Origins and Evolution of HTML

HTML stands for Hyper Text Markup Language. Hyper Text is text which contains links to other texts. It is a markup language consists of a set of markup tags. HTML uses markup tags to describe web pages. HTML markup tags are usually called HTML tags. HTML tags are keywords surrounded by angle brackets like. The first tag in a pair is the start tag, the second tag is the end tag. Start and end tags are also called opening tags and closing tags XHTML stands for EXtensible Hyper Text Markup Language. XHTML is almost identical to HTML 4.01.XHTML is a stricter and cleaner version of HTML 4.01.

Versions of HTML

Original intent of HTML: General layout of documents that could be displayed by a wide variety of computers.

HTML 1.0

• HTML 1.0 was the first release of HTML to the world
• The language was very limiting.

HTML 2.0

• HTML 2.0 included everything from the original 1.0 specifications
• HTML 2.0 was the standard for website design until January 1997
• Defined many core HTML features for the first time.

HTML 3.2 - 1997

• The browser-specific tags kept coming.
• First version developed and standardized exclusively by the W3C.

HTML 4.0 – 1997

• Introduced many new features and deprecated many older features.
• Support for HTML's new supporting presentational language, CSS.

HTML 4.01 - 1999

• A clean up of 4.0,
• it faces 2 problems
- it specifies loose syntax rules
- its specification does not define how a user agent (browser) is to recover when erraneous code is encountered.

Basic Syntax of HTML

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>** : Every HTML code must be enclosed between basic HTML tags. It begins with **<html>** and ends with **</html>** tag.

- **<head>**: The head tag comes next which contains all the header information of the web page or documents like the title of the page and other miscellaneous information. This information is enclosed within the head tag which opens with **<head>** and ends with **</head>**. The contents will of this tag will be explained in the later sections of the course.
- **<title>:** We can mention the title of a web page using the **<title>** tag. This is header information and hence is mentioned within the header tags. The tag begins with **<title>** and ends with **</title>.**
- **<body>:** Next step is the most important of all the tags. The body tag contains the actual body of the page which will be visible to all the users. This opens with **<body>** and ends with **</body>**. All content enclosed within this tag will be shown on the web page be it writings or images or audio or videos or even links. We will see later in the section how using various tags we may insert mentioned contents into our web pages.

```
<html>
<head>
   <!-- Information about the page -->
   <!--This is the comment tag-->
    <title>title of the page</title>
</head>
<body>
   <!--Contents of the webpage-->
</body>
</html>
```

Basic text markup

1. HTML Images Syntax

- The HTML <img> tag is used to embed an image in a web page.
- Images are not technically inserted into a web page; images are linked to web pages.
- The <img> tag creates a holding space for the referenced image.
- The <img> tag is empty, it contains attributes only, and does not have a closing tag.
- The <img> tag has two required attributes:
  - src - Specifies the path to the image
  - alt - Specifies an alternate text for the image

Syntax

<img src="*url*" alt="*alternatetext*">

The src Attribute
The required src attribute specifies the path (URL) to the image.

Example

<img src="img_chania.jpg" alt="Flowers in Chania">

2. HTML List

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:

Example

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

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:

Example

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

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:

Example

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

3. Define an HTML Table

A table in HTML consists of table cells inside rows and columns.

Example

A simple HTML table:

```
<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>
```

Table Cells

Each table cell is defined by a <td> and a </td> tag.

td stands for table data.

Everything between <td> and </td> are the content of the table cell.

Example

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
</table>
```

Table Rows

Each table row starts with a <tr> and ends with a </tr> tag.

tr stands for table row.

Example

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
```

```
  </tr>
</table>
```

Table Headers

Sometimes you want your cells to be table header cells. In those cases use the <th> tag instead of the <td> tag:

th stands for table header.

Example

Let the first row be table header cells:

```
<table>
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

Example

First name:
John
Last name:
Doe

Submit

The <form> Element

The HTML <form> element is used to create an HTML form for user input:

<form>
.
form elements
.
</form>

Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

- The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.
- The <input> Element
- The HTML <input> element is the most used form element.
- An <input> element can be displayed in many ways, depending on the type attribute.
- Here are some examples:

Text Fields

The <input type="text"> defines a single-line input field for text input.

Example

A form with input fields for text:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

- The <label> Element
- Notice the use of the <label> element in the example above.
- The <label> tag defines a label for many form elements.
- The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focuses on the input element.
- The <label> element also helps users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

- The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

Radio Buttons

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

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

Example

A form with radio buttons:

```
<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
```

```
  <label for="javascript">JavaScript</label>
</form>
```

This is how the HTML code above will be displayed in a browser:
Choose your favourite Web language:

- HTML
- CSS
- JavaScript

Checkboxes

The <input type="checkbox"> defines a **checkbox**.
Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

A form with checkboxes:

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

This is how the HTML code above will be displayed in a browser:

- I have a bike
- I have a car
- I have a boat

The Submit Button

- The <input type="submit"> defines a button for submitting the form data to a form-handler.
- The form-handler is typically a file on the server with a script for processing input data.
- The form-handler is specified in the form's action attribute.

Example

A form with a submit button:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
```

```
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:
John
Last name:
Doe

Submit

- The Name Attribute for <input>
- Notice that each input field must have a name attribute to be submitted.
- If the name attribute is omitted, the value of the input field will not be sent at all.

Example

This example will not submit the value of the "First name" input field:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" value="John"><br><br>
  <input type="submit" value="Submit">
</form>
```

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

Creating Frames

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

Note − The <frame> tag deprecated in HTML5. Do not use this element.

Example

Following is the example to create three horizontal frames −

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

```
  <head>
    <title>HTML Frames</title>
  </head>

  <frameset rows = "10%,80%,10%">
    <frame name = "top" src = "/html/top_frame.htm" />
    <frame name = "main" src = "/html/main_frame.htm" />
    <frame name = "bottom" src = "/html/bottom_frame.htm" />

    <noframes>
      <body>Your browser does not support frames.</body>
    </noframes>

  </frameset>

</html>
```

HTML5 - Overview

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

Browser Support

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

New Features

HTML5 introduces a number of new elements and attributes that can help you in building modern websites. Here is a set of some of the most prominent features introduced in HTML5.

- **New Semantic Elements** − These are like <header>, <footer>, and <section>.
- **Forms 2.0** − Improvements to HTML web forms where new attributes have been introduced for <input> tag.
- **Persistent Local Storage** − To achieve without resorting to third-party plugins.
- **WebSocket** − A next-generation bidirectional communication technology for web applications.
- **Server-Sent Events** − HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
- **Canvas** − This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video** − You can embed audio or video on your webpages without resorting to third-party plugins.
- **Geolocation** − Now visitors can choose to share their physical location with your web application.
- **Microdata** − This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.

- **Drag and drop** − Drag and drop the items from one location to another location on the same webpage.

CSS Introduction

What is CSS?

- 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

Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

Level soft style sheet

Using CSS

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the style attribute inside HTML elements
- **Internal** - by using a <style> element in the <head> section
- **External** - by using a <link> element to link to an external CSS file

The most common way to add CSS, is to keep the styles in external CSS files.

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.

The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

Example

```
<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>
```

Internal 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.

The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to red. In addition, the page will be displayed with a "powderblue" background color:

Example

```
<!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>
```

External CSS

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

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

Example

```
<!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>
```

The 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 what the "styles.css" file looks like:

"styles.css":

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

Style Specification Formats

Format depends on the level of the style sheet

• Format for Inline Style sheets
• Inline: Style sheet appears as the values of the style attribute of a tag

• **General form:**

style = "property_1: value_1;

property_2: value_2; …

property_n: value_n;"

Example:

<h1 style"=color:red; font-family: ">India is my country>/h1>

Format for Document-level / Embedded / Internal style sheet

• A style is applied to the entire HTML file.

• Use it when you need to modify all instances of particular element (e.g., h1) in a web page.

- Style sheet appears as a list of rules that are the content of a <style> tag within the header of adocument

General form:

<style type = "text/css">

rule list

</style>

- The <style> tag must include the type attribute, set to "text/css"
- Each style rule in a rule list has two parts:

- Selector: indicates the tag or tags affected by the rule

- List of property value pairs

  - Each property/value pair has the form: property: value
  - Pairs are separated by semicolons

- General Form of the rules in rule list:

selector {property_1:value_1; property_2:value_2;……
property_n:value_n;}

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

The CSS element Selector

The element selector selects HTML elements based on the element name.

Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

```
p {
  text-align: center;
  color: red;
}
```

The CSS id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example

The CSS rule below will be applied to the HTML element with id="para1":

```
#para1 {
  text-align: center;
  color: red;
}
```

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.
To select elements with a specific class, write a period (.) character, followed by the class name.

Example

In this example all HTML elements with class="center" will be red and center-aligned:

```
.center {
  text-align: center;
  color: red;
}
```

You can also specify that only specific HTML elements should be affected by a class.

Example

In this example only <p> elements with class="center" will be red and center-aligned:

```
p.center {
  text-align: center;
  color: red;
}
```

The CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

Example

The CSS rule below will affect every HTML element on the page:

```
* {
  text-align: center;
  color: blue;
}
```

The CSS Grouping Selector

To group selectors, separate each selector with a comma.

Example

In this example we have grouped the selectors from the code above:

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

Property Value Forms

- CSS1 include 60 different properties in 7 categories:
    - Fonts
    - Lists
    - Alignment of text
    - Margins
    - Colors
    - Backgrounds
    - Borders

Property Values (values of properties)

The property value can appear in many forms.

- Keywords – large,medium, small, …
- Number values – integers,decimal numbers etc.
- Length - numbers, maybe with decimal points followed by two character abbreviation of a unit name
- Units:
    - px - pixels
    - in - inches
    - cm - centimeters
    - mm - millimeters
    - pt - points
    - pc - picas (12 points)
    - em – value of the current font size in pixels

- ex - height of the letter ‘x’
- No space is allowed between the number and the unit specification e.g., 1.5 in is illegal!Eg: 10px, 24pt etc.

- Percentage - just a number followed immediately by a percent sign. Eg: 70%
- URL values
  - Colors
  - **Color name**: eg: fuchsia
  - **rgb(n1, n2, n3)** :Eg: rgb(255,0,255)
  - Numbers can be decimal or percentages
  - Hexadecimal form: hexadecimal numbers must be preceded with pound(#) sign.Eg : #B0E0E6 stands for powder blue color.

Font Properties

Font-Families

- The font-family property is used to specify a list of font name.
- The browser will use the first font in the list that it supports. For example, the following could bespecified.

font- family: Arial, Helvetica, Courier

If a font name has more than one word, it should be single-quoted. ‘ Times New Roman‘

Font-size

Sets the size of fonts.There are two categories of font-size values, absolute and relative.

In the case of absolute category the size value could be given as length value in points, picas or pixels orkeywords from the list xx-small, x-small, small, medium, large and x-large.

Eg: **font-size: 10pt**

The relative size values are smaller and larger, which adjust the font size relative to the font size of theparent element.

Eg: **font-size:1.2em**

This sets the font size to 1.2 times the font size of the parent element.1.2em and 120% are same.

Font Variant

The default value of the font-variantproperty is normal,which specifies the usual character font. This property can be set to small-capsto specify small capital letters.

Font-style

- Most commonly used to

specify italic.Eg: font-style: italic

Font-weight

- used to specify degrees of boldness.

Eg: font-weight: bold

Possible values are bolder, lighter, bold, normal(default)

- Could specify as a multiple of 100 (100 – 900) where 400 is same as normal.

```
<!DOCTYPE html>
  <!—fonts.html
          An example to illustrate font properties -->
  <html lang=”en”>
          <head><title>Font Properties</title>
  <meta charset=”utf-8” />
  <style type”text/css”>
  p.major {font-size: 14pt;
                  font-style: italic;
                  font-family: ‘Times New Roman’ ;}
  p.minor {font: 10pt bold ‘courier New’;}
  h2 {font-family: ‘Times New Roman’;font-
  size: 24pt;font-weight: bold;}
    h3 {font-family: ‘Courier New’;}
  </style>
  </head>
    <body>
```

If a job is worth doing,it’s worth doing right

< p class = “minor”>

Two wrong don’t make a right,but they certainly can get you in a lot oftrouble.

<h2> Chapter 1 Introduction</h2>

<h3> 1.1 The basics of computer networks</h3>

</body>

</html>

List properties

- Property Name: list-style-typecan applied to both ordered and unordered list.

Unordered lists

- Bullet can be a disc (default), a square, or a circle. Set it on either the <ul> or <li> tag. On
  , it applies to list items.

<style type = “text/CSS”> ul

{list-style-type:square}

</style>

……

<h3> Some Common Single-Engine Aircraft </h3>

 Cessna Skyhawk

 Beechcraft Bonanza

 Piper Cherokee

On <li>, list-style-typeapplies to just that item.

<h3> Some Common Single-Engine Aircraft </h3>


```
<li style = "list-style-type: disc">Cessna
        Skyhawk </li>
<li style = "list-style-type: square">Beechcraft
        Bonanza </li>
<li style = "list-style-type: circle">Piper
        Cherokee </li>
</ul>
```

ORDERED LIST

- *On ordered lists* - list-style-type property can be used to change the sequence values

Property value	***Sequence type***	***First four***
Decimal	Arabic numerals	1, 2, 3, 4
upper-alpha	Uc letters	A, B, C, D
lower-alpha	Lc letters	a, b, c, d
upper-roman	Uc Roman	I, II, III, IV
lower-roman	Lc Roman	i, ii, iii, iv

Alignment of Text

- The text-indentproperty allows indentation (first line of a paragraph can be intended)
- This property takes either a length or a % value

```
<style type = “text/css”>
p.indent        {text-indent: 0.5in}
</style>
……
<p class =“indent”>
Now is the time for all good Web programmers to begin using cascading style sheets for all presentationdetails in their documents. No more deprecated tags and attributes, just nice, precise style sheets.
</p>
```

- The text-alignproperty has the possible values, left (the default), center, right,or justify

- The float property is used to specify that text should flow around another element often an imageor table.The floatproperty has the possible values, left, right, and none(the default).If we have an element we want on the right, with text flowing on its left, we use the default text-alignvalue (left) for the text and the right value for floaton the element we want on the right.

Colors

• *Color is a problem for the Web for two reasons*:

1. Monitors vary widely
2. Browsers vary widely

- There are three color collections

1. There is a set of 17 colors that are guaranteed to be displayable by all graphical browsers on all color monitors

Black	000000	green	008000
Silver	C0C0C0	lime	00FF00
Gray	808080	olive	808000
White	FFFFFF	yellow	FFFF00
maroon	800000	navy	000080
Red	FF0000	blue	0000FF
Purple	800080	teal	008080
Fuchia	FF00FF	aqua	00FFFF

2.Color Properties

• The colorproperty specifies the foreground color of elements

```
<style type = “text/css”> th.red
    {color: red} th.orange
    {color: orange}
</style>
<table>
    <tr>
```

```
<th class = "red"> Apple </th>
<th class = "orange"> Orange </th>
<th class = "orange"> Screwdriver </th>
</tr>
</table>
```

- The background-color property specifies the background color of elements.

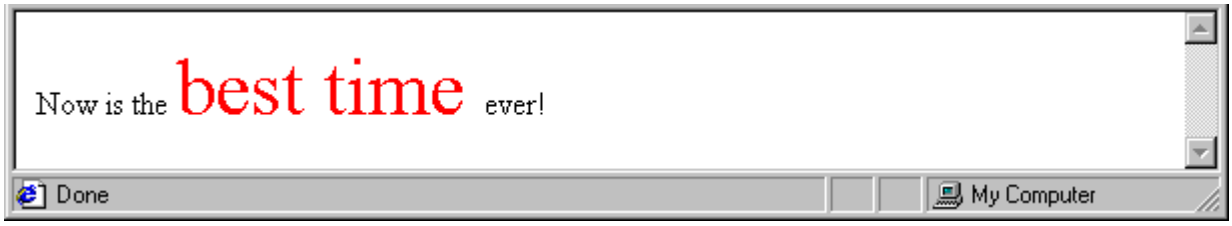
The and tags

One problem with the font properties is that they apply to whole elements, which are often too large .In many situations we want to apply special font properties to less than a whole paragraph of text. For eg: it isoften useful to have a word or phrase in a line appear in a different font size or color . The tag is designed for just this purpose. Unlike most other tags there is no default layout for the content.The defaultmeaning of is to leave the content as it is.

```
<p>
Now is the <span> best time </span> ever!
</p>
```

Use to apply a document style sheet to its content

```
<style type = "text/css" > bigred {font-
    size: 24pt;
        font-family: Ariel; color: red}
</style>
<p>
    Now is the <span class = "bigred"> best time </span> ever!
</p>
```



- The tag is similar to other HTML tags, they can be nested and they have id

and classattributes.

- Another tag that is useful for style specifications:

It is common for documents to have sections, each consisting of some number of paragraphs that have theirown presentation styles. Used to create document sections (or divisions) for which style can be specified.

- e.g., A section of five paragraphs for which you want some particular style.

```
<div class = "primary">
  <p>
  ...
  </p>
  <p>
  ...
  </p>
  <p>
  ...
  </p>
</div>
```

Javascript

Overview of JavaScript

Originally developed at Netscape by Brendan Eich, as LiveScript

Initially named Mocha, later renamed as LiveScript

Became a joint venture of Netscape and Sun in 1995, renamed JavaScript

Now standardized by the European Computer Manufacturers Association as ECMA-262 (also ISO 16262)

Javascript is divided into three parts

1) Core Javascript - Heart of the language , including its operators, expressions, statements and subprogram.

2) Client-side JavaScript - Collections of JavaScript code scripts , not programs, support the control of browsers and interaction with the user

3) Server side Javascript - Collection of objects that make the language useful on a web server.

Web Browsers & HTML- JAVASCRIPT Doc

• When a JavaScript code is encountered in the HTML document, the browser uses its JavaScript interpreter to “execute” the script. Output from the script becomes the next markup to be rendered. When the end of the script is reached, the browser goes back to reading the HTML document and displaying its content.

• There are two different ways to embed JavaScript in an HTML document:

– implicitly and explicitly.

• In explicit embedding, the JavaScript code physically resides in the HTML document.

• In implicit embedding the JavaScript can be placed in its own file, separate from the HTML document.

• When JavaScript scripts are explicitly embedded, they can appear in either part of an HTML document—the head or the body

• The interpreter notes the existence of scripts that appear in the head of a document, but it does not interpret them while processing the head. This method is useful when using Javascript functions. The function can be defined in the head of the HTML document.Scripts that are found in the body of a document are interpreted as they are found.

Object Orientation and JavaScript

• JavaScript is NOT an object-oriented programming language, it is an object based language.

• JavaScript does not have classes

Without classes Javscript does not support class-based inheritance

• Without class based inheritance it cannot support polymorphism

• It uses a technique that can be used to simulate some aspects of inheritance

• This is done with prototype object. This form of inheritance is called prototype-based inheritance.

• JavaScript has primitives for simple types

• The root object in JavaScript is Object – all objects are derived from Object

• All JavaScript objects are accessed through references

General Syntactic Characteristics

• JavaScript are embedded in HTML documents

• Either directly, as in General Syntactic Characteristics

• JavaScript are embedded in HTML documents

• Either directly, as in

<script type = "text/javaScript">

<! -- JavaScript script //-- >

</script>

Or indirectly, as a file specified in the src attribute of <script>, as in

<script type = "text/javaScript" src = "myScript.js">

</script>

Identifier form: begin with a letter or underscore, or dollar sign ($) followed by any number of letters, underscores, and digits

• Case sensitive

• 25 reserved words, plus future reserved words Eg: Break, continue, switch, case, if ,else, while, for…

• Comments: both // and /* … */

<!--

-- JavaScript script –

//-->

• Scripts are usually hidden from browsers that do not support JavaScript interpreters by putting them in special comments.

• Semicolons can be a problem. They are “somewhat” optiona

Primitives

1.Primitive Types

Five primitive types: Number, String, Boolean, Undefined, or Null.

Javascript includes predefined objects that are closely related to primitive types named Number ,string and Boolean. These objects are called wrapper objects. Because each objects contains a property that stores a value of the corresponding primitive type. The purpose of Wrapper objects are used to provide properties and methods that are convenient to use with the values of primitive types. Javascript coerces values between the number type primitive values and number objects and between the string type primitive values and string

2. Numeric & String Literals Numeric literals – like Java All numeric values are stored in double-precision floating point. Numeric values in Javascript are often called numbers. Numeric literals can be either integer or floating point values. 72,7.2,.72,7E2,7e2 etc are valid numeric literals.

String literals are delimited by either ' or " .Can include escape sequences (e.g., \n,\t).

Var str1=”Hello”;

Var str2=’Hello’; To include a single quote character in a string delimited by single quotes, the embedded single quote must be preceded by a backslash.

‘You \’re the best person I \’ve ever seen’

A double quote can be embedded in a double quoted string literal by preceding it with a backslash.

• A null string (a string with no characters) can be denoted with ‘ ‘ or “ “.

• All string literals are primitive types.

3. Other Primitive Types

Boolean values are true and false.Used for evaluating Boolean Expressions.

Var boo1= true;

Var boo2=false;

The only value of type Null is reserved word null which means no value. A variable is Null if it has not been explicitly declared or assigned a value.

The only value of type Undefined is undefined. If a variable has been explicitly declared, but not assigned a value it has the value undefined.

A variable declared without a value will have the value undefined.

ex) var carName;

Document.write(carName)

JavaScript Variable

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

Example:

```
<script>
var x = 10;
var y = 20;
var z=x+y;
document.write(z);
</script>
```

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>
```

Numeric Operators

Arithmetic Operators

Operator	Description	Example	Result
+	Addition	x=2 y=2 x+y	4
-	Subtraction	x=5 y=2 x-y	3
*	Multiplication	x=5 y=4 x*y	20
/	Division	15/5 5/2	3 2,5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

• Eg:increment operator

If the variable a has the value 7,

(++a)*3 =24

(a++)*3 = 21

The *precedence rules* of a language specify which operator is evaluated first when two operators with different precedence are adjacent in an expression.

The *associativity rules* of a language specify which operator is evaluated first when two operators with the same precedence are adjacent in an expression.

- **Precedence and associativity**
- **Operator** **Associativity**

++,--,unary-,unary+	**Right(though it is irrelevant)**
***,/,%**	**Left**
Binary +,Binary -	**Left**
>,<,>=,<=	**Left**
==,!=	**Left**
===,!===	**Left**
&&	**Left**
||	**Left**
=,+=,-=,*=,/=,&&=,||=,%=	**Right**

- Example of operator precedence and associativity

```
var a=2,
b=4,c,d;
C=3+a*b;
//* is first, so c is now 11 d=b/a/2;
// associates left , so d is now 1 Paranthesis
```

can be used to force any desired precedence.

(a+b)*c

Addition will be done before multiplication

3.2 Assignment Operators

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y

Comparison Operators

Operator	Description	Example
==	is equal to	5==8 returns false
===	is equal to (checks for both value and type)	x=5 y="5" x==y returns true x===y returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

4 Logical Operators

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
||	or	x=6 y=3 (x==5 || y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

Screen Output and Keyboard Input

Every web page resides inside a browser window which can be considered as an object.An object of window is created automatically by the browser. The javascript model for the browser display window is
the **Window** object. A *document* object represents HTML document. The model for the browser display window is the **Window** object. It represents the browser window.

The Window object has two properties, document and window, which refer to the **Document** and **Window** objects , respectively .

The **Document** object has several methods and properties. One method is **write**, which is used to create output.

The parameter is a string, often catenated from parts, some of which are variables e.g., document.write("Answer: " + result + "
");

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

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

Using

document.write()

document.write()

for writing

Example

```
<!DOCTYPE html>
<html><body>
<h1>My First Web Page</h1>
<p>My first paragraph.</p>
<script type = “text/javascript” > document.write(5 + 6);
</script>
</body>
</html>
```

The **Window** object has three methods for creating dialog boxes for user interactions , **alert, confirm and prompt**

The **alert** method opens a dialog window and displays its parameter in that window. It also displays an **OK**
Button

```
alert(“The sum is :”  +sum + “\n”);
```

The confirm method opens a dialog window in which the method displays its string parameter along with two buttons **OK** and **Cancel** to offer the user the choice of continuing some process.

```
Var question = confirm(“Do you want to continue this download”);
```

The **prompt** method used to create dialog window that contains a textbox used to collect string of input from from the user .The window also includes 2 buttons: **OK** and **Cancel**.The prompt takes 2 parameters: ***the string*** that prompts user for input and a ***default string*** in case user does not type a string before pressing one of the 2 buttons.

```
name=prompt(“What is your name?” ,””);
```

Alert

Parameter in alert is plain text, not HTML.Opens a dialog box which displays the parameter string and an OK button.

```
<!DOCTYPE html>

<html lang =“en”>
<head><title>hi.html</title>

<meta charset = “utf-8” />
<body>
<h1>My First Web Page</h1>
<p>My first paragraph.</p>
<script>
    window.alert(5 + 6);
</script>
</body>
</html>
```

Confirm

Opens a dialog box and displays the parameter and two buttons, OK and Cancel

```
if (confirm("Press a button!"))

  {
        txt = "You pressed OK!";
} else {
        txt = "You pressed Cancel!";

} window.alert(txt);
```

Prompt

Opens a dialog box and displays its string parameter, along with a text box and two buttons, OK and Cancel.

The second parameter is for a default response if the user presses OK without typing a response in the text box (waits for OK) .

```
var person = prompt("Please enter your name", ""); if (person ==
null || person == "") {
        txt = "User cancelled the prompt.";
} else {
        txt = "Hello " + person + "! How are you?";
}

window.alert(txt);
```

Sum of two numbers

```
<script type= “text/javascript”>

var no1 = prompt("Please enter No1", ""); var no2 =

prompt("Please enter No2", ""); if (no1 != "" || no2

!= "")

  {
    var res = Number(no1)+Number(no2);

} else {

        res= "Failed ";
}

window.alert(res);
```

```
</script>
```