologies.
By default the <canvas> element has 300px of width and 150px of height without any border and content. However, custom width and height can be defined using the CSS height and width property whereas the border can be applied using the CSS border property.

Canvas Coordinates

The canvas is a two-dimensional rectangular area. The coordinates of the top-left corner of the canvas are (0, 0) which is known as origin, and the coordinates of the bottom-right corner are (*canvas width*, *canvas height*). Here's a simple demonstration of canvas default coordinate system.

Here is the base template for drawing paths and shapes onto the 2D HTML5 canvas.



Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Drawing on Canvas</title>
<script>
    window.onload = function() {
        var canvas = document.getElementById("myCanvas");
        var context = canvas.getContext("2d");
        // draw stuff here
    };
</script>
</head>
<body>
    <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```

Differences between SVG and Canvas

At first sight, both SVG and Canvas are doing the same thing, drawing vector artwork using coordinate points. However, there are differences between them. Let’s have a look at them.

SVG is a language for describing 2D graphics in XML, whereas Canvas is used to draw 2D graphics on the fly (with JavaScript).

SVG is like a "draw" program. The drawing is a drawing instruction for each shape, and any part of any shape can be changed. Drawings are shape-oriented.

Canvas is like a "paint" program. When the pixels hit the screen, that is your drawing. You cannot change shapes except by overwriting them with other pixels.

Paintings are pixel-oriented.
SVG is XML based, which means that every element is available within the SVG DOM. In SVG, the drawn shape is remembered as an object. In Canvas, the browser forgets the drawn form immediately after it has been drawn. If you need to make changes in the drawing, you should draw it from scratch.

MEDIA TAGS

The media attribute specifies what media/device the linked document is optimized for. This attribute is used to specify that the target URL is designed for special devices (like iPhone) , speech or print media. This attribute can accept several values.

Audio on the Web

Before HTML5, audio files could only be played in a browser with a plug-in (like flash). The HTML5 <audio> element specifies a standard way to embed audio in a web page. To play an audio file in HTML, use the <audio> element:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
/audio>
```

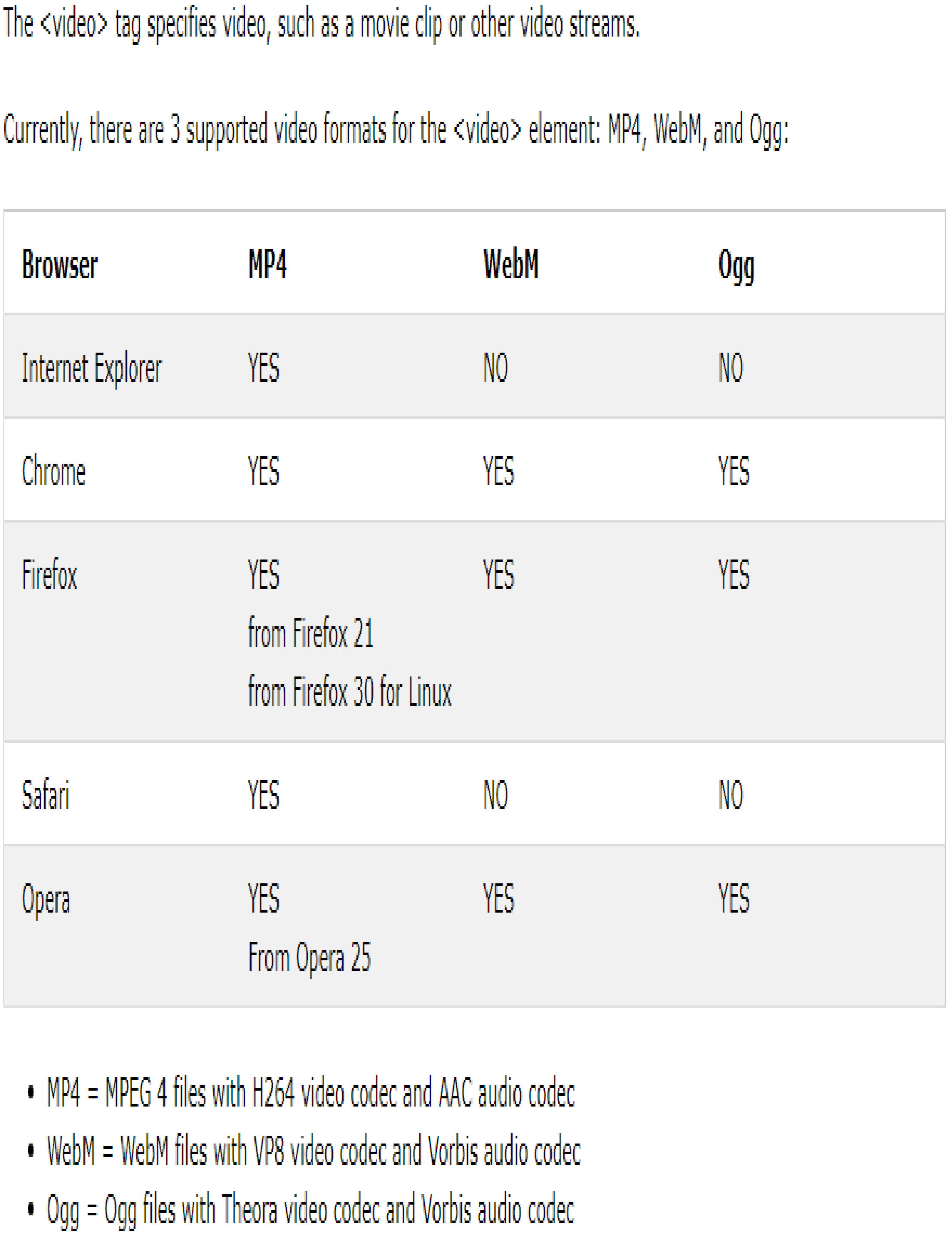
The controls attribute adds audio controls, like play, pause, and volume.

The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

HTML <video> Tag

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
  </video>
```



The <video> tag specifies video, such as a movie clip or other video streams.

Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg:

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES

Firefox	YES
	from Firefox 21
	from Firefox 30 for Linux
		YES	YES

Safari	YES	NO	NO
Opera	YES From Opera 25	YES	YES

- MP4 = MPEG 4 files with H264 video codec and AAC audio codec
- WebM = WebM files with VP8 video codec and Vorbis audio codec
- Ogg = Ogg files with Theora video codec and Vorbis audio codec

HTML Plug-ins

The purpose of a plug-in is to extend the functionality of a web browser.
HTML Helpers (Plug-ins)
Helper applications (plug-ins) are computer programs that extend the standard functionality of a web browser.

Examples of well-known plug-ins are Java applets.

Plug-ins can be added to web pages with the <object> tag or the <embed> tag.

Plug-ins can be used for many purposes: display maps, scan for viruses, verify your bank id, etc.

Plugins are great resources because they allow your site to **do** things it otherwise wouldn't be able to **do**. But the more **plugins** you have, the greater the risk of problems occurring within your site. **Plugin** problems can be random and unexpected, or more specific issues related to **plugin** or WordPress upgrades.

The <object> Element

The <object> element is supported by all browsers.

The <object> element defines an embedded object within an HTML document.

It is used to embed plug-ins (like Java applets, PDF readers, Flash Players) in web pages.

Example :

```
a. <object width="100%" height="500px" data="snippet.html"></object>
b. <object width="400" height="50" data="bookmark.swf"></object>
```

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 action="Script URL" method="GET|POST">
    form elements like input, textarea etc.
</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes:

Attribute	Description
action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
enctype	You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are: • **application/x-www-form-urlencoded** - This is the standard method most forms use in simple scenarios. • **mutlipart/form-data** - This is used when you want to upload binary data in the form of files like image, word file etc.

Note: You can refer to **Perl & CGI** for a detail on how form data upload works.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form:

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls

- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms:

- **Single-line text input controls -** This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.
- **Password input controls -** This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
- **Multi-line text input controls -** This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name:

```
<!DOCTYPE html>
<html>
<head>
<title>Text Input Control</title>
</head>
<body>
<form >
First name:  <input type="text" name="first_name" />
<br>
Last name:  <input type="text" name="last_name" />
</form>
</body>
</html>
```

This will produce following result:

First name:
Last name:

Attributes

Following is the list of attributes for <input> tag for creating text field.

Attribute	Description
type	Indicates the type of input control and for text input control it will be set to **text**.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	This can be used to provide an initial value inside the control.
size	Allows to specify the width of the text-input control in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box.

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input> tag but type attribute is set to **password**.

Example

Here is a basic example of a single-line password input used to take user password:

```
<!DOCTYPE html>
<html>
<head>
<title>Password Input Control</title>
</head>
<body>
<form >
User ID :  <input type="text" name="user_id" />
<br>
Password:  <input type="password" name="password" />
</form>
</body>
</html>
```

This will produce following result:

User ID :
Password:

Attributes

Following is the list of attributes for <input> tag for creating password field.

Attribute	Description
type	Indicates the type of input control and for password input control it will be set to **password.**
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	This can be used to provide an initial value inside the control.
size	Allows to specify the width of the text-input control in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box.

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description:

```
<!DOCTYPE html>
<html>
<head>
<title>Multiple-Line Input Control</title>
</head>
```

```
<body>
<form>
Description : <br />
<textarea rows="5" cols="50" name="description">
Enter description here...
</textarea>
</form>
</body>
</html>
```

This will produce following result:

Description :
Enter description here...

Attributes

Following is the list of attributes for <textarea> tag.

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
rows	Indicates the number of rows of text area box.
cols	Indicates the number of columns of text area box

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **checkbox**.

Example

Here is an example HTML code for a form with two checkboxes:

```
<!DOCTYPE html>
<html>
<head>
<title>Checkbox Control</title>
</head>
<body>
<form>
<input type="checkbox" name="maths" value="on"> Maths
<input type="checkbox" name="physics" value="on"> Physics
</form>
</body>
</html>
```

This will produce following result:

Maths Physics

Attributes

Following is the list of attributes for <checkbox> tag.

Attribute	Description
type	Indicates the type of input control and for checkbox input control it will be set to **checkbox.**
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the checkbox is selected.
checked	Set to *checked* if you want to select it by default.

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **radio**.

Example

Here is example HTML code for a form with two radio buttons:

```
<!DOCTYPE html>
<html>
<head>
<title>Radio Box Control</title>
</head>
<body>
<form>
<input type="radio" name="subject" value="maths"> Maths
<input type="radio" name="subject" value="physics"> Physics
</form>
</body>
</html>
```

This will produce following result:

Maths Physics

Attributes

Following is the list of attributes for radio button.

Attribute	Description
type	Indicates the type of input control and for checkbox input control it will be set to **radio.**
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the radio box is selected.
checked	Set to *checked* if you want to select it by default.

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

```
<!DOCTYPE html>
<html>
<head>
<title>Select Box Control</title>
</head>
<body>
<form>
<select name="dropdown">
<option value="Maths" selected>Maths</option>
<option value="Physics">Physics</option>
</select>
</form>
</body>
</html>
```

This will produce following result:

Maths

Attributes

Following is the list of important attributes of <select> tag:

value	The value that will be used if an option in the select box box is selected.
selected	Specifies that this option should be the initially selected value when the page loads.
label	An alternative way of labeling options

File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to **file**.

Example

Here is example HTML code for a form with one file upload box:

```
<!DOCTYPE html>
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
```

```
<form>
<input type="file" name="fileupload" accept="image/*" />
</form>
</body>
</html>
```

This will produce following result:

Attributes

Following is the list of important attributes of file upload box:

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
accept	Specifies the types of files that the server accepts.

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input> tag by setting its type attribute to **button**. The type attribute can take the following values:

Type	Description
submit	This creates a button that automatically submits a form.
reset	This creates a button that automatically resets form controls to their initial values.
button	This creates a button that is used to trigger a client-side script when the user clicks that button.
image	This creates a clickable button but we can use an image as background of the button.

Example

Here is example HTML code for a form with three types of buttons:

```
<!DOCTYPE html>
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
<form>
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset"  value="Reset" />
<input type="button" name="ok" value="OK"  />
<input type="image" name="imagebutton" src="/html/images/logo.png" />
</form>
</body>
</html>
```

This will produce following result:

Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page has be displayed next based on the passed current page.

Example

Here is example HTML code to show the usage of hidden control:

```
<!DOCTYPE html>
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
<form>
<p>This is page 10</p>
<input type="hidden" name="pagename" value="10" />
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset"  value="Reset" />
</form>
</body>
</html>
```

This will produce following result:

This is page 10