

UNIT-1

Web Essentials and Mark-up language- HTML

Outline

The Internet, basic internet protocols, the World Wide Web, HTTP Request message, HTTP response message, web clients, web servers

HTML: Introduction, history and versions. HTML elements: headings, paragraphs, line break, colors and fonts, links, frames, lists, tables, images and forms

Difference between HTML and HTML5.

CSS: Introduction to Style Sheet, CSS features, CSS core syntax, Style sheets and HTML, Style rule cascading and inheritance

Text properties. Bootstrap.

Web Technology

- The methods by which computers communicate with each other through the use of markup languages and multimedia packages is known as **Web Technology**.

Web Technologies

- HTML
- XHTML
- CSS
- XML
- JavaScript
- VBSCRIPT
- DOM
- DHTML
- AJAX
- E4X
- WMLScript
- SQL
- ASP
- ADO
- PHP
- .NET
- SMIL
- SVG
- FLASH
- Java Applets
- Java Servlets
- Java Server Page

What is Internet?

- The Inter-network or Internet is essentially a **global network of computing resources that allow devices and networks to communicate with each other.**

What is WWW?

- WWW stands for **World Wide Web**.
- A technical definition of the World Wide Web is – **All the resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP)**.
- The Interlinked hypertext documents are accessed using HTTP Protocol
- It uses the Client - Server architecture

- **In simple terms, The World Wide Web is a way of exchanging information between computers over the Internet.**

What is HTTP?

- HTTP stands for **Hypertext Transfer Protocol**.
- This is the protocol being used to transfer **hypertext documents** that makes the World Wide Web possible.
- A standard web address such as **http://www.google.com** is called a **URL** and here the prefix **http** indicates the protocol being used.

What is URL?

- URL stands for Uniform Resource Locator, and is used to specify addresses on the World Wide Web.
- A URL will have the following format –
 - **protocol://www.hostname/other_information**
- The protocol is followed by a colon, two slashes, and then the domain name. The **domain name is the computer on which the given resource is located.**

What is Website?

- It is a collection of **various pages linked together in meaningful way** to describe a body of information & it is **written in HTML markup language**.
- Each page available on the website is called a ***web page*** and first page of any website is called ***home page*** for that site.

What is Web Server?

- Every Website sits on a computer known as a Web server.
- This server is always connected to the internet.
- Every Web server that is connected to the Internet is given a unique address. For example, 68.178.157.132
- When you register a Web address, also known as a domain name, (Eg. eportal.incometax.gov.in) **you have to specify the IP address of the Web server that will host the website.**
- **Examples of Web Servers**
 - Apache Tomcat
 - Glassfish
 - Microsoft IIS

What is Web Browser?

- Web Browsers are software installed on your PC and used to access the Web.
- **Examples of Web Browsers**
 - Netscape Navigator,
 - Microsoft Internet Explorer
 - Mozilla Firefox.

What is HTML?

- HTML stands for **Hyper Text Markup Language**.
- This is the **language in which we write web pages** for any Website.
- This is a subset of Standard Generalized Mark-Up Language (SGML) for electronic publishing, the specific standard used for the World Wide Web.

What is Hyperlink?

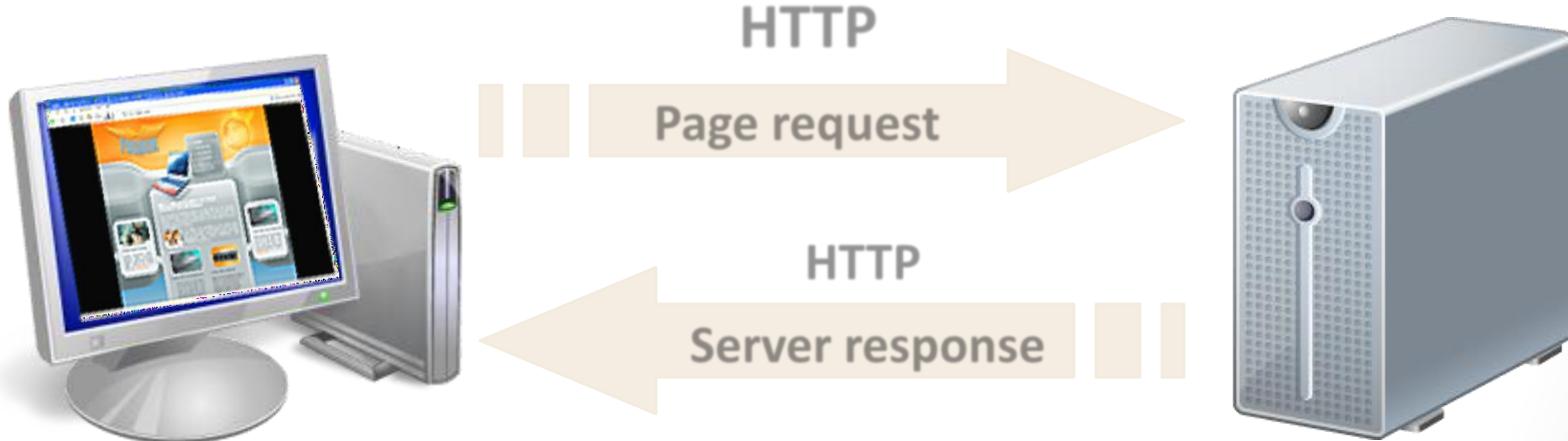
- A **hyperlink or simply a link** is a **selectable element** in an electronic document **that serves as an access point to other electronic resources.**
- Typically, you click the hyperlink to access the linked resource.
- Familiar hyperlinks include clickable text links, buttons, icons, image maps, etc.

Web - Domain Names & Extension Types

- A domain name is the part of your Internet address that comes after "www". For example, scopus.com
- Some Domain Extensions are as mentioned below:
 - **.com** – Stands for company/commercial, but it can be used for any website.
 - **.net** – Stands for network and is usually used for a network of sites.
 - **.org** – Stands for organization and is supposed to be for non-profit bodies.
 - **.us, .in** – They are based on your country names so that you can go for country specific domain extensions
 - **.biz** – A newer extension on the Internet and can be used to indicate that this site is purely **related to business**.

How the Web works?

- WWW uses classical client / server architecture
 - HTTP is text-based request-response protocol



**Client running a
Web Browser**

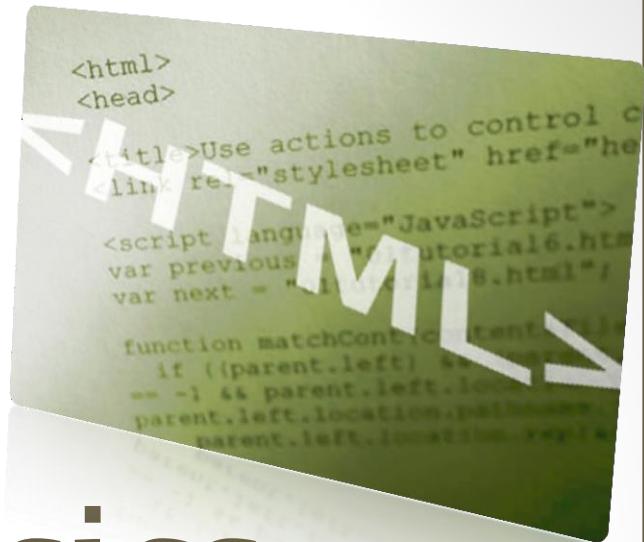
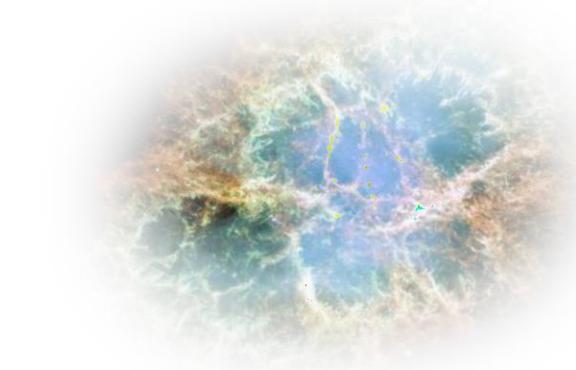
**Server running Web
Server Software
(IIS, Apache, etc.)**

How to choose a Technology?

- **Depends on:**
 - What is the type of content?
 - Who is your audience?
 - Who will modify your content?
 - What are your Future Plans?
 - Availability of technology?
 - Your previous experiences?
 - Portability and Data sharing

HTML Basics

Text, Images, Tables, Forms



What is HTML?

- HTML is the standard markup language for creating Web pages.
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by **tags**
- Browsers do not display the HTML tags, but use them to render the content of the page

A Simple HTML Document

Example

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

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

</body>
</html>
```

Explanation

- The **<html>** element is the root element of an HTML page
- The **<head>** element contains meta information about the document
- The **<title>** element specifies a title for the document
- The **<body>** element contains the visible page content
- The **<h1>** element defines a large heading
- The **<p>** element defines a paragraph

HTML Tags

- HTML tags are element names **surrounded by angle brackets**:

<tagname> content goes here...</tagname>

- HTML tags normally come **in pairs** like <p> and </p>
- The first tag in a pair is the **start tag**, the second tag is the **end tag**
- The **end tag** is written like the start tag, but with a **forward slash** inserted before the tag name

HTML Page Structure

```
<html>

  <head>
    <title>Page title</title>
  </head>

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

</html>
```

HTML Versions

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

Creating HTML Page

Write HTML Using [Notepad or TextEdit](#)

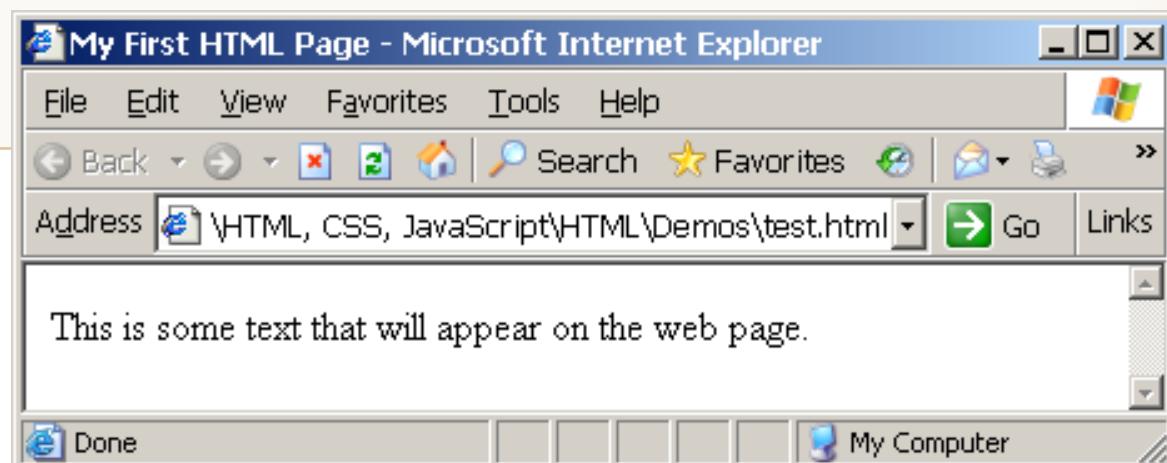
Save the file on your computer using [.html or .htm extension](#) and set the encoding to UTF-8

View the HTML Page in Your [Browser](#)

First HTML Page

test.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```



First HTML Page: Tags

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```

Opening tag

Closing tag

An HTML element consists of an opening tag, a closing tag and the content inside.

First HTML Page: Header

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```

HTML header

First HTML Page: Body

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```

HTML body

HTML Headings-

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

HTML Code

```
<!DOCTYPE html>
<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>

</body>
</html>
```

Output

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

Headings and Paragraphs

- Heading Tags (h1 – h6)

```
<h1>Heading 1</h1>
<h2>Sub heading 2</h2>
<h3>Sub heading 3</h3>
```

- Paragraph Tags

```
<p>This is my first paragraph</p>
<p>This is my second paragraph</p>
```

- Sections: div

```
<div style="background: skyblue;">
    This is a div </div>
```

Text Formatting

- Text formatting tags are used to modify the text between the opening tag and the closing tag
 - Ex. **Hello** makes “Hello” appear in bold

	bold
<i></i>	<i>italicized</i>
<u></u>	<u>underlined</u>
<sup></sup>	Sample ^{superscript}
<sub></sub>	Sample _{subscript}
	strong
<pre></pre>	Displays HTML text as it is
<blockquote cite=" " > </blockquote>	Quoted text block from other source
	Deleted text – strike through effect

Text Formatting Example

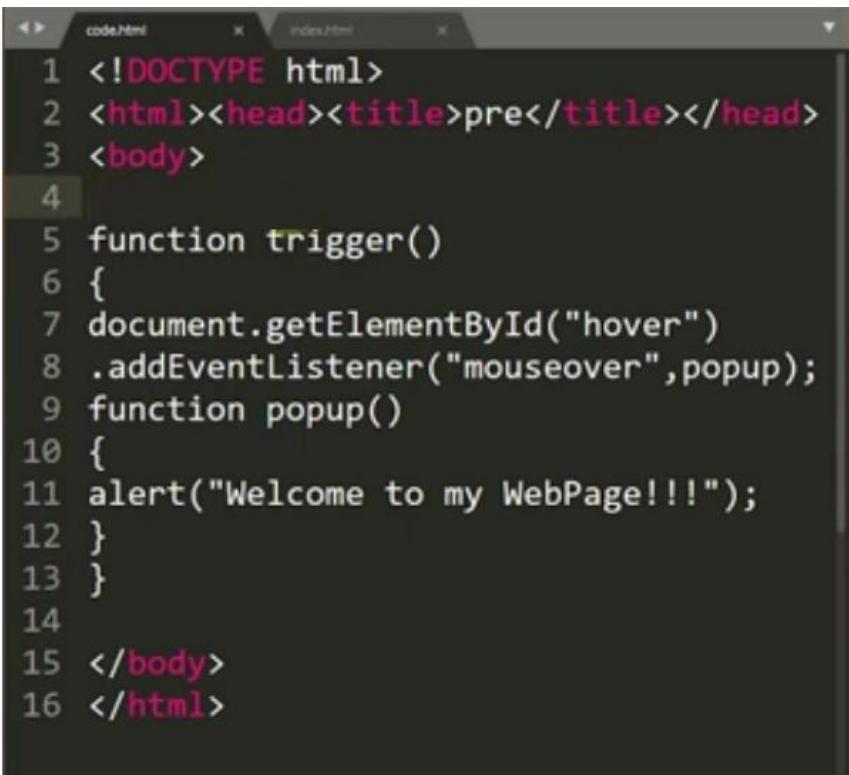
HTML Code

```
<!DOCTYPE html>
<html>
<body>
<b>This text is bold.</b>
<br/>
<i>This text is Italic.</i>
<br/>
<strong>This text is strong</strong>
<br/>
This is <sup>superscripted</sup> text.
</body>
</html>
```

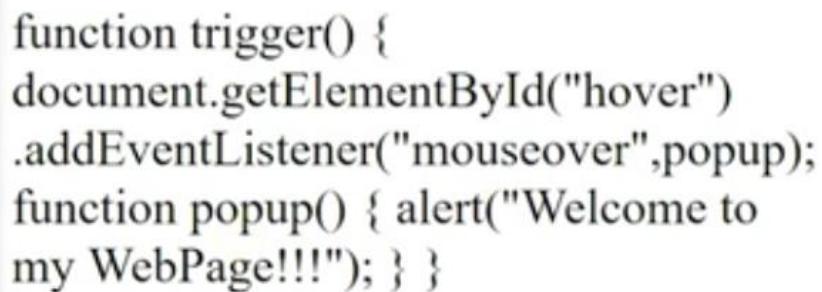
Output

This text is bold.
This text is Italic.
This text is strong
This is superscripted text.

<pre> tag



```
1 <!DOCTYPE html>
2 <html><head><title>pre</title></head>
3 <body>
4
5 function trigger()
6 {
7 document.getElementById("hover")
8 .addEventListener("mouseover",popup);
9 function popup()
10 {
11 alert("Welcome to my WebPage!!!");
12 }
13 }
14
15 </body>
16 </html>
```



```
function trigger() {
document.getElementById("hover")
.addEventListener("mouseover",popup);
function popup() { alert("Welcome to
my WebPage!!!!"); } }
```



```
1 <!DOCTYPE html>
2 <html><head><title>pre</title></head>
3 <body>
4 <pre>
5 function trigger()
6 {
7 document.getElementById("hover")
8 .addEventListener("mouseover",popup);
9 function popup()
10 {
11 alert("Welcome to my WebPage!!!");
12 }
13 }
14 </pre>
15 </body>
16 </html>
```

```
function trigger()
{
document.getElementById("hover")
.addEventListener("mouseover",popup);
function popup()
{
alert("Welcome to my WebPage!!!");
}
}
```

Also can be used to display other contents as it is, Eg.any application or mail contents, etc.

Hyperlinks: <a> Tag

- Link to a document called `form.html` on the same server in the same directory:

```
<a href="form.html">Fill Our Form</a>
```

- Link to a document called `parent.html` on the same server in the parent directory:

```
<a href=".../parent.html">Parent</a>
```

- Link to a document called `cat.html` on the same server in the subdirectory `stuff`:

```
<a href="stuff/cat.html">Catalog</a>
```

- Link to an external Web site:

```
<a href="http://www.devbg.org" target="_blank">BASD</a>
```

Links to the Same Document – Example

```
<h1>Table of Contents</h1>
```

```
<p><a href="#section1">Introduction</a><br />
```

```
<a href="#section2">Some background</A><br />
```

```
<a href="#section2.1">Project History</a><br />
```

```
<!-- The document text follows here -->
```

```
<h2 id="section1">Introduction</h2>
```

```
... Section 1 follows here ...
```

```
<h2 id="section2">Some background</h2>
```

```
... Section 2 follows here ...
```

```
<h3 id="section2.1">Project History</h3>
```

```
... Section 2.1 follows here ...
```

Images: tag

- ◆ Inserting an image with tag:

```

```

- ◆ Image attributes:

src	Location of image file (relative or absolute)
alt	Substitute text for display (e.g. in text mode)
height	Number of pixels of the height
width	Number of pixels of the width
border	Size of border, 0 for no border

- ◆ Example:

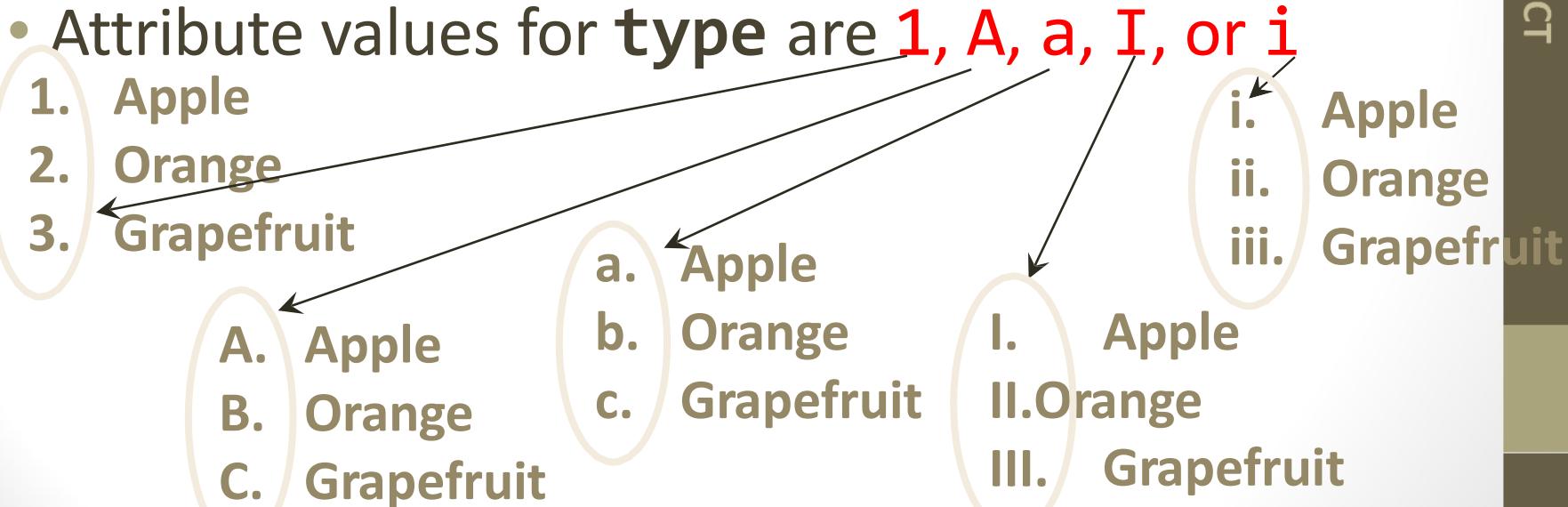
```

```

Ordered Lists: Tag

- Create an Ordered List using :

```
<ol type="1">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ol>
```



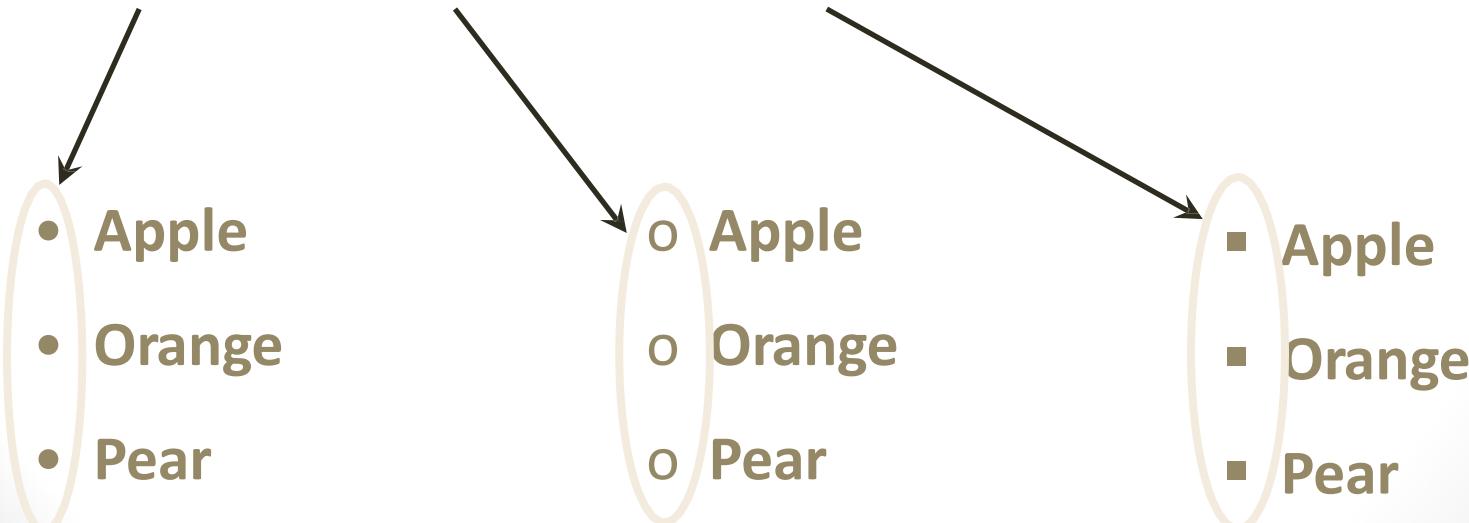
Unordered Lists: Tag

- Create an Unordered List using :

```
<ul type="disk" >  
  <li>Apple</li>  
  <li>Orange</li>  
  <li>Grapefruit</li>  
</ul>
```

- Attribute values for **type** are:

- **disc, circle or square**



Definition lists: <dl> tag

- Create definition lists using <dl>
 - It stores pairs of text or terms and its associated definition
 - Text is in <dt> tag and its definition is in <dd> tag

```
<dl>
  <dt>HTML</dt>
  <dd>It is a markup language...</dd>
  <dt>CSS</dt>
  <dd>Language used to design...</dd>
</dl>
```

- Definition is automatically indented



file:///C:/Users/Vijayend

HTML:

It is a markup language...

CSS:

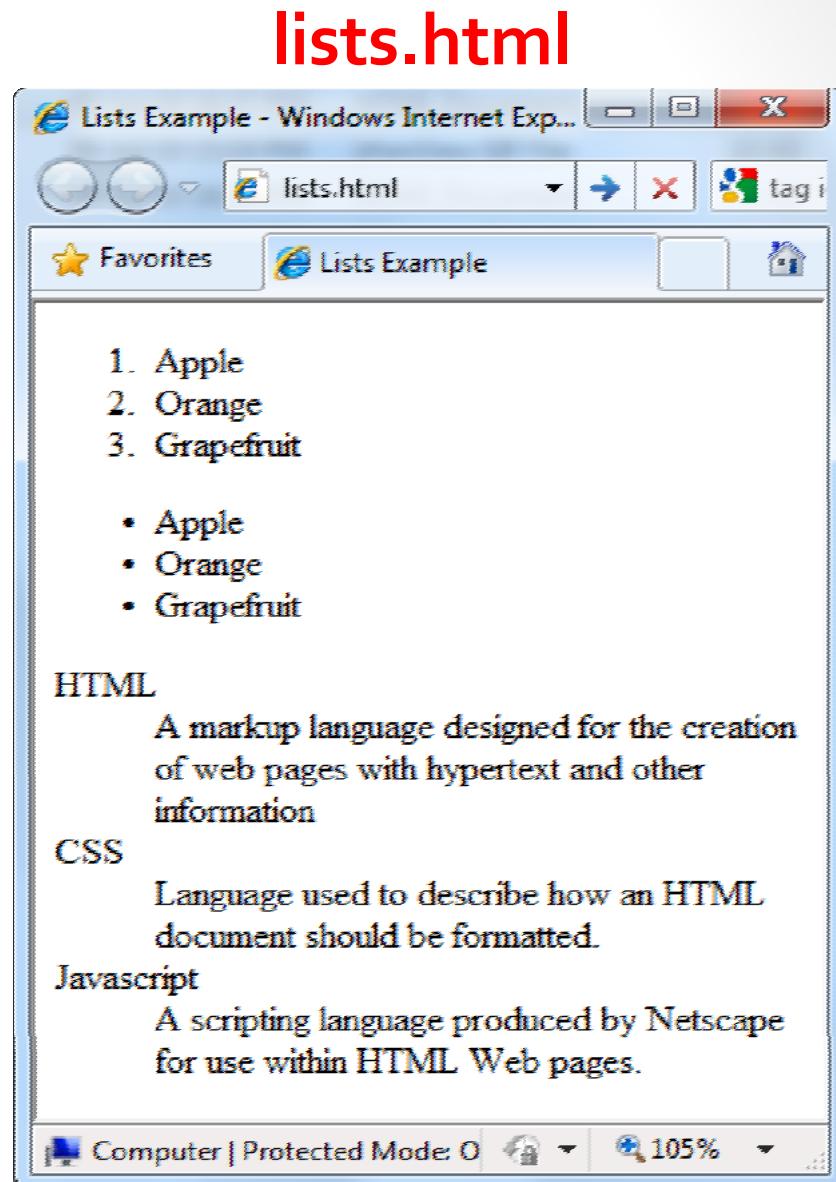
Language used to design...

Lists – Example

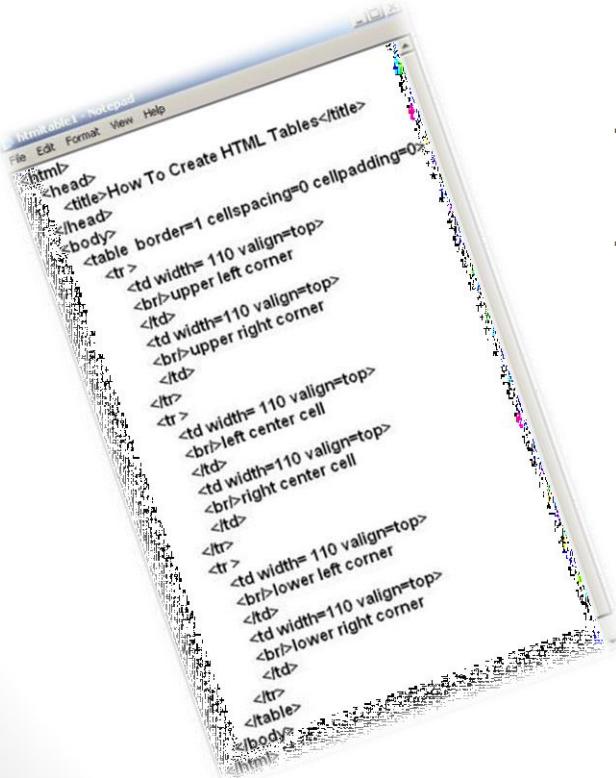
```
<ol type="1">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ol>

<ul type="disc">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ul>

<dl>
  <dt>HTML</dt>
  <dd>A markup lang...</dd>
</dl>
```



HTML Tables



Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data

HTML Table Tags

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

HTML Tables

- Start and end of a table

```
<table> ... </table>
```

- Start and end of a row

```
<tr> ... </tr>
```

- Start and end of a cell in a row

```
<td> ... </td>
```

Simple HTML Tables – Example

```
<html>
<body>

<table width=100% border = "1" bgcolor = "yellow">
<tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
</tr>
<tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
</tr>
</table>

</body>
</html>
```

Firstname	Lastname	Age
Jill	Smith	50

Continued...

- If width attribute is not set then it takes default width according to content.
- Attribute values of width:
 - pixels: It sets the width of table in terms of pixels.
 - %: It sets the width of table in terms of percentage

```
<table>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$80</td>
    </tr>
  </tbody>
```

```
<tfoot>
  <tr>
    <td>Sum</td>
    <td>$180</td>
  </tr>
</tfoot>
</table>
```

Month	Savings
January	\$100
February	\$80
Sum	\$180

```

<table>
  <colgroup>
    <col span="2" style="background-color: red; text-align: center">
    <col style="background-color: yellow">
  </colgroup>
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>$53</td>
  </tr>
  <tr>
    <td>5869207</td>
    <td>My first CSS</td>
    <td>$49</td>
  </tr>
</table>

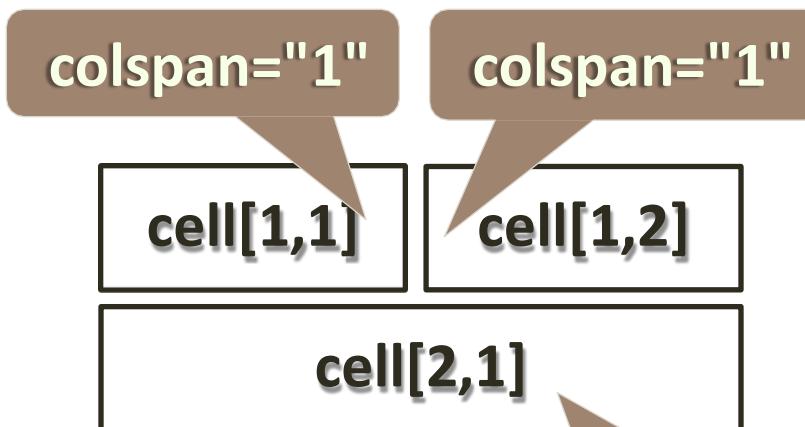
```

ISBN	Title	Price
3476896	My first HTML	\$53
5869207	My first CSS	\$49

Column and Row Span

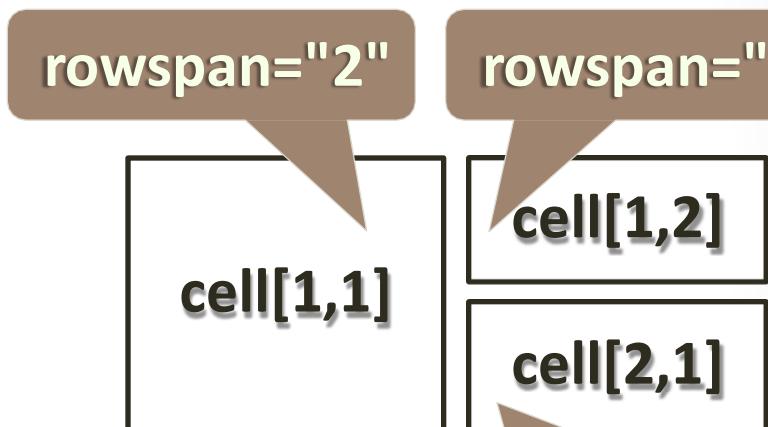
- Table cells have two important attributes:

◆ **colspan**



◆ **Defines how many columns the cell occupies**

◆ **rowspan**



◆ **Defines how many rows the cell occupies**

HTML Tables – colspan Example

```
<html>
<body>

<table width=50% border = "1" >
  <tr>
    <th colspan=2>Firstname</th>
    <th>Age</th>
  </tr>

  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
</table>

</body>
</html>
```

Firstname		Age
Jill	Smith	50

HTML Tables – rowspan Example

```
<h2 align=center> Cell that spans two rows: </h2>
<table style="width:50%" border="1" align=center>
<tr>
<th>Name:</th>
<td>Bill Gates</td>
</tr>
<tr>
<th rowspan="2">Telephone:</th>
<td>55577854</td>
</tr>
<tr>
<td>55577855</td>
</tr>
</table>
```

Cell that spans two rows:

Name:	Bill Gates
Telephone:	55577854
	55577855



HTML Forms

Entering User Data from a Web Page

Registration Form

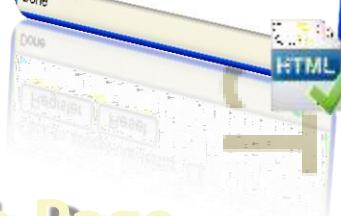
User name:

Password:

Gender: Male Female

Click to accept our terms:

Register Reset



HTML Form

- The **<form>** Tag: Popular way of interacting with users
- The HTML **<form>** tag defines a form that is used to collect user inputs:
- **Syntax**

```
<form>  
    form elements  
</form>
```

- **Form elements** are different types of **input elements**, like
 - text fields, password field, email field, dropdown list, etc.
 - checkboxes,
 - radio buttons,
 - submit buttons, and more...

The <input> Element

- The **<input> tag** element is the most important form element.
- Here are some examples:
 - <input type="text">
 - <input type="password">
 - <input type="submit">
 - <input type="button">
 - <input type="radio">
 - <input type="reset">
 - <input type="checkbox">
 - ...etc

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

Input type Text

- <input type="text">
It defines a **one-line text input field**:
- Example
- This is how the above HTML code will be displayed in a browser:

```
<form>  
First name:<br>  
<input type="text" name="firstname"><br>  
Last name:<br>  
<input type="text" name="lastname">  
</form>
```

First name:

Last name:

Input Type Password

- <input type="password">
- defines a **password field**:
- Example

```
<form>  
  User name:<br>  
  <input type="text" name="username"><br>  
  User password:<br>  
  <input type="password" name="psw">  
</form>
```

User name:

User password:

The characters in a password field are masked (shown as asterisks or circles).

Input type Submit

- It defines a button for **submitting the form data** to a **form-handler**.
- The form-handler is specified in the **form's action attribute**:
- **Example**

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

First name:


```
<input type="text" name="firstname" value="Mickey" placeholder="Type  
name here"> <br>
```

Last name:


```
<input type="text" name="lastname" value="Mouse" placeholder= "Type name  
here"><br><br>
```

```
<input type="submit" value="Submit">  
</form>
```

The image shows a simple HTML form. It contains two text input fields: one for the first name with the placeholder "Type name here" and the value "Mickey", and another for the last name with the placeholder "Type name here" and the value "Mouse". Below these fields is a single "Submit" button.

First name:	Mickey
Last name:	Mouse
Submit	

Input Type Reset

- **<input type="reset">** defines a **reset button** that will reset all form values to their default values:
- Just click the "Reset" button and the form-data will be reset.
- Example

```
<form action="/action_page.php">
First name:<br>
<input type="text" name="firstname" >
<br>
Last name:<br>
<input type="text" name="lastname" >
<br><br>
<input type="submit" value="Submit">
<input type="reset">
</form>
```

First name:

Last name:

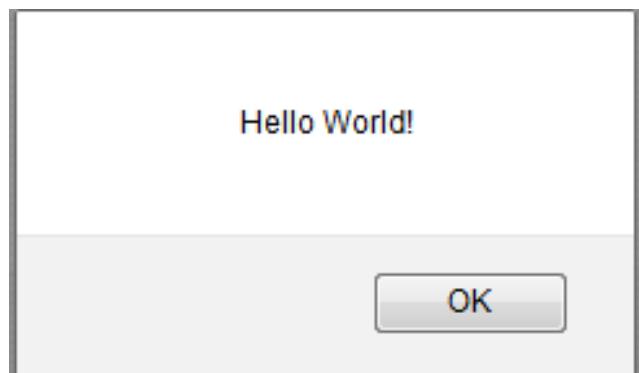
Submit Reset

Input Type Button

- <input type="button"> defines a **button**:
- Example
- <input type="button" value="Click Me!" onclick="alert('Hello World!')">



- After clicking above button it shows output as below:



Input Type Radio

- `<input type="radio">` defines a **radio button**.
- Radio buttons let a user **select ONLY ONE** of a limited number of choices:
- `<form>`
`<input type="radio" name="gender" value="male" checked> Male
`
`<input type="radio" name="gender" value="female"> Female
`
`<input type="radio" name="gender" value="other"> Other`
`</form>`
- This is how the HTML code above will be displayed in a browser:

Male
 Female
 Other

Input Type Checkbox

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

```
<form>
<input type="checkbox" name="vehicle1" value="Bike"> I have a bike
<br>
<input type="checkbox" name="vehicle2" value="Car"> I have a car
</form>
```

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

I have a bike
 I have a car

HTML Input Attributes

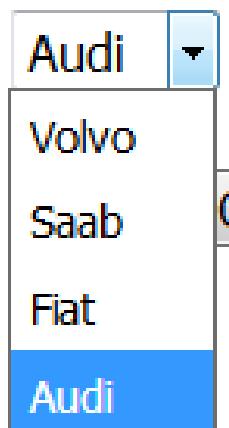
- **The value Attribute**
 - Holds the current value of the <input> element's
- **The readonly Attribute:**
 - Doesn't allow to edit/change the <input> element's value
- **The disabled Attribute:**
 - The disabled attribute can be set to keep a user from using the <input> element until some other condition has been met (like selecting a checkbox, etc.).
 - Then, a JavaScript could remove the disabled value and make the <input> element usable.

Continued..

- The **size** Attribute:
 - It specifies the *visible width*, in *characters*, of an `<input>` element.
 - default size is 20
 - (used only for types- text, search, tel, url, email, and password)
- The **maxlength** Attribute:
 - It specifies the *maximum number of characters allowed* in the `<input>` element.

The <select> Element (Dropdown menus)

- The <select> element defines a **drop-down** list:
- ```
<select name="cars">
 <option value="volvo">Volvo</option>
 <option value="saab">Saab</option>
 <option value="fiat">Fiat</option>
 <option value="audi">Audi</option>
</select>
```



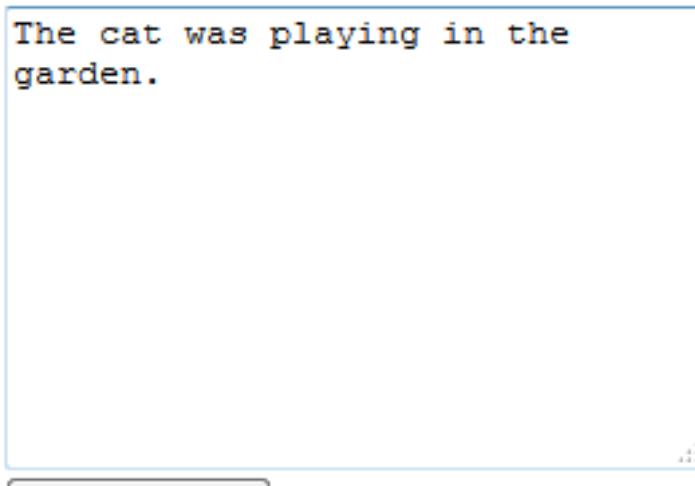
# The <select> Element with multiple selection

- <select name="cars" size="4" multiple>  
    <option value="volvo">Volvo</option>  
    <option value="saab">Saab</option>  
    <option value="fiat">Fiat</option>  
    <option value="audi">Audi</option>  
  </select>



# The <textarea> Element

- `<textarea name="message" rows="10" cols="30">`  
The cat was playing in the garden.  
`</textarea>`



The cat was playing in the  
garden.

- The **rows** attribute specifies the **visible number of lines** in a text area.
- The **cols** attribute specifies the **visible width** of a text area.

# HTML Forms – Example

## form.html

```
<form method="post" action="apply_now.jsp">
 <input name="subject" type="hidden" value="Class" />
 <fieldset><legend>Academic information</legend>
 <label for="degree">Degree</label>
 <select name="degree" id="degree">
 <option value="BA">Bachelor of Art</option>
 <option value="BS">Bachelor of Science</option>
 <option value="MBA" selected="selected">Master of
 Business Administration</option>
 </select>

 <label for="studentid">Student ID</label>
 <input type="password" name="studentid" />
 </fieldset>
 <fieldset><legend>Personal Details</legend>
 <label for="fname">First Name</label>
 <input type="text" name="fname" id="fname" />

 <label for="lname">Last Name</label>
 <input type="text" name="lname" id="lname" />
```

# HTML Forms – Example (2)

## form.html (continued)

```


 Gender:
 <input name="gender" type="radio" id="gm" value="m" />
 <label for="gm">Male</label>
 <input name="gender" type="radio" id="gf" value="f" />
 <label for="gf">Female</label>

 <label for="email">Email</label>
 <input type="text" name="email" id="email" />
</fieldset>
<p>
 <textarea name="terms" cols="30" rows="4"
 readonly="readonly">TERMS AND CONDITIONS...</textarea>
</p>

 <input type="submit" name="submit" value="Send Form" />
 <input type="reset" value="Clear Form" />
</form>
```

# HTML Forms – Example(3)

## form.html (continued)

The screenshot shows a Mozilla Firefox window displaying an HTML form. The title bar reads "HTML Forms Example - Mozilla Firefox". The menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The toolbar includes Back, Forward, Stop, Home, and search fields for file:///C:/work/D... and Google.

**Academic information**

Degree: Master of Business Administration

Student ID: [Input field]

Classes attended:

- Geography
- Mathematics
- English

**Personal Details**

First Name: [Input field]

Last Name: [Input field]

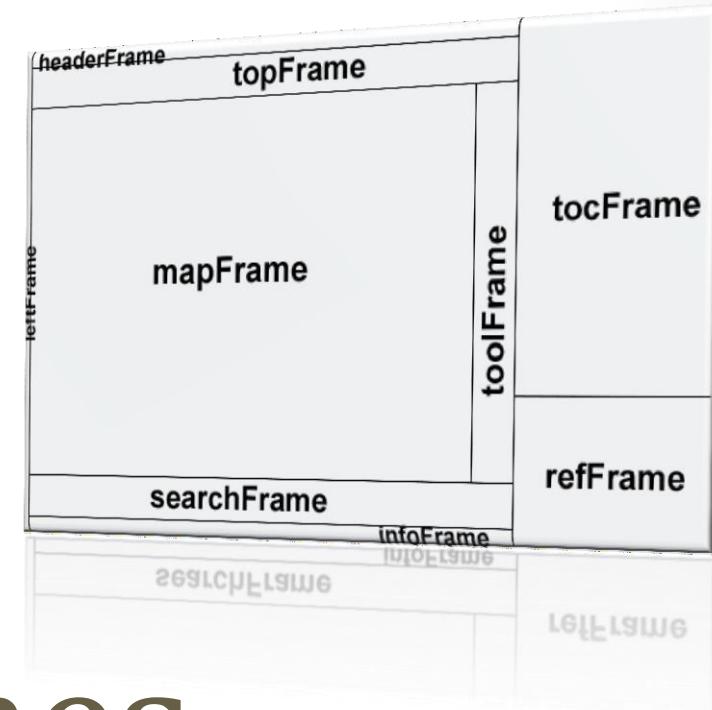
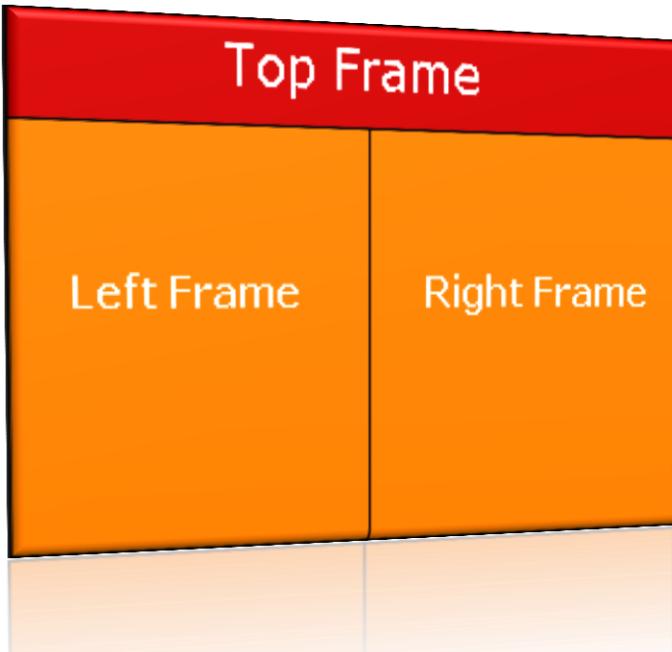
Gender:  Male  Female

Email: [Input field]

**TERMS AND CONDITIONS...**

**Buttons:** Send Form, Clear Form

At the bottom, status bars show "Done", "Fiddler: Disabled", "0 errors / 0 warnings", and "ra s. Gaikwad, PICT".



# HTML Frames

<frameset>, <frame> and <iframe>

# HTML Frames

- Frames provide a way **to show multiple HTML documents in a single Web page.**
- The web-page can be split into separate views (i.e. frames) horizontally and vertically.
- Frames were popular in the early days of HTML development, but now their usage is rejected.
- Frames are no more supported by all user agents (i.e. browsers, search engines, etc.)

# HTML <frame> Tag

- **Example:**

A simple three-framed page:

- ```
<frameset cols="25%, 50%, 25%">
    <frame src="frame_a.html">
    <frame src="frame_b.html">
    <frame src="frame_c.html">
</frameset>
```



- Each <frame> in a <frameset> can have different attributes, such as border, scrolling, the ability to resize, etc.

HTML

HTML5

Doctype declaration in Html is too longer

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

character encoding declaration in Html is also longer <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

No concept of **media** in case of HTML

HTML does not allow drag and drop effects

It is almost impossible to get true **GeoLocation of users** browsing any website **especially when it comes to mobile devices.**

HTML offers local storage instead of cookies.

Not possible to draw any shapes like circle, rectangle, triangle, etc.

Allows use of **browser cache** as the temporary storage

DOCTYPE declaration in Html5 is very simple "<!DOCTYPE html>

character encoding (charset) declaration is also very simple <meta charset="UTF-8">

Audio and Videos are **integral part of HTML5** e.g. <audio> and <video> tags.

HTML5 allows drag and drop effects

JS GeoLocation API in HTML5 helps identify location of users browsing any website (provided user allows it)

HTML5 uses cookies to store data.

Using the **<canvas>** element of HTML5 you can draw shapes like circle, rectangle, triangle via scripts.

HTML5 can use **application cache, web SQL database and web storage**



Cascading Style Sheets (CSS)

```
171 #content .article img.left.border {  
172     padding: 0 9px 9px 0;  
173     border-right: 1px dotted #999;  
174     border-bottom: 1px dotted #999; }  
175 #content .article blockquote {  
176     margin-left: 10px;  
177     padding-left: 10px;  
178     border-left: 3px solid #252525; }  
179 #content .article ul {  
180     padding-left: 1em;  
181     list-style-type: circle; }
```

Introduction of CSS

- Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of **making web pages presentable**.
- **CSS handles the look and feel part of a web page.**
- **Using CSS, you can control :**
 - the color of the text,
 - the style of fonts,
 - the spacing between paragraphs,
 - how columns are sized and laid out,
 - what background images or colors are used,
 - variations in display for different devices and screen sizes,
 -and variety of other effects

Advantages of CSS

- CSS saves time
- Pages load faster
- Easy maintenance
- Superior styles to HTML
- Multiple Device Compatibility
- Global web standards
- Platform Independence

CSS - Syntax

- A **CSS comprises of style rules** that are interpreted by the browser and then applied to the corresponding elements in your document.
- **Style rule is made of three parts –**
- **Selector** – A selector is an HTML tag at which a style will be applied. This could be any tag like `<p>`, `<h1>` or `<table>` etc.
- **Property** - A property is a type of attribute of the selected HTML tag. Put simply, **all the HTML attributes are converted into CSS properties**. They could be *color*, *border* etc.
- **Value** - Values are assigned to these properties. For example, *color* property can have value either *red* or *#F1F1F1*

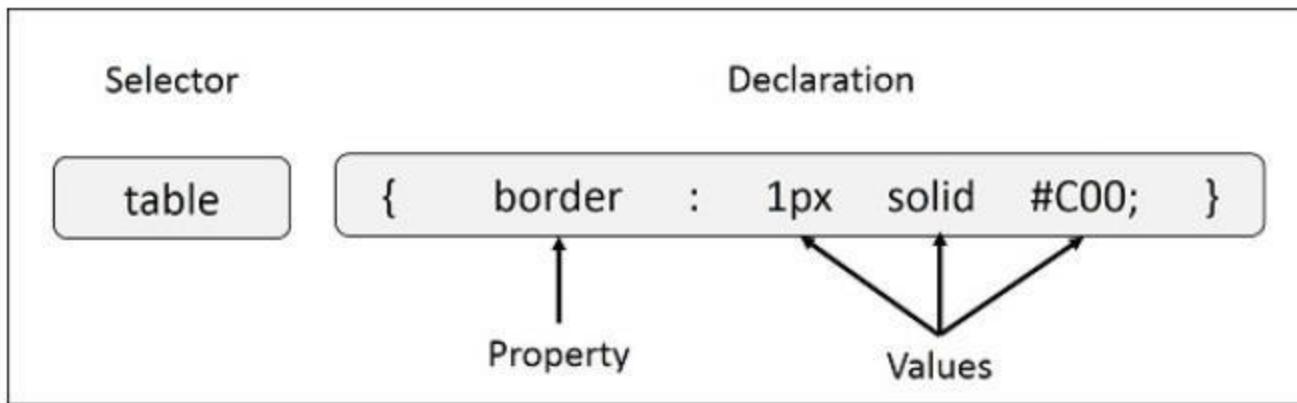
CSS - Syntax

- **Syntax:**

```
selector { property: value }
```

- **Example:**

```
table{ border :1px solid black; }
```



CSS selectors (1)

- CSS selectors are used for "finding" (or selecting) HTML elements **based on their *element name, id, class, attribute*** and more.
- **The element Selector:**
 - **The element selector selects elements based on the element's name.**
 - You can select all `<p>` elements on a page in following way:
(in this case, all `<p>` elements will be center-aligned, with a red text color):
 - **Example:**
 - `p {
 text-align:
 center;
 color: red;
}`

Example with output

HTML Code with CSS

```
<html>
<head>
<style>
p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

Output

Every paragraph will be affected by the style.

Me too!

And me!

CSS selectors (2)

- **The *id* Selector:**
- The **id selector** uses the **id attribute** of an HTML element **to select a specific element.**
- The ***id*** of an element should be unique within a page, so that the ***id*** selector is used to select one unique element.
- To select an element with a specific ***id***, **write a hash (#)** character, followed by the ***id*** of the element.
- The style rule below will be applied to the HTML element with **id="para1":**
- **Example**
- **#para1 {**
 text-align: center;
 color: red;
}

Example with output

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>
```

HTML Code with CSS



Hello World!

This paragraph is not affected by the style.

Output

CSS selectors (3)

- **The *class* Selector:**
- The class selector selects elements with a specific class attribute.
- **To select elements with a specific class, write a period (.) character, followed by the name of the class.**
- In the example below, all HTML elements with class="abc" will be red and center-aligned:
- **Example**
- ```
.abc{
 text-align: center;
 color: red;
}
```

# Example with output

## HTML Code with CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
 text-align: center;
 color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```

## Output

**Red and center-aligned heading**

Red and center-aligned paragraph.

# CSS selectors (4)

- **The tag+class Selector continued....**
- You can also specify that only specific HTML elements should be affected by a class.
- In the example below, only `<p>` elements with `class="myclass"` will be in red and center-aligned:
- **Example**

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

# CSS selectors (5)

- **Grouping Selectors:**
- If you have HTML elements with the same style definitions, like this:

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

- then, it will be better to group the selectors, to minimize the code:

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

# Example with output

## HTML Code with CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
 text-align: center;
 color: red;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>

</body>
</html>
```

## Output

Hello World!  
Smaller heading!  
This is a paragraph.

# CSS selectors (6)

- **The attribute Selector:**
- It is used to **select HTML elements with a specified attribute**:
- Example:

```
a[target] {
 color: red;
}
```

```
<html>
<style>
 a[target] {
 color: red;
 }
</style>
```

```
<h1>
 Click1

 Click2

</h1>
</html>
```



file:///C:/Users/Vijayendra/Desktop

**Click1** **Click2**

# Insert CSS in HTML

- **Three Ways to Insert CSS:**
  1. External style sheet
  2. Internal style sheet
  3. Inline style

# Internal Style Sheet

- An internal style sheet may be used if one single page has a unique style.
- Internal styles are defined within the `<style>` element, inside the `<head>` section of an HTML page:
- Example

```
<head>
<style>
body {
 background-color: linen;
}

h1 {
 color: maroon;
 margin-left: 40px;
}
</style>
</head>
```

# External Style Sheet

- With an external style sheet, you can change the look of an entire website by changing just one file.
  - Each page must just include a reference to the external style sheet file inside the `<link>` element.
  - The `<link>` element goes inside the `<head>` section.
- 
- Example:**
  - `<head>`  
`<link rel="stylesheet" type="text/css" href="mystyle.css">`  
`</head>`

# External Style Sheet - Example

HTML Code

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

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

</body>
</html>
```

CSS File named- mystyle.css

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

h1 {
 color: navy;
 margin-left: 20px;
}
```

# Inline Styles

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant HTML element. The style attribute can contain any CSS property.
- The example below shows how to change the color and the left margin of a <h1> element:
- **Example**

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

<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

# Text-related CSS Properties

- **color** – specifies the color of the text
- **font-size** – size of font: **xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger** or numeric value
- **font-family** – comma separated font names
  - Example: **verdana, sans-serif**, etc.
  - The browser loads the first one that is available
  - There should always be at least one generic font
- **font-weight** can be **normal, bold, bolder, lighter** or a number in range [100 ... 900]

# CSS Rules for Fonts (2)

- **font-style** – styles the font
  - Values: **normal, italic, oblique**
- **text-decoration** – decorates the text
  - Values: **none, underline, line-through, overline, blink**
- **text-align** – defines the alignment of text or other content
  - Values: **left, right, center, justify**

# Short-hand for font Property

- **font**
  - Short-hand rule for setting multiple font properties at the same time :

```
font:italic red bold 12px/16px verdana
```

Above shorthand is equal to writing this:

```
font-style: italic;
color: red;
font-weight: bold;
font-size: 12px;
line-height: 16px;
font-family: verdana;
```

# Backgrounds

- **background-image:** `url("back.gif");`
  - URL of image to be used as background
- **background-color:** color name | hex color value
  - Used to specify background color of any HTML element
- **background-repeat:** repeat-x | repeat-y | repeat | no-repeat
  - used for repeating the small images
- **background-attachment:** fixed / scroll
  - Used to fix the background image while scrolling through the web page

# continued...

- **background-position:** specifies vertical and horizontal position of the background image.
  - **Vertical position:** top, center, bottom
  - **Horizontal position:** left, center, right
- Both can be specified in percentage or pixel values also:

**background-position: top left;**

**background-position: 5px 50px;**

# Borders

- **border-width:** thin, medium, thick or numeric value in pixels (e.g. 10px)
- **border-color:** color alias or RGB value
- **border-style:** none, hidden, dotted, dashed, solid, double, groove, ridge, etc.
- **Each property can be defined separately for left, top, bottom and right side borders**  
**border-top-style**, **border-left-color**, ...

# Border Short-hand Property

- Shorthand rule for setting border properties at once:

**border: 1px solid red**

is equal to writing:

**border-width:1px;  
border-color:red;  
border-style:solid;**

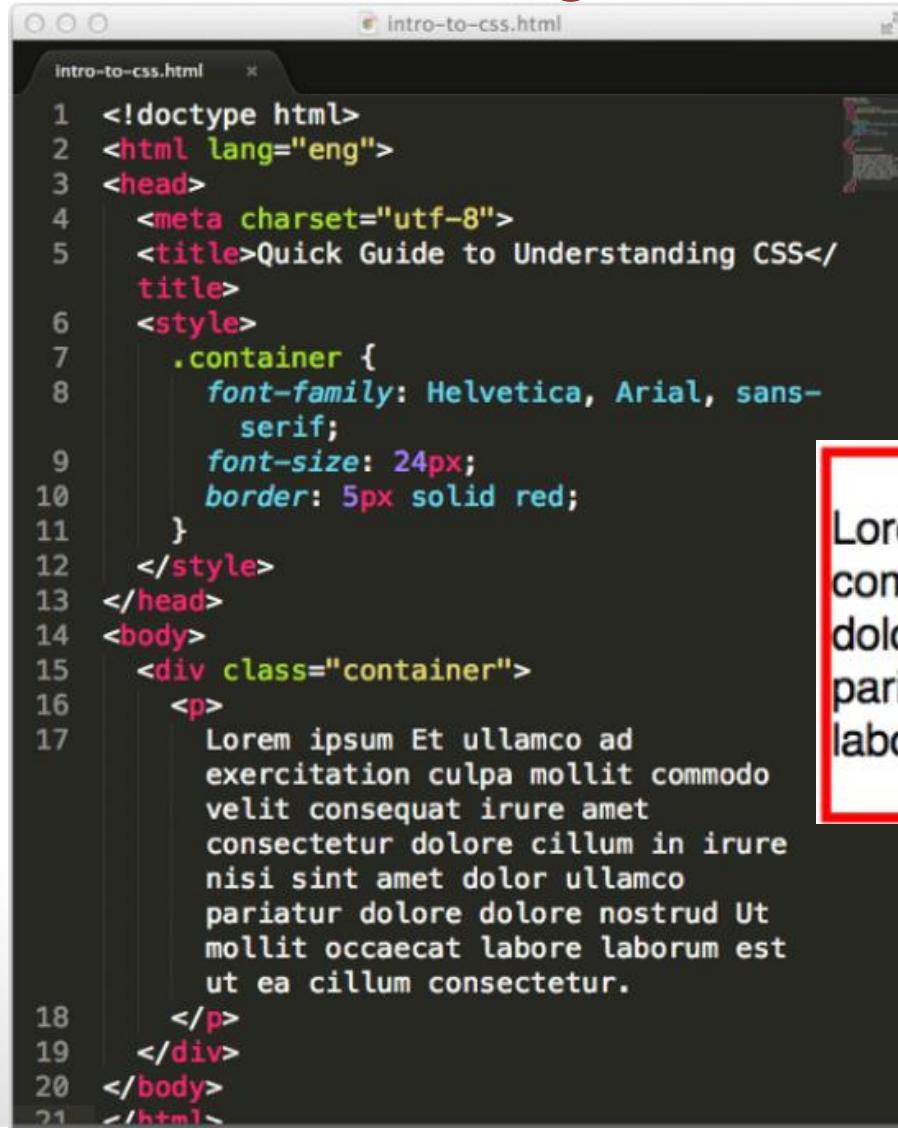
- Specify different borders for the sides via shorthand rules:  
border-top, border-left, border-right, border-bottom
- When to avoid border:0

# Width and Height of elements

- **width** – defines numerical value for the **width of an element**,  
e.g. 200px
- **height** – defines numerical value for the **height of an element**,  
e.g. 100px
- By default the height of an element is defined by its contents

# Style rule cascading and inheritance

- CSS inheritance is not different than what we understand through the term “inheritance”.



```
intro-to-css.html
1 <!doctype html>
2 <html lang="eng">
3 <head>
4 <meta charset="utf-8">
5 <title>Quick Guide to Understanding CSS</
 title>
6 <style>
7 .container {
8 font-family: Helvetica, Arial, sans-
9 serif;
10 font-size: 24px;
11 border: 5px solid red;
12 }
13 </style>
14 </head>
15 <body>
16 <div class="container">
17 <p>
18 Lorem ipsum Et ullamco ad exercitation culpa mollit
19 commodo velit consequat irure amet consectetur
20 dolore cillum in irure nisi sint amet dolor ullamco
21 pariatur dolore dolore nostrud Ut mollit occaecat
22 labore laborum est ut ea cillum consectetur.
23 </p>
24 </div>
25 </body>
26 </html>
```

Lorem ipsum Et ullamco ad exercitation culpa mollit  
commodo velit consequat irure amet consectetur  
dolore cillum in irure nisi sint amet dolor ullamco  
pariatur dolore dolore nostrud Ut mollit occaecat  
labore laborum est ut ea cillum consectetur.

- CSS class called “container” has associated styles with it (namely, font-family, font-size, and border).
- We applied the “container” class to a `<div>` tag, and within the `<div>` tag, we’ve included a `<p>` tag.
- So, the text inside the `<p>` tag is getting styled with the font-family and font-size styles, etc. even though the “container” class is applied only to the `<div>` tag.
- This happens due to CSS inheritance i.e. because the `<p>` tag is a child tag of the `<div>` tag, it inherited those styles as well.

- There is border on only the <div> tag **but not on <p> tag.** Why..??
- So, not all CSS styles are inheritable from parent to children.
- In this case, the “border” style is one of those styles that doesn’t get automatically inherited from parent tag to child tags.

# Style rule cascading

- Cascading refers to the fact that cumulative styles across multiple CSS rules are applied to each and every HTML tag.
  - **CSS styles are directly applied to a particular tag.**

```
6 <style>
7 p {
8 font-family: Helvetica, Arial, sans-
9 serif;
10 font-size: 14px;
11 }
12 .special-paragraph {
13 font-size: 24px;
14 color: red;
15 }
16 </style>
17 </head>
18 <body>
19 <p>
20 Lorem ipsum Et ullamco ad exercitation
21 culpa mollit commodo velit consequat
22 irure amet consectetur dolore cillum
23 in irure nisi sint amet dolor.
24 </p>
25 <p class="special-paragraph">
26 Lorem ipsum Ut Excepteur nulla Ut ea
27 sint Excepteur nisi proident in
28 reprehenderit.
29 </p>
30 </body>
```

**Et ullamco ad exercitation culpa mollit commodo velit consequat irure  
amet consectetur dolore cillum in irure nisi sint amet dolor.**

**Ut Excep**teur nulla Ut ea sint  
Excep**teur nisi proident in reprehenderit.******

One rule is applied for all `<p>` tags and another for tags with the “special-paragraph” is class.

**Both** `<p>` tags have the “font-family” and “font-size” styles applied to it.

But the `<p>` tag with the “special-paragraph” class applied to it **overrode** the “font-size: 14px” style from the previous CSS rule with its own “font-size: 24px” style

In addition to overriding a style, the “special-paragraph” class gave the **second `<p>` tag an additional “color: red” style.**

“Cascading” part of CSS comes when there are **multiple styles** that are **directly applied to a HTML tag.**

So, in the case of multiple styles, the actual final style applied to a HTML tag is the most “**specific**” one.

# Bootstrap

- Bootstrap is the most popular **HTML, CSS** and **JavaScript framework** for developing **responsive, mobile-first** websites and web applications.

## Why use Bootstrap?

- Increase development speed
- Assure responsiveness
- Prevent repetition between projects
- Add consistency
- Ensure cross browser compatibility
- Large community
- Customizable

## What is Responsive Web Design?

Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

# Where to Get Bootstrap?

There are two ways to start using Bootstrap on your own website:

## 1. Download Bootstrap from [getbootstrap.com](http://getbootstrap.com):

If you want to download and host Bootstrap yourself, go to [getbootstrap.com](http://getbootstrap.com), and follow the instructions there.

## 2. Include Bootstrap from a CDN :

### MaxCDN:

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
 href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->
<script
 src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
```

# Create First Web Page With Bootstrap

## 1. Add the HTML5 doctype:

- Bootstrap uses HTML elements and CSS properties that require the HTML5 doctype.
- Always include the HTML5 doctype at the beginning of the page, along with the *lang* attribute and the correct character set:

```
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8">
 </head>
 </html>
```

# Continued...

## 2. Bootstrap-3 is mobile-first:

- Bootstrap-3 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.
- To ensure proper rendering and touch zooming, add the following `<meta>` tag inside the `<head>` element:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1` part sets the initial zoom level when the page is first loaded by the browser.

# Continued...

## 3. Containers

Bootstrap also requires a containing element to wrap site contents.

There are two container classes to choose from:

1. The `.container` class provides a responsive **fixed width container**
2. The `.container-fluid` class provides a **full width container**, spanning the entire width of the viewport

# Basic Bootstrap Page

```
<!DOCTYPE html>
<html lang="en">
<head>
 <title>Bootstrap Example</title>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="stylesheet"
 href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
 <script
 src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
 <script
 src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>
```

```
<div class="container">
 <h1>My First Bootstrap Page</h1>
 <p>This is some text.</p>
</div>
</body>
</html>
```

# Bootstrap Tables

## Bootstrap Basic Table

A basic Bootstrap table has a light padding and only horizontal dividers.

The `.table` class adds basic styling to a table:

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

```
<div class="container">
 <h2>Basic Table</h2>
 <p>The .table class adds basic styling (light
padding and only horizontal dividers) to a
table:</p>
 <table class="table">
 <thead>
 <tr>
 <th>Firstname</th>
 <th>Lastname</th>
 <th>Email</th>
 </tr>
 </thead>
 <tbody>
 <tr>
 <td>John</td>
 <td>Doe</td>
 <td>john@example.com</td>
 </tr>
 <tr>
 <td>Mary</td>
 <td>Moe</td>
 <td>mary@example.com</td>
 </tr>
 <tr>
 <td>July</td>
 <td>Dooley</td>
 <td>july@example.com</td>
 </tr>
 </tbody>
 </table>
</div>
```

# Striped Rows

The `.table-striped` class adds zebra-stripes to a table:

```
<table class="table table-striped">
```

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

# Bordered Table

The `.table-bordered` class adds borders on all sides of the table and cells:

```
<table class="table table-bordered">
```

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

# Bootstrap Alerts

## Alerts

Bootstrap provides an easy way to create predefined alert messages:

**Success!** This alert box indicates a successful or positive action. ×

**Info!** This alert box indicates a neutral informative change or action. ×

**Warning!** This alert box indicates a warning that might need attention. ×

**Danger!** This alert box indicates a dangerous or potentially negative action. ×

Alerts are created with the `.alert` class, followed by one of the four contextual classes `.alert-success`, `.alert-info`, `.alert-warning` or `.alert-danger`:

```
<div class="alert alert-success">
 Success! Indicates a successful or positive action.
</div>
```

```
<div class="alert alert-info">
 Info! Indicates a neutral informative change or action.
</div>
```

```
<div class="alert alert-warning">
 Warning! Indicates a warning that might need attention.
</div>
```

```
<div class="alert alert-danger">
 Danger! Indicates a dangerous or potentially negative
action.
</div>
```

# For complete Bootstrap tutorials:

<https://www.w3schools.com/bootstrap/>