

# HTML

## Web Authoring



eMail:-ssp@ucsc.cmb.ac.lk

**Dr. Samantha Mathara Arachchi**

B.Sc,Pg.Dip(Com.Tech.) Col.,Pg.Dip.(IM),M.Sc.(IM),PhD (MY), CTHE (Col), SEDA(UK), MCS(SL) MACM,MIEEE

- In the World Wide Web (WWW), A client(Web browser) communicates with a server(Web server) to request for hypertext documents with hypertext transfer protocol(HTTP).

## HTML History

Since the early days of the World Wide Web, there have been many versions of HTML:

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	<u>WHATWG HTML5 Living Standard</u>
2014	<u>W3C Recommendation: HTML5</u>
2016	W3C Candidate Recommendation: HTML 5.1
2017	<u>W3C Recommendation: HTML5.1 2nd Edition</u>
2017	<u>W3C Recommendation: HTML5.2</u>

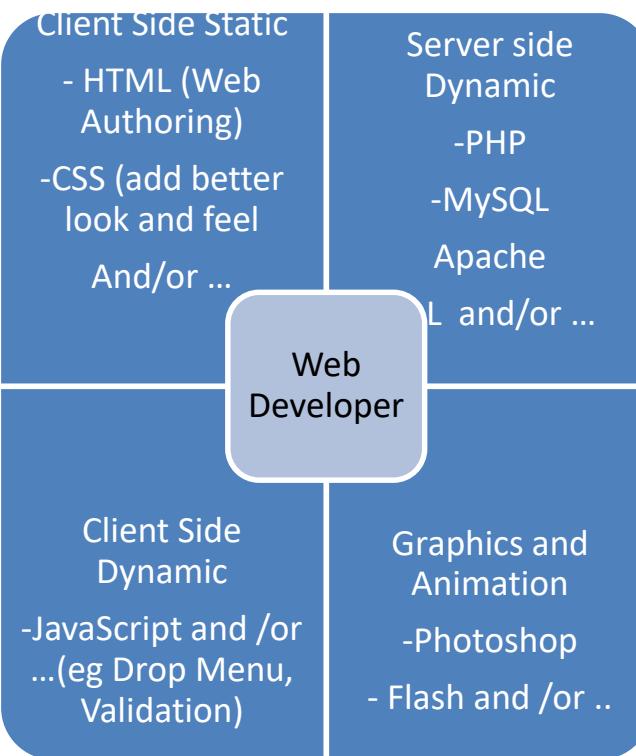
## Browsers:

- Google Chrome
- Mozilla Firefox
- Opera Web Browser
- Safari Web Browser
- Internet Explorer
- Slimjet Browser
- Netscape Browser



**Client:**

frontend and backend



HTTP Response (Web site)

HTTP Request Using URL (<http://www.yahoo.com>)



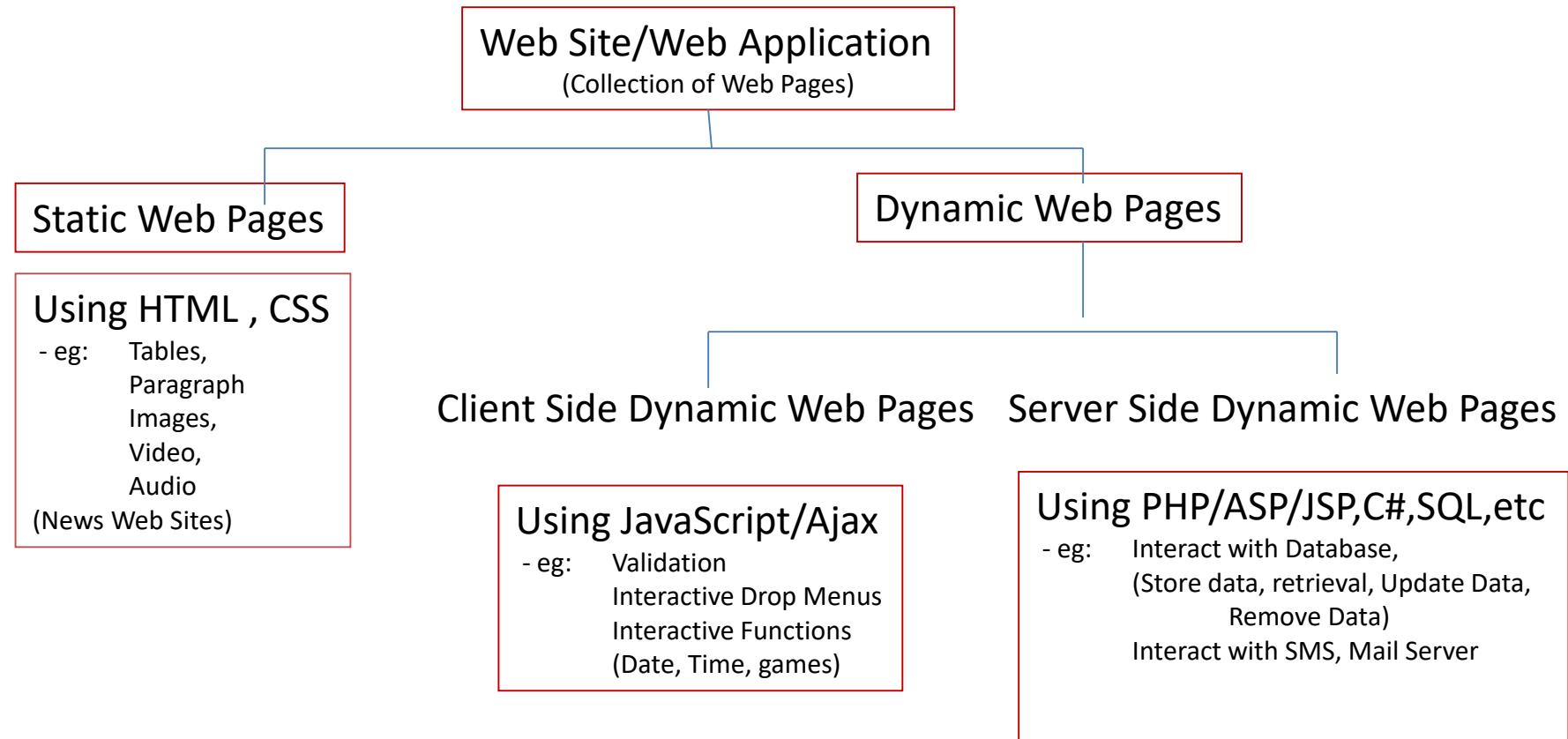
**Servers**

Uploading using FTP Mechanism



- Web Server (Apache/IIS (internet information Services)/ Tomcat (Apache group/PSW (Microsoft Personal Web Server)- upload all files (eg. **HTML, CSS, JS, Audio, Video, Images, Animations**)
- Database Server (MySQL/Oracle) –store data (**create Database, tables**)
- Video Server (Kodi – Home Theater **Software**. , PLEX – Media **Server**, Subsonic – Personal Media Streamer. Madsonic – Music Streamer. ... Emby – Open Media Solution. , Gerbera – UPnP Media **Server** . , Tvmobili – Smart TV Media **Server** . , OpenELEC – Open Embedded Linux Entertainment Center
- Mail Server (send and receive email using standard email (Zimbra, PostFix, Kolab, Mozilla Thunderbird)
- SMS server- (**SmsGate Pro, Sms Server, Sms Gateway, SMS software**)
- File and Print Servers

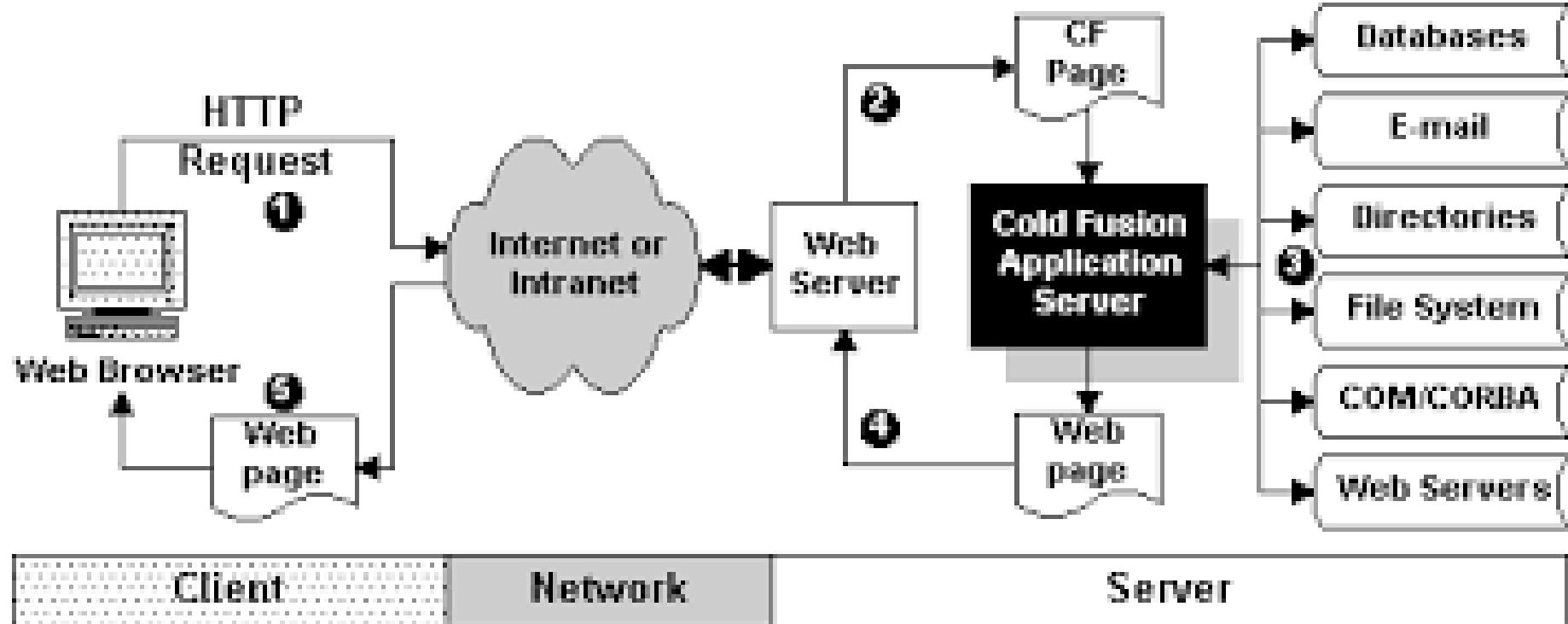
**Web Developer/Programmer**



- A **Uniform Resource Locator**, abbreviated **URL** is a specific character string that constitutes a reference to a resource.
  - In most web browsers, the URL of a web page is displayed on top inside an address bar. An example of a typical URL would be "http://en.example.org/wiki/Main\_Page".
  - Or
  - [https://en.example.org/wiki/Main\\_Page](https://en.example.org/wiki/Main_Page)
- Explanation:
- eg: **scheme://host.domain:port/path/filename**
- scheme** - defines the **type** of Internet service (most common is **http**)
- host** - defines the **domain host** (default host for http is **www**)
- domain** - defines the Internet **domain name** ([w3schools.com](http://w3schools.com))
- port** - defines the **port number** at the host (default for http is **80**)
- path** - defines a **path** at the server (If omitted: the root directory of the site)
- filename** - defines the name of a document or resource

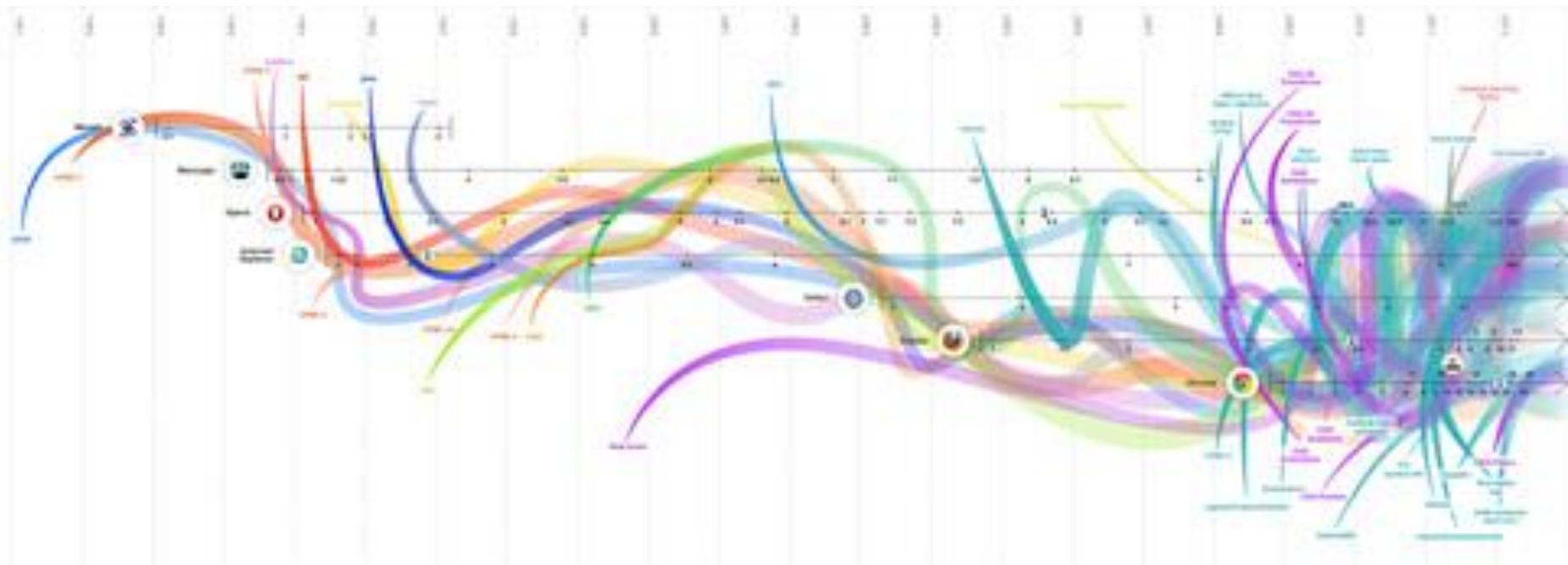


# How Web Works



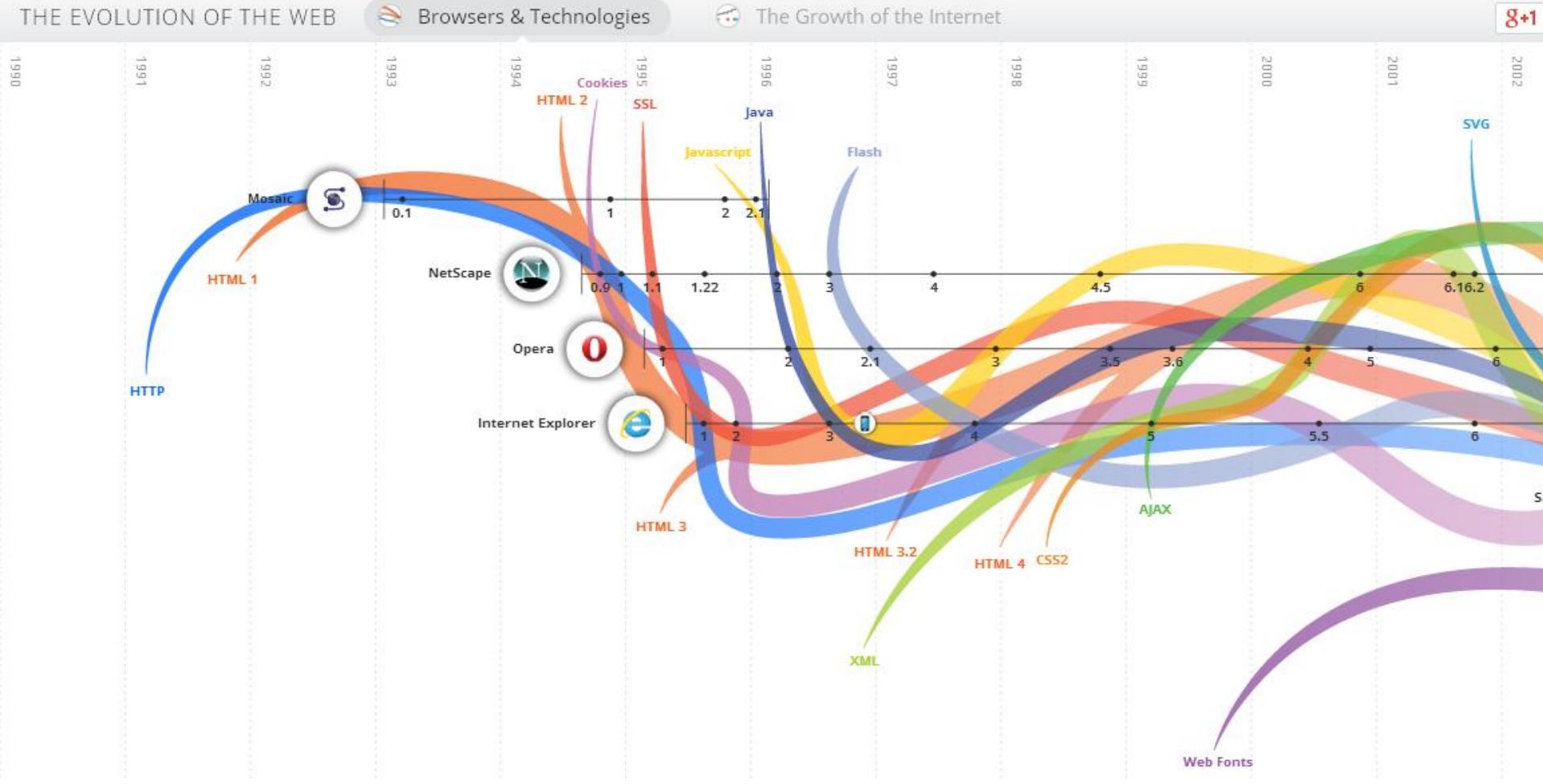
# How Web Works ...

- User types a URL in the web browser.
- Web browser contacts the domain name service DNS to find the IP Address of the server associated with the domain name in URL.
- Web browser creates a TCP socket connection to the server using the servers IP address and Port number.
- Web browser sends an HTTP request to the server asking for the file pointed by the URL.
- Server determines the file to be sent.
- Server sends response code and the document.

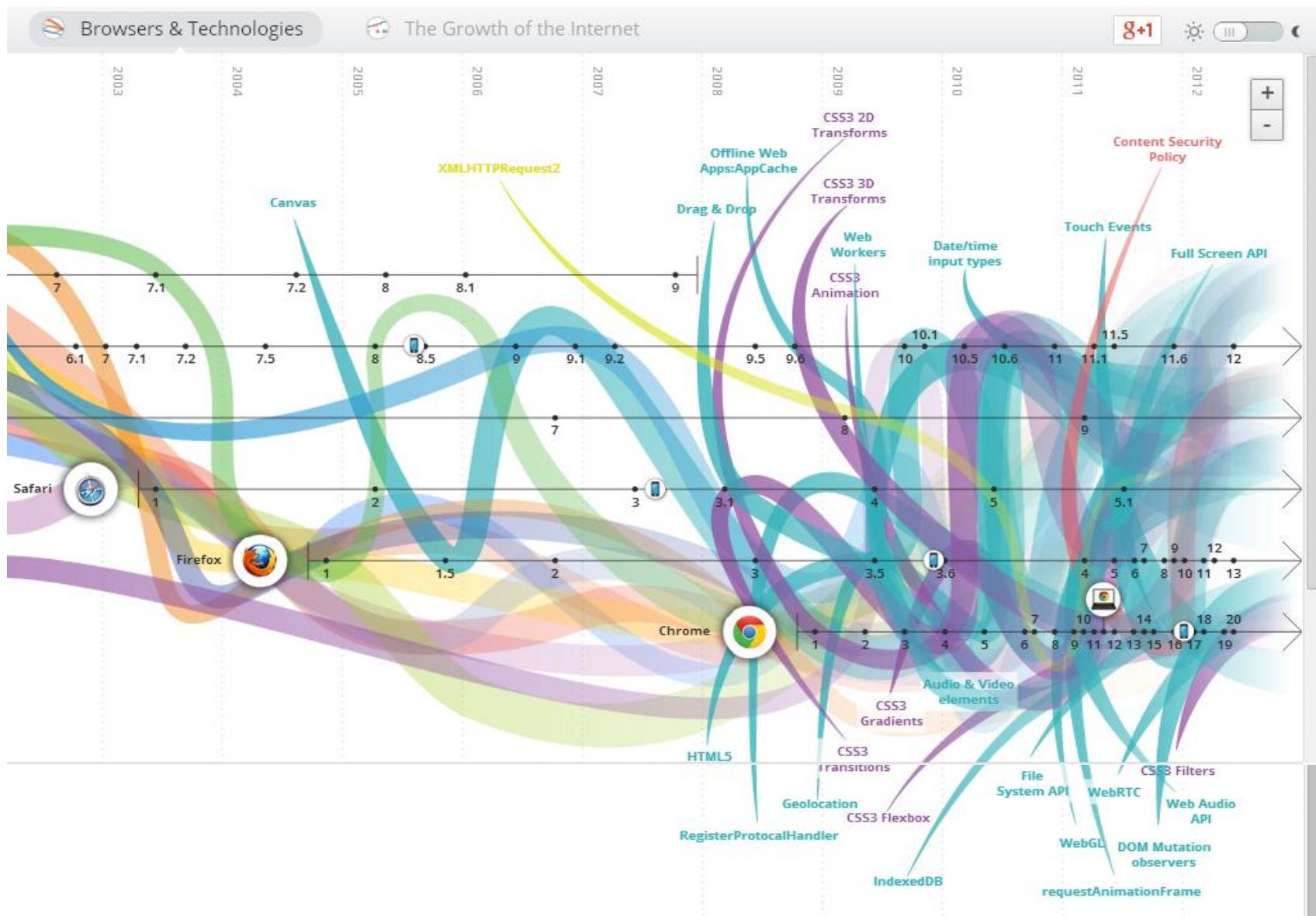


## THE EVOLUTION OF THE WEB

<http://www.evolutionoftheweb.com/#/evolution/day>



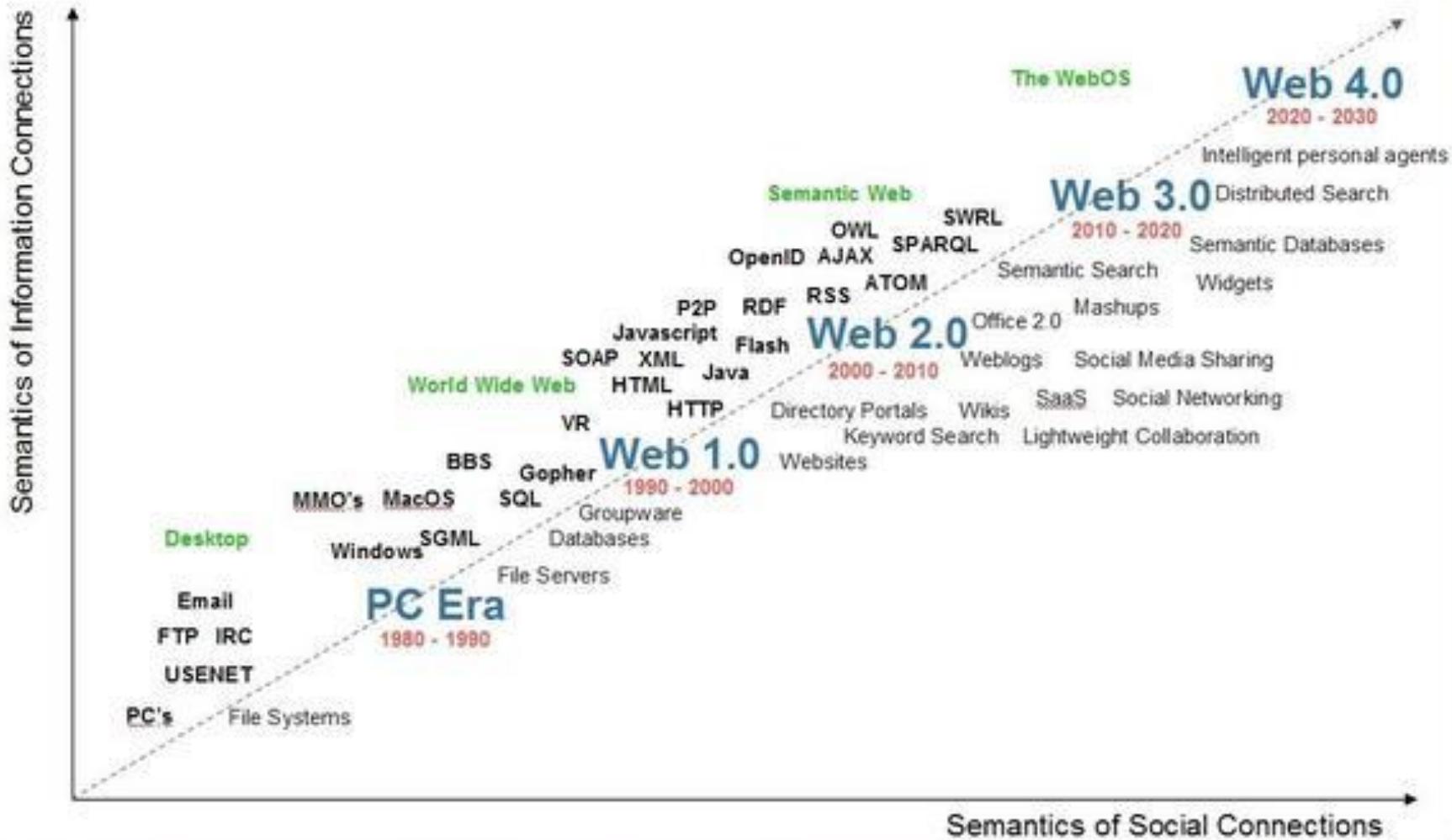
# The Evolution of the Web



# Web 1.0, 2.0 and 3.0 and 4.0

- Web 1.0 (**The shopping carts & static web**)
  - It is the “readable” phrase of the World Wide Web with flat data. In Web 1.0, there is only limited interaction between sites and web users.
  - Web 1.0 is simply an information portal where users passively receive information without being given the opportunity to post reviews, comments, and feedback.
  - The best examples of this 1.0 web era are millions of static websites which mushroomed during the dot-com boom (which eventually has led to the [dotcom bubble](#)). There was no active communication or information flow from consumer (of the information) to producer (of the information).

# Web 1.0, 2.0 and 3.0 and 4.0



Source: Radar Networks & Nova Spivack, 2007 – [www.radarnetworks.com](http://www.radarnetworks.com)

# Web 1.0, 2.0 and 3.0 and 4.0 ...

- **Web 2.0 (The writing and participating web)**
  - It is the “writable” phrase of the World Wide Web with interactive data. Unlike Web 1.0, Web 2.0 facilitates interaction between web users and sites, so it allows users to interact more freely with each other.
  - Web 2.0 encourages participation, collaboration, and information sharing. Examples of Web 2.0 applications are Youtube, Wiki, Flickr, Facebook, and so on.

# Web 1.0, 2.0 and 3.0 and 4.0 ...

- Web 3.0 (**The semantic executing web**)
  - It is the “executable” phrase of Word Wide Web with dynamic applications, interactive services, and “machine-to-machine” interaction.
  - Web 3.0 is a semantic web which refers to the future.
  - *Semantic markup* refers to the communication gap between human web users and computerized applications.
  - In Web 3.0, computers can interpret information like humans and intelligently generate and distribute useful content tailored to the needs of users.
  - One example of Web 3.0 is [Tivo](#), a digital video recorder. Its recording program can search the web and read what it finds to you based on your preferences.
- The following attributes are going to be a part of Web 3.0: Contextual Search
  - Tailor made Search
  - Personalized Search
  - Evolution of 3D Web
  - Deductive Reasoning

# Web 1.0, 2.0 and 3.0 and 4.0 ...

- **Web 4.0 (Mobile Web)**
  - The next step is not really a new version, but is a alternate version of what we already have. Web needed to adapt to it's mobile surroundings. Web 4.0 connects all devices in the real and virtual world in real-time.
- **Web 5.0- Open, Linked and Intelligent Web = Emotional Web**

# Web 1.0, 2.0 and 3.0 and 4.0 ...

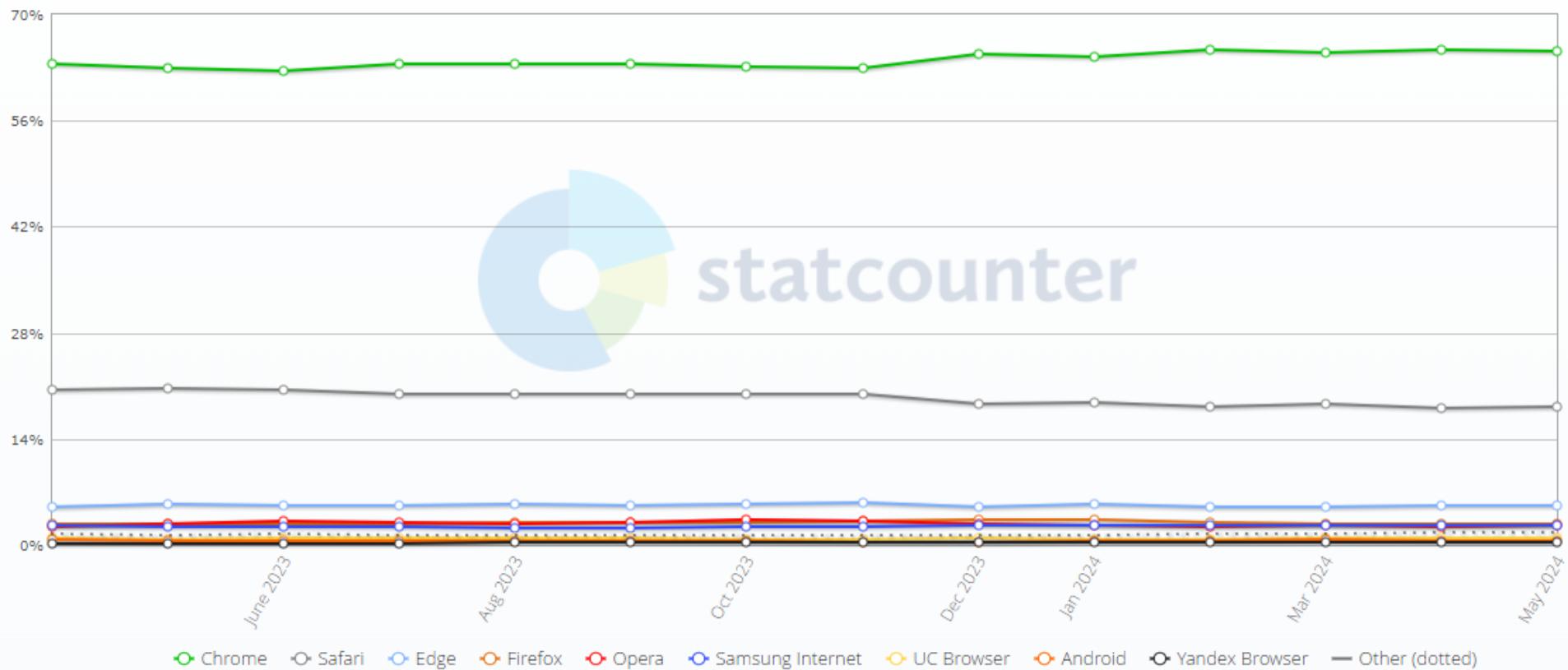
## Web 1.0 / 2.0 / 3.0 Summary

Crawl	Walk	Run
Web 1.0	Web 2.0	Web 3.0
Mostly Read-Only	Wildly Read-Write	Portable & Personal
Company Focus	Community Focus	Individual Focus
Home Pages	Blogs / Wikis	Lifestreams / Waves
Owning Content	Sharing Content	Consolidating Content
Web Forms	Web Applications	Smart Applications
Directories	Tagging	User Behavior
Page Views	Cost Per Click	User Engagement
Banner Advertising	Interactive Advertising	Behavioral Advertising
Britannica Online	Wikipedia	The Semantic Web
HTML / Portals	XML / RSS	RDF / RDFS / OWL

# Browser Market share

Browser Market Share Worldwide

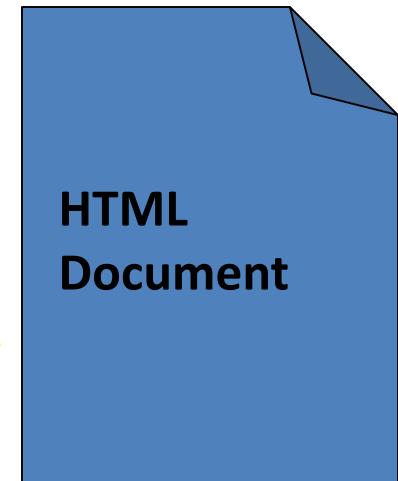
Apr 2023 - May 2024



- HTML- HyperText Markup Language
- HTML is not a programming language, it is a markup language
- A markup language is a set of markup tags
- HTML uses markup tags to describe web pages
- Use of HTML
  - Display text data
  - Call up other Web Pages (Link between Pages)
  - Display multimedia data

- Creating directly using a text editor;
  - Notepad
  - Wordpad
- Use an HTML Editor
  - FrontPage
  - HomePage
  - Dreamweaver

- Grammar
  - Tag
    - A tag is a code used in the HTML document
    - A tag is single-byte UPPERCASE/lowercase letters enclosed by “<” and “>” ( less than (<) and greater than (>))
    - <HTML> tag at the beginning of the document
    - </HTML> tag at the end of the document



**Header**



**Body**

- Header is the starting section of the HTML document
- Description of the name of the web page is enclosed between the **<HEAD>** tag and the **</HEAD>** tag
- Main contents of the HTML document
- Tags related to the display of the web page are contained between the **<BODY>** tag and the **</BODY>** tag

- <HTML> - </HTML>
  - These tags define the start and end of the HTML Document
  - All the other tags are specified between them
- <HEAD> - </HEAD>
  - These tags define the attribute of the HTML document
  - They form the header section of the HTML document
- <TITLE> - </TITLE>
  - These tags define the title of the Web page
  - They are specified between <HEAD> - </HEAD>.
  - The title defined by these tags is shown on the title bar of the browser.
- <BODY> - </BODY>
  - These tags defined the body of the HTML document

- <head>
  - Title of web page
  - Embed Styles
  - Link to the external CSS file
  - Write JavaScript Code
  - Link to external JavaScript File
  - Write PHP code
  - To Include Meta tag <meta>
  - Introduce based target <base>
  - To Include frame tags <frameset> <frame>
- </head>

- <body>
  - Author content
    - Paragraphs <p> </p>
    - Table <table> </table>
  - Embed Pictures <img />
  - Embed Audio files <embed> <audio>
  - Embed Video Files <embed> <video>
  - Create Hyperlinks <a> </a>
- </body>

Head Section

```
<HTML>
<HEAD>
```

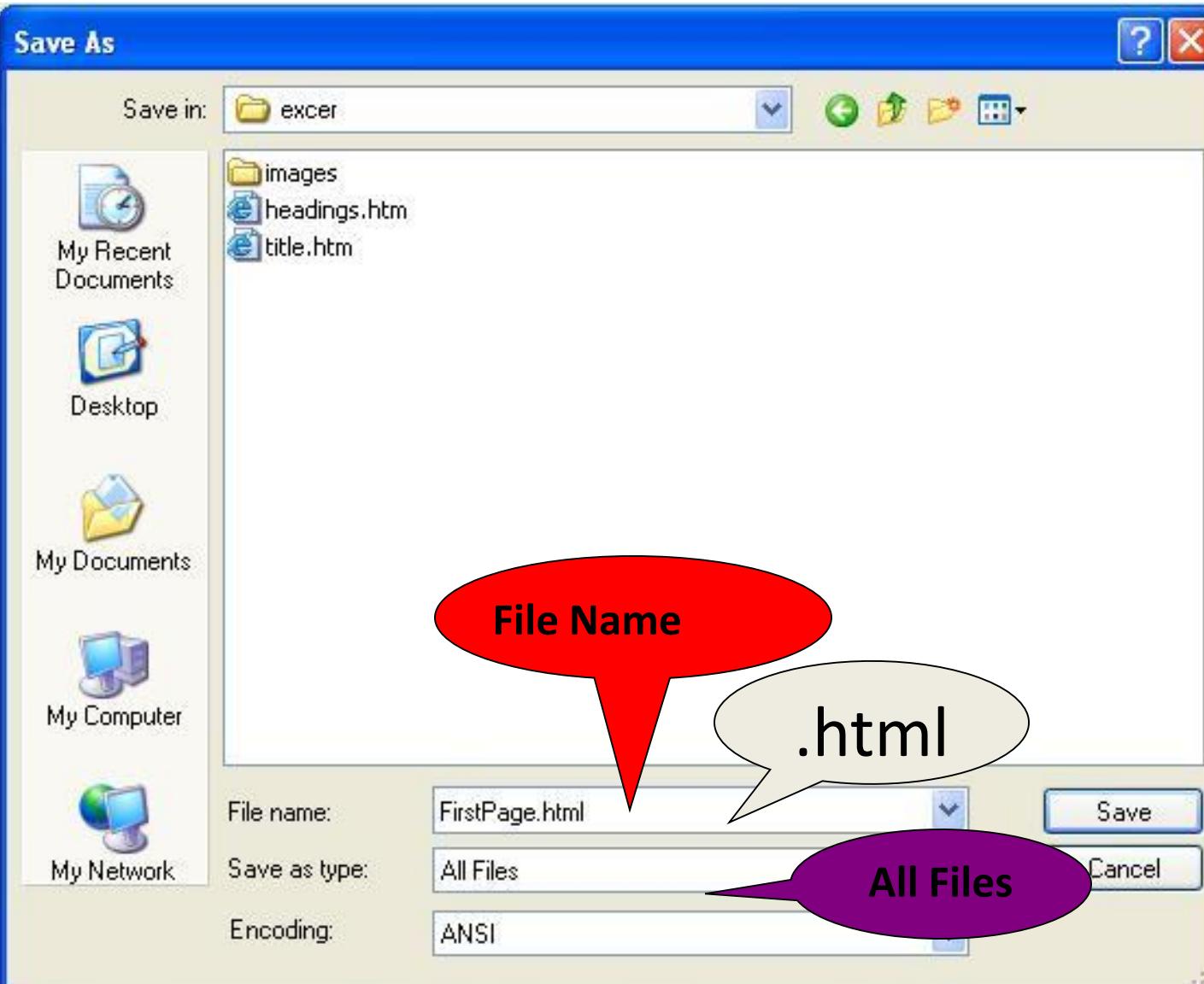
Body Section

```
<TITLE>My First Web Page</TITLE>
</HEAD>
<BODY>

<P> This is my first web page. Let's enjoy with HTML</P>

</BODY>
</HTML>
```

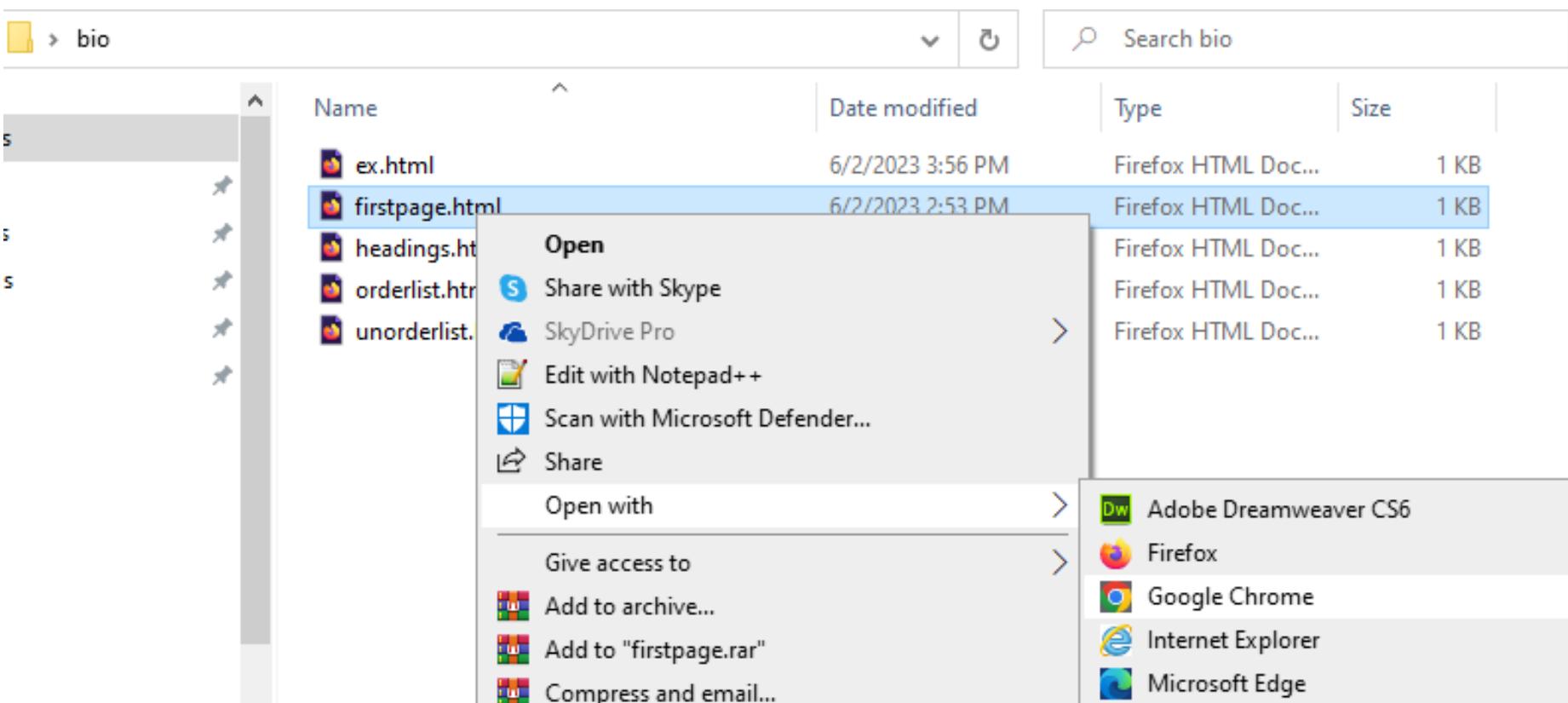




Possible ways of naming:

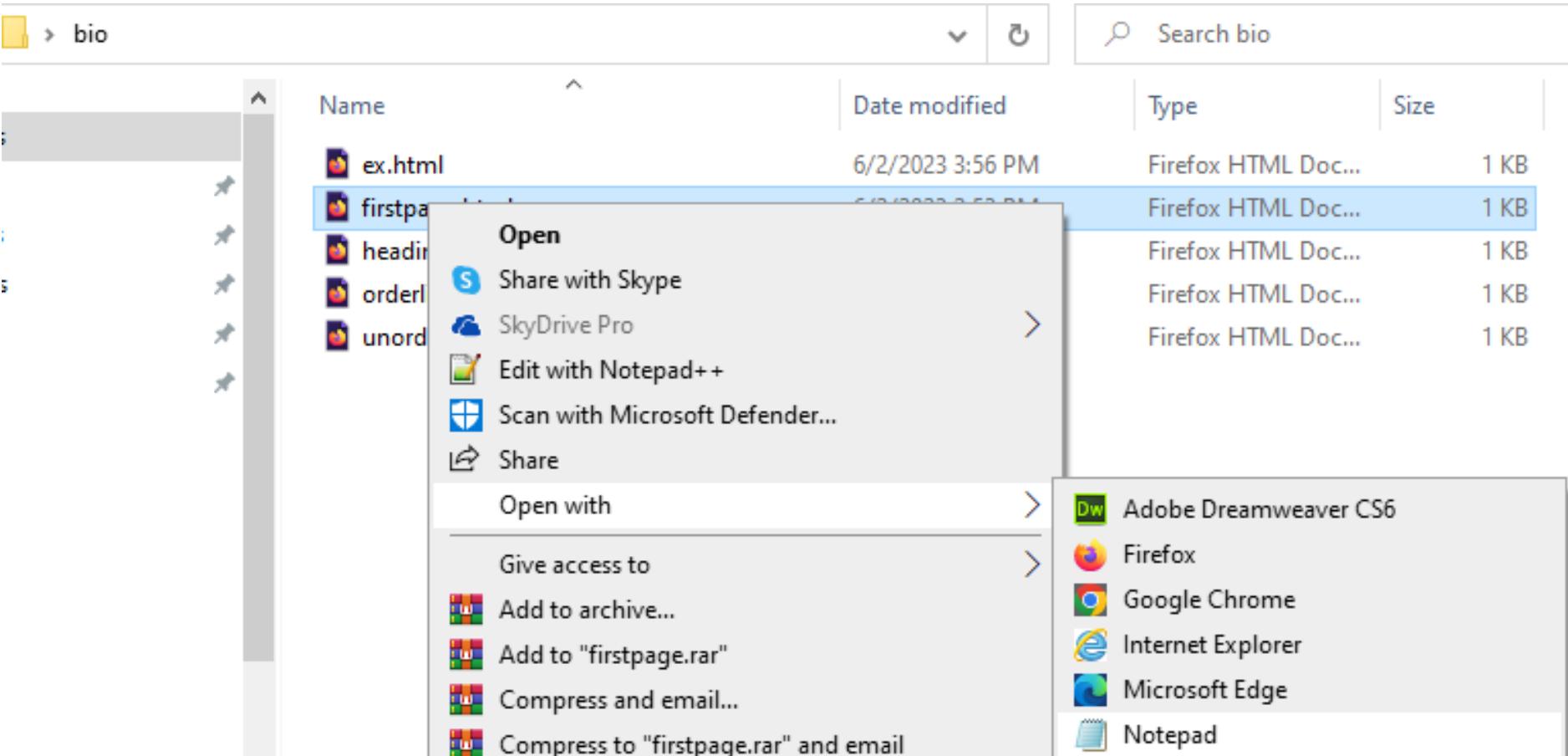
1. Firstpage.html ✓
2. FirstPage.html ✓
3. first-page.html ✓
4. First\_page.html ✓

first page.html X

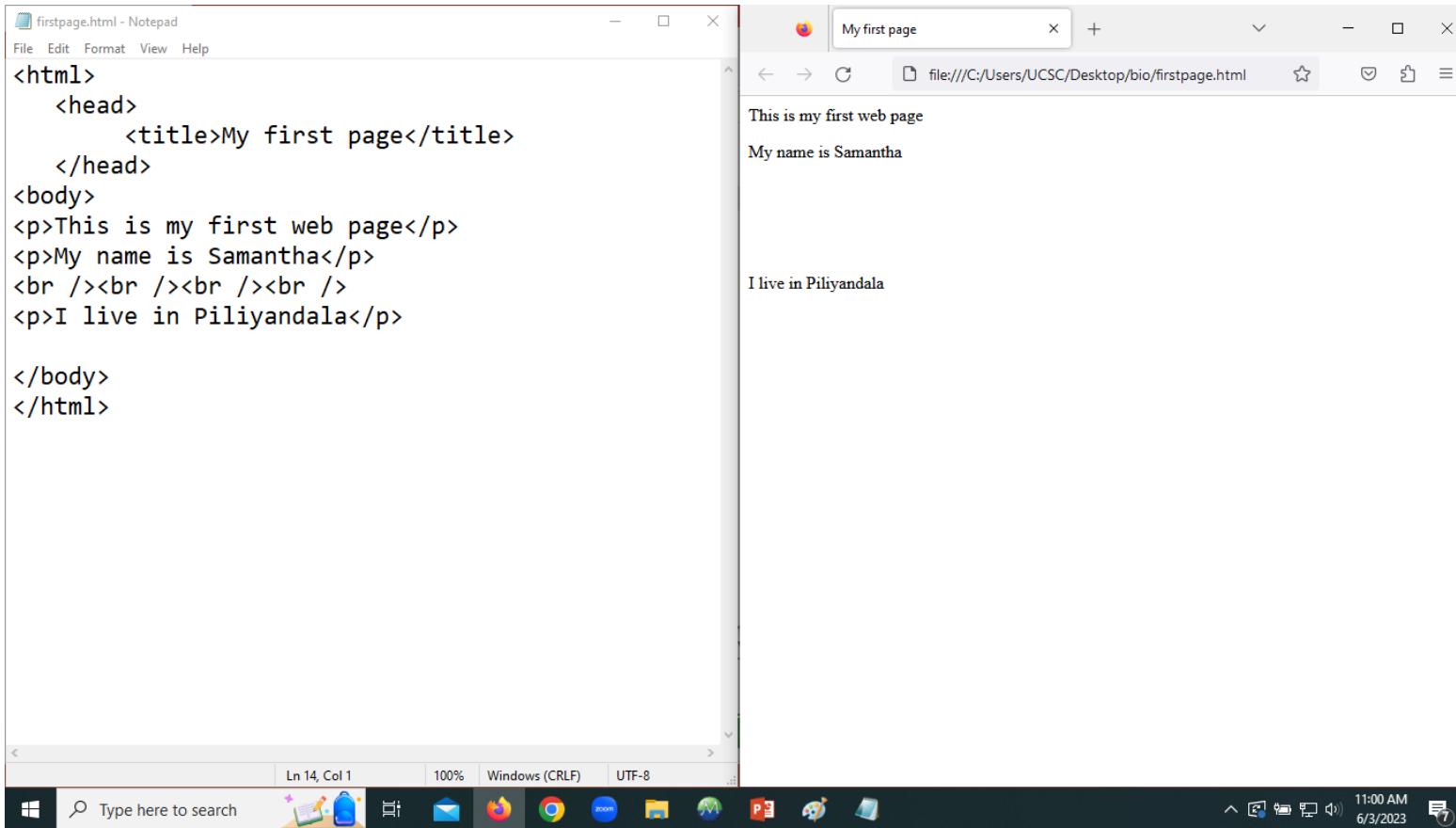


**Steps: Open your folder>> select your web file>> right click>> Open with >> select a browser**

# How to re-open the notepad



**Steps: Open your folder>> select your web file>> right click>> Open with >> select notepad**



**Steps: \*after opening both files (web page and notepad) >>click on notepad>>press window key and Left arrow key simultaneously to position on the left side.**

**\* Then select the browser page >> press window key and right arrow key simultaneously to position on the right side**

- Grammatically , they belong to any of the patterns described follows

- 1 • Basic Tag **<TAG> .. </TAG>**

Opening /Start tag

character String

Closing/End tag

**<TITLE> My First Page (Content) </TITLE>**

For this type of tag an opening tag is used before a character string and a closing tag is used after it.

2

- Single Tag <TAG /> or <TAG>

For this type of tag an opening tag is used either before or after a character string.

Eg. -<br /> Line Break Tag

-<hr /> horizontal rule

-<img /> Insert an Image

- ✓ HTML elements with no content are called **empty elements**.
- ✓ <br> is an empty element without a closing tag (the <br> tag defines a line break).
- ✓ Empty elements can be "closed" in the opening tag like this: <br />.
- ✓ HTML5 does not require empty elements to be closed. But if you want stricter validation, or you need to make your document readable by XML parsers, you should close all HTML elements.

3

- Attribute Tag

**<TAG attribute attribute ..> .. </TAG>**

```
<font color="#aa1122" size="16px" type="arial">attribute tag</font>
```

- HTML elements can have attributes
- Attributes provide additional information about an element
- Attributes are always specified in the start tag
- Attributes come in name/value pairs like: name="value"
- For this type of tag is used to specify attributes within a tag for detailed specification
- Depending on the type of tag, the attributes that can be used are different.
- Multiple attributes can be specified in random order
- Attributes are separated by a single-byte space.

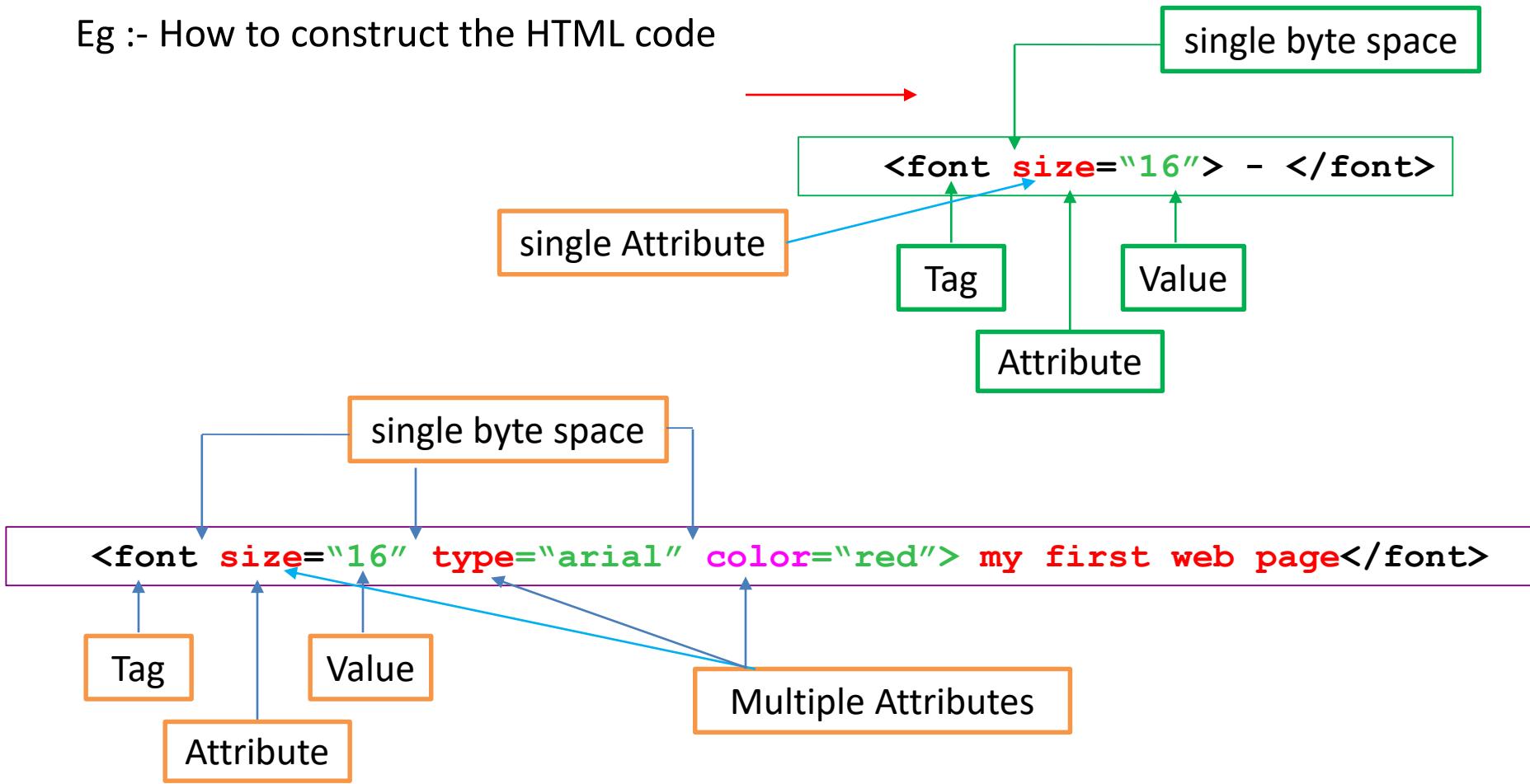
Eg <BODY BGCOLOR="#FFFFFF" BACKGROUND="image.gif"> ....</BODY>

## Text Formatting

- **<FONT FACE=“SPECIFY A TYPE”> - </FONT>**
  - These tag is used to specify the font Type
    - Eg <FONT FACE= “Arial”> My First Web Page</Font>
- **<FONT SIZE=“SPECIFY A SIZE”> - </FONT>**
  - These tag is used to specify the font size
    - » Eg <FONT SIZE = “16”> My First Web Page</Font>
- **<FONT COLOR=“SPECIFY A COLOR”> - </FONT>**
  - These tag is used to specify the color
    - Eg <FONT color = “red”> My First Web Page</Font>

- <FONT SIZE="SPECIFY A SIZE" TYPE="SPECIFY A TYPE" COLOR="SPECIFY A COLOR"> - </FONT>
  - These attributes are used to specify the font size, color and type

Eg :- How to construct the HTML code



# Single or Double Quotes?

- Using quotes are the most common. Omitting quotes can produce errors.
- The HTML5 standard does not require quotes around attribute values.
- Double style quotes are the most common in HTML, but single style can also be used.
- In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes:

```
<p title='Nimal "Sampath" Perera'>
```

```
<p title="Nimal 'Sampath' Perera">
```

# HTML Quotation and Citation Elements

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

# HTML Quotation and Citation Elements

```
<html>
<body>

<p>Browsers usually insert quotation marks around the q element.</p>

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

</body>
</html>
```

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

# blockquote elements

```
<html>
<body>

<p>Browsers usually indent blockquote elements.</p>

<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature. The world's leading
conservation organization, WWF works in 100 countries and is supported by
1.2 million members in the United States and close to 5 million globally.
</blockquote>

</body>
</html>
```

Browsers usually indent blockquote elements.

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

# Marking up abbreviations

```
<html>
<body>

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

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

</body>
</html>
```

The WHO was founded in 1948.

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

# bi-directional override (bdo) - Tag

```
<html>
<body>

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

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

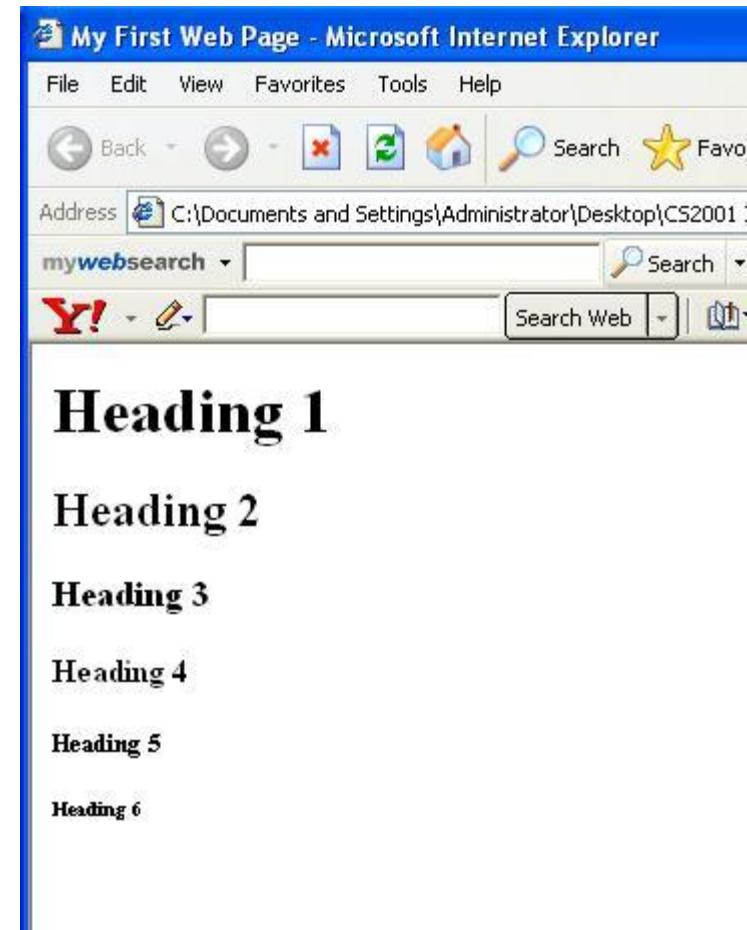
</body>
</html>
```

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

tfel ot thgir morf nettirw eb lliw enil sihT

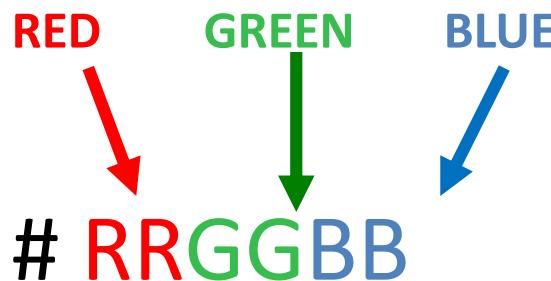
- These tags are used to display the headings
- The numbers indicate the **levels of the headings** (Relative size) from 1-6)

```
<body>
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
</body>
```



- <BR>
- This tag is used to define line breaks <br/>
  
- <!-- -->
- This tag is used to comment out line
- Eg: <!-- table 1.1-->

- Six digit hexadecimal numbers



# ff0000 - RED

# 00ff00 - GREEN

# 000000 - BLACK

# ffffff - WHITE

• A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color. All values must be between 00 and FF.

• For example, the #0000FF value is rendered as blue, because the blue component is set to its highest value (FF) and the others are set to the lowest value (00).

Color	Color HEX	Color RGB	Color Name
Red	#FF0000	rgb(255,0,0)	Red
Green	#00FF00	rgb(0,255,0)	Green
Blue	#0000FF	rgb(0,0,255)	Blue

Variation 0~9 or a~f → 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f

This 216 cross platform web safe color palette was originally created to ensure that all computers would display all colors correctly when running a 256 color palette.

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

<font color="red"> Colombo </font>

<font color="#ac3d5f"> Colombo </font>

## ■ **<I> - </I>**

- These tag is used to display italic text

- Eg **<I>My First Web Page</I>**

- **<B> - </B>**

- These tag is used to display bold text

- » Eg **<B>My First Web Page</B>**

## ■ **<U> - </U>**

- These tag is used to display underline text

- Eg **<U> My First Web Page</U>**

## ■ **<SUP> - </SUP>**

- These tag is used to display Superscript text
  - Eg 2<SUP>nd</SUP>

- **<SUB> - </SUB>**

- These tag is used to display subscript text
    - » Eg H<SUB>2</SUB>

## ■ **<STRIKE> - </STRIKE>**

- These tag is used to display strike out text
  - Eg <STRIKE> My First Web Page</STRIKE>



My First Web Page

# HTML <mark> Element

```
<html>
<body>

<h2>HTML <mark>Marked</mark>
Formatting</h2>

</body>
</html>
```

HTML **Marked** Formatting

# The <marquee> Tag Attributes

```
<body>
  <marquee>This is basic example of marquee</marquee>

  <marquee width = "50%">This example will take only 50% width</marquee>

  <marquee direction = "right">This text will scroll from left to right</marquee>

  <marquee direction = "up">This text will scroll from bottom to up</marquee>

</body>
</html>
```

## ■ <P> - </P>

- These tags define the paragraph
- They force a line break and insert a blank line before and after the paragraph.
- <p> element has a **title** attribute.

<BODY>

<P> This is my first paragraph </p>

</BODY>

<p title="Sri Lanka">

Sri Lanka (formerly Ceylon), a tiny island nation south of India in the Indian Ocean, is a rugged land of rainforest, diverse wildlife and endless beaches.

</p>

```
blockquote {  
padding : 10px;  
border : 1px solid #6395a5;  
background-color : #bfd5e2;  
text-align : justify;  
font-style : italic;  
}
```

<blockquote><p>You are free to use this template in any way you like. All I ask for is that you leave the web site design credit links in the footer area of this template intact.</p></blockquote>

<blockquote>:-indicates that the enclosed text is an extended quotation. Usually, this is rendered visually by **indentation**

- <HR>
- This tag is used to display a border (horizontal rule)
  - Attribute of the <HR> tag can specify the thickness. Length, left and right alignment.
    - <HR WIDTH="70%" ALIGN="LEFT">

## ■ <OL> - </OL>

- These tags are used to display the **ordered list**
- They are used in combination with <LI> tag

## ■ <UL> - </UL>

- These tags are used to display the **unordered list**
- They are used in combination with <LI> tag

## ■ <LI>

- This tag is used in the scope of the <OL> tags or the <UL> tags described above for the display of a **list item**.
- One <LI> tag is used for each list item
- <!--nested tag-->

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

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

```
<ol type="a" start="8">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

## ■ **<DL> - </DL>**

- These tags are used to display the **Description Lists**
- They are used in combination with the **<DT>** and **<DD>**tags described below

## ■ **<DT> - </DT>** (element to define the description term)

- These tags are used in the scope of the **<DL>** tags described above for the display of the definition term. **</DT>** Can be omitted

## ■ **<DD> - </DD>** (element to define the description data)

- These tags are used in the scope of the **<DL>** tags described above for the display of definition
- The definition is displayed at an indent position from the definition term. **</DD>** can be omitted

The **<dl>** tag defines a description list.

The **<dl>** tag is used in conjunction with **<dt>** (defines terms/names) and **<dd>** (describes each term/name).

```
<DL>
```

```
  <DT>HTML</DT>
```

```
  <DD>HyperText Markup Language</DD>
```

```
  <DT>OS</DT>
```

```
  <DD>Operating System</DD>
```

```
  <DT>CSS</DT>
```

```
  <DD>Cascading Style Sheets </DD>
```

```
</DL>
```

## ■ **<TABLE> - </TABLE>**

- These tags define the table
- They are used in combination with the **<TR>** and **<TH>**
- An attribute of the **<TABLE>**
  - **<TABLE BORDER="2">**

## ■ **<TR> - </TR>**

- These tags define the rows of the table.

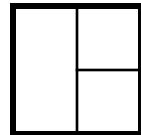
## ■ **<TD> - </TD>**

- These tags define the data (cell) of the rows.
- Adjust the Column width: **<td width=100> - </td>**

## ■ **<TH> - </TH>**

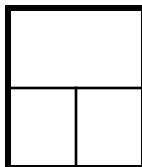
- These tags define the Item names of the columns of the table.

## ■ “ROWSPAN”



- The “ROWSPAN” attribute that specifies a cell spanning across multiple rows
- `<TD ROWSPAN="number of row"> - </TD>`

## ■ “COLSPAN”



- The “COLSPAN” attribute specifies a cell across multiple columns
- `<TD COLSPAN="number of column"> - </TD>`

## ■ <CAPTION>-</CAPTION>

- These tags define the caption of the table
- They are used between the `<TABLE>` tag and the first `<TR>` tag

# Creating Advanced TABLES using HTML 5

- The HEAD element `<thead>--</thead>`
  - Contains rows, headings and cells that make up the head of the table
- The TFOOT element `<tfoot>--</tfoot>`
  - Creates a footer of the table that appears at the end of the table
- The TBODY element `<tbody>--</tbody>`
  - Contains body of the table, that is the table data

# Creating Advanced TABLES using HTML 5

No	Name	Address	Age
01	Nimal	Colombo	20
02	Sumudu	Kandy	22
03	Gayani	Galle	21
Student's Personal Details			

]

Heading Section <thead> - </thead>

]

Body Section <tbody> - </tbody>

]

Footer Section <tfoot> - </tfoot>

# Creating Advanced TABLES using HTML 5 ...

```
<html>
<head> <title>TbodyTheadTfoot</title>
</head>
<body>
<table border=1>
<thead>
  <tr>
    <th>No</th>
    <th>Name</th>
    <th>Address</th>
  </tr>
</thead>
<tfoot>
  <tr>
    <td colspan=3>Information about data</td>
  </tr>
</tfoot>
<tbody>
  <tr>
    <td>1</td>
    <td>Saman</td>
    <td>Colombo 1</td>
  </tr>
```

```
<tr>
  <td>2</td>
  <td>Gayan</td>
  <td>Kandy</td>
</tr>
</tbody>
</table>
```

No	Name	Address
1	Saman	Colombo 1
2	Gayan	Kandy
Information about data		

# Creating Blank cell

```
<html>
<head> <title>TbodyTheadTfoot</title>
</head>
<body>
<table border=1>
<thead>
  <tr>
    <th>No</th>
    <th>Name</th>
    <th>Address</th>
  </tr>
</thead>
<tfoot>
  <tr>
    <td colspan=3>Inforation about data</td>
  </tr>
</tfoot>
<tbody>
  <tr>
    <td>1</td>
    <td></td>
    <td>Colombo 1</td>
  </tr>
```

```
<tr>
  <td>2</td>
  <td>Gayan</td>
  <td>Kandy</td>
</tr>
</tbody>
</table>
```

No	Name	Address
1		Colombo 1
2	Gayan	Kandy
Information about data		

- **<IMG SRC=“image file name”>**

- This tag is used to display an image

- **“ALT”**

- The “ALT” attribute that specifies the character strings to be displayed on a text based browser in place of the image
    - <IMG SRC=“image name” ALT=“character string”>

- **WIDTH and HEIGHT attributes**

- The “WIDTH” attribute and the “HEIGHT” attribute that specifies the display size of the image
    - <IMG SRC=“image name” WIDTH=“width of the image” HEIGHT=“height of the image”>

## ■ ALIGN

- The “ALIGN” attribute that specifies the alignment of the text to be displayed next to the image.
  - <IMG SRC=“image name” ALIGN=“position of the character string”>
  - (TOP, MIDDLE & BOTTOM)

```
<img src=abc.gif width=100 height=150 alt="Map of Sri Lanka">
```

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

Example

```

```

# <figure> and </figure>

<figure>



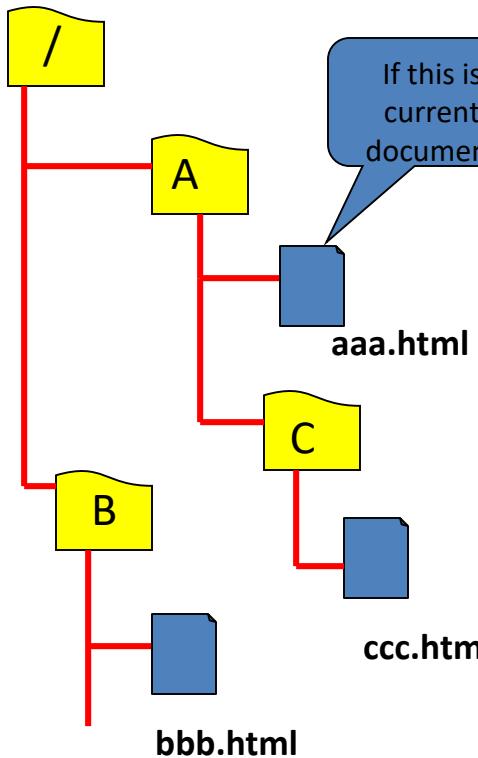
<figcaption>Fig.1 - Trulli, Puglia, Italy.</figcaption>

</figure>



Fig.1 - Trulli, Puglia, Italy.

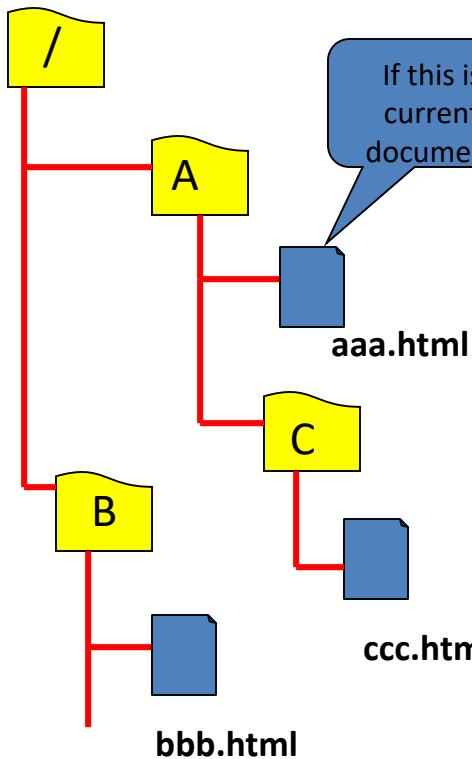
Go to Settings to activate Window



(I)

**■ Absolute Path:****<A HREF=" /A/C/ccc.html">****■ Absolute Path for different directory:****<A HREF=" /B/bbb.html">**

- The absolute path shows exactly where the file is on the computer.
- In HTML, you start every absolute pathname with a slash(/)
- Then you type the directory names on the computer starting with the topmost directory in the folder hierarchy
- Even the absolute pathnames of files located on different hard disks begin with a slash.



(II)

**■ Relative Path:**

```
<A HREF=" C/ccc.html">
```

or

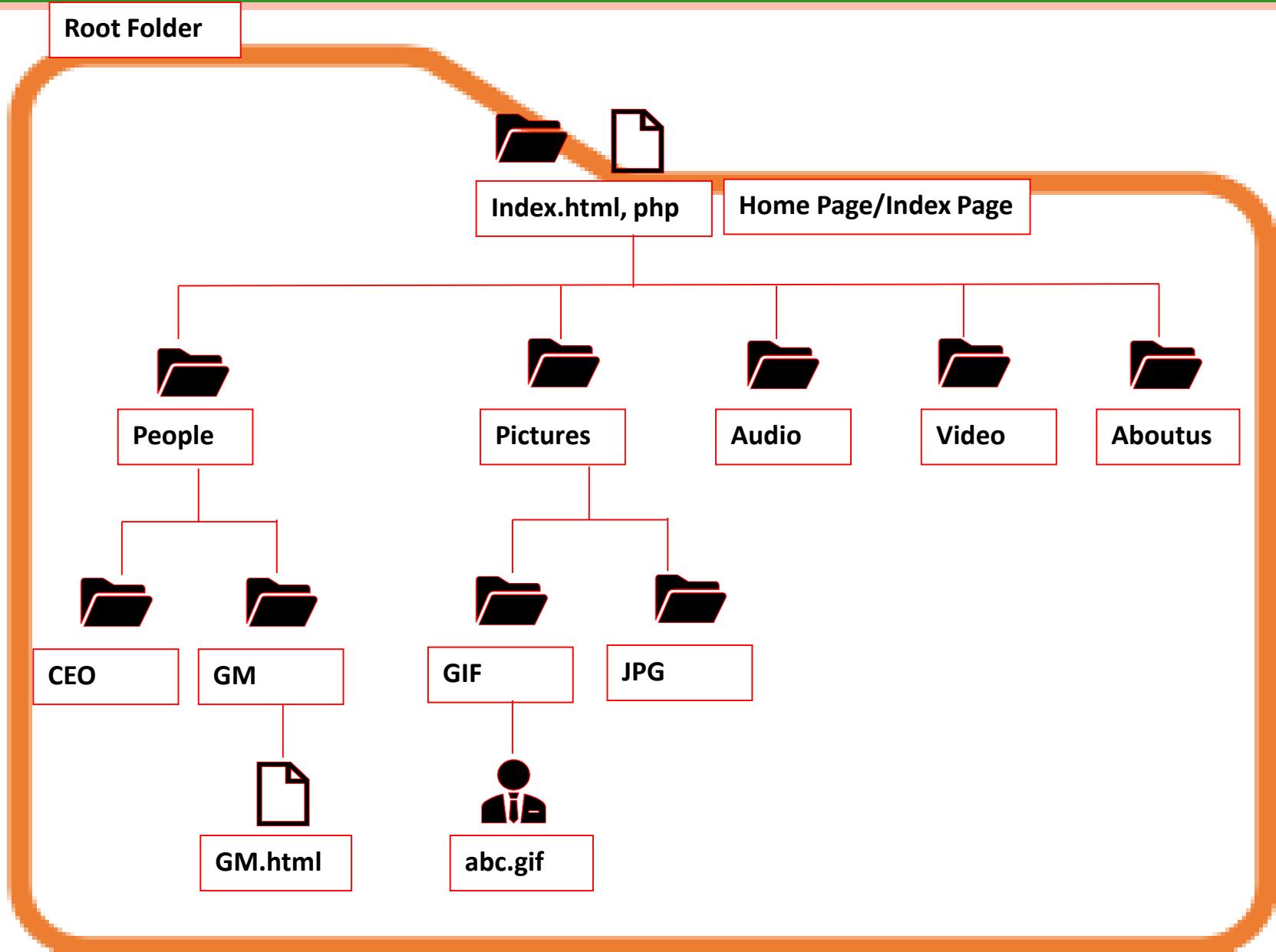
```
<A HREF=" ./C/ccc.html">
```

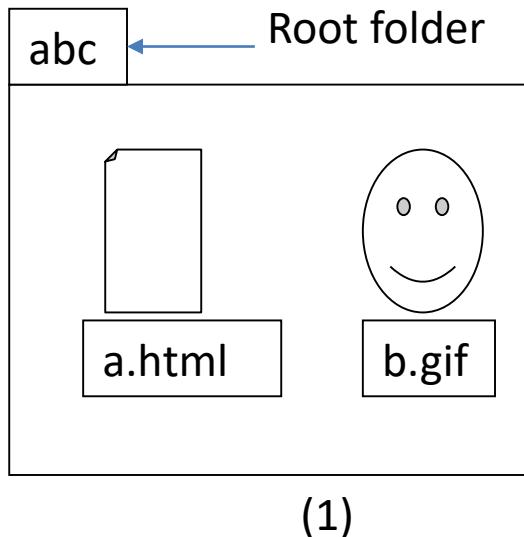
**■ Relative Path from various directory:**

```
<A HREF=" ../B/bbb.html">
```

- To reference a file in a folder above the current folder in the folder hierarchy (One level up)
- To reference a file in a folder below the current folder in the folder hierarchy then just specify the name of the subfolder.

# Folder Structure

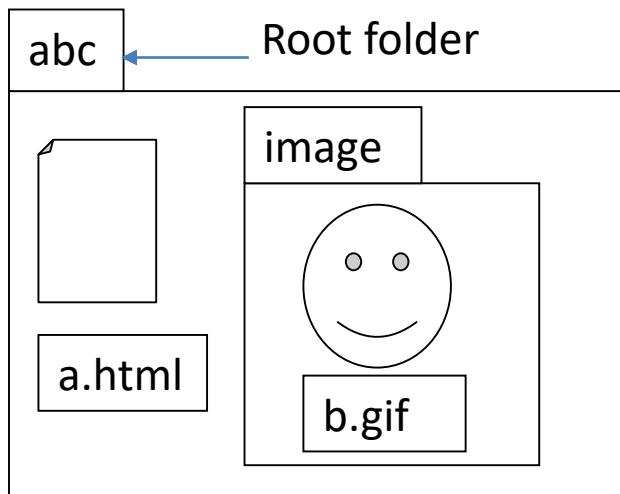




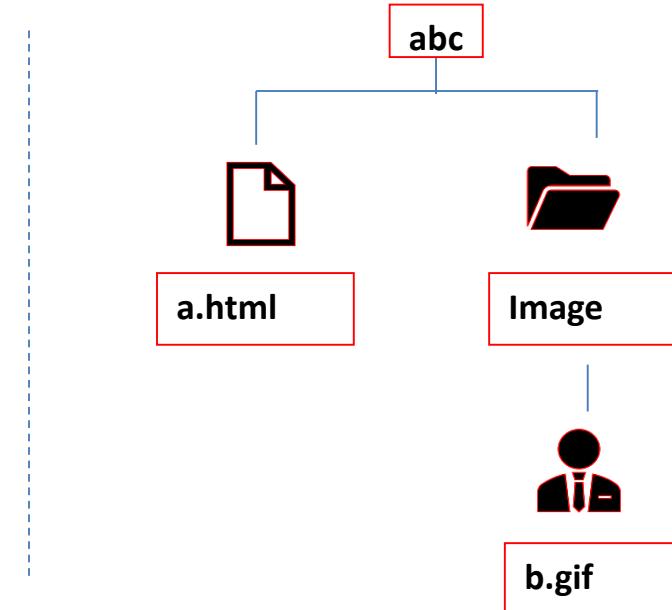
**Situation 1:** Save both files (a.html and b.gif) inside the “abc” folder, Let’s assume “abc” Is root folder.

```

```



(2)

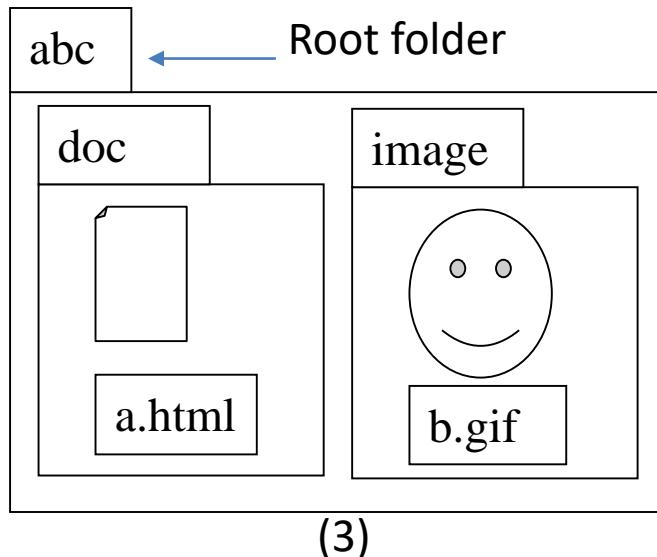


**Situation 2:** Create a subfolder ("Image") inside the root folder and save image files (b.gif) there and the HTML file(a.html) in the root folder ("abc")

```

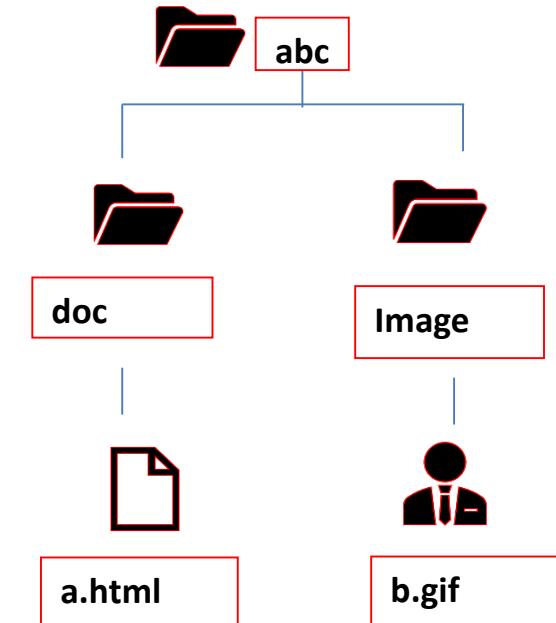
```

Folder name



(3)

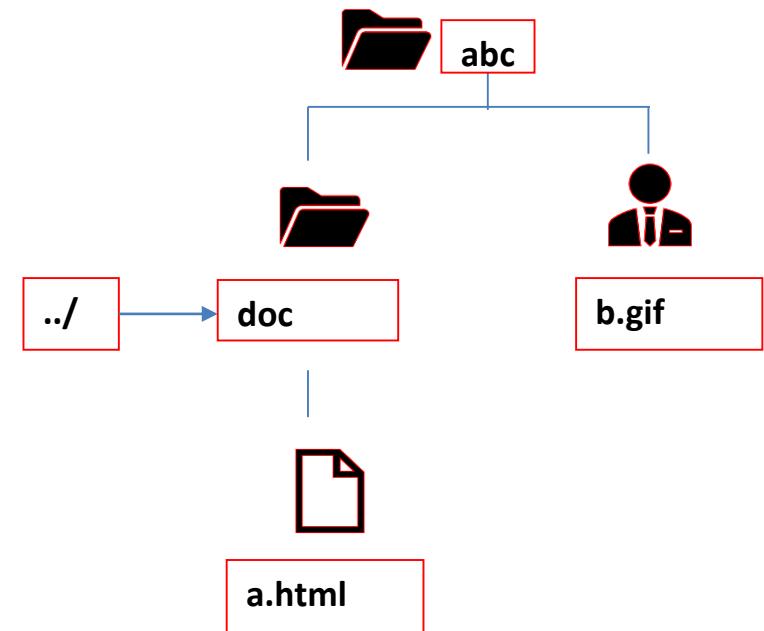
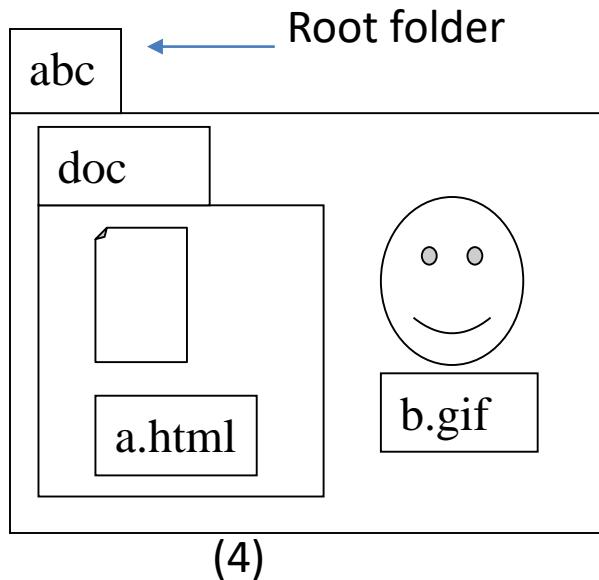
**Situation 3:** Create subfolders ("Image and doc") inside the root folder and save image file (b.gif) inside the image folder and HTML file (a.html) inside the "doc" folder.



```

```

Folder name



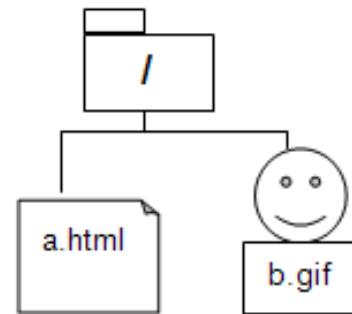
**Situation 4:** Create a subfolder ("doc") inside the root folder and save HTML files (a.html) there and the image file (b.gif) in the root folder ("abc").

```

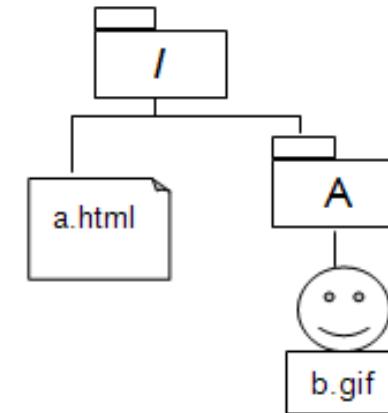
```

# Specifying Links to Other Pages

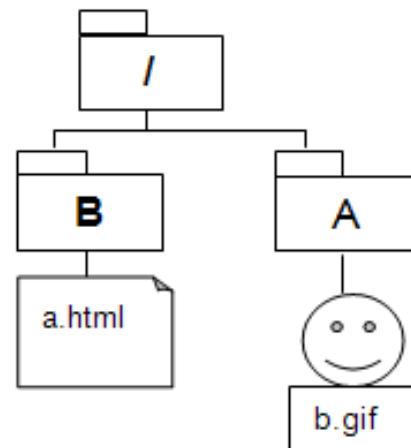
(Note: “/” = root folder)



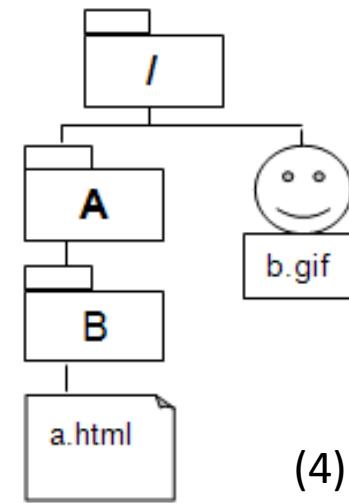
(1)



(2)



(3)



(4)

## ■ <A HREF="file name">-</A>

- These tags are used to set up hyperlinks for calling up other web pages.
- To call up Web pages from other directory, the following path must be specified.
  - Specify Relative Path –(Specify a relative position from the directory that stores the calling Web page.)
    - Path name/file name
  - Specify Absolute Path – (Specify an absolute position from the root directory)
    - /directory name (in the root directory)/directory name/file name

# HTML Links - Create a Bookmark

- HTML bookmarks are used to allow readers to jump to specific parts of a Web page.
- Bookmarks can be useful if your webpage is very long.
- To make a bookmark, you must first create the bookmark, and then add a link to it.
- When the link is clicked, the page will scroll to the location with the bookmark.

First, create a bookmark with the `id` attribute:

```
<h2 id="C4">Chapter 4</h2>
```

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

```
<a href="#C4">Jump to Chapter 4</a>
```

**■ <A HREF =“file name #anchor name”>-</A>**

- Eg `<a href="#B">Part B</a>`
  - These tags and the option assign a name to an **arbitrary position** in the HTML document.
  - Using a link defined by the `<A>` tags described next, the position specified by these tags can be display at the beginning of the browser screen.
  - Anchor name can be any alphabets

**■ <A NAME =“anchor name”>-</A>**

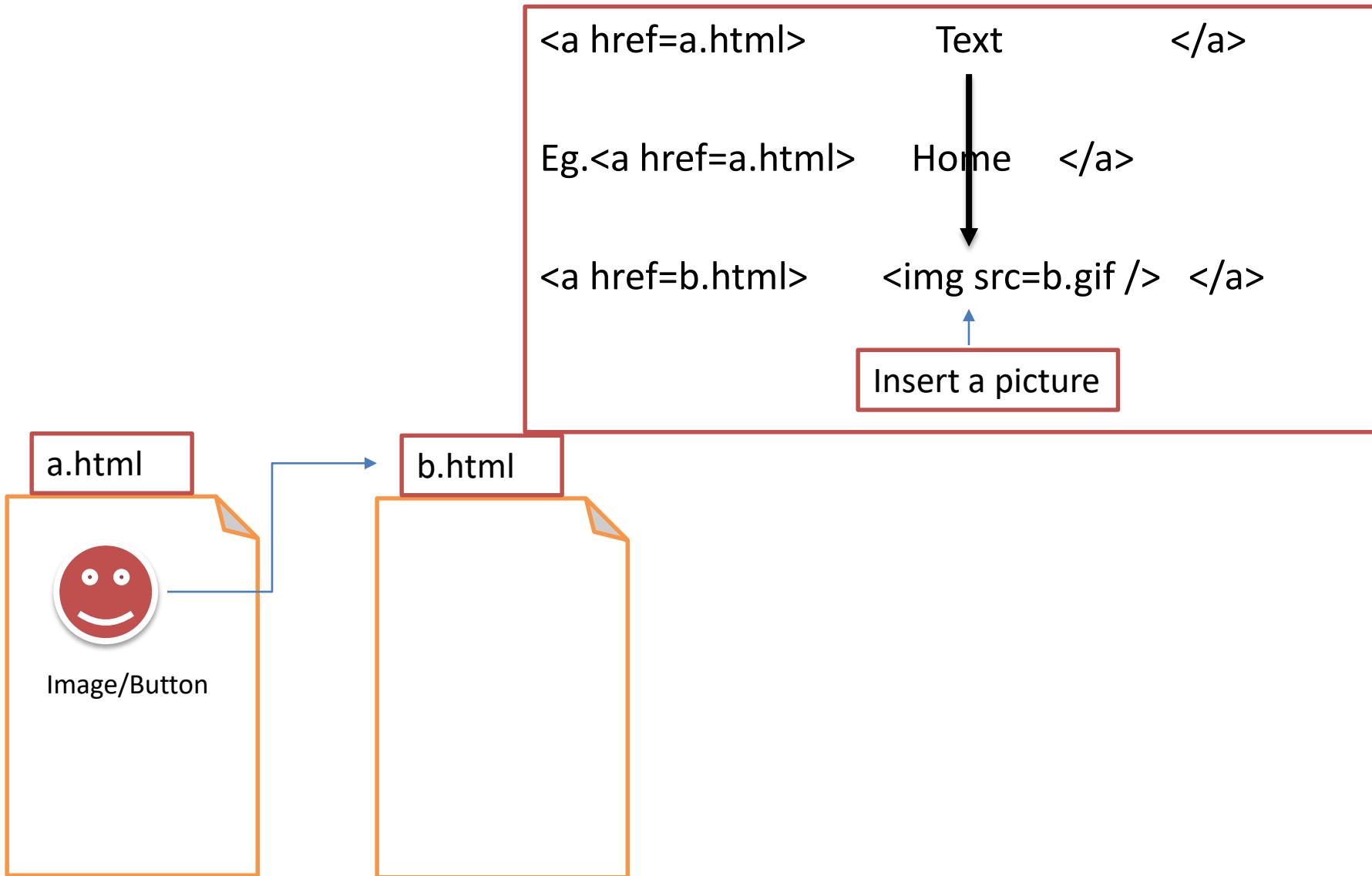
- Eg `<a name="B">Part B</a>`
  - These tags define a link to display the part of a Web page.

- For Link page:
  - Syntax :`<a href="page name.htm#section name">Introduction</a>`
  - Eg:`<a href="page2.html#d">Introduction</a>`
- Source Page:
  - `<a name=d><h1>Introduction</h1></a>`

Note: No space between webpage and the section name

```
<A HREF="#UCSC">About the UCSC</A>  
<A HREF="#ADMTC">About the ADMTC</A>  
:  
:  
:  
  
<H2><A NAME="UCSC">About the UCSC (Paragraph)  
<H2><A NAME="ADMTC">About the UCSC (Paragraph)
```

# Example ...(Image Link-Button)



- Others:
- <A NAME =“mailto:e-mail address”>-</A>
  - Specifies an e-mail program to be started up
  - mailto protocol and the e-mail address as the URL with the HREF attribute.

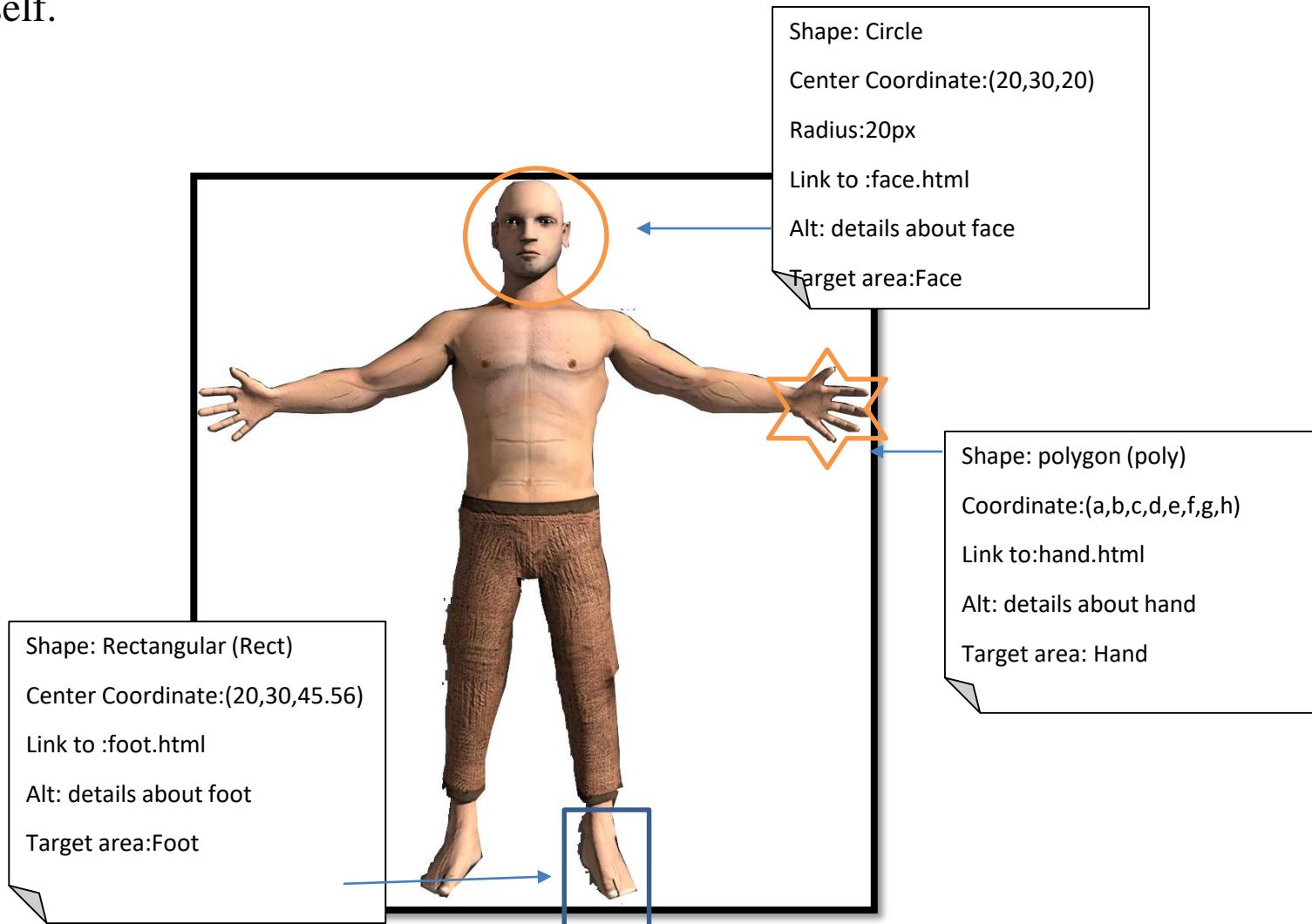
```
<ADDRESS>
<A HREF=“mailto:user00@flm.co.jp”>Development Team</A>
</ADDRESS>
```

- <A HREF =“specify external media file”>-</A>
  - These <A> tags are used to display external media data
  - <A HREF =“clip01.avi”>Play the production information video-</A>

## ■ What is an Image Map

- The image map is a function that calls up another Web page when any part of its image is clicked.
- There are two ways to use this image map function
  - Client side Image Map
    - This method makes use of the tag function in the HTML document
  - Server side Image Map
    - This method makes use of the application (CGI/SSI) on the WWW server to process data

make a couple of hyperlinks to open different html pages on the same image (“human.gif”) itself.



- <IMG SRC="image file name" ALT="Comment" USEMAP="#name of the map">
- These tags and the USEMAP attribute are used to display images to be used as the image map.
- <MAP NAME="name of map"> - </MAP>
- These tags are used to set up the image map
- The <AREA> tags described below are specified in the scope of these tags.

```
<IMG SRC="can02.gif" ALT="Can Image" USEMAP="#CAN">

<MAP NAME="CAN">

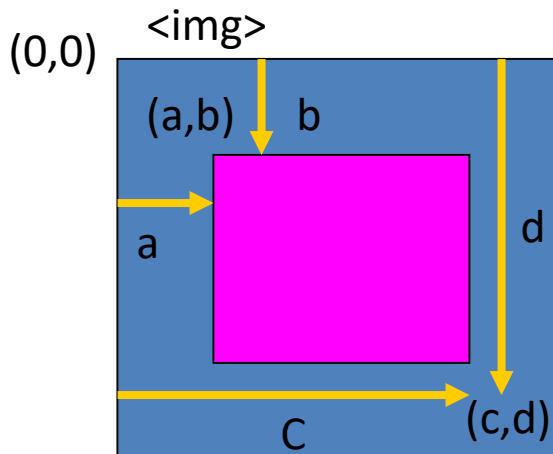
<AREA SHAPE="RECT" COORDS="8,38,87,63" HREF="Comet01.html" alt=xxx>
<AREA SHAPE="CIRCLE" COORDS="8,38,20" HREF="Comet02.html" alt=yyy >
<AREA SHAPE="POLY" COORDS="8,38,87,63" HREF="http://www.yahoo.com" alt=bbb >

</MAP>
```

**<AREA SHAPE="shape" COORDS="specify coordinates" HREF="file name or URL"  
ALT="comment">**

- <IMG SRC="image file name" ALT="Comment" USEMAP="#name of the map">
- These tags and the USEMAP attribute are used to display images to be used as the image map.

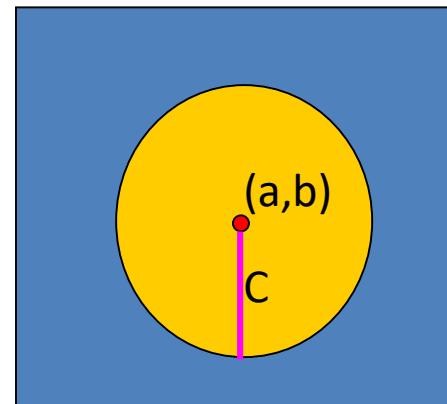
**Shapes:** RECT



Shape="rect"

Coords="a,b,c,d"

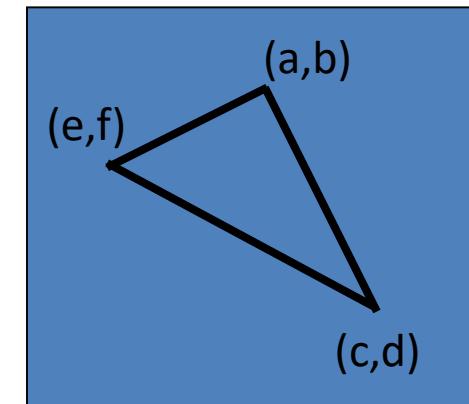
CIRCLE



Shape="circle"

Coords="a,b,c"  
(Centre & Radius –upto circumference of a circle)

POLY



Shape="poly"

Coords="a,b,c,d,e,f"  
(coordinate of each corner point)

```
<AREA SHAPE="shape" COORDS="specify coordinates" HREF="file  
name or URL" ALT="comment">
```

# Forms

## Registration Page

Name:	Nimal	Textbox: Type=text
Password	.....	Textbox: Type=password
Country	Nepal ▾	List box, Combo Box
Fruits	Mango Banana Apple	List box, with multiple values
Gender:	<input checked="" type="radio"/> Female <input type="radio"/> Male	Radio Button: (Allow to select one item out of the given list)
Hobbies	<input type="checkbox"/> Cricket <input type="checkbox"/> Swimming <input type="checkbox"/> Rugby <input type="checkbox"/> Football	Check box(Allow to select many items out of the given list)
Photo	Choose File No file chosen	File field
Comments	write your comment here.	Text Area : write a Comment : (allow to type many lines)
	Cancel Send	Buttons

Browser Interface

Table Structure

Name	Password	Country	Fruits	Gender	Hobbies
Takashi	****	Japan	Mango	Male	Swimming
Nimal	*****	Sri Lanka	Banana	Male	Cricket
Anne	****	England	Apple	Female	Reading Books

```
<FORM METHOD="POST" ACTION=mailto:e-mail address ENCTYPE="text/plain"> - </FORM>
```

#### Type of Data

- These tags and the attributes are used to define the data screen (HTML form)
- The input data is sent as e-mail to the address specified for the ACTION attribute.
- Form Attribute types are as follows:
  - Text
  - password
  - Radio
  - Check box
  - Submit
  - Reset

```
<INPUT TYPE="TEXT" NAME="textbox name" SIZE="length of the textbox">
```

- The `<INPUT>` tag and the attributes are used to define the textbox
- `<INPUT>` tag, when the attribute `TYPE="PASSWORD"` is specified, “**“\*”** will be displayed in the place of the input password.

```
<INPUT TYPE="PASSWORD" NAME="textbox name" SIZE="length of the textbox">
```

<SELECT NAME="pull-down Menu name">-</SELECT>

- These tags are used to define the pull down menu
- They are used in combination with the following <option> tags
- They are described in the scope of the <FORM> tags.
- An attribute of the “SELECT” tag can be set for multiple items to be selected from the pull-down menu.
- The “MULTIPLE” attribute that allows section of multiple items

<SELECT NAME="pull-down Menu name" MULTIPLE>-</SELECT>

- The “SIZE” attribute that specifies the number of items to be displayed in the list

<SELECT NAME="pull-down Menu name" SIZE="number of items">-</SELECT>

**<OPTION> Item Name (Value to be sent)**

- This tag defines an item of the pull-down menu. Selection list option)
- It is described in the scope of the **<SELECT>** tags
- An attribute of the **<OPTION>** tag can be set for a list item to be selected by default (default selection item)

```
<select name ="fruit">
<option> Banana</option>
<option> Pineapple</option>
<option selected> Star fruit</option>
<option> Mango</option>
</select>
```

```
<select name="vegetable" multiple size="3">
<option> Go-ya </option>
<option> Papaya </option>
<option> Carrot </option>
<option> Egg </option>
<option> Fish </option>
</select>
```

```
<INPUT TYPE="RADIO" NAME="radio button group name"  
VALUE="value to be sent">-
```

- This tag defines the radio button (Exclusive input). It is specified in the scope of the <FORM> tags

```
<INPUT TYPE="RADIO" NAME="gender" VALUE="WOMAN">Female<BR>  
<INPUT TYPE="RADIO" NAME="gender" VALUE="MAN">Male<BR>
```

```
<INPUT TYPE="CHECKBOX" NAME="checkbox name">
```

- This tag defines the checkbox. It is specified in the scope of the <FORM> tags.
- In the e-mail to be sent, the selected item is described as "checkbox name="on".

```
<INPUT TYPE="CHECKBOX" NAME="Colombo"> Colombo
```

```
<INPUT TYPE="CHECKBOX" NAME="Kandy"> Kandy
```

```
<INPUT TYPE="CHECKBOX" NAME="Rathnapura"> Rathnapura
```

```
<INPUT TYPE="CHECKBOX" NAME="Mathara"> Mathara
```

```
<TEXTAREA NAME="name of the input area" ROWS="height of the input area"  
COLS="width of the input area"> - </TEXTAREA>
```

- These tags define an input area that contains more than 1 line of input
- They are specified in the scope of the <FORM> tags
- The character string described in the scope of the <TEXTAREA> tags is displayed in the area as the initial value
- In the e-mail to be sent, it is described as “name of input area=input sentences”



```
<TEXTAREA NAME="message" ROWS="5" COLS="60">  
  
Please write your comments here  
  
</TEXTAREA>
```

<INPUT TYPE="SUBMIT" VALUE="name of send button">

- This tag and the attributes define the button for sending input or selected data
- Press this button and e-mail will be sent to the address specified by the ACTION attribute of the <FORM> tag.

<INPUT TYPE="RESET" VALUE="name of cancel button">

- This tag and the attributes define the button for clearing the input data

```
<INPUT TYPE="SUBMIT" VALUE="Send">
```

```
<INPUT TYPE="RESET" VALUE="Cancel">
```

# HTML5 Input Types

- HTML5 added several new input types:
  - color
  - date
  - datetime
  - datetime-local
  - email
  - month
  - number
  - range
  - search
  - tel
  - time
  - url
  - week

# HTML5 Input Type : Number

<p>Numeric restrictions will apply in the input field. </p>



```
<form action="action_page.php">  
    Quantity (between 1 and 5):  
    <input type="number" name="quantity" min="1" max="5">  
    <input type="submit">  
</form>
```

Numeric restrictions will apply in the input field.

Quantity (between 1 and 5):

```
<form action="action_page.php">  
    Quantity:  
    <input type="number" name="points" min="0" max="100" step="10" value="30">  
    <input type="submit">  
</form>
```

Fixed steps will apply in the input field.

Quantity:

# HTML5 Input Type : Date



```
<form>
  Birthday:
  <input type="date" name="bday">
</form>
```

A date picker can pop-up when you enter the input field.

Birthday:  ▼ ▲ Submit

October, 2015 ▾

◀ ⏪ ⏩ ▶

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

# HTML5 Input Type : Date -add restrictions to the input



```
<form>
```

Enter a date before 1980-01-01:

```
<input type="date" name="bday" max="1979-12-31"><br>
```

Enter a date after 2000-01-01:

```
<input type="date" name="bday" min="2000-01-02"><br>
```

```
</form>
```

Enter a date before 1980-01-01:

Enter a date after 2000-01-01:

October, 2015 ▾

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

# HTML5 Input Type : Date -add restrictions to the input



```
<form action="action_page.php">  
    Birthday (month and year):  
    <input type="month" name="bdaymonth">  
    <input type="submit">  
</form>
```

Depending on browser support:  
A date picker can pop-up when you enter the input field.

Birthday (month and year):  Submit

October, 2015 ▾

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

# HTML5 Input Type : Date -add restrictions to the input



```
<form>
```

Select a week:

```
  <input type="week" name="week_year">  
</form>
```

```
<form>
```

Select a time:

```
  <input type="time" name="usr_time">  
</form>
```

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

Birthday (date and time):

```
  <input type="datetime-local"  
        name="bdays">  
  <input type="submit" value="Send">  
</form>
```

```
<form>
```

E-mail:

```
  <input type="email" name="email">  
</form>
```



Birthday (month and year): March 2021

Select a week: Week 25, 2021

Select a time: 10:35 AM

Birthday (date and time): 06/16/2021 11:24 AM

E-mail: ww@Abc.com

Search Google: qweqeqewq

Add your homepage: qewe

# HTML5 Input Type : Date -add restrictions to the input

```
<form>  
    Search Google:  
    <input type="search" name="googlesearch">  
</form>
```



```
<form>  
    Add your homepage:  
    <input type="url" name="homepage">  
</form>
```



```
<form action="action_page.php">  
    <input type="text" name="fname" placeholder="First name"><br>  
    <input type="text" name="lname" placeholder="Last name"><br>  
    <input type="submit" value="Submit">  
</form>
```

First name
Last name
Submit

### Example

Group related elements in a form:

```
<form>
  <fieldset>
    <legend>Personalia:</legend>
    Name: <input type="text"><br>
    Email: <input type="text"><br>
    Date of birth: <input type="text">
  </fieldset>
</form>
```

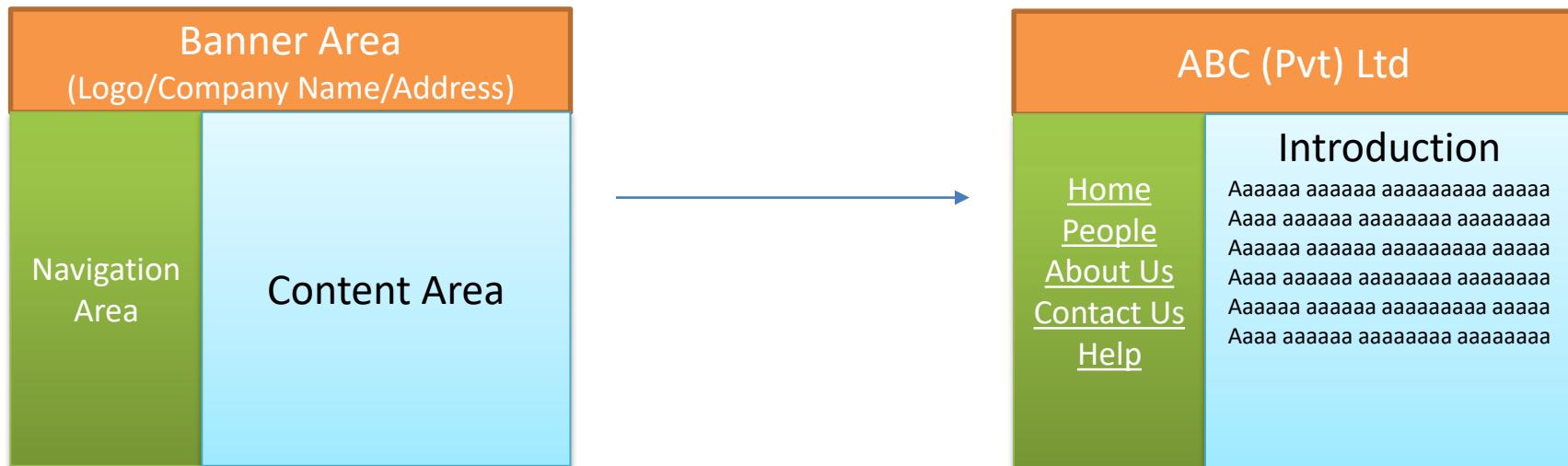
Personalia:

Name:

Email:

Date of birth:

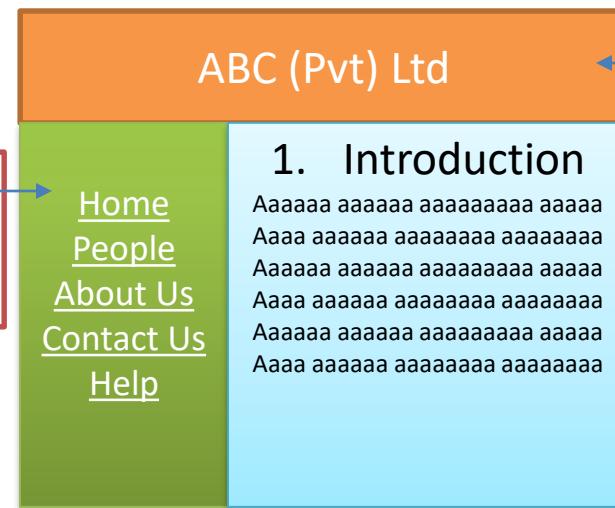
- What is the Frame Function?
  - This function splits the browser screen into multiple frames, which display separate web pages.
  - To use the frame function, define the method for splitting the browser and the file names of the web pages to be displayed in different frames in the HTML file used for frame setting



Navigation:-

1. Local Navigation (Sub Links) Eg: Profile>>Dean, HOD,
2. Global Navigation (Common Links) eg: Home, Help, About us

Frame Name: left  
Include Page: left.html  
Include Content: Navigation



Frame Name: top (any name)  
Include Page: top.html  
Include Content: Banner

Frame Name: main  
Include Page: main.html  
Include Content: Introduction

- <FRAMESET>-</FRAMESET>

- These tags are used to define the split screen. In the HTML file used for frame setting, these tags are used in place of the <BODY> tags.
  - The method of split and the size of each frame are set up in the options
  - The COLS attribute that splits the screen into left and right frames
  - The ROWS attribute the split the screen into top and bottom frames

<FRAMESET ROWS="size of the top frame, size of the bottom frame"> - </FRAMESET>

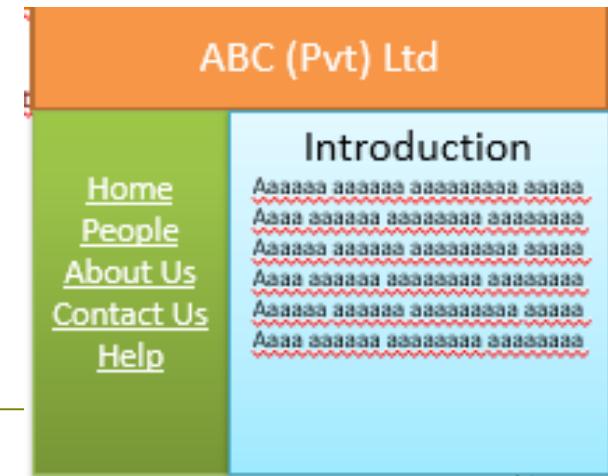
- <FRAME SRC="HTML file name" NAME="frame name">
  - This Tag defines the web page to be displayed in a frame. It is specified in the scope of the <FRAMESET> tags.
  - Frames are displayed from the left or the top in the order of description.

<A HREF="HTML file name" TARGET="frame name">-</A>

- The target frame name that display the HTML file is specified in the “TARGET” attribute
- The target frame name is the frame name assigned by the “NAME” attribute of the <FRAME> tags.
- The frame that display the linked web page is specified by the “TARGET” attribute.

- <NOFRAMES> - </NOFRAMES>
  - These tags define the content of the web page to be displayed for browsers that do not support the frame function

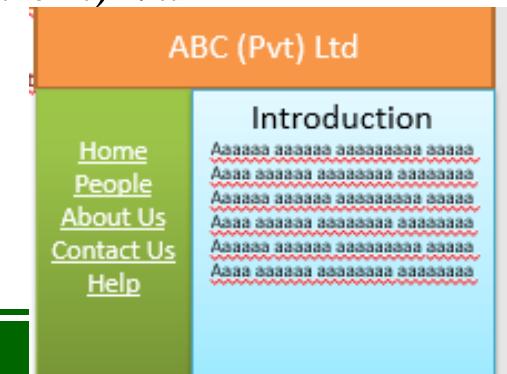
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-
1252"> <meta http-equiv="Content-Language" content="en-us">
<title>Main Entrance</title>
<!-- frames -->
<frameset rows="159,*" border="0">
<frame name="top" src="top.html" marginwidth="0" marginheight="0"
scrolling="no" frameborder="0" noresize="noresize">
<frameset cols="159,*" border="0">
<frame name="left" src="left.html" marginwidth="0" marginheight="0"
scrolling="no" frameborder="0" noresize="noresize">
<frame name="main" src="main.html" marginwidth="0" marginheight="0"
scrolling="auto" frameborder="0" noresize="noresize">
</frameset>
</frameset>
</head>
<body>
<noframes>
</noframes>
</body>
</html>
```



- TARGET attribute. This attribute is not found in the frame definition but is used within normal HTML documents to direct data to specific frames
- TARGET attribute is used to direct data to different frame
- consider the TARGET attribute in terms of frames and the <A> tag.
- In order to direct data to a different frame, the receiving frame must have been named in the frame definition.

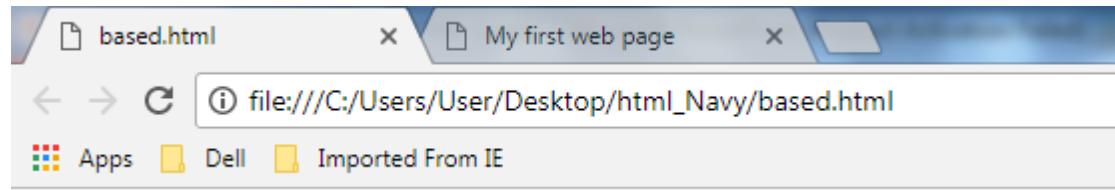
# Targeting frames ...

- 11.2 How to give links (navigation page for left.html) <html>
- <head>
- <title>left navigation page</title>
- </head>
- <body topmargin="0" leftmargin="0" rightmargin="0" bgcolor="#aaccbb">
- 
- <a href="page01.html" target="main">Page 01</a>
- <br>
- <a href="page02.html" target="main">Page 02</a>
- <br>
- <a href="page03.html" target="\_blank">page03-(target\_blank)</a>
- <br>
- <a href="page04.html" target="\_self">page04-target\_self)</a>
- <br>
- <a href="page05.html" target="\_parent">page05-target\_parent)</a>
- <br>
- </body>
- </html>



- There are some special target names, called frame target names. These all begin with an underscore ( \_ ) and are used to load pages into groups of frames.
- **\_blank** The user agent should load the designated document in a new, unnamed window.
- **\_self** The user agent should load the document in the same frame as the element that refers to this target.
- **\_parent** The user agent should load the document into the immediate FRAMESET parent of the current frame. This value is equivalent to **\_self** if the current frame has no parent.
- **\_top** The user agent should load the document into the full, original window (thus canceling all other frames). This value is equivalent to **\_self** if the current frame has no parent

# Based Tag for target



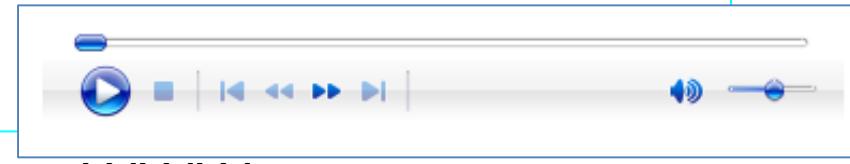
```
<html>
<head>
<base target="_self">
</head>
<body>
<p>
<a href="firstpage.html" target="_blank">This link</a>
will load in a new window because the target attribute is set to "_blank".
</p>
<p>
<a href="firstpage.html">This link</a>
will also load in a new window even without a target attribute because base target is “_self”.
</p>
</body>
</html>
```

- <A HREF="mailto:e-mail address > - </A>
  - These tags define to send an e-mail to somebody

<A HREF="mailto:saman17lk@yahoo.com">saman</a>

- How to embed a Music file (.wav, .mid)

```
<HTML>
<HEAD>
<TITLE>EMBEDDED</TITLE>
</HEAD>
<BODY>
<EMBED SRC="MUSIC.MID" WIDTH="144" HEIGHT="60" PLAY="TRUE"      LOOP="TRUE"
QUALITY="HIGH"><!-- -or- -->
<EMBED SRC="MUSIC.MID" WIDTH="144" HEIGHT="60" PLAY="TRUE"
LOOP="-1" QUALITY="HIGH" autostart=false>
</BODY>
</HTML>
```



- Adding Multimedia bgsound to WWW

```
<HTML>
<HEAD>
<TITLE>Background music</TITLE>
</HEAD>
<BODY bgcolor="#2233ad">
<bgsound src="SOUND.WAV">
</BODY>
</HTML>
```

- How to embed a Movie file
- Avi,wma files can use

```
<HTML>
<HEAD>
<TITLE>EMBEDDED</TITLE>
</HEAD>
<BODY>
<EMBED SRC="SRILANKA.AVI" WIDTH="300" HEIGHT="350" PLAY="TRUE"
       LOOP="TRUE" QUALITY="HIGH">
</BODY>
</HTML>
```

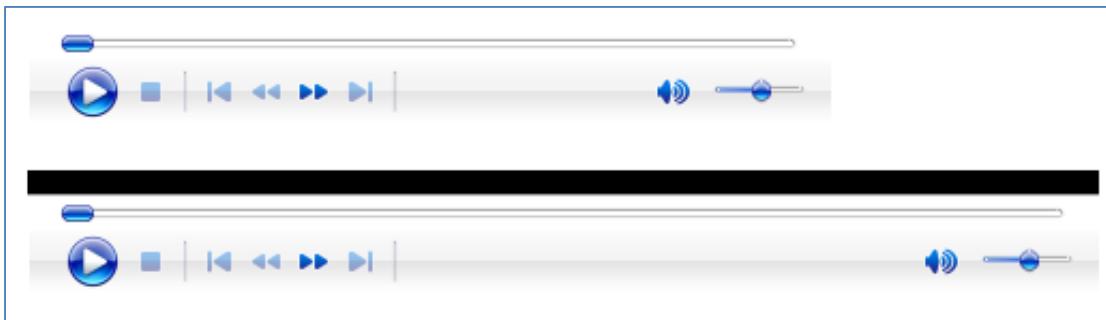


- Using HTML 5
- For Audio

```
<html>
<body>

<audio width="320" height="40" controls>
  <source src="a.mp3" type="audio/mp3" />
  Your browser does not support the audio tag.
</audio>

</body>
</html>
```



- Using HTML 5
- For Video

```
<html>
<body>

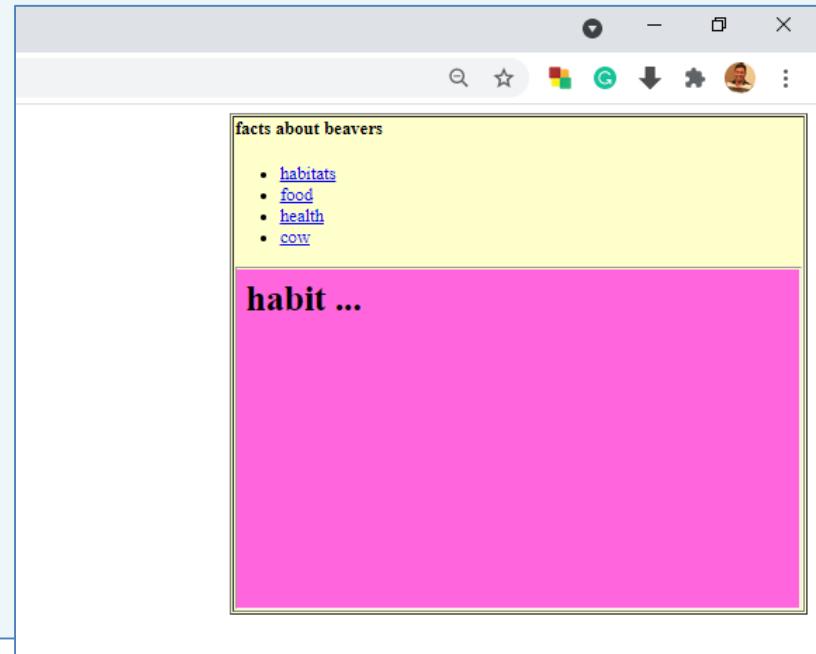
<video width="320" height="240" controls>
  <source src="abc.mp4" type="video/mp4">
</video>

</body>
</html>
```

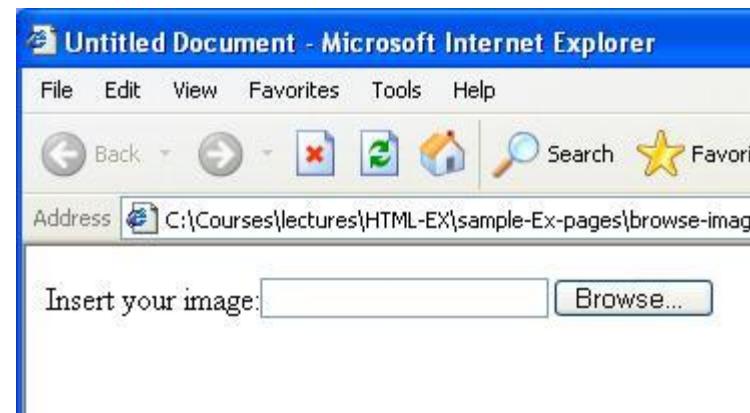


```
<body>
<TABLE ALIGN=RIGHT BORDER=1 BGCOLOR="#FFFFCC">
<TR>
<TD><H4>Facts About Beavers</H4>
<UL>
<LI><A HREF="habitat.html" TARGET="abc">Habitats</A> </LI>
<LI><A HREF="food.html" TARGET="abc">Food</A></LI>
<LI><A HREF="health.html" TARGET="abc">Health</A></LI>
</UL>

<IFRAME
NAME="abc"
SRC="Habitat.html"
WIDTH=300 HEIGHT=200>
</IFRAME></TD>
</TR></TABLE>
</body>
</html>
```



```
<body>
<!--forms then file field -->
<p>Insert your image:<input type="file" name="as"></p>
</body>
```



```
<a href="applicationform.doc">Download your Application (doc)
</a> or
<a href="applicationform.docx">Download your Application
(doc) </a>

<a href="applicationform.pdf">PDF </a>
```

# <code>-</code>

## Example

Regular text, followed by code.

<p>This is how we declare a Javascript variable:<br/> <code>var  
i = 0;</code> </p>

Result:

This is how we declare a Javascript variable:

var i = 0;

- <?xml version="1.0" encoding="us-ascii"?>
- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
- <html lang="en" xml:lang="en">
- <head>
- <title>On Going Events</title>
- <link href="../css/default.css" rel="stylesheet" type="text/css"  
  title="Default" />
- <link href="../css/black.css" rel="alternate stylesheet" type="text/css"  
  title="High Contrast" />
- </head>
- <body>
- </body>
- </html>

<!DOCTYPE > Defines the document type and tells the browser which version of the markup language the page is written in. This tag goes before the <html> start tag.

- Describe metadata within an HTML document

```
<head>
<meta name="description" content="Free Web tutorials" />
<meta name="keywords" content="HTML,CSS,XML,JavaScript" />
<meta name="author" content="Hege Refsnes" />
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
</head>
```

## ■ Definition and Usage

- Metadata is information about data.
- The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parseable.
- Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.
- The <meta> tag always goes inside the head element.
- The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

# What is Character Encoding?

- ASCII was the first character encoding standard (also called character set). It defines 127 different alphanumeric characters that could be used on the internet.
- ASCII supported numbers (0-9), English letters (A-Z), and some special characters like ! \$ + - ( ) @ < > .
- ANSI (Windows-1252) was the original Windows character set. It supported 256 different character codes.
- ISO-8859-1 was the default character set for HTML 4. It also supported 256 different character codes.
- Because ANSI and ISO were limited, the default character encoding was changed to UTF-8 in HTML5.
- UTF-8 (Unicode) covers almost all of the characters and symbols in the world.

- The most interesting new HTML5 elements are:
- New **semantic elements** like `<header>`, `<footer>`, `<article>`, and `<section>`.
- New **attributes of form elements** like number, date, time, calendar, and range.
- New **graphic elements**: `<svg>` and `<canvas>`.
- New **multimedia elements**: `<audio>` and `<video>`.

# New Semantic/Structural Elements

- HTML5 offers new elements for better document structure:

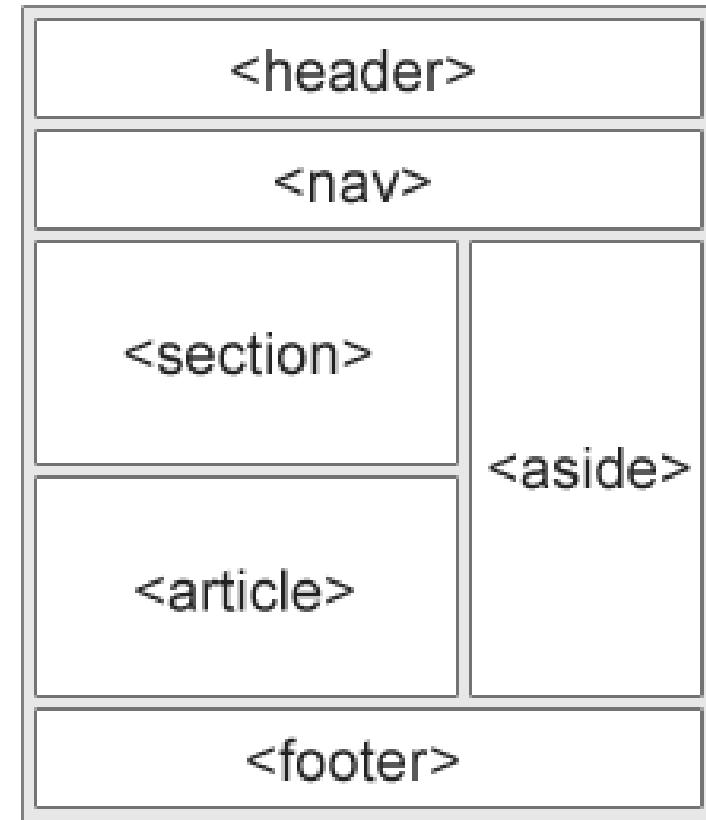
Tag	Description
<article>	Defines an article in a document
<aside>	Defines content aside from the page content
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<main>	Defines the main content of a document
<mark>	Defines marked/highlighted text
<menuitem>	Defines a command/menu item that the user can invoke from a popup menu
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links
<progress>	Represents the progress of a task
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time
<wbr>	Defines a possible line-break

# What are Semantic Elements?

- Semantics is the study of the meanings of words and phrases in a language.
- Semantic elements = elements with a meaning.
- A semantic element clearly describes its meaning to both the browser and the developer.
- Examples of **non-semantic** elements:
  - <div> and <span> - Tells nothing about its content.
- Examples of **semantic** elements:
  - <form>, <table>, and <article> - Clearly defines its content.

# New Semantic Elements in HTML5

- 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:
  - <article>
  - <aside>
  - <details>
  - <figcaption>
  - <figure>
  - <footer>
  - <header>
  - <main>
  - <mark>
  - <nav>
  - <section>
  - <summary>
  - <time>



# HTML5 <section> Element

- The <section> element defines a section in a document.
- According to W3C's HTML5 documentation: "A section is a thematic grouping of content, typically with a heading."
- A home page could normally be split into sections for introduction, content, and contact information.

```
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF)
is....</p>
</section>
```



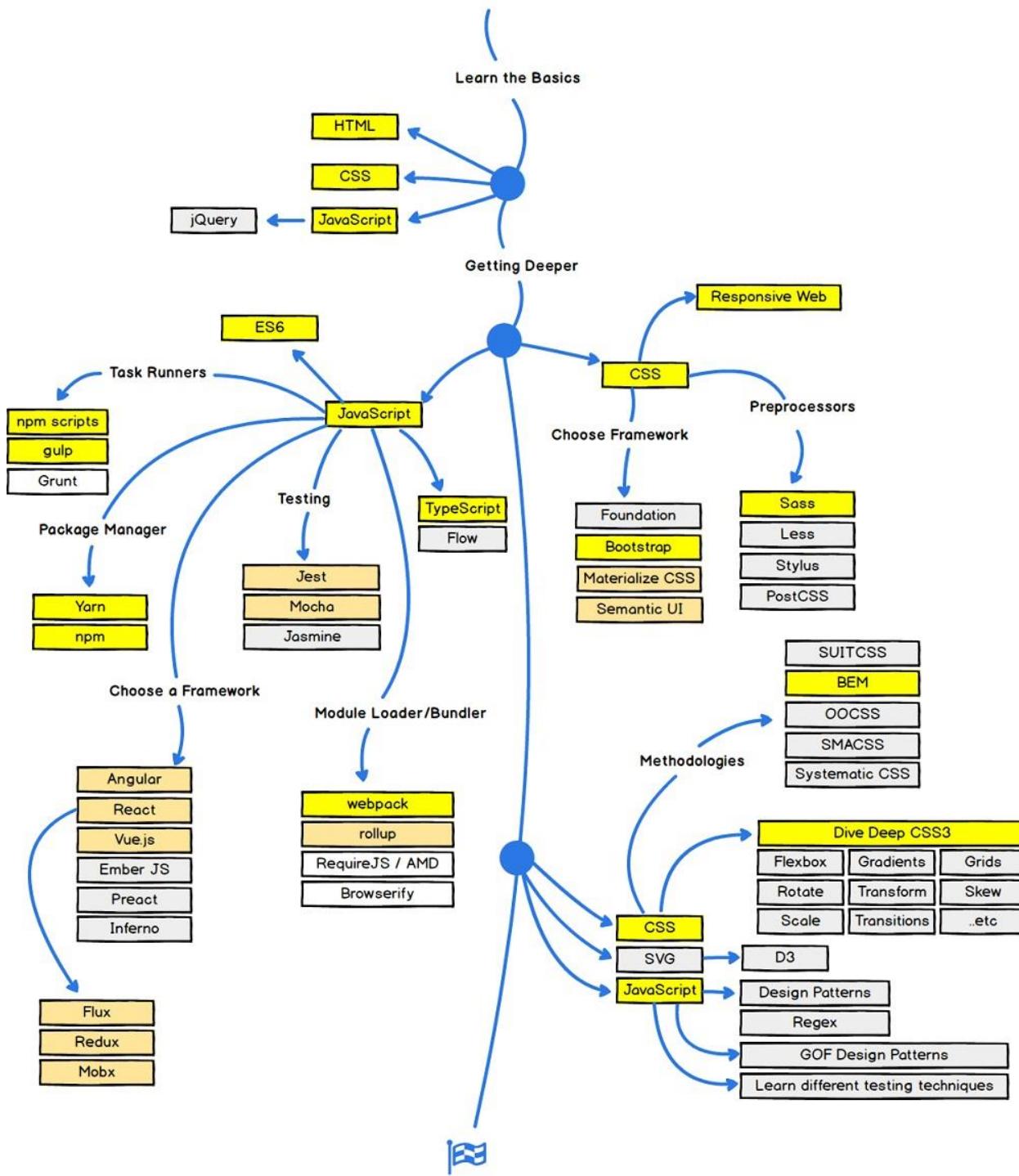
[https://www.w3schools.com/html/html5 semantic elements.asp](https://www.w3schools.com/html/html5_semantic_elements.asp)

- [http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp)

- **Markup Validation Services v0.7.1** <http://validator.w3.org/>
  - [\(W3C markup validation service\)](#)
- **CSS validator**
  - <http://jigsaw.w3.org/css-validator/>

# How Browsers Work: Behind the scenes of modern web browsers

- [https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/#HTML Parser](https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/#HTML_Parser)



# Full Stack Web Developer

A full stack web developer is a person who can develop both **client** and **server** software.

In addition to mastering HTML and CSS, he/she also knows how to:

- Program a **browser** (like using JavaScript, jQuery, Angular, or Vue)
- Program a **server** (like using PHP, ASP, Python, or Node)
- Program a **database** (like using SQL, SQLite, or MongoDB)

# Popular Stacks

- LAMP stack: JavaScript - Linux - Apache - MySQL - PHP
- LEMP stack: JavaScript - Linux - Nginx - MySQL - PHP
- MEAN stack: JavaScript - MongoDB - Express - AngularJS - Node.js
- Django stack: JavaScript - Python - Django - MySQL
- Ruby on Rails: JavaScript - Ruby - SQLite - PHP

# Client Software (Front End)

- [HTML](#)
- [CSS](#)
- [Bootstrap](#)
- [W3.CSS](#)
- [JavaScript](#)
- [ES5](#)
- [HTML DOM](#)
- [JSON](#)
- [XML](#)
- [jQuery](#)
- [Angular](#)
- [React](#)
- [Backbone.js](#)
- [Express.js](#)
- [Ember.js](#)
- [Redux](#)
- [Storybook](#)
- [GraphQL](#)
- [Meteor.js](#)
- [Grunt](#)
- [Gulp](#)

# Server Software (Back End)

- [PHP](#)
- [ASP](#)
- [C++](#)
- [C#](#)
- [Java](#)
- [Python](#)
- [Node.js](#)
- [Ruby](#)
- [REST](#)
- [GO](#)
- [SQL](#)
- [MongoDB](#)
- [Firebase.com](#)
- [Sass](#)
- [Less](#)
- [Parse.com](#)
- [PaaS \(Azure and Heroku\)](#)

# Advantages & Disadvantages

The advantage of being a full stack web developer is:

- You can master all the techniques involved in a development project
- You can make a prototype very rapidly
- You can provide help to all the team members
- You can reduce the cost of the project
- You can reduce the time used for team communication
- You can switch between front and back end development based on requirements
- You can better understand all aspects of new and upcoming technologies

## Disadvantages

- The solution chosen can be wrong for the project
- The solution chosen can be dependent on developer skills
- The solution can generate a key person risk
- Being a full stack developer is increasingly complex

# Reference and further reading

- <https://www.w3schools.com/html/>
- <https://www.tutorialspoint.com/html/index.htm>
- <https://www.javatpoint.com/html-tutorial>