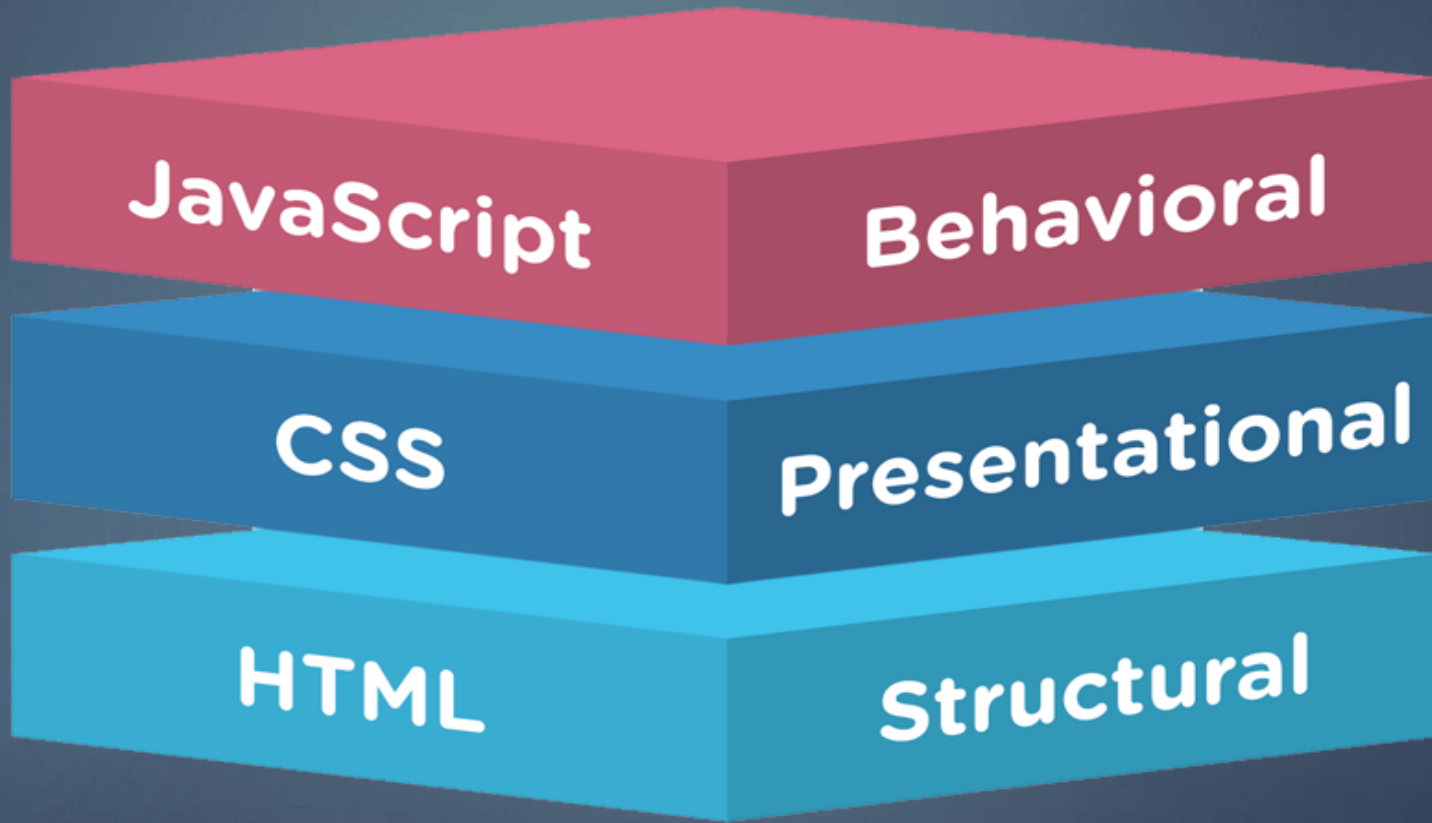


HTML , CSS

HTML vs CSS vs JAVASCRIPT

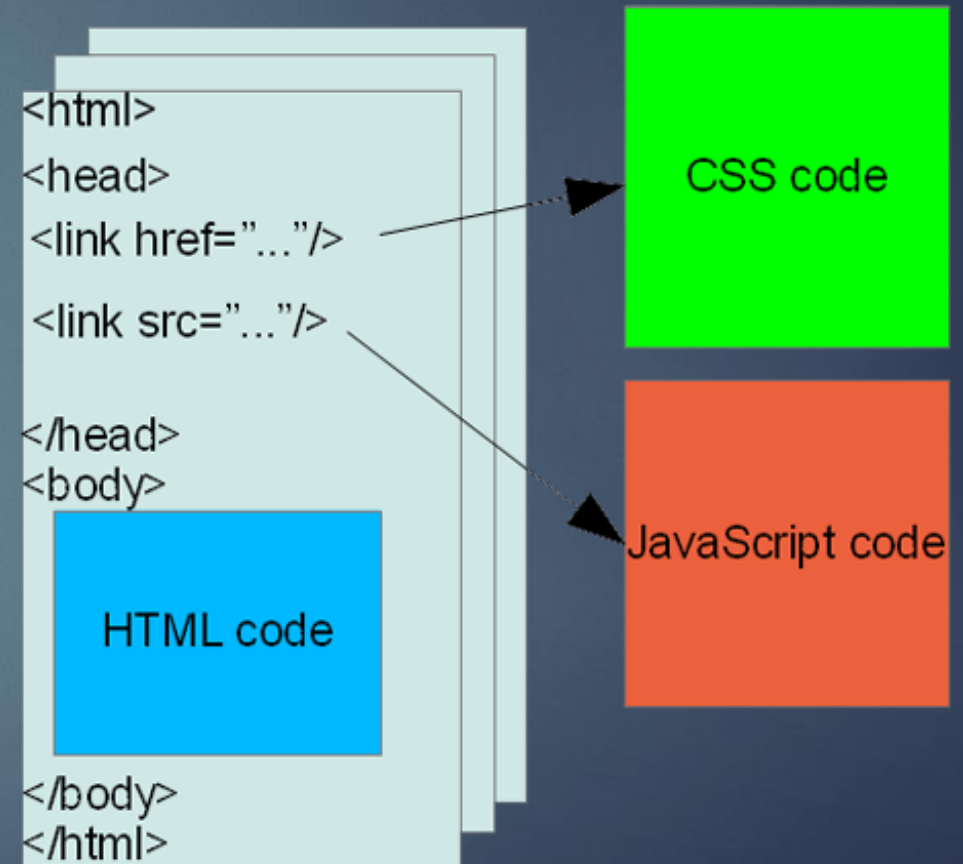
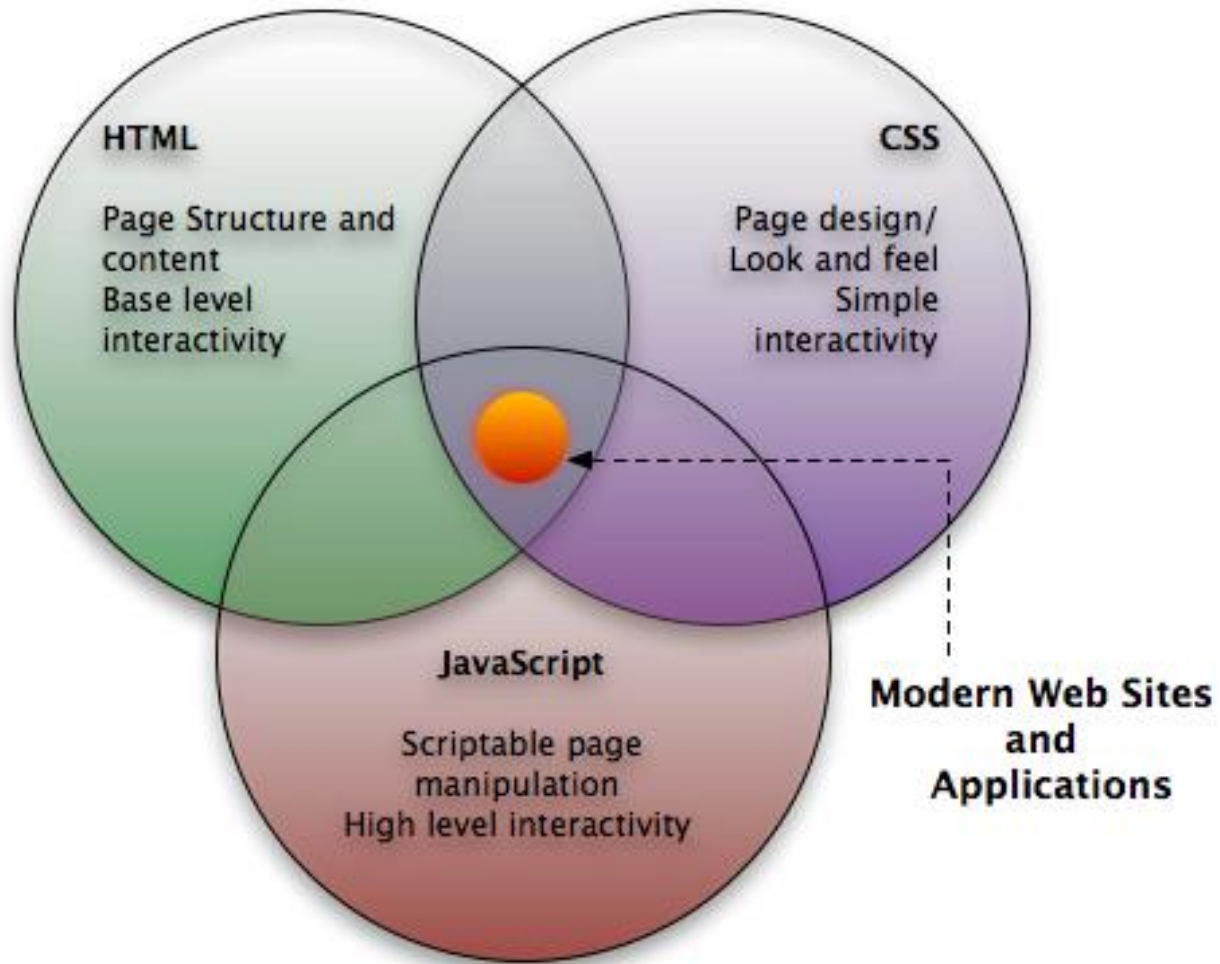
Overview



HTML vs CSS vs JAVASCRIPT

3

Overview



What is HTML?

- ▶ Tim Berners-Lee was the author of html, with his team at CERN.
- ▶ The HTML that Tim invented was strongly based on SGML (Standard Generalized Mark-up Language).
- ▶ Hypertext Markup Language (First Version of HTML) was formally published on June 1993.
- ▶ Platform independent.
- ▶ HTML is a format that tells a computer how to display a web page.
- ▶ The documents themselves are plain text files (ASCII) with special "tags" or codes that a web browser knows how to interpret and display on your screen.

HTML Versions

5

- ▶ HTML 2.0
 - ▶ HTML 2.0 was developed by the Internet Engineering Task Force HTML Working Group in 1996.
- ▶ HTML 3.2
- ▶ HTML 4.0
- ▶ HTML 4.01
 - ▶ HTML 4.01 was released as a W3C Recommendation 24. December 1999.
 - ▶ HTML 4.01 is a minor update of corrections and bug fixes in HTML 4.0.
 - ▶ W3C will not continue to develop HTML. Future W3C work will be focusing on XHTML.

- ▶ XHTML 1.0
 - ▶ XHTML 1.0 reformulates HTML 4.01 in XML.
 - ▶ XHTML 1.0 was released as a W3C Recommendation 20. January 2000.
- ▶ HTML5
 - ▶ Web Hypertext Application Technology Working Group(WHATWG) and W3C came up with this in 2007
 - ▶ HTML 5 is a combination of HTML 4.01 and XHTML 1.0.
 - ▶ Many browsers are going to start supporting this in the future.
 - ▶ HTML 5 is backwards compatible.

You can write your HTML code in almost any available text editor, including notepad.

Open source text editor

Brackets <http://brackets.io/> Notepad++ <https://notepad-plus-plus.org/>

or

We'll use Eclipse which has built-in HTML Editor.

HTML Document will always be saved in **.html** extension or an **.htm** extension.

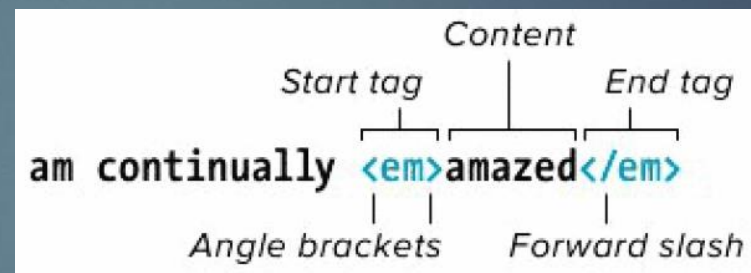
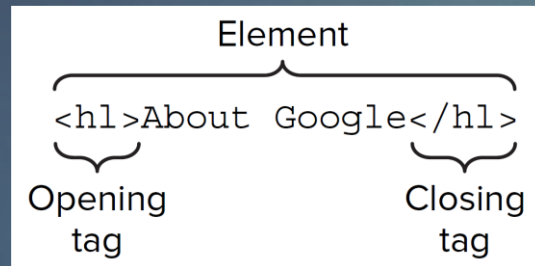
HTML Tags and Elements

Tags are enclosed in angle brackets `< >`

For Eg.: `<html>` Opening Tag, `</html>` Closing Tag.

Element is the combination of (opening & closing Tags and the content between them).

For Eg.:



`<p>`Part of this text is ``bold``. `</p>` is a PARAGRAPH element that contains a BOLD element

An HTML document is a collection of elements (text/media with context).

Empty tags vs Container tags

Some elements which does not requires **closing tags**, are known as Empty Tags or Elements.

For Eg.: ``

`
` begining of new line. **BR** stands for **BReak**.

`<hr />` puts a line across the page. **HR** stands for **H**orizontal **R**ule.

The elements which requires **opening** and **closing tags**, are known as Container Tags or Elements.

For Eg.: `<h1>` This is a heading `</h1>`

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

HTML Attributes and Values

HTML elements can have attributes which provides additional information about an element. Always specified in the opening tag and should contained value.

For Eg.:



```
<!DOCTYPE html>
<html>
<head>
<title>Align Attribute Example</title>
</head>
<body>
<p align="left">This is left aligned</p>
<p align="center">This is center aligned</p>
<p align="right">This is right aligned</p>
</body>
</html>
```

This is left aligned

This is center aligned

This is right aligned

Some Important Attributes

11

Attribute	Options	Function
title	User Defined	"Pop-up" title of the elements.
href	User Defined	The link address is specified in the href attribute opens.
class	User Defined	Classifies an element for use with Cascading Style Sheets.
id	User Defined	Names an element for use with Cascading Style Sheets.
bgcolor	numeric, hexadecimal, RGB values	Places a background color behind an element
background	URL	Places a background image behind an element
align	right, left, center	Horizontally aligns tags
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
width	Numeric Value	Specifies the width of tables, images, or table cells.
height	Numeric Value	Specifies the height of tables, images, or table cells.

Structural Elements

12

A standard HTML document has two main structural elements

head contains setup information for the browser & the Web page

For E.g., the title for the browser window, style definitions, JavaScript code, ...

body contains the actual content to be displayed in the Web page

```
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>My first HTML document</title>
</head>
  <body>
    <p> Hello world! </p>
  </body>
</html>
```

Comments and doctype

13

HTML has a mechanism for embedding comments that are not displayed when the page is rendered in a browser.

Eg.: `<!-- This is comment text -->`

Besides tags, text content, and entities, an HTML document must contain a doctype declaration as the first line. For

Eg.:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>My first HTML document</title>
  </head>
  <body>
    <p> Hello world! </p>
  </body>
</html>
```

Current version of HTML is 5 and it makes use of the following declaration: `<!DOCTYPE html>`

doctype

- ▶ DOCTYPE tells the consuming user agent (web browsers, web crawlers, validation tools) what type of document the file is. Using it ensures that the consumer correctly parses the HTML as you intended it.
- ▶ HTML 4.01 Strict
 - ▶ This DTD contains all HTML elements and attributes, but does NOT INCLUDE presentational or deprecated elements (like font).
 - ▶ `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- ▶ HTML 4.01 Transitional
 - ▶ This DTD contains all HTML elements and attributes, INCLUDING presentational and deprecated elements (like font).
 - ▶ `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`

doctype

- ▶ HTML 4.01 Frameset

- ▶ This DTD is equal to HTML 4.01 Transitional, but allows the use of frameset content.

- ▶ `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">`

- ▶ HTML 5

- ▶ `<!DOCTYPE html>`

<head> and <body> Elements

16

- THE **<HEAD>** ELEMENT IS WHERE YOU INCLUDE A **<TITLE>** ELEMENT (THAT APPEARS IN THE TITLE BAR OF THE BROWSER).
- YOU CAN ALSO INCLUDE LOTS OF OTHER TYPE OF INFORMATION IN THE **<HEAD>** ELEMENT.
 - Cascading Style sheet information, or a link to an external style sheet (or several).
 - “Meta” data, such as who authored the page, the type of content, and clues that search engines may (or may not) use to help categorize your page.
 - JavaScript code.
- THE **<BODY>** ELEMENT CONTAINS THE MAIN BULK OF THE MATERIAL TO BE DISPLAYED ON THE WEBPAGE.
 - Paragraphs.
 - Tables and lists.
 - Images.
 - JavaScript code.
 - PHP code can be included here too (if passed through a PHP parser before being served to the client’s browser).
 - Other embedded objects (videos, etc).

<head> Elements

17

Meta tags

The <meta> tag provides metadata about the HTML document.

Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

Some examples –

Example 1 - Define keywords for search engines:

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

Example 3 - Define the author of a page:

```
<meta name="author" content="Hege Refsnes">
```

Example 4 - Refresh document every 30 seconds:

```
<meta http-equiv="refresh" content="30">
```

<head> Elements (Cont.)

18

Title Tag

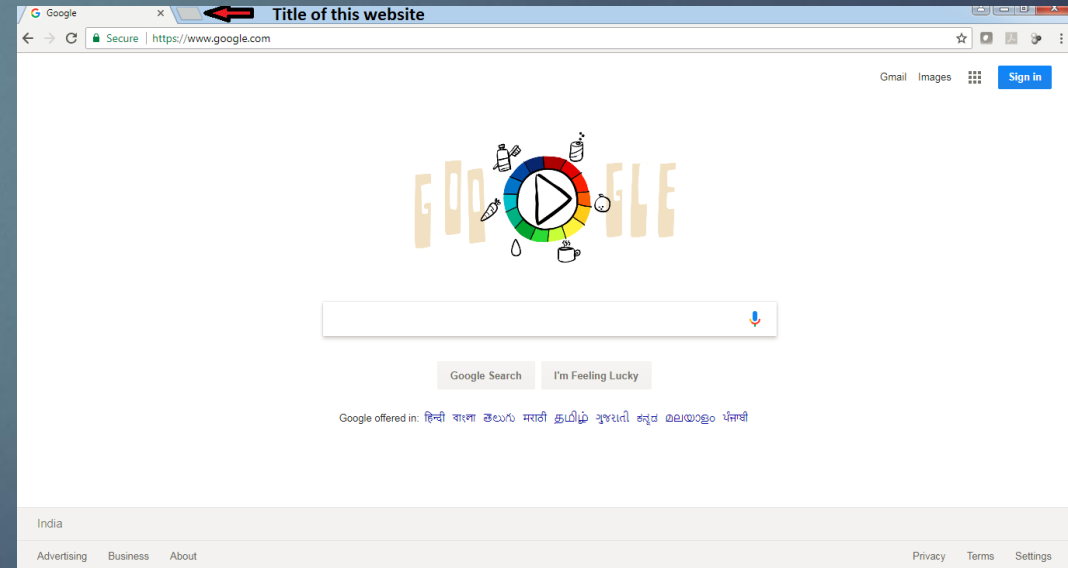
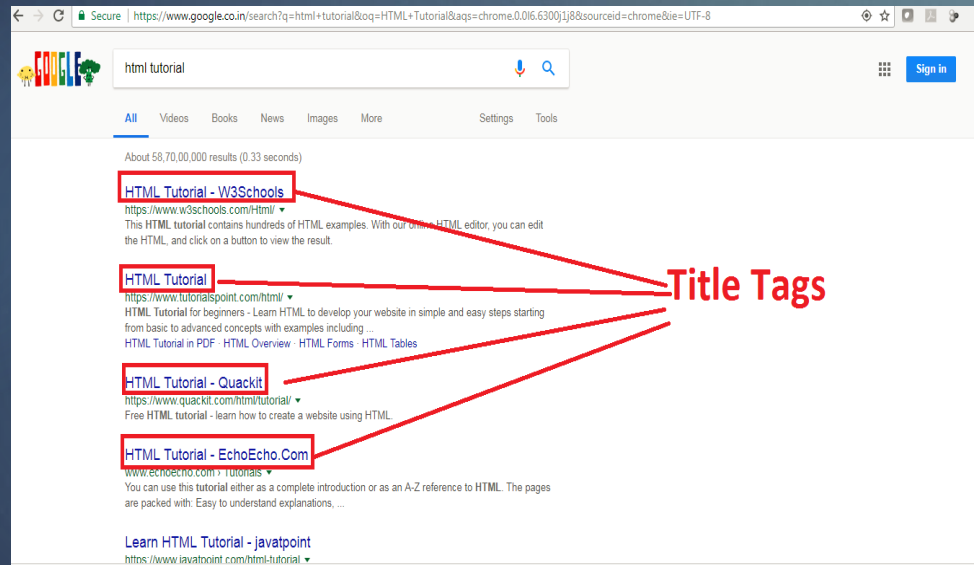
The <title> tag is required in all HTML documents and it defines the title of the document.

The <title> element: Defines a title in the browser toolbar.

Provides a title for the page when it is added to favorites.

Displays a title for the page in search-engine results.

Eg.:



<head> Elements (Cont.)

19

S
U
j
a
t
a

B
a
t
r
a

Link Tag

The <link> tag defines a link between a document and an external resource.

In HTML the <link> tag has no end tag.

Some Imp. Attributes –
charset, - To know browser, which character encoding is used.
href, - hyperlink.
rel, - Relation between linked document.
target. – It specifies where to open the linked document.

Example –

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

<head> Elements (Cont.)

Script Tags

The <script> tag is used to define a client-side script, such as a JavaScript.

The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

Example -

```
<html>
<head>
<title>Align Attribute Example</title>
</head>
<body>

<p id="demo"></p>
<script>document.getElementById("demo").innerHTML
= "Hello JavaScript!"; </script>

</body>
</html>
```

Hello JavaScript!

<head> Elements (Cont.)

21

Style Tag

The <style> tag is used to define style information for an HTML document.

Inside the <style> element you specify how HTML elements should render in a browser.

Example-

```
<html>
<head>
h1 {color:red;}
p {color:blue;}
</head>
<body>
<h1>A heading</h1>
<p>A paragraph.</p>
</body>
</html>
```

This is a heading

This is a paragraph.

Elements for the BODY section

22

Block-level elements

The BODY of a document consists of multiple block elements. If plain text is found inside the body, it is assumed to be inside a paragraph P. See the syntax rules for an explanation of the syntax used in the overview.

Headings

H1 - Level 1 header
H2 - Level 2 header
H3 - Level 3 header
H4 - Level 4 header
H5 - Level 5 header
H6 - Level 6 header

Text containers

P - Paragraph
PRE - Preformatted text
BLOCKQUOTE - Large quotation
ADDRESS - Address information

Text Level Elements

Logical Markups
Physical Markups
Special Markups

Lists

UL - Unordered list
OL - Ordered list
DIR - Directory list
MENU - Menu item list
LI - List item
DL - Definition list
DT - Definition term
DD - Definition

Others

DIV - Logical division
CENTER - Centered division
FORM - Input form
HR - Horizontal rule
TABLE - Tables

Elements for the BODY section

Headings

23

There are 6 types of heading tags.

Eg.: –

```
<html>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
<p><b>Tip:</b> Use h1 to h6 elements only for headings.
Do not use them just to make text bold or big. Use other tags
for that.</p>
</body>
</html>
```

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

Tip: Use h1 to h6 elements only for headings. Do not use them just to make text bold or big. Use other tags for that.

Elements for the BODY section (Cont.)

24

<p> - Paragraph Tag and <pre> - Preformatted Tag

<p> Tag - Another way to structure your text in paragraph forms.

<Pre> Tag - is used to apply structural exactness.

Eg.:

```
<html>
<body>
<p>This is a paragraph of text.</p>
<p>This is a second paragraph of text.</p>
<pre>This is preformatted text with    exact space,
line and breaks.</pre>
</body>
</html>
```

This is a paragraph of text.

This is a second paragraph of text.

This is preformatted text with exact space,
line and breaks.

Elements for the BODY section (Cont.)

25

<blockquote> Tag and <address> Tag

Blockquote Tag - Indicates that the enclosed text is an extended quotation.

Address Tag - Address Information of the Author/Owner.

Eg.:

```
<html>
<body>
<blockquote cite="http://www.sujatabatra.com/">
  <p>This is a quotation taken from the Sujata Batra.</p>
</blockquote>

<address>
Written by
<a href="mailto:sujatabatra@hotmail.com">Sujata
Batra</a>.<br>
Visit us at:<br>
www.SujataBatra.com<br>
S1, TowerX, Some Business Bay,<br>
Outer Ring Road, Delhi.
</address>
</body>
</html>
```

This is a quotation taken from the Sujata Batra.

Written by Sujata Batra.

Visit us at:

www.SujataBatra.com

S1, TowerX, Some Business Bay,

Outer ring road, Delhi.

Elements for the BODY section (Cont.)

26

S
U
j
a
t
a

B
a
t
r
a

Text Formatting Elements

Physical markup

TT - Teletype

I - Italics

B - Bold

U - Underline

STRIKE - Strikeout

BIG - Larger text

SMALL - Smaller text

SUB - Subscript

SUP - Superscript

Logical markup

EM - Emphasized text

STRONG - Strongly emphasized

DFN - Definition of a term

CODE - Code fragment

SAMP - Sample text

KBD - Keyboard input

VAR - Variable

CITE - Short citation

Special markup

A - Anchor

IMG - Image

BASEFONT - Default font size

APPLET - Java applet

PARAM - Parameters for Java applet

FONT - Font modification

BR - Line break

MAP - Client-side imagemap

AREA - Hotzone in imagemap

Elements for the BODY section (Cont.)

27

S
U
j
a
t
a

B
a
t
r
a

Text Formatting Elements (Physical Markup)

Tag	Description
<code>....</code>	- bold.
<code><i>.....</i></code>	- italic.
<code><u>....</u></code>	- underline.
<code><strike>...</strike></code>	- strikethrough.
<code><sub>....</sub></code>	- subscript.
<code><sup>....</sup></code>	- superscript.
<code><big>....</big></code>	- bigger font (one font size bigger).
<code><small>....</small></code>	- small font (one font size smaller).
<code><tt>....</tt></code>	- typewriter (monospaced).

Elements for the BODY section (Cont.)

28

S
U
j
a
t
a
B
a
t
r
a

Text Formatting Elements (Physical Markup)

```
<html>
<body>
<b> Sujata Training. </b> <br>
<i> Java Training. </i> <br>
<u> Powered by – Sujata Batra. </u> <br>
<strike> Text. </strike> <br>
<small> Copyright &copy; Sujata Batra Pvt.Ltd. </small>
<sub> Subscript. </sub>
<sup> Superscript. </sup>
</body>
</html>
```

Sujata Training.

Java Training.

Powered by – Sujata Batra.

~~Text.~~

Copyright © Sujata Batra Pvt.Ltd. Subscript. ^{Superscript.}

Elements for the BODY section (Cont.)

29

Text Formatting Elements (Logical Markup)

Tag		Description
<code></code>	-	Emphasized
<code></code>	-	Strongly emphasized
<code><dfn></code>	-	A definition
<code><code></code>	-	Represents computer code
<code><kbd></code>	-	keyboard characters
<code><var></code>	-	Program variable
<code><cite></code>	-	A citation

Elements for the BODY section (Cont.)

30

Text Formatting Elements (Logical Markup)

```
<html>
```

```
<body>
```

```
<em> Sujata Training. </em> <br>
```

```
<strong> Java Training. </strong> <br>
```

```
<dfn> Powered by - SujataBatra. </dfn> <br>
```

```
<code> Text. </code> <br>
```

```
<kbd> Subscript. </kbd>
```

```
<var> Superscript. </var>
```

```
<cite> Superscript. </cite>
```

```
</body>
```

```
</html>
```

Sujata Training.

Java Training.

Powered by - SujataBatra.

Text.

Subscript. *Superscript. Superscript.*

Elements for the BODY section (Cont.)

31

S
U
j
a
t
a

B
a
t
r
a

Text Formatting Elements (Special markup)

Links and Navigation

Anchor Element-

An anchor can be used to create a link to another document (with the href attribute).

Types –

External : `Our Best Friend`

Internal : `contact`

Image Tag-

The syntax for the tag to insert image into the webpage is-

``

Eg.: ``

Elements for the BODY section (Cont.)

32

S
U
j
a
t
a
B
a
t
r
a

Unordered List and Odered Lists

Unordered Lists - tag. Item lists in tag. The list items will be marked with bullets.

Ordered Lists - tag. Item lists in tag. . The list items will be marked with numbers.

Eg.:

```
<html>
<body>
<h2>Unordered List </h2>
<ul>
  <li>Java</li>
  <li>Python</li>
  <li>Ruby</li>
</ul>
<h2>Ordered List </h2>
<ol>
  <li>Java</li>
  <li>Python</li>
  <li>Ruby</li>
</ol>
</body>
</html>
```

Unordered List

- Java
- Python
- Ruby

Ordered List

1. Java
2. Python
3. Ruby

Elements for the BODY section (Cont.)

33

S
U
j
a
t
a

B
a
t
r
a

Div Tag

<div> tag – Used to defines a division or a section in an HTML document. And to group block-elements to format them with CSS.

Eg.:

```
<html>
<body>
<div style="color:#00FF00">
  <h2>Sujata Academy</h2>
  <p>Welcome to Java Training</p>
</div>
</body>
</html>
```

Sujata Academy
Welcome to Java Training.

Elements for the BODY section (Cont.)

34

Table Element

<table> Tag :

<tr>	Table R ow -	Defines a new row,
<td>	Table D ata -	Defines a single cell,
<th>	Table H eadings -	Defines header cell.

```
<html>
<body>
<style> table, th, td { border: 1px solid black; }
</style>
<table>
  <tr>
    <th>Day</th> <th>Session</th>
  </tr>
  <tr>
    <td>Thursday</td> <td>HTML</td>
  </tr>
  <tr>
    <td>Friday</td> <td>CSS</td>
  </tr>
</table>
</body>
</html>
```

Day	Session
Thursday	HTML
Friday	CSS

Tables

- ▶ Display data in a tabular format.
- ▶ Helps in positioning the contents of the page in a more structured way.
- ▶ `<TABLE> ... </TABLE>` : define a table.
- ▶ Some attributes
 - ▶ `ALIGN = LEFT | RIGHT | CENTER`
 - ▶ `BORDER = n (Number of Pixels)`
 - ▶ `BGCOLOR = "color" | "#rrggbb"`
 - ▶ `CELLSPACING = n (Number of Pixels)`
 - ▶ `CELLPADDING = n (Number of Pixels)`
 - ▶ `WIDTH= % Of Parent | n (pixels)`

Table structure

```
<TABLE BORDER=1>      <!-- start of table definition -->
<CAPTION> caption contents </CAPTION>  <!--caption definition -->

<TR>                  <!-- start of header row definition -->
    <TH> first header cell contents </TH>
    <TH> last header cell contents </TH>

</TR>                  <!-- end of header row definition -->
<TR>                  <!-- start of first row definition -->
    <TD> first row, first cell contents </TD>
    <TD> first row, last cell contents </TD>

</TR>                  <!-- end of first row definition -->
<TR>                  <!-- start of last row definition -->
    <TD> last row, first cell contents </TD>
    <TD> last row, last cell contents </TD>

</TR>                  <!-- end of last row definition -->
</TABLE>               <!-- end of table definition -->
```

Creating tables

Simple sample table :

Heading1	Heading2
Cell 1	Cell 2
Cell 3	Cell 4

```
<TABLE BORDER=1 CELLSPACING=1 CELLPADDING=1 WIDTH=30%>
```

```
<CAPTION> Simple sample table </CAPTION>
```

```
<TR>
```

```
<TH>Heading1</TH>
```

```
<TH>Heading2</TH>
```

```
</TR>
```

```
<TR>
```

```
<TD>Cell 1</TD>
```

```
<TD>Cell 2</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>Cell 4</TD>
```

```
<TD>Cell 5</TD>
```

```
</TR>
```

```
</TABLE>
```


Creating tables

```
<TABLE WIDTH="100%" BORDER=1 BGCOLOR=gray>
  <TR ALIGN=CENTER >
    <TD COLSPAN=2>This cell spans 2 columns!</TD>
    <TD> Cell </TD></TR>
  <TR ALIGN=CENTER >
    <TD ROWSPAN=3>This cell spans 3 rows!!</TD>
    <TD> Cell </TD>
    <TD> Cell </TD></TR>
  <TR ALIGN=CENTER >
    <TD> Cell </TD>
    <TD> Cell </TD></TR>
  <TR ALIGN=CENTER >
    <TD> Cell </TD>
    <TD> Cell </TD>
  </TR>
</TABLE>
```

This cell spans 2 columns!		Cell
This cell spans 3 rows!!	Cell	Cell
	Cell	Cell
	Cell	Cell

Elements for the BODY section (Cont.)

39

S
U
j
a
t
a

B
a
t
r
a

Form Elements

<form> - It is a method of accepting inputs from user. A form is an area that can contain form elements.

Eg.:

```
<form name="form1" action="abc.asp" method=get>
```

```
<!-- form elements -->
```

```
</forms>
```

Name- is used for future manipulation of data by scripting language.

Action- indicates a program on the server that will be executed when this form is submitted. Mostly it will be an ASP or a CGI script.

Method- indicates the way the form is submitted to the server - popular options are GET/POST.

Elements for the BODY section (Cont.)

40

Form Elements

Form Elements	Description
Text Field	Can create a Text Field by using Input Element with Type Attribute.
Pass Word Field	When text is entered in Pass Word Field it shows * * * * Symbol
Combo Box	It can have multiple values and it allows user to select one value at a time
List Box	It can have multiple values and allows user to select more than one value at a time
Radio Button	Can create a Radio Button by using Input Element with Value and Name Attribute
Check Box	Can create Check box by Using Input Element
Command Button	This is useful for submitting any data that is helpful in transferring data across different interfaces

Elements for the BODY section (Cont.)

41

Form Elements

```
<html>
<body>
<form name="frm">
  Enter Your Login ID : <input type=textsize=20><br>
  Enter Your Pass Word : <input type=Password maxlength=8
size=20><br>
  <select name=combo1>
    <option>Value1</option>
    <option>Value2</option>
    <option>Value3</option>
  </select> <br><br><br>
  <select name=combo1 multiple>
    <option>Value1</option>
    <option>Value2</option>
    <option>Value3</option>
  </select><br><br>
  Select Gender
  <input type=Radio value=Male name=Checked>Male
  <input type=Radio value=Female name=Checked>Female <br>

  Select Hobbies
  <input type= Checkbox >Cricket
  <input type= Checkbox > Reading
  <input type= Checkbox > Watching TV<br><br>

  <input type=submit value="Load Data">
  <input type=button value="Update Data">

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

Enter Your Login ID :

Enter Your Pass Word :

Value1 ▾

Value1
Value2
Value3

Select Gender ☒ Male ☐ Female

Select Hobbies ☒ Cricket ☐ Reading ☐ Watching TV

Load Data Update Data

Elements for the BODY section (Cont.)

Character Entities

42

Some characters like the < character, have a special meaning in HTML, and therefore cannot be used in the text. The most common character entities:

Result	Description	Entity Name
 	non-breaking space	
<	less than	<
>	greater than	>
&	ampersand	&
“	quotation mark	"
‘	apostrophe	'

Some Other Commonly Used Character Entities

©	copyright	©
®	registered trademark	®
£	pound	£
¥	yen	¥

HTML



Overview

44

S
U
j
a
t
a

B
a
t
r
a

1. Advance version of HTML.
2. In 2008, the first HTML5 public draft was released
3. HTML5 W3C Final Recommendation was released 28. October 2014.
4. New elements, attributes, and behaviors were introduced.
5. It helps to create more powerful website and interactive web applications.
6. HTML5 comes with XML syntax.
7. HTML5 is to compete with Flash and Silverlight.
8. Empowering Mobile devices.

Technical Advantages Over Previous Version.

45

1. Audio and Videos are integral part of HTML5 specifications e.g. <audio> and<video> tags.
2. Vector graphics is integral part of HTML5 e.g. SVG and canvas.
3. JS GeoLocation API in HTML5 helps identify location of user browsing any website (provided user allows it).
4. Full duplex communication channels can be established with Server using Web Sockets.
5. Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5.
6. Application Cache, Web SQL database and Web storage is available as client side storage.
7. Retain Backward Compatibility with previous versions of HTML5.

HTML5 Technology Functions

46

S
U
j
a
t
a
B
a
t
r
a

Semantics: allowing you to describe more precisely what your content is.

Connectivity: allowing you to communicate with the server in new and innovative ways.

Offline & Storage: allowing webpages to store data on the client-side locally and operate offline more efficiently.

Multimedia: making video and audio first-class citizens in the Open Web.

2D/3D Graphics & Effects: allowing a much more diverse range of presentation options.

Performance & Integration: providing greater speed optimization and better usage of computer hardware.

Device Access: allowing for the usage of various input and output devices.

Styling: letting authors write more sophisticated themes.

HTML5 New Tags and Elements

HTML5 Introduces 28 New Elements, Some of them are mentioned here.

47

S

U

J

a

t

a

B

a

t

r

a

Navigation:

<article>
<aside>
<header>
<hgroup>
<footer>
<figure>
<figcaption>
<nav>
<section>

Multimedia/Interactivity:

<audio>
<canvas>
<embed>
<source>
<track>
<video>

New <input> types:

color
date
datetime
datetime-local
email
month
number
range
search
tel
time
url
week

Miscellaneous:

<bdi>
<command>
<datalist>
<details>
<keygen>
<mark>
<meter>
<output>
<progress>
<summary>
<rp>
<rt>
<ruby>
<time>
<wbr>

Elements removed in HTML5

48

S
u
j
a
t
a

B
a
t
r
a

Element	Use instead
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
<center>	CSS
<dir>	
	CSS
<frame>	
<frameset>	
<noframes>	
<strike>	CSS
<tt>	CSS

Migration from HTML4 to HTML5

49

S
U
j
a
t
a

B
a
t
r
a

HTML4	HTML5
<div id="header">	<header>
<div id="menu">	<nav>
<div id="content">	<section>
<div id="post">	<article>
<div id="footer">	<footer>

Defining HTML5 Documents

50

S
U
j
a
t
a

B
a
t
r
a

Remember the DOCTYPE declaration-

```
<!DOCTYPE html>
```

Again, HTML5 simplifies this line:

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

The default character encoding (charset) declaration

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

Semantic Elements

51

A semantic element clearly describes its meaning to both the browser and the developer.

Eg. of non-semantic elements: `<div>` and `` - Tells nothing about its content.

Eg. of semantic elements: `<form>`, `<table>`, and `` - Clearly defines its content.

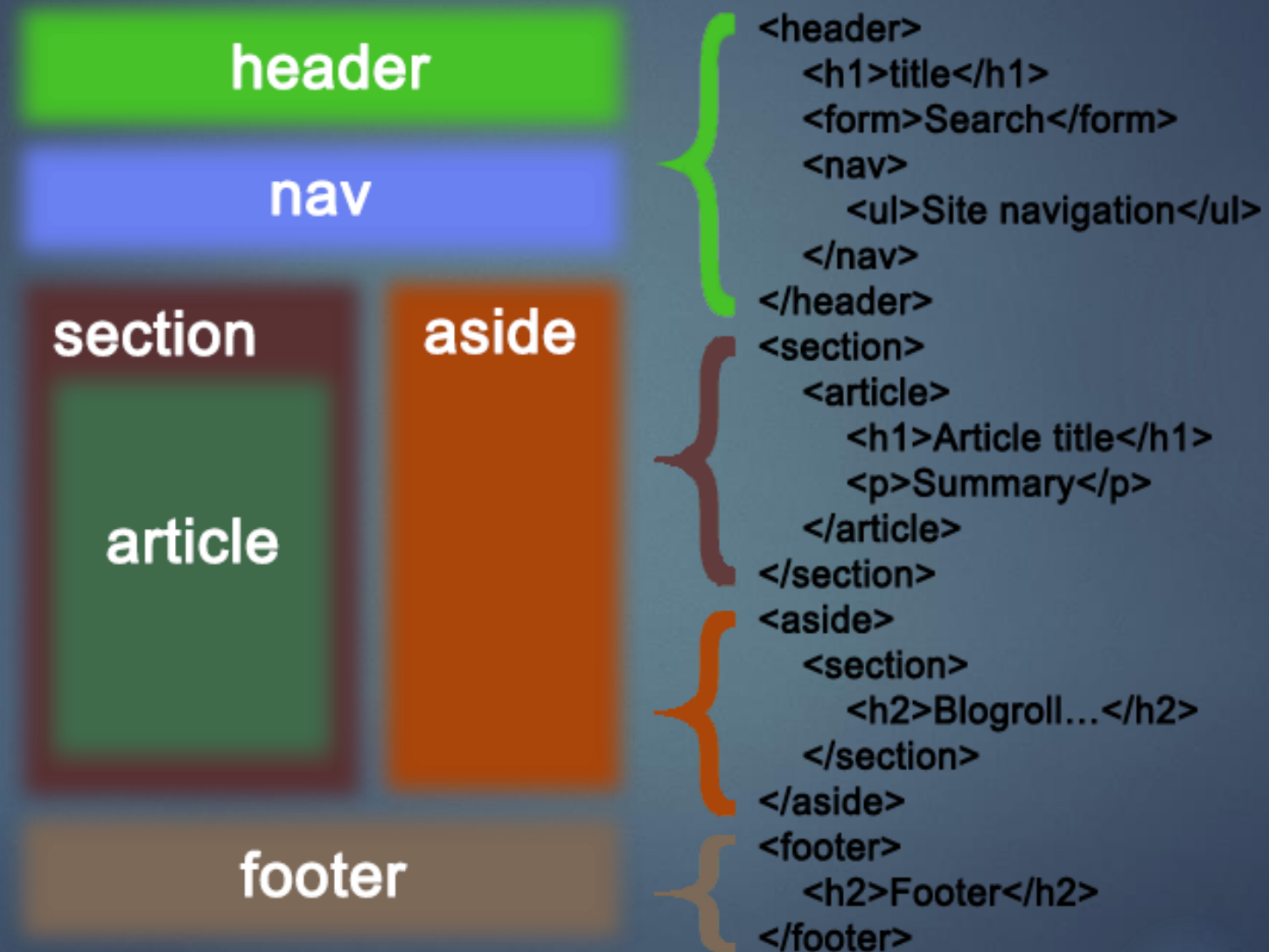
Many web sites contain HTML code like: `<div id="nav">` `<div class="header">` `<div id="footer">` to indicate navigation, header, and footer.

HTML5 offers new semantic elements to define different parts of a web page:

<code><article></code>	<code><header></code>
<code><aside></code>	<code><main></code>
	<code><mark></code>
<code><details></code>	<code><nav></code>
<code><figcaption></code>	<code><section></code>
<code><figure></code>	<code><summary></code>
<code><footer></code>	<code><time></code>

Semantic Elements

52



Graphics API (Canvas and SVG)

53

S
U
j
a
t
a

B
a
t
r
a

Previously possible with Flash, VML, Silverlight.

Very complex to do in JavaScript without plugins (for example, rounded corners or diagonal lines).

Provide native drawing functionality on the Web.

Completely integrated into HTML5 documents (Part of DOM).

Can be styled with CSS.

Can be controlled with JavaScript.

(Canvas and SVG)

Both have their own unique features and can be used combined.

54

Canvas	SVG
Low level	High Level
Immediate mode	Retained mode
Fixed size	Scalable
Best for keyboard-based apps	Best for mouse-based apps
Animation (no object storage)	Medium animation
Pixels	XML object model
No interaction	User interaction (hit detection, events on the tree)

Canvas

55

<canvas> element as “a resolution-dependent bitmap canvas which can be used for rendering graphs, game graphics, or other visual images on the fly.” A canvas is a rectangle in your page where you can use JavaScript to draw anything you want and CSS for styling. In 2D context and 3D context (Web GL).

Eg.:

```
<canvas id="myCanvas" width="200" height="100" style="border:4px solid #d3d3d3;">
```

Your browser does not support the HTML5 canvas tag. </canvas>

```
<script>
```

```
var c = document.getElementById("myCanvas");
```

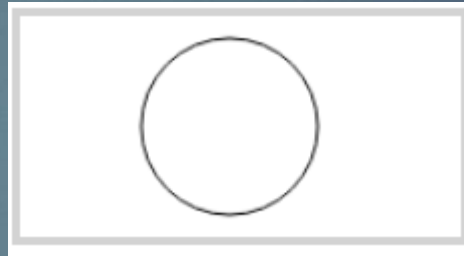
```
var ctx = c.getContext("2d");
```

```
ctx.beginPath();
```

```
ctx.arc(95,50,40,0,2*Math.PI);
```

```
ctx.stroke();
```

```
</script>
```



SVG – Scalable Vector Graphics

56

S
U
j
a
t
a

B
a
t
r
a

<svg> element Modularized, XML-based language for describing 2D vector and mixed vector/raster graphics. You can zoom SVG graphics to any level.

Eg.:

```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40"  
    stroke="blue" stroke-width="4" fill="orange" />
```

Sorry, your browser does not support inline SVG.

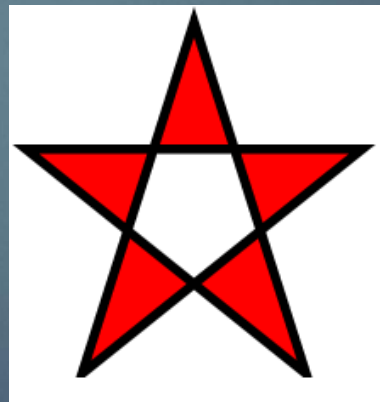
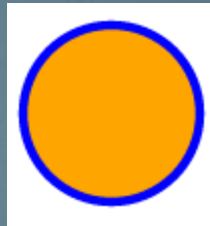
```
</svg> <br>
```

```
<svg width="300" height="200">  
  <polygon points="100,10 40,198 190,78 10,78  
160,198"
```

```
  style="fill:red;stroke:black;stroke-width:5;fill-  
rule:evenodd;" />
```

Sorry, your browser does not support inline SVG.

```
</svg>
```



HTML5 Media Elements - Audio and Video

57

S
u
j
a
t
a

B
a
t
r
a

<audio> and <video> - are two new HTML5 media elements can be controlled using Audio/Video API, have native support in the browser (Embedded Codecs).

AV Containers and Codecs

1. Audio and Video containers
H264 and Ogg.
2. Audio and Video codecs (algorithm used to encode and decode an audio or video stream)
Audio – AAC, MP3, Vorbis.
Video – H264, MP4, Theora.
3. You can add multiple formats per (Audio/Video).

Example

58

S
U
j
a
t
a

B
a
t
r
a

- ▶ **HTML <video> Element**

```
<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>
```

- ▶ **HTML <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>
```

HTML5 Plug-ins

59

S
U
j
a
t
a

B
a
t
r
a

To extend the functionality of the HTML browser. Plug-ins are also known as Helper Applications. Popular example of plug-ins are Java applets. Plug-ins can be added to web pages with the `<object>` tag or the `<embed>` tag.

1. **`<object>` Element** – It is used to embed plug-ins (like Java applets, PDF readers, Flash Players) in web pages.

Eg.: `<object width="200" height="60" data="SujataTraining.swf"></object>`

2. **`<embed>` Element** – Used to embed object within an HTML document. It does not have closing tag.

Eg.: `<embed width="200" height="60" src="SujataTraining.swf">`

1. Currently limited HTML5 IDE support

- SEEdit (Text Editor)

2. HTML5 simplicity reduces work.

3. Advanced browser development tools allow for “semi-rapid” development and live coding.

- Chrome Developer Console.

- Safari Web Inspector.

- Firefox Firebug Add-on and Opera Firefly.

4. More IDE support as specification solidifies.

CSS

What is CSS ?

62

CSS stands for “Cascading Style Sheets”

Cascading: refers to the procedure that determines which style will apply to a certain section, if you have more than one style rule.

Style: how you want a certain part of your page to look. You can set things like color, margins, font, etc for things like tables, paragraphs, and headings.

Sheets: the “sheets” are like templates, or a set of rules, for determining how the webpage will look.

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML.

CSS

History

- CSS1 was the first edition introduced in 1996.
- CSS2 was published in 1998 and provides enhancement over CSS1.
- CSS2.1 was the last 2nd generation edition of CSS.
- CSS 3 is the latest edition. Several new functionalities have been provided through CSS3.

Functions like rounded corners, background decoration, box shadows, which are demonstrated in the subsequent sections, are introduced in this version.

CSS

64

S
U
j
a
t
a

B
a
t
r
a

Advantages

- A web application will contains hundreds of web pages, which are created using HTML.
- Formatting these HTML pages will be a laborious process, as formatting elements need to be applied to each and every page.
- CSS saves lots of work as we can change the appearance and layout of all the web pages by editing just one single CSS file.

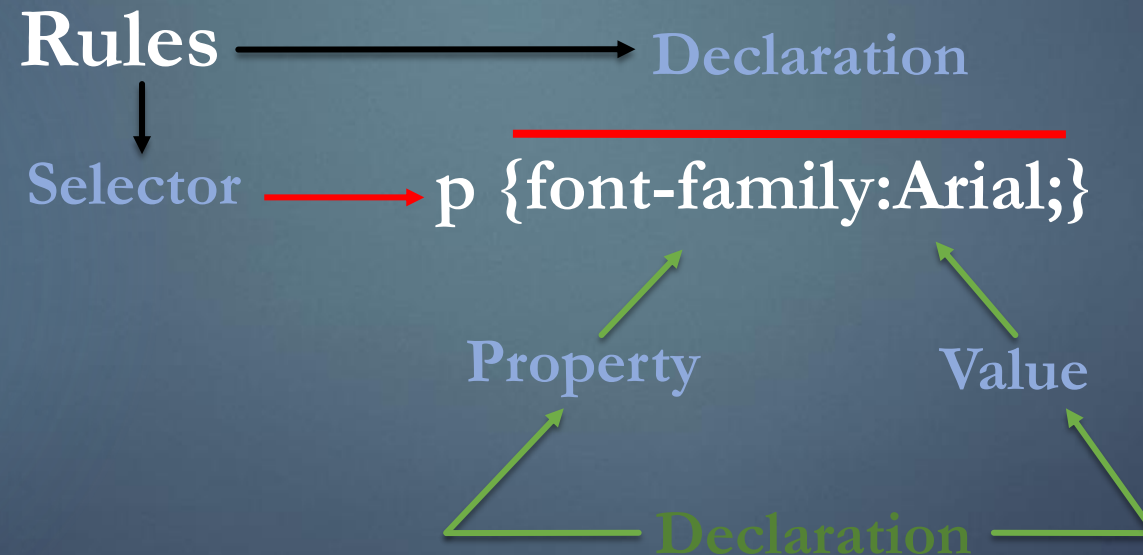
CSS Syntax Rules

65

Rule have two parts - Selector and declaration.

Selector: The HTML element you want to add style to.
<p> <h1> <table> etc

Declaration: The statement of style for that element. Made up of property and value.



CSS Style Example

66

Selector - I want the text color of my paragraph to be red and the background color to be black.

```
<html>
<head>
<style> p {font-family:Arial; color: red;
background-color:black;} </style>
</head>
<body>
<p> <b> Welcome to Snapdeal Academy </b>
</p>
</body>
</html>
```

Welcome to Snapdeal Academy

CSS Selectors

67

CSS selectors allow you to select and manipulate HTML elements based on their id, class, type, attribute, and more.

Examples –

Declaring a CSS Rule for a **Class** Attribute

the HTML

```
<a class="pdf" href="brochure.pdf">Brochure</a>
```

the CSS

```
.pdf {background: url(images/pdf.gif) no-repeat left 50%;}
```

use a **period** when writing a rule for a **class**

Declaring a CSS Rule for an **Id** Attribute

the HTML

```
<div id="wrapper">Main Content</div>
```

the CSS

```
#wrapper {width: 750px; margin: 0 auto;}
```

use a **pound sign** when writing a rule for a **id**

CSS Selectors (Cont.)

68

S
U
j
a
t
a

B
a
t
r
a

Declaring a CSS Rule for a Elements Attribute

It will style all the content of that element which you are selecting.

Example-

The HTML

```
<p> Welcome to the Sujata Academy </p>
```

```
<p> <b><i> Powered by – Sujata Batra </i></b></p>
```

The CSS

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

CSS Selectors (Cont.)

69

Grouping Selectors

You can group all the selectors of same style to minimize the code. The selectors should be separated with comma.

Example-

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

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

Grouped Selectors-

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

Inserting a StyleSheet

70

S
U
j
a
t
a

B
a
t
r
a

You can do in three different ways-

1. External Style Sheet

Styles are specified in an external CSS file. you can change the looks of entire website by using single external style sheet.

Eg.: `<head> <link rel="stylesheet" type="text/css" href="ex1.css" /> </head>`

2. Internal Style Sheet

To Apply specific styles to a single HTML file inside the head section of an HTML page.

Eg.: `<style> p { text-align:left; font-size:24px; } </style>`

3. Inline Styles

Styles are specified inside an HTML tag/element.

Eg.: `<p style="font-family:Algerian; font-size:28px;"> Demo of Inline Style </p>`

Inserting a StyleSheet

71

S
U
j
a
t
a

B
a
t
r
a

Multiple Style Sheets – It can be referenced inside an HTML document.

The question is, what styles will be applicable when there is more than one style specified?

All styles cascade into a new virtual style sheet by applying the following rules, where the higher number has the greater priority:

1. Browser default.
2. External Stylesheet.
3. Internal Stylesheet (styles defined in head section).
4. Inline Style (styles defined in an HTML element).

Formatting with CSS Properties

72

S
U
j
a
t
a

B
a
t
r
a

CSS Background

We can use CSS Background properties to define the background effects of an element.

The following properties can be used for background effects :

- background-color
 - The background-color property is used to specify the background color of the element.
- background-image
 - The background-image property is used to set an image as a background of an element.
- background-repeat
- background-attachment
 - used to specify if the background image is fixed or scroll with the rest of the page in browser window.
- background-position
 - used to define the initial position of the background image.
 - By default, the background image is placed on the top-left of the webpage.

Formatting with CSS Properties

73

S
U
j
a
t
a

B
a
t
r
a

CSS Background Image

You can use an image as the background for an element using background-image property.

Example-

```
body{  
    background-image:url('java.png');  
}
```

By default, the image is repeated, both horizontally and vertically, so as to cover the entire body (or the element on which it is applied).

Formatting with CSS Properties

74

S
U
j
a
t
a
B
a
t
r
a

CSS Background Color

The **background-color** property is used to specify the background color of an element.

Example-

```
body {  
    background-color:darkblue;  
}
```

Similarly, we can specify the background for any element (wherever applicable).

```
p {  
    background-color:orange;  
}
```

Formatting with CSS Properties

75

S
U
j
a
t
a

B
a
t
r
a

CSS Background Position

If the background image disturbs the text, i.e. if the text cannot be read clearly due to the image in the background, we can set the position of the background image.

Example-

```
body {  
    background-image:url("SomeImage.jpg");  
    background-repeat:no-repeat;  
    background-position:right top;  
}
```

Formatting with CSS Properties

76

S
U
j
a
t
a

B
a
t
r
a

CSS Background Shorthand

You can also specify all the properties in a single property.
This property is known as shorthand property.

For specifying shorthand property, you just need to use **background**.

Example-

```
body {  
    background:cyan url("SomeImage.jpg") no-repeat right top;  
}
```

Formatting with CSS Properties

77

S
U
j
a
t
a

B
a
t
r
a

Text Formatting

The following properties can be used for formatting text :

1. Text Color
2. Text Alignment
3. Text Decoration
4. Text Transformation
5. Text Indentation

Formatting with CSS Properties

78

S
U
j
a
t
a

B
a
t
r
a

Text Alignment

We can either align the text to the left, right, center or we can make it justified.

Example-

```
p { text-align:left;}  
h1 {text-align:center;}
```

Text Color

The color property is used to set the color of text.

Example-

```
body { color:blue;}  
p1 {color:magenta;}
```

Formatting with CSS Properties

79

S
U
j
a
t
a

B
a
t
r
a

Text Decoration

You can use **text-decoration** property to set or remove decorations from text.

Example-

```
p {text-decoration:overline;}  
p {text-decoration:line-through;}  
p {text-decoration:underline;}
```

Text Transformation

You can use text-transform property to specify uppercase and lowercase letters of any text.

Example-

```
h1 {text-transform:uppercase;}  
h2 {text-transform:lowercase;}  
p {text-transform:capitalize;}
```

Formatting with CSS Properties

80

CSS Font

CSS font properties are used to define the font family, size, style and boldness of the text.
In CSS, there are two types of font family names:

generic family - a group of font families with a similar look (like "Serif" or "Monospace").
font family - a specific font family (like "Times New Roman" or "Arial").

Comments in CSS

`/* comment */` - This is comment used in CSS.

Formatting with CSS Properties

81

S
U
j
a
t
a

B
a
t
r
a

CSS Font Family

The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

Example :

```
p { font-family:"Arial", Times, "Sans-serif";}
```

CSS Font Style

You can use the property font-style to specify mostly italic text. It has three values – Normal, Italic, Oblique (similar to italic).

Formatting with CSS Properties

82

S
U
j
a
t
a

B
a
t
r
a

CSS Font Size

You can use the **font-size** property to set the size of text. The font-size value can be absolute or it can be relative.

Example-

```
h1 {  
    font-size: 30px;  
}
```

```
p {  
    font-size: 14px;  
}
```


Formatting with CSS Properties

83

S
U
j
a
t
a

B
a
t
r
a

CSS Font Size with em (Relative Size)

You may face resizing problems, when you use older versions of browsers. To avoid such problems, you can use set font size using em, instead of pixels.

The em size unit is a W3C recommendation. 1 em is equal to the current font size. The default text size is 16 px. So, the default size of 1 em is 16 px.

Example

```
h2 {  
    font-size: 1.875em; /* 30px/16=1.875em */  
}
```

```
p {  
    font-size: 0.875em; /* 14px/16=0.875em */  
}
```

CSS Units

- ▶ CSS has several different units for expressing a length.
- ▶ Many CSS properties take "length" values, such as width, margin, padding, font-size, border-width, etc.
- ▶ Length is a number followed by a length unit, such as 10px, 2em, etc.
- ▶ A whitespace cannot appear between the number and the unit. However, if the value is 0, the unit can be omitted.
- ▶ There are two types of length units:
 - ▶ absolute
 - ▶ relative.

Absolute Lengths

- ▶ The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

- ▶ Pixels (px) are relative to the viewing device. For low-dpi devices, 1px is one device pixel (dot) of the display. For printers and high resolution screens 1px implies multiple device pixels.

Relative Lengths

- ▶ Relative length units specify a length relative to another length property. Relative length units scales better between different rendering mediums.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

- ▶ Viewport = the browser window size. If the viewport is 50cm wide, 1vw = 0.5cm.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  font-size: 16px;
  line-height: 1em;
}

div {
  font-size: 30px;
  border: 1px solid black;
}

span {
  font-size: 0.5em;
}
</style>
</head>
<body>
```

```
<p>These paragraphs have a calculated line-height of: 2x16px = 32px.</p>
<p>These paragraphs have a calculated line-height of: 2x16px = 32px.</p>
<p>These paragraphs have a calculated line-height of: 2x16px = 32px.</p>
<div>The font-size of the div element is set to 30px. <span>The span
element inside the div element has a font-size of 0.5em, which equals to
0.5x30 = 15px</span>.</div>
```

These paragraphs have a calculated line-height of: $2 \times 16\text{px} = 32\text{px}$.

These paragraphs have a calculated line-height of: $2 \times 16\text{px} = 32\text{px}$.

These paragraphs have a calculated line-height of: $2 \times 16\text{px} = 32\text{px}$.

The font-size of the div element is set to 30px. The span
element inside the div element has a font-size of 0.5em, which equals to $0.5 \times 30 = 15\text{px}$.

Formatting with CSS Properties

88

S
U
j
a
t
a

B
a
t
r
a

CSS Links

You can use CSS styles to style any link. Links can be styled in different ways by using any CSS property like color, font-family etc.

Links can be in one of the following states :

- **a: link** – Unvisited link
- **a: visited** – A link that the user has visited
- **a: hover** – A link over which the mouse pointer is moving
- **a: active** – A link, which has been just clicked

Links can be styled according to their states.

Formatting with CSS Properties

89

S
U
j
a
t
a

B
a
t
r
a

CSS Links

```
a {  
  font-weight: bold;  
}  
a:link {  
  color: black;  
}  
a:visited {  
  color: gray;  
}  
a:hover {  
  text-decoration: none;  
  color: white;  
  background-color: navy;  
}  
a:active {  
  color: aqua;  
  background-color: navy;  
}
```

Styling Links

link - before a visit

visited - after it has been visited

hover - when your mouse is over it but you have not clicked

active - you have clicked it and you have not yet seen the new page

Formatting with CSS Properties

90

S
U
j
a
t
a

B
a
t
r
a

CSS List

You can use CSS list properties for

- Setting different list item markers for ordered lists
- Setting different list item markers for unordered lists
- Set an image as the list item marker

Values-

- ❖ list-style-type
- ❖ list-style-image

Formatting with CSS Properties

91

S
U
j
a
t
a

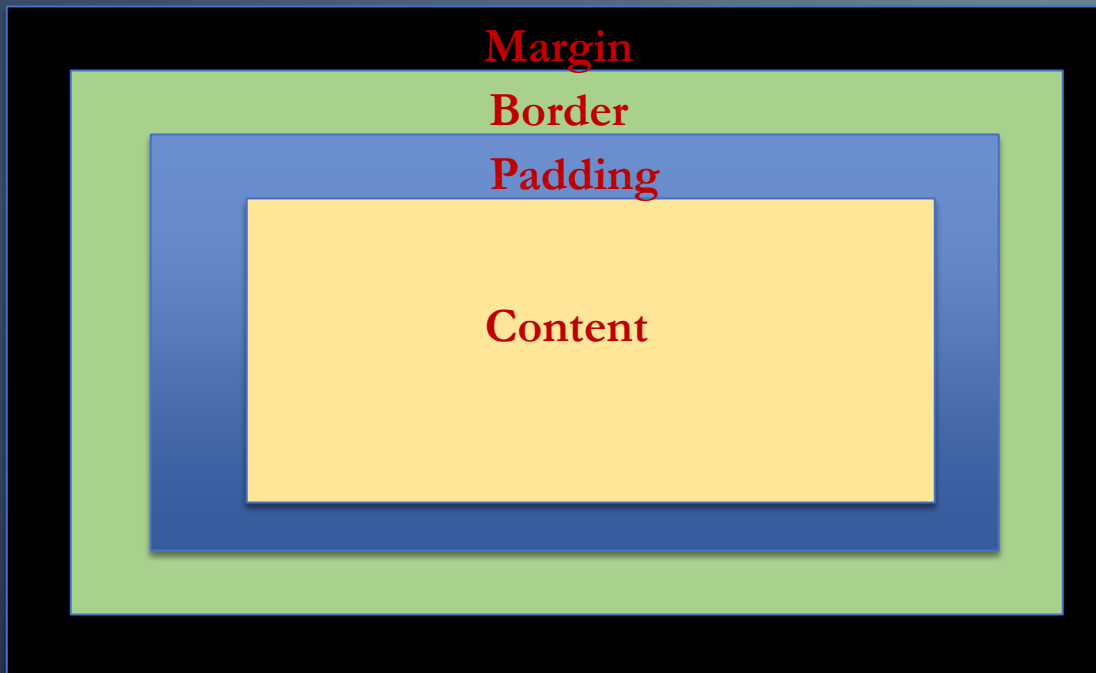
B
a
t
r
a

Box Model : Introduction

Box model is useful for designing the layout of an HTML Page.

CSS Box model describes a box that wraps around HTML elements.

Using this model, we can define the margins, borders, padding and the actual content. We can place border around elements and space elements in relation to each other.



Box Model : Illustration

You can set the height and width of an element using the **height** and **width** properties.

Formatting with CSS Properties

92

CSS Padding

You can use the CSS padding properties to define the space between the element border and the element content. It is possible to change the top, right, bottom and left padding independently using separate properties.

In shorthand; specify one, two, three, or four space-separated values:

- [value] [top, right, bottom, and left] 10px
 - [value] [value] [top and bottom] [left and right] 10px 20px
 - [value] [value] [value][top] [right and left] [bottom] 10px 20px 30px
 - [value] [value] [value] [value] [top] [right] [bottom] [left] 10px 20px 30px 40px
- **Note:** padding: 10px 20px 30px 40px; **is same as** padding-top: 10px; padding-right: 20px; padding-bottom: 30px; padding-left: 40px;

Formatting with CSS Properties

93

S
U
j
a
t
a

B
a
t
r
a

CSS Border

You can use the CSS Border properties to specify the style and color of an element's border.

- **border-style**
 - To make a border around an element.
 - Values: solid, dotted, dashed, double, groove, ridge, inset and outset.
- **border-width**
 - sets the width of the border
 - There are also properties for border-top-width, border-right-width, border-bottom-width and border-left-width.
- **border-color**
 - sets the color.

Border-width

- ▶ In shorthand; specify one, two, three, or four space-separated values:
 - ▶ [value] [top, right, bottom, and left] 10px
 - ▶ [value] [value] [top and bottom] [left and right] 10px 20px
 - ▶ [value] [value] [value] [top] [right and left] [bottom] 10px 20px 30px
 - ▶ [value] [value] [value] [value] [top] [right] [bottom] [left] 10px 20px 30px 40px
- ▶ **Note :** border-width: 10px 20px 30px 40px; **is same** as border-top-width: 10px; border-right-width: 20px; border-bottom-width: 30px; border-left-width: 40px;

Formatting with CSS Properties

95

CSS Margin

Using CSS Margin properties you can specify the space around elements.

- **One, two, three, or four space-separated values:**
 - [value] [top, right, bottom, and left] 10px
 - [value] [value] [top and bottom] [left and right] 10px 20px
 - [value] [value] [value] [top] [right and left] [bottom] 10px 20px 30px
 - [value] [value] [value] [value] [top] [right] [bottom] [left] 10px 20px 30px 40px
- **Note :** margin: 10px 20px 30px 40px; **is same** as margin-top: 10px; margin-right: 20px; margin-bottom: 30px; margin-left: 40px;, for example.

Formatting with CSS Properties

96

S
U
j
a
t
a

B
a
t
r
a

Pseudo-Class

A pseudo-class is used to define a special state of an element.

- Style an element when a user mouses over it.
- Style visited and unvisited links differently.

```
/* unvisited link */  
a:link { color: #FF0000;  
}
```

```
/* visited link */  
a:visited { color: #00FF00;  
}
```

```
/* mouse over link */  
a:hover { color: #FF00FF;  
}
```

```
/* selected link */  
a:active { color: #0000FF;  
}
```

Formatting with CSS Properties

97

S
U
j
a
t
g

Pseudo-Elements

A CSS pseudo-element is used to style specified parts of an element.

- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element

The **::first-line** pseudo-element is used to add a special style to the first line of a text. All Pseudo Elements-

<u>::after</u>	p::after	Insert content after every <p> element
<u>::before</u>		
<u>::first-letter</u>	p::first-letter	Selects the first letter of every <p> element
<u>::first-line</u>		
<u>::selection</u>	p::selection	Selects the portion of an element that is selected by a user

Formatting with CSS Properties

Media Types

The `@media` rule makes it possible to define different style rules for different media types in the same stylesheet.

Example-

```
@media screen {  
  p { font-family: verdana, sans-serif;  
        font-size: 20px;  
  } }
```

```
@media print {  
  p { font-family: georgia, serif;  
        font-size: 15px;  
        color: blue;  
  } }
```

CSS float

- ▶ The CSS float property is a positioning property.
- ▶ Used to push an element to the left or right, allowing other element to wrap around it.
- ▶ Generally used with images and layouts.

How it works

- ▶ A floated element may be moved as far to the left or the right as possible. Simply, it means that a floated element can display at extreme left or extreme right.
- ▶ The elements after the floating element will flow around it.
- ▶ The elements before the floating element will not be affected.
- ▶ If the image floated to the right, the texts flow around it, to the left and if the image floated to the left, the text flows around it, to the right.

CSS float Properties

Property	Description	Values
clear	The clear property is used to avoid elements after the floating elements which flow around it.	left, right, both, none, inherit
float	It specifies whether the box should float or not.	left, right, none, inherit

CSS Float Property Values

Value	Description
none	It specifies that the element is not floated, and will be displayed just where it occurs in the text. this is a default value.
left	It is used to float the element to the left.
right	It is used to float the element to the right.
initial	It sets the property to its initial value.
inherit	It is used to inherit this property from its parent element.

Final Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.header{margin:-8px -8px 0px;background-color:red;color:white;text-align:center;padding:10px;}

.container{width:100%}

.left{width:15%;float:left;}

.body{width:65%;float:left;background-color:pink;padding:5px;}

.right{width:15%;float:left;}

.footer{margin:-8px;clear:both;background-color:green;color:white;text-align:center;padding:10px;}
</style>
</head>
```

```
<body>
<div class="header"><h2>Sujata Training</h2></div>

<div class="container">
<div class="left">
<p>Left Page</p>
</div>
<div class="body">
<h1>Body Page</h1>
<p>Page Content goes here</p><p>Page Content goes here</p><p>Page Content goes here</p>
<p>Page Content goes here</p><p>Page Content goes here</p><p>Page Content goes here</p>
<p>Page Content goes here</p><p>Page Content goes here</p><p>Page Content goes here</p>
<p>Page Content goes here</p><p>Page Content goes here</p><p>Page Content goes here</p>
<p>Page Content goes here</p>
</div>
<div class="right">
<p>Right Page</p>
</div>
</div>

<div class="footer">
<p>Footer</p>
</div>

</body>
</html>
```



Writing for Web Audience

- ▶ Punch up headlines.
 - ▶ Web visitors scan first for headlines, so make every heading word meaningful.
- ▶ Emphasize key concepts.
 - ▶ Help readers scan for key concepts by emphasizing important information.
- ▶ Harness the power of lists.
 - ▶ Lists slow the reader down and bring attention to important information
- ▶ Create meaningful captions.
 - ▶ Web users focus on text over graphics, make sure to caption all graphics clearly.
- ▶ Simplify for understanding.
 - ▶ Reading from the screen is slower than reading from print, so make your users happy by giving them less to read.

Writing for Web Audience

- ▶ Invert the pyramid.
 - ▶ The inverted pyramid style is bottom-up. To write this way, start by stating the conclusion. Then build upon the conclusion by summarizing the most interesting and important supportive information. Next provide detail about each important point. Then close with background information.
- ▶ Write one idea per paragraph.
 - ▶ Make sure each paragraph contains one idea only, and summarize that idea in the first sentence.
- ▶ Make each page stand alone.
 - ▶ Don't expect that users will enter your Web site at the home page and work their way through the site in an organized manner. Thanks to the power of search engines and offsite links, visitors may enter your site on any page at all. Because of this, each page needs to stand alone, and your prose must not assume that they have already read any other page. Provide context to help users understand where the page fits within your Web site.

Writing for Web Audience

- ▶ Be current, accurate, and credible.
 - ▶ Capture the trust of your readers by offering information that is up-to-date and accurate .
 - ▶ A simple way to improve credibility: Skip the marketing hype. Replace it with well-written, interesting, and useful information.
 - ▶ Give credit where credit is due. If your site has multiple authors, give each one credit with a byline. Bylines are personal touches that add credibility to a Web site. And don't just quote information from a study or survey; you'll gain credibility by providing a direct link to the source.