WEB TECHNOLOGY

Sikhinam Nagamani
Assistant Professor
Computer Science & Engineering
RGUKT- Ongole-AP

Unit 1- Syllabus

Unit - I

Introduction to Web World: Recap on HTML, inserting Frames and frame sets, inserting hyperlinks, lists, tables and images,

JavaScript : Client side scripting with JavaScript, variables, functions, conditions, loops and repetition, Pop up boxes, objects, HTML DOM and web, Browser environments, form validation, Events and Event Listeners

INTRODUCTION

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages.

HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by Tim Berners-Lee in 1991.

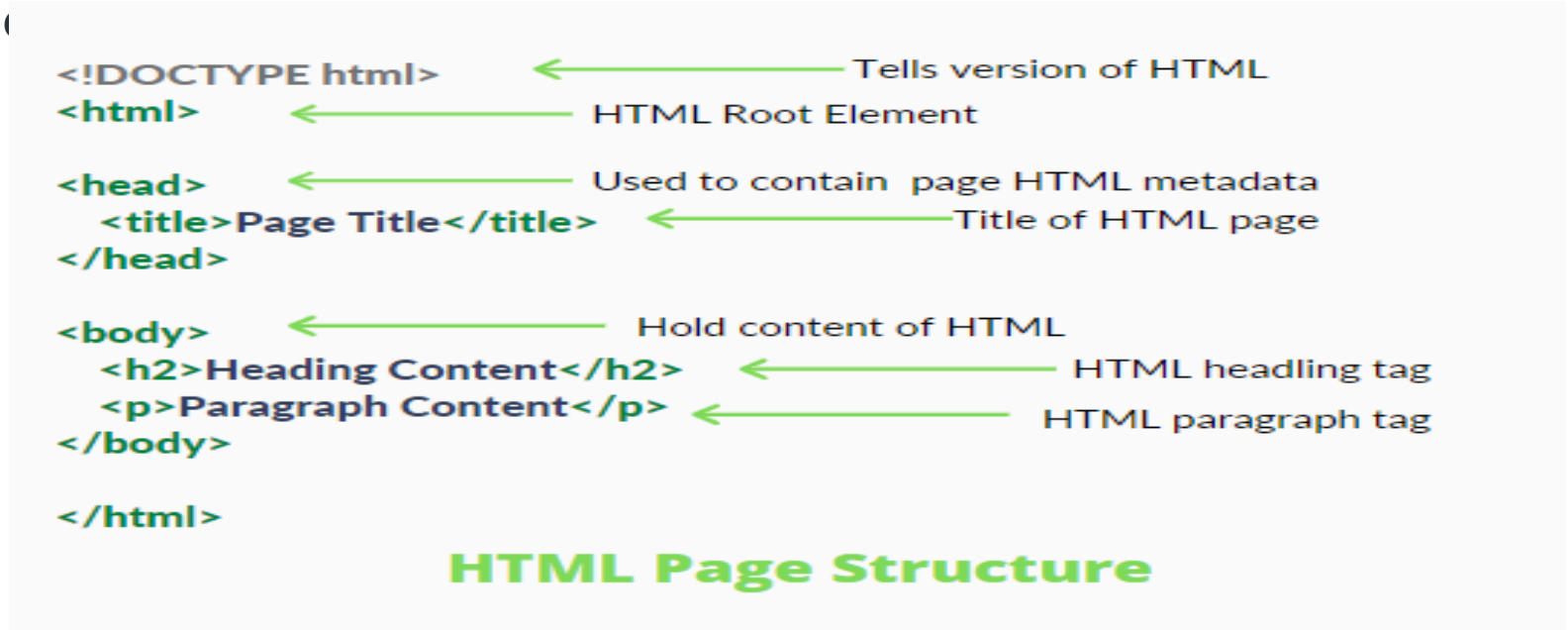
INTRODUCTION

Elements and Tags: HTML uses predefined tags and elements which tell the browser how to properly display the content.

Web Technology

INTRODUCTION

HTML page structure: The basic structure of an HTML page contains the essential building-block elements (i.e. doctype declaration, HTML, head, titl



INTRODUCTION

<DOCTYPE! html>: This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declaration is not case-sensitive.

<html>: This is called the HTML root element. All other elements are contained within it.

<head>: The head tag contains the “behind the scenes” elements for a webpage. Elements within the head aren’t visible on the front-end of a webpage. HTML elements used inside the <head> element include:

<style> <title> <base> <noscript> <script>
<meta> <link>

<body>: The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front-end.

An HTML document can be created using any text editor. Save the text file using **.html** or **.htm**. Once saved as an HTML document, the file can be opened as a webpage in the browser.

INTRODUCTION

```
<!DOCTYPE html>
<html>
<head>
    <title>Demo Web Page</title>
</head>

<body>
    <h1>Web Technology</h1>

<p>A computer science portal </p>

</body>
</html>
```

INTRODUCTION

Features of HTML:

- It is easy to learn and easy to use.
- Images, videos, and audio can be added to a web page.
- Hypertext can be added to the text.
- It is a markup language.

Why learn HTML?

- It is a simple markup language. Its implementation is easy.
- It is used to create a website.
- Helps in developing fundamentals about web programming.

INTRODUCTION

Advantages:

- HTML is used to build websites.
- It is supported by all browsers.
- It can be integrated with other languages like CSS, JavaScript, etc.

Disadvantages:

- HTML can only create static web pages. For dynamic web pages, other languages have to be used.
- A large amount of code has to be written to create a simple web page.
- The security feature is not good.

INTRODUCTION

HTML text editors

HTML text editors are used to create and modify web pages. HTML codes can be written in any text editors including the **notepad**. One just needs to write HTML in any text editor and save the file with an extension ".html". Some of the popular HTML text editors are given below:

- Notepad
- Notepad++
- Atom

HTML COMMENTS

HTML Comments

The comment tag (<!– Comment –>) is used to insert comments in the HTML code. It is a good practice of coding so that coder and the reader can get help to understand the code. It is helpful to understand the complex code. The comment tag is useful during the debugging of codes.

It helps the coder and reader to understand the piece of code used for especially in complex source code.

Types of HTML Comments: There are three types of comments in HTML which are:

- Single-line comment
- Multi-lines comment
- Using <comment> tag

HTML COMMENTS

```
<!DOCTYPE html>
<html>
<body>
        <!--This is single line comment -->
        <h2>This is single line comment</h2>
                <!– This is
                        Multi Line
                        Commen-->
                <comment>This is the Heading Tag<comment>
                </body>
                </html>
```

HTML BASICS

Basic HTML Document: Below mentioned are the basic HTML tags that divide the whole document into various parts like head, body, etc.

Every HTML document begins with a HTML document tag. Although this is not mandatory, it is a good convention to start the document with this below-mentioned tag.

<html> :

<head>:

<title>:

<body>:

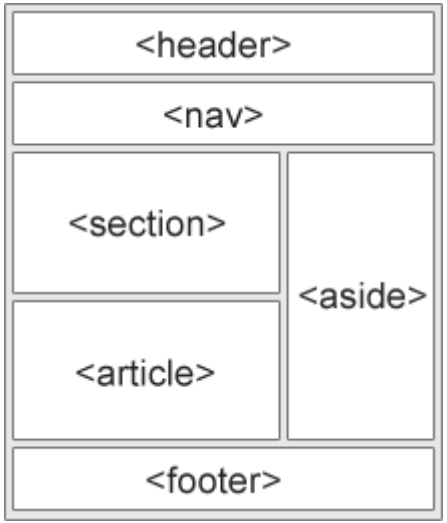
HTML Headings:

HTML Paragraph:

HTML Horizontal Lines

HTML Images:

HTML Layout



- <header> - Defines a header for a document or a section
- <nav> - Defines a set of navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent, self-contained content
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section

HTML Elements

An HTML element is the collection of start and end tags with the content inserted in between them.

<tagname> contents... </tagname>

Example 1: In this example is a starting tag, is an ending tag and it contains some content between the tags, which form an element.

```
<!DOCTYPE html>
<html>
        <head>
        <title>HTML Elements</title>
        </head>
<body>
        <h2>Welcome To WT</h2>

<p>Hi Every one!</p>

</body>

</html>
```

HTML Elements

Nested HTML Elements: The HTML element is use inside the another HTML Element is called nested HTML elements.

Example : This example describes the use of the Nested HTML elements. Here, <html> tag contains the <head> and <body>. The <head> and <body> tag contains another elements so it is called nested element.

Necessary to add end tag: It is necessary to add the end tag of an element. Otherwise, the displayed content may or may not be displayed correctly.

HTML Elements

Empty HTML Elements: HTML Elements without any content i.e, that do not print anything are called Empty elements. Empty HTML elements do not have an ending tag. For instance.
, <hr>, <link>, <input> etc are HTML elements.

```
<!DOCTYPE html>
<html>
<head>
        <title>Empty HTML Elements</title>
</head>
<body>
        <h2>Welcome To GfG</h2>
        <br />
<p>Hello Geeks.</p>

</body>
</html>
```

HTML Heading

An HTML heading tag is used to define the headings of a page. There are six levels of headings defined by HTML. These 6 heading elements are H1, H2, H3, H4, H5, and H6; with H1 being the highest level and H6 the least.

Syntax: <h1> Hi </h1>

Importance of Heading:

- Search Engines use headings for indexing the structure and content of the webpage.
- Headings are used for highlighting important topics.

HTML Heading

Changing the size of HTML Headings: The default size of HTML headings can be changed using the style attribute.

```
<!DOCTYPE html>
<html>

<body>
        <!-- Style attribute is used here-->
        <h1 style="font-size: 50px">H1 with new
size.</h1>
</body>

</html>
```

Horizontal rule: The <hr> tag is an empty tag, and it does not require an end tag. It is basically used to separate content.

```
<!DOCTYPE html>
<html>
<body>
        <h1>Heading 1</h1>
<p>I like HTML.</p>
        <!-- hr Tag is used here-->
        <hr />
        <h2>Heading 2</h2>
<p>I like CSS.</p>
        <!-- hr Tag is used here-->
        <hr />
        <h2>Heading 3</h2>
<p>I like Javascript.</p>
</body>
</html>
```

HTML Paragraphs

The ** ** tag in HTML defines a paragraph. These have both opening and closing tags. So anything mentioned within ** ** and ** ** is treated as a paragraph.

- As already mentioned, the tag automatically adds space before and after any paragraph, which is basically margins added by the browser.
- If a user adds multiple spaces, the browser reduces them to a single space.
- If a user adds multiple lines, the browser reduces them to a single line.

HTML Paragraphs

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

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

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

<p align="value">

```
<p align="center">Welcome </p>
```

<pre> tag: We have seen how the paragraph tag ignores all the changes 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 s

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

</body>
```

HTML Text Formatting

HTML facilitates the ability for formatting text just like we do in MS Word or any text editing software.

```
<body>
    <h2>Welcome To Web Technology</h2>
    <!--Text in Strong-->
    <strong>Hello Geeks</strong>
    <br>
    <!--Text in small-->
    <small>Hello Geeks</small>
    <br>
    <!--Text in Highlight-->
    <mark>Hello Geeks</mark>
</body>
```

HTML Text Formatting

Making text Bold or Strong:

```
<!--Text in Bold-->
  <p>
     <b> WelCome </b>
  </p>
  <!--Text in Strong-->
  <p>
     <strong> WelCome </strong>
  </p>
```

Making text *Italic* or *emphasiz*

```
<!--Text in Italics-->
    <p>
       <i>WelCome</i>
    </p>
    <!--Text in Emphasize-->
    <p>
       <em> WelCome </em>
    </p>
<!--Text in Highlight-->
<p>
<mark> WelCome </mark>
</p>
```

HTML Text Formatting

Making a text Subscript or Superscript:

```
<!--Text in Superscript-->
<p>Hello  <sup> WelCome </sup>
</p>

<!--Text in Subscript-->
<p>Hello  <sub> WelCome </sub>
</p>
```

Making text smaller:

```
<!--Text in small--
<p><small>Hello Web Technology</small>
</p>
```

Striking through the text:

```
<!--Text in Delete-->
<p> <del>Hello Web Technology</del>
</p>
```

HTML | Quotations

<q> element:

The <q> element is used to set a set of text inside the quotation marks

It has both open

```
<!--Inside quotes-->
<p><q>The quick brown fox jumps over the lazy
dog</q></p>
```

HTML Links

It is a connection from one web resource to another. A link has two ends, An anchor and direction. The link starts at the “source” anchor and points to the “destination” anchor, which may be any Web resource such as an image, a video clip, a sound bite, a program, an HTML document.

```
<!DOCTYPE html>
<html>
<h3>Example Of Adding a link</h3>
<body>
<p>Click on the following link</p>
<a href = "https://www.anylink.org">Welcome</a>
</body>
</html>
```

HTML
Images

Images can be uploaded by providing the file path relative to the location of the current web page file.

Adding images on a webpage: The <img> tag is used to add or embed the images to a webpage/website. The “img” tag is an empty tag, which means it can contain only a list of attributes and it has no closing tag.

```
<img src="url" alt="some_text" width="" height="“>

<img src="url" alt="some_text" width="" height="“
border=“”align=“right” >
```

HTML TABLES

HTML tables allow web developers to arrange data into rows and columns.

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial
Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico

```
<table border=“4” align=“center”>
```

HTML-FORMS

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

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>
```

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.

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
- Submit and Reset Button

Text Input Controls

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.

```
<body>
    <form >
      First name: <input type = "text" name = "first_name" />
      <br>
      Last name: <input type = "text" name = "last_name" />
    </form>
  </body>
```

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**.

```
<body>
    <form >
      User ID : <input type = "text" name = "user_id" />
      <br>
      Password: <input type = "password" name = "password" />
    </form>
  </body>
```

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.

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

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.**

```
<body>
    <form>
      <input type = "checkbox" name = "maths" value = "on"> Maths
      <input type = "checkbox" name = "physics" value = "on"> Physics
    </form>
  </body>
```

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**.

```
<body>
    <form>
      <input type = "radio" name = "subject" value = "maths"> Maths
      <input type = "radio" name = "subject" value = "physics"> Physics
    </form>
</body>
```

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.

```
<body>
    <form>
      <select name = "dropdown">
        <option value = "Maths" selected>Maths</option>
        <option value = "Physics">Physics</option>
      </select>
    </form>
  </body>
```

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**.

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

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**.

```
<body>
    <form>
      <input type = "submit" name = "submit" value = "Submit" />
      <input type = "reset" name = "reset"  value = "Reset" />
      <input type = "button" name = "ok" value = "OK" />
    </form>
  </body>
```

HTML-FRAMES

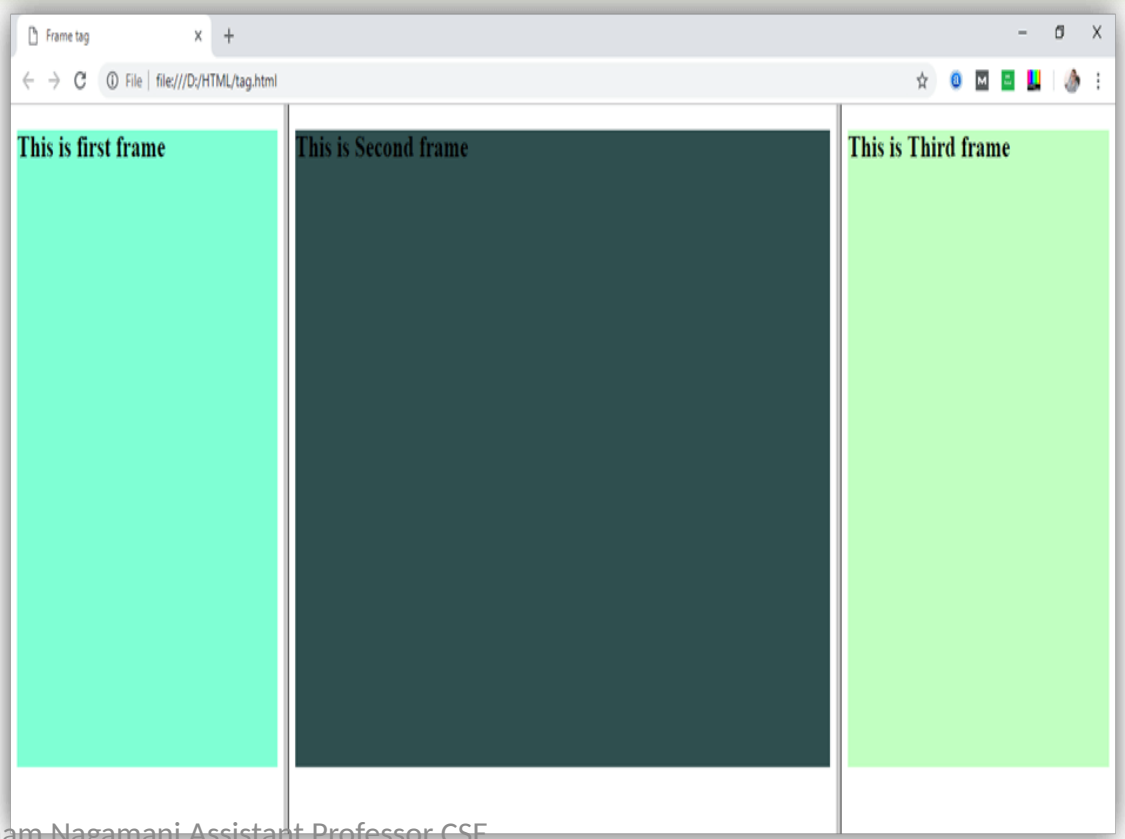
HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset.

DISWANTAGES

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back* button might not work as the user hopes.There are still few browsers that do not support frame technology.

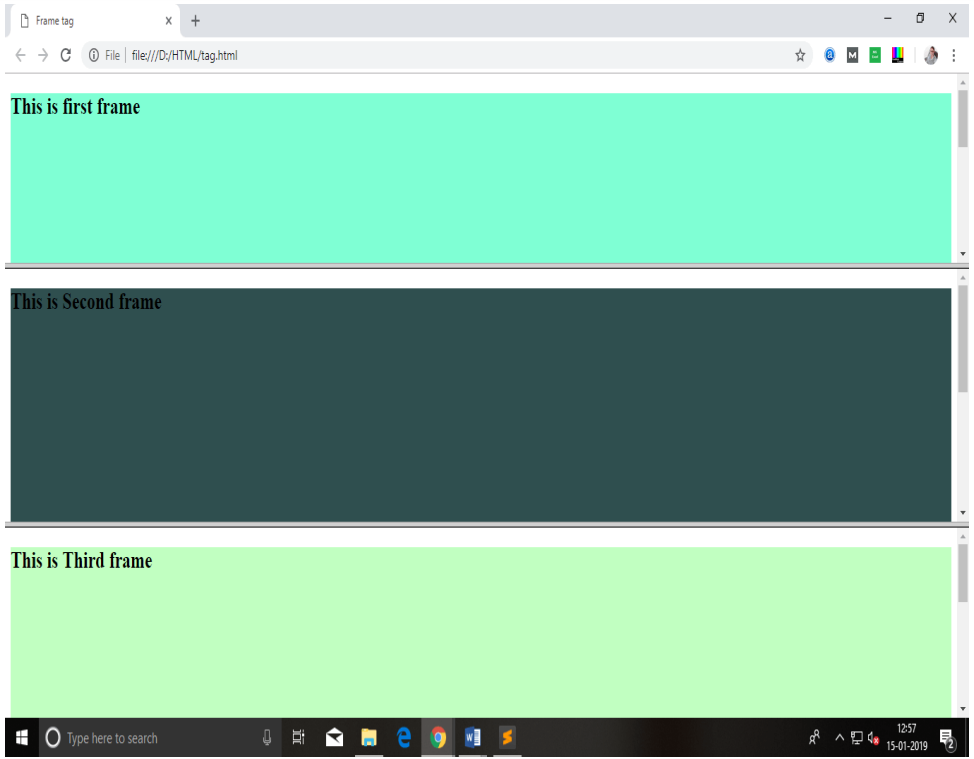
Create Vertical frames:

```
<html>
<head>
   <title>Frame tag</title>
</head>
  <frameset cols="25%,50%,25%">
    <frame src="frame1.html" >
    <frame src="frame2.html">
    <frame src="frame3.html">
  </frameset>
</html>
```



Create Horizontal frames:

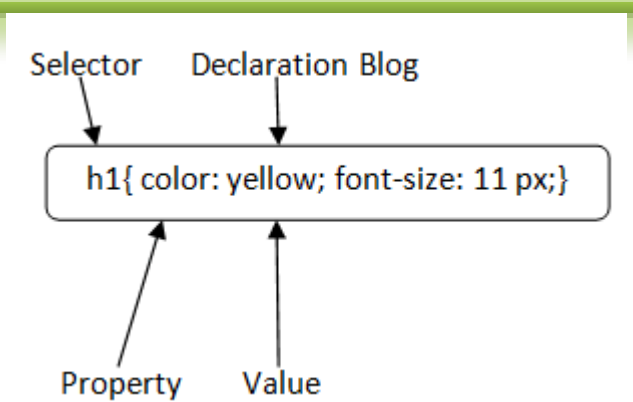
```
<html>
<head>
   <title>Frame tag</title>
</head>
  <frameset rows="30%, 40%, 30%">
    <frame name="top" src="frame1.html" >
    <frame name="main" src="frame2.html">
    <frame name="bottom" src="frame3.html">
  </frameset>
</html>
```



CSS

- CSS is the language we use to style an HTML document.
- CSS describes how HTML elements should be displayed.
- CSS stands for Cascading Style Sheets
- 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
- External stylesheets are stored in CSS files

CSS Syntax



Selector: Selector indicates the HTML element you want to style. It could be any tag like <h1>, <title> etc.

Declaration Block: The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

1.color: yellow;

2.font-size: 11 px;

Each declaration contains a property name and value, separated by a colon.

CSS Selector

CSS selectors are used *to select the content you want to style*. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.There are several different types of selectors in CSS.

1.CSS Element Selector
2.CSS Id Selector
3.CSS Class Selector
4.CSS Universal Selector
5.CSS Group Selector

1) CSS Element Selector

The element selector selects the HTML element by name.

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

2) CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

It is written with the hash character (#), followed by the id of the element.

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p id="para1">Hello Javatpoint.com</p>
<p>This paragraph will not be affected.</p>
</body>
</html>
```

3) CSS Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol) followed by the class name.

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is blue and center-aligned.</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</html>
```

CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is not affected</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</html>
```

4) CSS Universal Selector

The universal selector is used as a wildcard character. It selects all the elements on the pages.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  color: green;
  font-size: 20px;
}
</style>
</head>
<body>
<h2>This is heading</h2>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

5) CSS Group Selector

The grouping selector is used to select all the elements with the same style definitions.

Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

Let's see the CSS code without group selector.

```
h1,h2,p {
    text-align: center;
    color: blue;
}
```

```
h1 {
    text-align: center;
    color: blue;
}
h2 {
    text-align: center;
    color: blue;
}
p {
    text-align: center;
    color: blue;
}
```

How to add CSS

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

1.Inline CSS
2.Internal CSS
3.External CSS

1.Inline CSS

We can apply CSS in a single element by inline CSS technique.

If you want to use inline CSS, you should use the style attribute to the relevant tag.

Syntax:

<htmltag style="cssproperty1:value; cssproperty2:value;"**> </htmltag>**

Example:<**h2**style="color:red;margin- left:40px;">Inline CSS is applied on this heading.</**h2**>

2. Internal CSS

The internal style sheet is used to add a unique style for a single document. It is defined in <head> section of the HTML page inside the <style> tag.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: linen;
}
h1 {
    color: red;
    margin-left: 80px;
}
</style>
</head>
<body>
<h1>The internal style sheet is applied on this heading.</h1>
<p>This paragraph will not be affected.</p>
</body>
</html>
```

3. External CSS

The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.

It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

```
<head>
<link rel="stylesheet" type="text/
css" href="mystyle.css">
</head>
```

The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML elements.

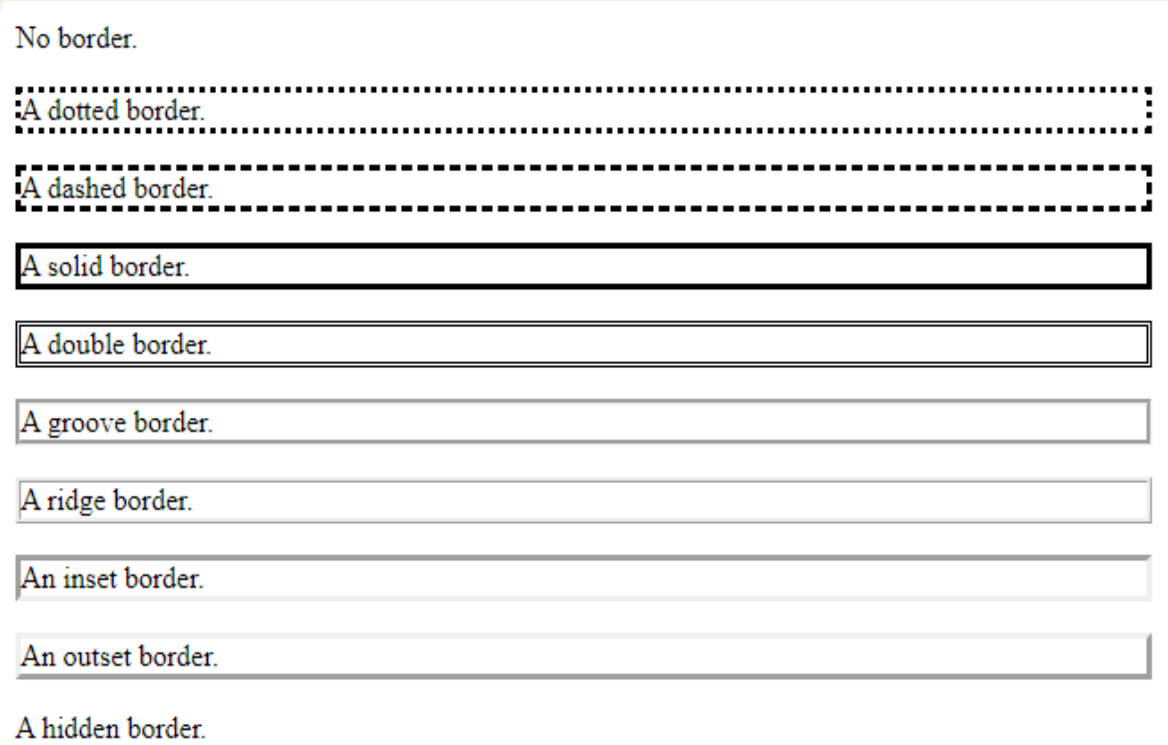
CSS Comments

CSS comments are generally written to explain your code. It is very helpful for the users who reads your code so that they can easily understand the code. Comments are ignored by browsers.

Comments are single or multiple lines statement and written within /*.....*/ .

CSS PROPERTIES

```
<head>
<style>
p.none {border-style: none;}
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.hidden {border-style: hidden;}
</style>
</head>
<body>
<p class="none">No border.</p>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="hidden">A hidden border.</p>
</body>
```



JavaScript

- JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages.
- It is an uninterpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document.
- With JavaScript, users can build modern web applications to interact directly without reloading the page every time.

Features of JavaScript

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. It is a case-sensitive language.
4. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
5. It provides good control to the users over the web browsers.

JavaScript provides 3 places to put the JavaScript code:

- Between the body tag of html
- Between the head tag of html
- In a javaScript)

```
<script type="text/javascript">
document.write("JavaScript is a simple language for java learners");
</script>
```

- The **script** tag specifies that we are using JavaScript.
- The **text/javascript** is the content type that provides information to the browser about the data.

1) JavaScript Example : code between the body tag

Below code displays the simple example of JavaScript that displays alert dialog box.

```
<script type="text/javascript">
 alert("Hello Javatpoint");
</script>
```

2) JavaScript Example : code between the head tag

In this example, we are creating a function msg(). To create function in JavaScript, you need to write function with function_name as given below.
To call function, you need to work on event. Here we are using onclick event to call msg() function.

```
<html>
<head>
<script type="text/javascript">
function msg(){
 alert("Hello Java");
}
</script>
</head>
<body>
<p>Welcome to Javascript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
</html>
```

External JavaScript file

We can create external JavaScript file and embed it in many html page.
It provides **code re usability** because single JavaScript file can be used in several html pages.
An external JavaScript file must be saved by .js extension. It is recommended to embed all JavaScript files into a single file. It increases the speed of the webpage.

Example:

message.js

```
function msg(){
 alert("Hello Javatpoi
nt");
}
```

index.html

```
<html>
<head>
<script type="text/javascript" src="message.js"></script>
</head>
<body>
<p>Welcome to JavaScript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
```

Advantages of External JavaScript

There will be following benefits if a user creates an external javascript:

1. It helps in the reusability of code in more than one HTML file.
2. It allows easy code readability.
3. It is time-efficient as web browsers cache the external js files, which further reduces the page loading time.
4. It enables both web designers and coders to work with html and js files parallelly and separately, i.e., without facing any code conflictions.
5. The length of the code reduces as only we need to specify the location of the js file.

Comment

The JavaScript comment is ignored by the JavaScript engine i.e. embedded in the browser.

1. **To make code easy to understand** It can be used to elaborate the code so that end user can easily understand the code.
2. **To avoid the unnecessary code** It can also be used to avoid the code being executed. Sometimes, we add the code to perform some action. But after sometime, there may be need to disable the code. In such case, it is better to use comments.

There are two types of comments in JavaScript.

1. Single-line Comment
2. Multi-line Comment

Single Line Comments

```
<script>
// It is single line comment
document.write("hello javascript");
</script>
```

Multi Line Comments

```
<script>
/* It is multi line comment.
It will not be displayed */
document.write("example of javascript
multiline comment");
</script>
```

JavaScript Variables

In a programming language, variables are used to store data values.

JavaScript uses the keywords var, let and const to declare variables.

An equal sign is used to assign values to variables.

In this example, x is defined as a variable. Then, x is assigned (given) the value 6:

```
let x;
x = 6;
```

Data Types	Description	Example
String	represents textual data	'hello', "hello world!" etc
Number	an integer or a floating-point number	3, 3.234, 3e-2 etc.
BigInt	an integer with arbitrary precision	900719925124740999n , 1n etc.
Boolean	Any of two values: true or false	true and false
undefined	a data type whose variable is not initialized	let a;
null	denotes a null value	let a = null;
Symbol	data type whose instances are unique and immutable	let value = Symbol('hello');
Object	key-value pairs of collection of data	let student = { };

JavaScript Variable

A **JavaScript variable** is simply a name of storage location. There are two types of variables in JavaScript :

1. local variable
2. global variable.

There are some rules while declaring a JavaScript variable (also known as identifiers).

1. Name must start with a letter (a to z or A to Z), underscore(_), or dollar($) sign.
2. After first letter we can use digits (0 to 9), for example value1.
3. JavaScript variables are case sensitive, for example x and X are different variables.

JavaScript local variable

A JavaScript local variable is declared inside block or function. It is accessible within the function or block only. For example:

```
<script>
function abc(){
var x=10;//local variable
}
</script>
```

JavaScript global variable

A **JavaScript global variable** is accessible from any function. A variable i.e. declared outside the function or declared with window object is known as global variable. For example:

```
<script>
var data=200;//gloabal variable
function a(){
document.writeln(data);
}
function b(){
document.writeln(data);
}
a();//calling JavaScript function
b();
</script>
```

JavaScript Functions

JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.

Advantage of JavaScript function

There are mainly two advantages of JavaScript functions.

1. **Code reusability**: We can call a function several times so it save coding.
2. **Less coding**: It makes our program compact. We don’t need to write many lines of code each time to perform a common task.

JavaScript Function Syntax

```
function functionName([arg1, arg2, ...argN]){
 //code to be executed
}
```

Simple example of function in JavaScript that does not has arguments.

```
<script>
function msg(){
alert("hello! this is message");
}
</script>
<input type="button" onclick="msg()"
value="call function"/>
```

JavaScript Function Arguments

call function by passing arguments. Let's see the example of function that has one argument.

```
<script>
function getcube(number){
alert(number*number*number);
}
</script>
<form>
<input type="button" value="click" onclick="g
etcube(4)"/>
</form>
```

64

Function with Return Value

We can call function that returns a value and use it in our program. Let's see the example of function that returns value.

```
<script>
function getInfo(){
return "hello java! How r u?";
}
</script>
<script>
document.write(getInfo());
</script>
```

JavaScript Loops

The **JavaScript loops** are used *to iterate the piece of code* using for, while, do while. It makes the code compact. It is mostly used in array.

There are three types of loops in JavaScript.

1.for loop
2.while loop
3.do-while loop

1) JavaScript For loop

The **JavaScript for loop** *iterates the elements for the fixed number of times*. It should be used if number of iteration is known. The syntax of for loop is given below.

```
for (initialization; condition; increment)
{
   code to be executed
}
```

2) JavaScript while loop

The **JavaScript while loop** *iterates the elements for the infinite number of times*. It should be used if number of iteration is not known. The syntax of while loop is given below.

```
while (condition)
{
   code to be executed
}
```

3) JavaScript do while loop

The **JavaScript do while loop** *iterates the elements for the infinite number of times* like while loop. But, code is *executed at least* once whether condition is true or false. The syntax of do while loop is given below.

```
do{
   code to be executed
}while (condition);
```

JavaScript Conditions

The **JavaScript if-else statement** is used *to execute the code whether condition is true or false*. There are three forms of if statement in JavaScript.

1.If Statement

2.If else statement

JavaScript If statement

It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

```
if(expression){
//content to be evaluated
}
```

JavaScript If...else Statement

It evaluates the content whether condition is true of false. The syntax of JavaScript if-else statement is given below.

```
if(expression){
//
content to be evaluated if condition is tr
ue
}
else{
```

The **JavaScript switch statement** is used *to execute one code from multiple expressions*. It is just like else if statement. But it is convenient than *if..else..if* because it can be used with numbers, characters etc.
The signature of JavaScript switch statement is given below.

```
switch(expression){
case value1:
 code to be executed;
 break;
case value2:
 code to be executed;
 break;
......

default:
 code to be executed if above values are not matched;
}
```

Application of JavaScript

JavaScript is used to create interactive websites. It is mainly used for:

- Client-side validation,
- Dynamic drop-down menus,
- Displaying date and time,
- Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
- Displaying clocks etc.
- It can change HTML Content
- It can change HTML attributes
- It can change style of HTML element
- Java script can validate HTML Content

JavaScript Can Change HTML Content

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

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!"'>Click Me!</button>

</body>
</html>
```

JavaScript Can Change HTML Styles (CSS)

Changing the style of an HTML element, is a variant of changing an HTML attribute:

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

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<button type="button"
onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!</button>


</body>
</html>
```

Java Script OUT puts

JavaScript Display Possibilities

JavaScript can "display" data in different ways:

Writing into an HTML element, using innerHTML.
Writing into the HTML output using document.write().
Writing into an alert box, using window.alert().
Writing into the browser console, using console.log().

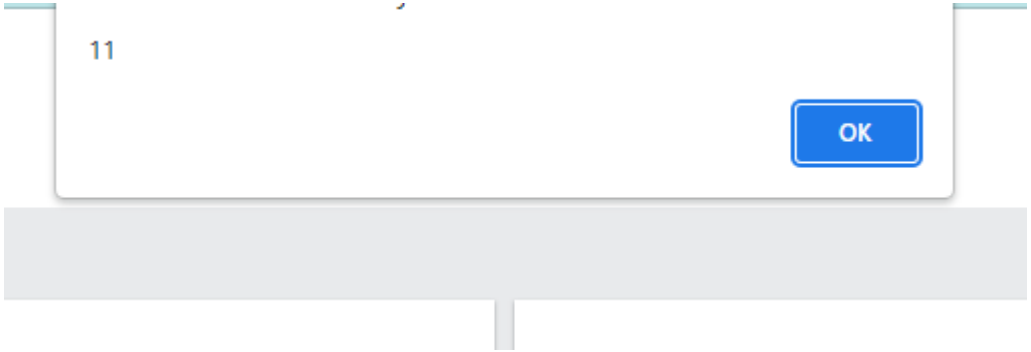
Using innerHTML

- To access an HTML element, JavaScript can use the document.getElementById(id) method.
- The id attribute defines the HTML element. The innerHTML property defines the HTML content:

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

<h1>My First Web Page</h1>
<p>My First Paragraph</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```

window.alert().

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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```



Using document.write()

For testing purposes, it is convenient to use document.write():

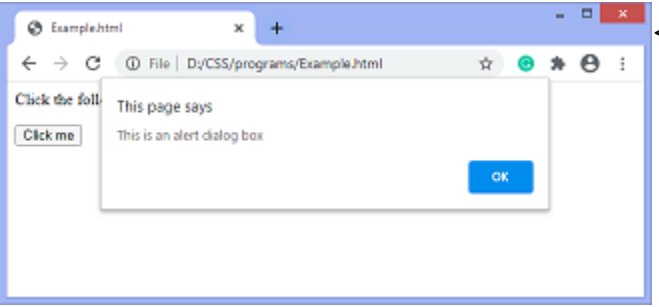
```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

Functions
Variables
Operators
Statements
Arrays
Objects
Strings

Browser object model in javaScript(BOM)

Method	Description
alert()	displays the alert box containing message with ok button.
confirm()	displays the confirm dialog box containing message with ok and cancel button.
prompt()	displays a dialog box to get input from the user.
open()	opens the new window.
close()	closes the current window.

- Window.alert(“Hello”)
- Window.confirm(“Hello”)
- Window.prompt(“Hello”)
- Open(“www.google.com”)
- Close() = toclose the window




```
<html>
  <head>
    <script type = "text/javascript">
        function fun() {
            alert ("This is an alert dialog box");
        }
    </script>
  </head>
<body>
    <p> Click the following button to see the effect </p>
    <form>
      <input type = "button" value = "Click me" onclick = "fun();" /
    </form>
  </body>
</html>
```

prompt() in javascript

```
<script type="text/javascript">
function msg(){
var v= prompt("Who are you?");
alert("I am "+v);

}
</script>

<input type="button" value="click" onclick="msg()"/
>
```

open() in javascript

```
<script type="text/javascript">
function msg(){
open("http://www.javascript.com");
}
</script>
<input type="button" value="javatpoint" onclic
"msg()"/>
```

The HTML DOM (Document Object Model)

Write()
GetelementByID
Getelementby name
Changing content
Changing Value
Changing Style

Form validation in javascript

- If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted:

Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- has the user filled in all required fields?
- has the user entered a valid date?
- has the user entered text in a numeric field?

Most often, the purpose of data validation is to ensure correct user input.

Validation can be defined by many different methods, and deployed in many different ways.

Server side validation is performed by a web server, after input has been sent to the server.
Client side validation is performed by a web browser, before input is sent to a web server.

```
<!DOCTYPE html>
<html>
<head>
<script>
function validateForm() {
  let x = document.forms["myForm"]["fname"].value;
  if (x == "") {
    alert("Name must be filled out");
    return false;
  }
}
</script>
</head>
<body>

<h2>JavaScript Validation</h2>

<form name="myForm" action="/action_page.php"
onsubmit="return validateForm()" method="post">
  Name: <input type="text" name="fname">
  <input type="submit" value="Submit">
</form>

</body>

</html>
```