# Introduction

- Web technology is related to the interface between web servers and their clients.
- Web technology incorporates tools and techniques for web development.
- Web Development is a broad term for the work involved in developing a web site for World Wide Web. This can include *web design, web content development, client-side/server-side scripting, web server, e-commerce development,* & many more.
- Web Development usually refers to the main non-design aspects of building websites: writing markup and coding.
- Web Development can range from simplest static single page of plain text to the most complex web-based internet applications, electronic businesses, or social network services.

# Introduction

- Web Design is a broad term used to encompass the way that content is delivered to an end-user through the World Wide Web, using a web browser.
- The intent of web design is to create a website – a collection of online content including documents and applications that reside on a web servers.
- A website may include text, images, sounds, and other content, and may be interactive.
- For the typical websites, the basic aspects of design are:
  - *The Content:* the substance, and information on the site should be relevant to the site and should target the area of the public that the website is concerned with.
  - *The Usability:* the site should be user-friendly, with the interface and navigation simple and reliable.
  - *The Appearance:* the graphics and text should include a single style that flows throughout, to show consistency. The style should be professional, appealing, and relevant.
  - *The Structure:* of the website as a whole.

# Web Page

- A web page is a document or information resource that is suitable for the World Wide Web and can be accessed through a web browser and displayed on a monitor or mobile device.
- This information is usually in HTML or XHTML format, and may provide navigation to other web pages via hypertext links.
- Web pages may be retrieved from a local computer or from a remote web server.
- The web server may restrict access only to a private network, e.g. a corporate intranet, or it may publish pages on the World Wide Web.
- Web pages are requested and served from web servers using Hypertext Transfer Protocol (HTTP).
- Web pages may consist of files of static text and other content stored within the web server's file system (static web pages), or may be constructed by server-side software when they are requested (dynamic web pages).

# Web Site

- A website or simply site, is a collection of related web pages containing images, videos, or other digital assets.
- A website is hosted on at least one web server, accessible via a network such as the Internet or a private local area network through an Internet address known as a Uniform Resource Locator (URL).
- All publicly accessible websites collectively constitute the World Wide Web.
- Web sites can be static or dynamic.

# Static Website

- A static website is one that has web pages stored on the server in the format that is sent to a client web browser, primarily coded in Hypertext Markup Language (HTML).
- Simple forms or marketing examples of websites, such as *classic website*, a *five-page website* or a *brochure website* are often static websites, because they present pre-defined, static information to the user.
- This type of website usually displays the same information to all visitors.
- Similar to handing out a printed brochure to customers or clients, a static website will generally provide consistent, standard information for an extended period of time.
- Although the website owner may make updates periodically, it is a manual process to edit the text, photos and other content and may require basic website design skills and software.
- In summary, visitors are not able to control what information they receive, and must instead settle for whatever content the website owner has decided to offer at that time.

# Dynamic Website

- A dynamic website is one that changes or customizes itself frequently and automatically, based on certain criteria.
- Dynamic websites can have two types of dynamic activity: Code and Content.
- Dynamic Code is invisible or behind the scenes and Dynamic Content is visible or fully displayed.
- The first type is a web page with dynamic code. The code is constructed dynamically on the fly using active programming language instead of plain, static HTML.
- The second type is a website with dynamic content displayed in plain view. Variable content is displayed dynamically on the fly based on certain criteria, usually by retrieving content stored in a database.

# Domain Names, DNS, & URLs

- IP addresses are not convenient for users to remember easily. So an IP address can be represented by a natural language convention called a Domain Name.
- Domain Name System (DNS) translates domain names into IP addresses. DNS is the "phone book" for the Internet, it maps between host names and IP addresses.
- A Uniform Resource Locator (URL), which is the address used by a Web browser to identify the location of content on the Web, also uses a domain name as part of the URL.
- In http://www.example.com/, the scheme is http, & the domain name is example.com.

# Hypertext Markup Language (HTML)

- It is not a programming language.
- A markup language specifies the *layout and style* of a document.
- It consists of a set of markup tags.
- HTML uses markup tags to describe web pages.
- HTML tags are keywords surrounded by angle brackets like <html>.
- Most HTML tags normally come in pairs like  & . The first tag is called the start tag (opening tag) & the second tag is called the end tag (closing tag).
- HTML documents contain HTML tags and plain text.
- HTML documents are also called Web pages.
- The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

# Hypertext Markup Language (HTML)

- A simple HTML document is given below:

```
<html>
    <head>
        <title>This is my first web page</title>
    </head>
    <body>
        <h1>My first heading</h1>
        <p>My first paragraph</p>
    </body>
</html>
```

- Save this page with .html or .htm extension.
- However, it is good practice to use .htm extension.

# HTML Elements

- HTML documents are defined by HTML elements.
- An HTML element is everything from the start tag to the end tag. For example, My first paragraph .
- An HTML element consists of start tag, end tag, and element content.
- The element content is everything between the start tag and end tag.
- Empty elements are closed in the start tag.
- Most HTML elements can have attributes. For example, src attribute of img tag.

# HTML Attributes

- Attributes provide additional information about HTML elements.
- Attributes are always specified in the start tag.
- Attributes come in name/value pair like name="value".
- For example, HTML links are defined with <a> tag and the link address is provided as an attribute href like

  <a href="http://www.tu.edu.np">BSc.CSIT</a>

- Always quote attribute values and use lowercase attributes.

# HTML Headings

- HTML headings are defined with the <h1> to <h6> tags.
- <h1> displays largest text and <h6> smallest.
- For example, <h1>My first heading</h1>.

# HTML Paragraphs

- HTML paragraphs are defined with  tag.
- For example, My first paragraph .

# HTML Comments

- We use comments to make our HTML code more readable and understandable.
- Comments are ignored by the browser and are not displayed.
- Comments are written between <!-- and -->.
- For example, <!–This is a comment -->.

# HTML Rules (Lines)

- We use <hr /> tag to create horizontal line.

# HTML Global Attributes

- HTML attributes give elements meaning and context.
- The global attributes below can be used on **any** HTML element.

| Attribute | Description |
|---|---|
| accesskey | Specifies a shortcut key to activate/focus an element |
| class | Specifies one or more class names for an element (refers to a class in a style sheet) |
| contenteditable | Specifies whether the content of an element is editable or not |
| data-* | Used to store custom data private to the page or application |
| dir | Specifies the text direction for the content in an element |
| draggable | Specifies whether an element is draggable or not |
| dropzone | Specifies whether the dragged data is copied, moved, or linked, when dropped |
| hidden | Specifies that an element is not yet, or is no longer, relevant |
| id | Specifies a unique id for an element |
| lang | Specifies the language of the element's content |
| spellcheck | Specifies whether the element is to have its spelling and grammar checked or not |
| style | Specifies an inline CSS style for an element |
| tabindex | Specifies the tabbing order of an element |
| title | Specifies extra information about an element |
| translate | Specifies whether the content of an element should be translated or not |

# HTML Formatting

- The **<b>** tag specifies bold text without any extra importance.
- The **<u>** tag represents some text that should be stylistically different from normal text, such as misspelled words or proper nouns in Chinese.
- The **<i>** tag defines a part of text in an alternate voice or mood. The content of the **<i>** tag is usually displayed in italic. The **<i>** tag can be used to indicate a technical term, a phrase from another language, a thought, or a ship name, etc.

# HTML Formatting

- The **<small>** tag defines smaller text (and other side comments).
- The **<sup>** tag defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW[1].
- The **<sub>** tag defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like $H_2O$.

# HTML Formatting

```
<html>
<body>
<b>This is a bold content.</b>
<i>This is an italic content.</i>
<u>This is an underlined content.</u>
<b><i><u>This content is bold, italic, and underlined.</b></i></u>
<p>After this content, <small>text comes in smaller size.</small></p>
<p>a<sup>2</sup>+b<sup>2</sup></p>
<p>H<sub>2</sub>0</p>
</body>
</html>
```

# HTML Spacing

- The **<br>** tag inserts a single line break. The **<br>** tag is an empty tag which means that it has no end tag.
- The **<pre>** tag defines preformatted text. Text in a **<pre>** element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.

# HTML Spacing

```
<html>
<body>
<pre>
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both      spaces and
line breaks
</pre>
<p>Text in a p element should use <br> line break tag to go to next line.</p>
</body>
</html>
```

# Formatting Text Phrases

- The **** tag is used to group inline-elements in a document. The **** tag provides no visual change by itself. The **** tag provides a way to add a hook to a part of a text or a part of a document.
- The **** tag is a phrase tag. It defines important text.
- If **<tt>** was used for marking up keyboard input, consider the <kbd> element; for variables, consider the <var> element; for computer code, consider the <code> element; and for computer output, consider the <samp> element, or use CSS instead.
- The **<tt>** tag defines teletype text.

# Formatting Text Phrases

```
<html>
<body>
<p>My mother has <span style="color:blue;font-weight:bold">blue</span>
eyes and my father has <span style="color:darkolivegreen;font-
weight:bold">dark green</span> eyes.</p>
<em>Emphasized text</em><br>
<strong>Strong text</strong><br>
<code>A piece of computer code</code><br>
<samp>Sample output from a computer program</samp><br>
<kbd>Keyboard input</kbd><br>
<var>Variable</var><br>
<tt>This text is teletype text.</tt>
</body>
</html>
```

# HTML Lists

- HTML supports *ordered, unordered* and *definition lists*.
- Ordered lists items are marked with numbers, letters, etc.
- We use <ol> tag for ordered list and each list item starts with <li> tag.

```
<ol type="A">
        <li>Apples</li>
        <li>Mangoes</li>
        <li>Oranges</li>
</ol>
```

**Output:**

A. Apples
B. Mangoes
C. Oranges

- If we do not use type attribute, items are marked with numbers.
- We use type="a" for lowercase letters list, type="I" for roman numbers list, and type="i" for lowercase numbers list.

# HTML Lists

- Unordered lists items are marked with bullets.
- We use <ul> tag for unordered list and each list item starts with <li> tag.

```
<ul type="circle">
        <li>Apples</li>
        <li>Mangoes</li>
        <li>Oranges</li>
</ul>
```

**Output:**

- Apples
- Mangoes
- Oranges

- If we do not use type attribute, items are marked with discs.
- We use type="circle" for circle bullets list, and type="square" for square bullets list.

# HTML Lists

- Definition lists is the list of items with a description of each item.
- We use <dl> tag for definition list, <dt> for definition term, and <dd> for definition description.

```
<dl>
        <dt>Coffee</dt>
            <dd>Black Hot Drink</dd>
        <dt>Milk</dt>
            <dd>White Cold Drink</dd>
</dl>
```

**Output:**

Coffee
    Black Hot Drink
Milk
    White Cold Drink

# HTML Links

- A link is the address to a resource on the web.
- HTML links are defined using an anchor tag (<a>).
- We can use this tag to point to a resource (an HTML page, an image, a sound file, a movie, etc.) and an address inside a document.
- We can use *href* attribute to define the link address. For example,

```
<a href="http://www.tu.edu.np">BSc.CSIT</a>
```

- We can use *target* attribute to define where the linked document will be opened. For example, the following will open the document in a new window.

```
<a href="http://www.tu.edu.np" target="_blank">BSc.CSIT</a>
```

# HTML Div

The  tag defines a division or a section in an TML document.

The  tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.

```
<style>
.myDiv {
  border: 5px outset red;
  background-color: lightblue;
  text-align: center;
}
</style>
</head>
<body>

<div class="myDiv">
  <h2>This is a heading in a div element</h2>
  <p>This is some text in a div element.</p>
</div>
```

# HTML Images

- HTML images are defined with <img> tag.
- To display an image on a page, you need to use the src attribute.
- We can also use width and height attributes with img tag.

```
<img src = "photo.jpg" width = "104" height = "142"/>
```

- We can use alt attribute to define an alternate text for an image.

```
<img src="photo.jpg" width="104" height="142" alt="My best photo"/>
```

- The "alt" attribute tells the reader what he or she is missing on a page if the browser can't load images by displaying the alternate text instead of the image.
- It is a good practice to include the "alt" attribute for each image on a page, to improve the display and usefulness of your document for people who have text-only browsers.

# HTML Image Map

- The **<map>** tag defines an image-map.
- An image-map is an image with clickable areas.
- The idea behind an image map is that you should be able to perform different actions depending on where in the image you click.
- To create an image map you need an image, and a map containing some rules that describe the clickable areas.
- The image is inserted using the **<img>** tag. The only difference from other images is that you must add a **usemap** attribute:
- The **usemap** value starts with a hash tag # followed by the name of the image map, and is used to create a relationship between the image and the image map.

# HTML Image Map

- Then add a **<map>** element.
- The **<map>** element is used to create an image map, and is linked to the image by using the name attribute:
- The name attribute must have the same value as the **usemap** attribute.
- Then add the clickable areas.
- A clickable area is defined using an **<area>** element.
- You must define the shape of the area, and you can choose one of these values:
  - rect - defines a rectangular region
  - circle - defines a circular region
  - poly - defines a polygonal region
  - default - defines the entire region
- You must define some coordinates to be able to place the clickable area onto the image.
- The coordinates come in pairs, one for the x-axis and one for the y-axis.

# Image Map

```
<html>
<body>
<h2>Image Maps</h2>
<p>Click on the computer, the phone, or the cup of coffee to go to a new page and read more about the topic:</p>
<img src="D:\Teaching Stuffs\Texas International College\BScCSIT\5thSemester\Resources\workplace.jpg" alt="Workplace" usemap="#workmap" width="400" height="379">
<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer“ href="https://en.wikipedia.org/wiki/Computer">
  <area shape="rect" coords="290,172,333,250" alt="Phone“ href="https://en.wikipedia.org/wiki/IPhone">
  <area shape="circle" coords="337,300,44" alt="Cup of coffee“ href="https://en.wikipedia.org/wiki/Coffee">
</map>
</body>
</html>
```

# HTML Tables

- Tables are defined with the <table> tag.
- A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag).
- The letters td stands for "*table data*", which is the content of a data cell.
- A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.
- We use border attribute to display table with border.
- Headings in a table are defined with <th> tag.

# HTML Tables

```
<table border="1">
<tr>
        <th>Heading</th>
        <th>Another Heading</th>
</tr>
<tr>
        <td>row 1, cell 1</td>
        <td>row 1, cell 2</td>
</tr>
<tr>
        <td>row 2, cell 1</td>
        <td>row 2, cell 2</td>
</tr>
</table>
```

**Output:**

| Heading | Another Heading |
|---|---|
| row 1, cell 1 | row 1, cell 2 |
| row 2, cell 1 | row 2, cell 2 |

# HTML Tables

- We can use <caption> tag inside a <table> to display caption for a table.
- We can define table cells that span more than one row or one column using colspan and rowspan attributes respectively.

```
<td colspan="2">Data</td>
```

- We can use cellpadding and cellspacing attributes to create white space between the cell content and its borders, and to increase the distance between cells respectively.

```
<table border="1" cellpadding="10">and<table border="1" cellspacing="10">
```

- We can use align attribute to align the contents of a cell.

```
<td align="left">Data</td>
```

# HTML Tables Alignment – row

```
<html>
<body>
<table border=2 cellpadding=5 cellspacing=2
style="width:100%">
 <tr>
  <th style="text-align: center;">Firstname</th>
  <th style="text-align: center;">Lastname</th>
  <th style="text-align: center;">Age</th>
 </tr>
 <tr>
  <td style="text-align: left;">Jill</td>
  <td style="text-align: left;">Smith</td>
  <td style="text-align: left;">50</td>
 </tr>
 <tr>
  <td style="text-align: right;">Eve</td>
  <td style="text-align: right;">Jackson</td>
  <td style="text-align: right;">94</td>
 </tr>
</table>
</body>
</html>
```

**Output:**

| Firstname | Lastname | Age |
|---|---|---|
| Jill | Smith | 50 |
| Eve | Jackson | 94 |

# HTML Tables Alignment – entire

```
<html>
<body>
<table border=2 cellpadding=5 cellspacing=2 style="width:100%; text-align:center; border-collapse: collapse;">
 <tr>
  <th>Firstname</th>
  <th>Lastname</th>
  <th>Age</th>
 </tr>
 <tr>
  <td>Jill</td>
  <td>Smith</td>
  <td>50</td>
 </tr>
 <tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>94</td>
 </tr>
</table>
</body>
</html>
```

**Output:**

| Firstname | Lastname | Age |
|---|---|---|
| Jill | Smith | 50 |
| Eve | Jackson | 94 |

# HTML Tables – colspan

```
<html>
<head>
<body>
<table border=1 style="width:100%;
border-collapse:collapse;">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
</body>
</html>
```

**Output:**

| Name | Telephone | |
|---|---|---|
| Bill Gates | 55577854 | 55577855 |

# HTML Tables – rowspan

```
<html>
<head>
<body>
<table border=1 style="width:100%; border-collapse:collapse;">
  <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>
</body>
</html>
```

**Output:**

| | |
|---|---|
| Name: | Bill Gates |
| Telephone: | 55577854 |
| | 55577855 |

# HTML Tables – caption

```
<html>
<body>
<table border=2 cellpadding=5 cellspacing=2
style="width:100%; text-align:center; border-collapse:
collapse;">
 <caption>Monthly savings</caption>
 <tr>
  <th>Month</th>
  <th>Savings</th>
 </tr>
 <tr>
  <td>January</td>
  <td>$100</td>
 </tr>
 <tr>
  <td>February</td>
  <td>$50</td>
 </tr>
</table>
</body>
</html>
```

**Output:**

Monthly savings

| Month | Savings |
|---|---|
| January | $100 |
| February | $50 |

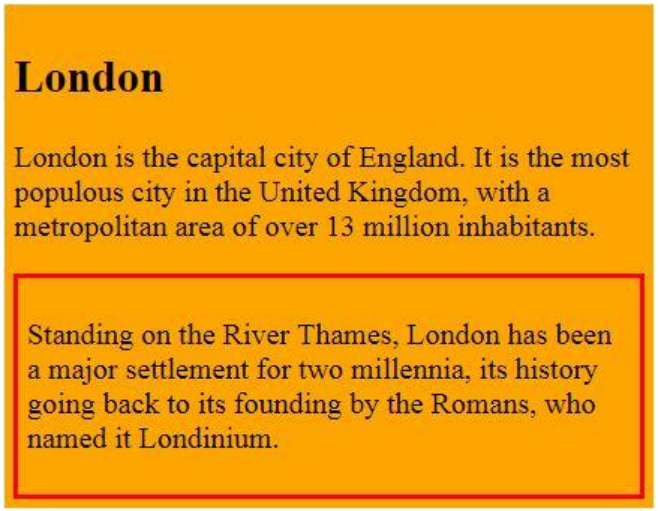
# HTML Div

- The  element defines logical divisions within the document.
- When you use a  element, you are indicating that the enclosed content is specific section of the page and you can format the section with CSS.
- When used together with CSS, the  element can be used to style blocks of content.

# HTML Div

```
<div style="background-color:orange;color:black;padding:5px;">
<h2>London</h2>
<p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
<div style="border:2px solid red;padding:5px">
<p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>
</div>
```

**Output:**



# HTML Frame

- The **<frame>** tag defines one particular window (frame) within a **<frameset>**.
- Each **<frame>** in a **<frameset>** can have different attributes, such as border, scrolling, the ability to resize, etc.

```
<html>
<frameset cols="25%,*,25%">
  <frame src="https://www.sharesansar.com/">
  <frame src="https://www.onlinekhabar.com/">
  <frame src="http://nepalipatro.com.np/">
</frameset>
</html>
```

# HTML Forms

- Forms are used to select different types of user input.
- A form is an area that contains different form elements (text fields, drop-down menus, radio buttons, checkboxes, etc.).
- Form elements are elements that allow the user to enter information in a form.
- A form is defined with the <form> tag.
- The most commonly used form tag is <input> tag.
- The type of input is specified with the type attribute within the <input> tag.

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

**Output:**

First name:

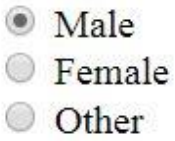
Last name:

# HTML Forms

- Another input type is radio button.
- Radio buttons are used when you want the user to select one of a limited number of choices.

```
<form>
<input type="radio" name="gender" value="male" checked> Male<br>
<input type="radio" name="gender" value="female"> Female<br>
<input type="radio" name="gender" value="other"> Other<br>
</form>
```

**Output:**

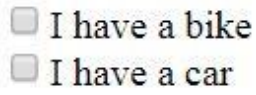


# HTML Forms

- Another input type is checkboxes.
- Checkboxes are used when you want to select one or more options of a limited number of choices.

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

**Output:**



# HTML Forms

- Another input type is submit button.
- When the user clicks on the "Submit" button, the content of the form is sent to the server.
- The form's action attribute defines the name of the file to send the content to.
- The file defined in the action attribute usually does something with the received input.

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

**Output:**

First name:

Last name:

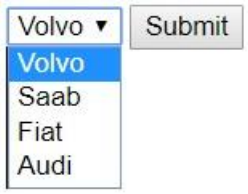
Submit

# HTML Forms

- We can create a simple drop-down box on an HTML page.
- A drop-down box is a selectable list.

```
<form action="/submit.php">
        <select name="cars">
        <option value="volvo">Volvo</option>
        <option value="saab">Saab</option>
        <option value="fiat">Fiat</option>
        <option value="audi">Audi</option>
        </select>
        <input type="submit">
</form>
```

**Output:**



# HTML Forms – Input Types

Here are the different input types you can use in HTML:

- <input type="button">
- <input type="checkbox">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="hidden">
- <input type="image">
- <input type="month">
- <input type="number">
- <input type="password">
- <input type="radio">
- <input type="range">
- <input type="reset">
- <input type="search">
- <input type="submit">
- <input type="tel">
- <input type="text">
- <input type="time">
- <input type="url">
- <input type="week">

# HTML Forms – Input Types

```
<html>
<body>
<form action="/action_page.php">
User name: (Input Type text)<br>
<input type="text" name="userid"><br>
User password: (Input Type password)<br>
<input type="password" name="psw"> <br><br>
Gender: (Input Type radio)<br>
<input type="radio" name="gender" value="male" checked> Male<br>
<input type="radio" name="gender" value="female"> Female<br>
<input type="radio" name="gender" value="other"> Other<br><br>
Multiple Choice Option: (Input Type checkbox)<br>
<input type="checkbox" name="vehicle1" value="Bike">I have a bike<br>
<input type="checkbox" name="vehicle2" value="Car">I have a car<br><br>
Choose from dropdown: (select tag)
<select name="cars">
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="fiat">Fiat</option>
<option value="audi">Audi</option>
</select><br><br>
<input type="submit"  value="SUBMIT">
<input type="reset">
</form>
</form>
</body>
</html>
```

# HTML Meta Tags

- Metadata is data (information) about data.
- The **<meta>** tag provides metadata about the HTML document and always go inside the **<head>** element.
- Metadata is always passed as name/value pairs.
- Metadata will not be displayed on the page, but will be machine parseable.
- The purpose of the meta element is to provide meta-information about the document.
- Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.
- Meta elements are purely used for search engine's use and to provide some additional information about your pages.
- We use three attributes *name*, *content*, and *http-equiv* with <meta> tag.

# HTML Meta Tags

- Define a description of your web page:

```
<meta name="description" content="Free Web Tutorials">
```

- Define keywords for search engines:

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

- Define the author of a page:

```
<meta name="author" content="John Doe">
```

- Refresh document every 30 seconds:

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

# HTML Audio Tag

- The **<audio>** tag defines sound, such as music or other audio streams.
- Currently, there are 3 supported file formats for the **<audio>** element – MP3, WAV, and OGG.

```
<html>
<body>
<audio controls>
<source src="D:\iPhone songs\Dil chahta hai.mp3"
type="audio/mpeg">
</audio>
</body>
</html>
```

# HTML Video Tag

- The **<video>** tag specifies video, such as a movie clip or other video streams.
- Currently, there are 3 supported video formats for the **<video>** element – MP4, WebM, and Ogg.

```
<html>
<body>
<video width="320" height="240" controls>
  <source src="C:\Users\Mr. Nepal\Videos\sample video.mp4"
type="video/mp4">
</video>
</body>
</html>
```

# HTML <nav> Tag

```
<html>
<body>
<nav>
<a href="https://www.facebook.com/">Facebook</a> |
<a href="https://www.youtube.com/">Youtube</a> |
<a href="https://twitter.com/">Twitter</a> |
<a href="https://www.instagram.com/">Instagram</a>
</nav>
</body>
</html>
```

# HTML Head

- The head element contains general information, also called meta-information, about a document.
- The elements inside the head element should not be displayed by a browser.
- According to the HTML standard, only a few tags are legal inside the head section. These are: *<base>*, *<link>*, *<meta>*, *<title>*, *<style>*, and *<script>*.
- These element should only be used once and must start immediately after the opening <html> tag and end directly before the opening <body> tag.

# HTML Head

```
<head>
        <title>Page Title</title>
        <style>
        body {background-color: aqua;}
        h1 {color: red;}
        p {color: blue;}
        </style>
</head>
<body>
        <h1>This is a Heading</h1>
        <p>This is a paragraph.</p>
</body>
```

**Output:**

# HTML Events

- Events trigger actions in the browser, like starting a JavaScript when a user clicks on an HTML element.
- Below is a list of attributes that can be inserted to HTML tags to define event actions.
- **Window Events** (Only valid in body and frameset elements)

| Attribute | Value | Description |
|---|---|---|
| Onload | *Script* | Script to be run when a document loads |
| Onunload | *Script* | Script to be run when a document unloads |

# HTML Events

- **Form Element Events** (Only valid in form elements)

| Attribute | Value | Description |
|---|---|---|
| Onchange | *Script* | Script to be run when element changes |
| Onsubmit | *Script* | Script to be run when the form is submitted |
| Onreset | *Script* | Script to be run when the form is reset |
| Onselect | *Script* | Script to be run when the element is selected |
| Onblur | *Script* | Script to be run when the element loses focus |
| Onfocus | *Script* | Script to be run when the element gets focus |

# HTML Events

- **Keyboard Events** (Not valid in *base*, *bdo*, *br*, *frame*, *frameset*, *head*, *html*, *iframe*, *meta*, *param*, *script*, *style*, and *title* elements)

| Attribute | Value | Description |
|---|---|---|
| Onkeydown | *Script* | What to do when key is pressed |
| Onkeypress | *Script* | What to do when key is pressed and released |
| Onkeyup | *Script* | What to do when key is released |

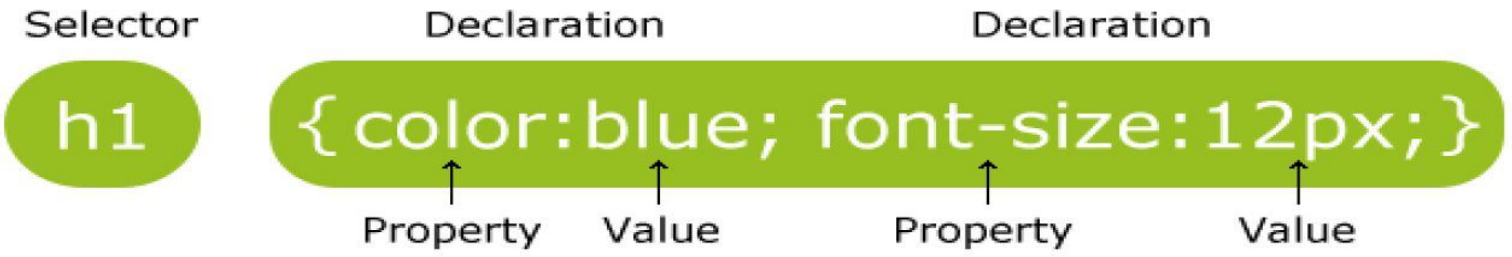
# HTML Events

- **Mouse Events** (Not valid in *base*, *bdo*, *br*, *frame*, *frameset*, *head*, *html*, *iframe*, *meta*, *param*, *script*, *style*, and *title* elements)

| Attribute | Value | Description |
|---|---|---|
| Onclick | *Script* | What to do on a mouse click |
| Ondblclick | *Script* | What to do on a mouse double-click |
| Onmousedown | *Script* | What to do when mouse button is pressed |
| Onmousemove | *Script* | What to do when mouse pointer moves |
| Onmouseout | *Script* | What to do when mouse pointer moves out of an element |
| Onmouseover | *Script* | What to do when mouse pointer moves over an element |
| Onmouseup | *Script* | What to do when mouse button is released |

# Cascading Style Sheets (CSS)

- It was first developed in 1997, as a way for Web developers to define the *look and feel* of their Web pages.
- It was intended to allow developers to separate content from design and layout so that HTML could perform more of the function without worry about the design and layout.
- It is used to separate style from content.
- A CSS rule has two main parts: a *selector* and one or more *declarations*.
- Selector is normally the HTML element you want to style and each declaration consists of a *property* and *value*.
- The *property* is the style attribute we want to use and each property has a *value* associated with it.



# Inserting CSS

- We can use style sheets in three different ways in our HTML document.
- There are three style sheets:
  - External Style Sheet
  - Internal Style Sheet
  - Inline Style

## Multiple Styles will cascade into one

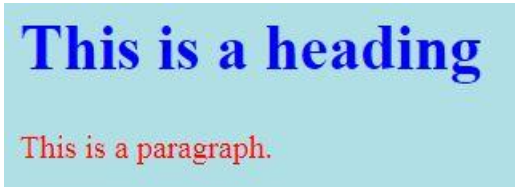
- Styles can be specified:
  - Inside an HTML element
  - Inside the head section of an HTML page
  - In an external CSS file

# External Style Sheet

- If we want to apply the same style to many pages, we use external style sheet.
- With an external style sheet, you can change the look of an entire Web site by changing one style sheet file.
- Each page must link to the style sheet using the <link> tag.
- The <link> tag goes inside the head section.

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

**Output:**




- An external style sheet can be written in any text editor.
- The file should not contain any html tags.
- Your style sheet should be saved with a .css extension.

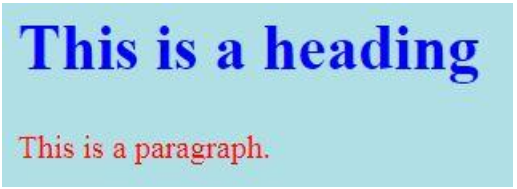
```
body{background-color: powderblue;}
h1{color:blue;}
p{color:red;}
```

# Internal Style Sheet

- If you want a unique style to a single document, an internal style sheet should be used.
- You define internal styles in the head section of an HTML page, by using the <style> tag.

```
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

**Output:**



# Inline Styles

- If you want a unique style to a single element, an inline style sheet should be used.
- An inline style loses many of the advantages of style sheets by mixing content with presentation.
- To use inline styles you use the style attribute in the relevant tag.
- The style attribute can contain any CSS property.

```
<body>
<h1 style="color:blue;">This is a Blue Heading</h1>
</body
```

**Output:**

**This is a Blue Heading**

# Advantages of CSS

- **Quick loading and less bandwidth** – the reason why CSS is preferred by web developers is because it is lighter than the table layout which consumes a lot of bandwidth. The style sheet is loaded only once and is stored in the cache memory. This way the subsequent pages load faster.
- **Multiple Browsers Compatible** – CSS is compatible with all web browsers. Therefore, a website based on CSS will appear similar in all the web browsers.
- **Ability to position elements anywhere** – with CSS, you can position the elements anywhere in the webpage. Thus you can ask the developer to change the position of a particular element if it does not go well with the way you had wanted it to be.

# Advantages of CSS

- **Customization** – CSS sheets allow the users to customize the web page. The CSS sheets are stored externally and allow the user to make requisite changes on their own. A number of modern browsers give the liberty of defining their own style sheets to the users. Thus a user can change font properties easily.
- **Search engine friendly** – the CSS style sheets make it easier for a website to feature in major search engine results. Since it allows elements to be positioned anywhere, the user can project the main contents first, so that it is easily captured by web spiders. It also provides cleaner HTML codes thus cutting down the job of web spider to search the real content from junk code.
- **Print friendly web pages** – another important feature of CSS is that it provides print friendly web pages. This is to say that the web pages can be easily printed. The colors, images and other things which are difficult to print can be eliminated and printed easily.

# Advantages of CSS

- **Consistency** – CSS provides complete consistency to the web pages. It implies that all the expressions and texts will get their characteristics from external style sheet. Thus a developer need not worry about the change in characteristics of the elements because they can be easily altered at any stage of web development by using CSS.
- **Portability** – CSS also lends portability to the content. With the help of CSS, you can make separate style sheets for different media. Thus the content can be presented in a flexible way. It also allows you to redefine the characteristics of elements in a website to suit the need of the situation.

# Id and Class Selectors

- The id selector is used to specify a style for a single, unique element.
- The id selector uses id attribute of the HTML element and is defined with "#".

```
<html>
<head>
<style>
#p01{color:blue;}
</style>
</head>
<body>
<p>This is a paragraph.</p>
<p id="p01">I am different.</p>
</body>
</html>
```

**Output:**

This is a paragraph.

I am different.

# Id and Class Selectors

- The class selector is used to specify a style for a group of elements.
- This allows you to set a particular style for any HTML elements with the same class.
- The class selector uses the HTML class attribute, and is defined with a "."

```
<html>
<head>
<style>
p.error{color:red;}
</style>
</head>
<body>
<p>This is a paragraph.</p>
<p class="error">I am different.</p>
<p>This is a paragraph.</p>
<p class="error">I am different too.</p>
</body>
</html>
```

**Output:**

This is a paragraph.

I am different.

This is a paragraph.

I am different too.

# Grouping Selectors

- In style sheets there are often elements with the same style.
- To minimize the code, you can group selectors by separating each selector with a comma.
- If you have 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;}
```

- It will be better to group the selectors, to minimize the code, like this:

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

# Grouping Selectors

- Example 1:

```
<head>
<style>
h1{text-align:center; color:red;}
h2{text-align:center; color:red;}
p{text-align:center; color:red;}
</style>
</head>
<body>
<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>
</body>
```

- Example 2:

```
<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>
```

**Output:**

# Nesting Selectors

- You can nest selectors, for example a <ul> tag has <ul> and also a <li> tag; you can apply style to the <li> tag by nesting.

  ul li {padding:10px;}

- This would give all <li> tags in an unordered list a padding of 10px.
- Similarly, the following would style a <th> element with a black background, providing it's in a <table> tag and then a <tr> tag.

  table tr th {background-color:#000;}

# Cascading Order

- What style will be used when there is more than one style specified for an HTML element?
- Style sheets cascade in this order (3rd having the highest priority)
  - External Style Sheets
  - Internal Style Sheets
  - Inline Styles
- This list could be summed up into one basic rule: If two styles come into conflict, the last one used will take precedence.
  - Inline styles must be placed in the body of the HTML document, while internal style sheets must be placed in the head of the HTML document. As a result, inline styles are always the last ones used and therefore take precedence.
  - The browser treats all external style sheets as occurring before internal style sheets, regardless of where the link is placed. Therefore, internal style sheets are always a higher priority than external because they occur later in the document (according to the browser).

# CSS Colors

- In CSS, a color can be specified by using a color name.
- You can set the background color for HTML elements.
- You can set the color of text.
- You can set the color of borders.
- In CSS, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values.
- In CSS, a color can be specified as an RGB value, using this formula: **rgb(*red, green, blue*)**
- In CSS, a color can be specified using a hexadecimal value in the form: ***#rrggbb***

# CSS Colors

- In CSS, a color can be specified using hue, saturation, and lightness (HSL) in the form: **hsl(*hue*, *saturation*, *lightness*)**
- RGBA color values are an extension of RGB color values with an alpha channel – which specifies the opacity for a color.
- An RGBA color value is specified with: **rgba(*red, green, blue, alpha*)**
- HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.
- An HSLA color value is specified with: **hsla(*hue, saturation, lightness, alpha*)**

# CSS Colors

```
<html>
<body>
<p style="border:2px solid Tomato; background-color:hsl(147, 50%, 47%); color:#0000ff;">
Bard of Blood - Excommunicated RAW agent Kabir Anand, a courageous analyst and a sleeper agent take on a covert mission in Balochistan when four Indian spies are captured; the mission soon turns south and Kabir must face his demons from the past.
</p>
</body>
</html>
```

# CSS Background

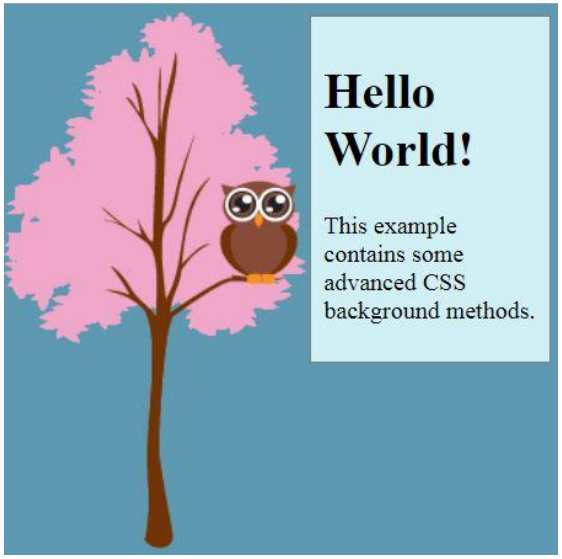
- Background properties are used to define the background effects of an HTML element.
- CSS properties used to define background effects are: background-color, background-image, background-repeat, background-attachment, and background-position.

| Property | Description |
|---|---|
| background | Sets all the background properties in one declaration (shorthand property) |
| background-attachment | Sets whether a background image is fixed or scrolls with the rest of the page |
| background-clip | Specifies the painting area of the background |
| background-color | Sets the background color of an element |
| background-image | Sets the background image for an element |
| background-origin | Specifies where the background image(s) is/are positioned |
| background-position | Sets the starting position of a background image |
| background-repeat | Sets how a background image will be repeated |
| background-size | Specifies the size of the background image(s) |

# CSS Background

```
<head>
<style>
body {margin-left: 180px; background: #5d9ab2 url("https://f-scope.net/images/img_treepng.png") no-repeat top left;}
.center_div {
  border: 1px solid gray;
  margin-left: auto;
  margin-right: auto;
  width: 90%;
  background-color: #d0f0f6;
  text-align: left;
  padding: 8px;
}
</style>
</head>
<body>
<div class="center_div">
  <h1>Hello World!</h1>
  <p>This example contains some advanced CSS background methods.</p>
</div>
</body>
```

**Output:**



# CSS Borders

- The CSS border properties allow you to specify the style, width, and color of an element's border.
- The border-style property specifies what kind of border to display. The following values are allowed:
  - dotted - Defines a dotted border
  - dashed - Defines a dashed border
  - solid - Defines a solid border
  - double - Defines a double border
  - groove - Defines a 3D grooved border. The effect depends on the border-color value
  - ridge - Defines a 3D ridged border. The effect depends on the border-color value
  - inset - Defines a 3D inset border. The effect depends on the border-color value
  - outset - Defines a 3D outset border. The effect depends on the border-color value
  - none - Defines no border
  - hidden - Defines a hidden border
- The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

# CSS Borders

```
<html>
<body>
<h2>The border-style Property</h2>
<p>This property specifies what kind of border to display:</p>
<p style="border-style: dotted;">A dotted border.</p>
<p style="border-style: dashed;;">A dashed border.</p>
<p style="border-style: solid;">A solid border.</p>
<p style="border-style: double">A double border.</p>
<p style="border-style: groove;">A groove border.</p>
<p style="border-style: ridge;">A ridge border.</p>
<p style="border-style: inset;">An inset border.</p>
<p style="border-style: outset;">An outset border.</p>
<p style="border-style: none;">No border.</p>
<p style="border-style: hidden;">A hidden border.</p>
<p style="border-style: dotted dashed solid double;">A mixed border.</p>
</body>
</html>
```

# CSS Text

- The **color** property is used to set the color of the text. The color is specified by:
  - A color name - like "red"
  - A HEX value - like "#ff0000"
  - An RGB value - like "rgb(255,0,0)“
- The **text-align** property is used to set the horizontal alignment of a text. A text can be left or right aligned, centered, or justified.
- The **text-transform** property is used to specify uppercase and lowercase letters in a text. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.
- The **letter-spacing** property is used to specify the space between the characters in a text.
- The **word-spacing** property is used to specify the space between the words in a text.

# CSS Text

```
<html>
<head>
<style>
h1 {color: rgb(255,0,0); text-align: center; text-transform: uppercase; letter-spacing: 3px;}
h2 {color: #0000ff; text-align: left; text-transform: lowercase; word-spacing: 10px;}
h3 {color: green; text-align: right; text-transform: capitalize; }
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
</body>
</html>
```

# CSS Font

- The CSS **font** properties define the font family, boldness, size, and the style of a text.
- The font family of a text is set with the **font-family** property.
- The **font-style** property is mostly used to specify italic text.
- This property has three values:
  - normal - The text is shown normally
  - italic - The text is shown in italics
  - oblique - The text is "leaning" (oblique is very similar to italic, but less supported)
- The **font-size** property sets the size of the text.

# CSS Font

```
<html>
<body>
<p style="font-style: normal; font-family: arial; font-size: 30px;">This is a paragraph in normal style.</p>
<p style="font-style: italic; font-size: 20px;">This is a paragraph in italic style.</p>
<p style="font-style: oblique; font-size: 40px;">This is a paragraph in oblique style.</p>
</body>
</html>
```

# CSS List

- The **list-style-type:none** property can also be used to remove the markers/bullets.
- Note that the list also has default margin and padding. To remove this, add margin:0 and padding:0 to <ul> or <ol>
- We can also style lists with colors, to make them look a little more interesting.

# CSS List

```
<html>
<body>
<p>Default list:</p>
<ul style="background: #ff9999; padding: 20px;">
 <li style="background: #3399ff;">Coffee</li>
 <li style="background: #ffe5e5;">Tea</li>
 <li style="background: #3399ff;">Coca Cola</li>
</ul>
<p>Remove bullets, margin and padding:</p>
<ul style="list-style-type: none; margin: 0; padding: 0;">
 <li>Coffee</li>
 <li>Tea</li>
 <li>Coca Cola</li>
</ul>
</body>
</html>
```

# CSS Table

- The look of an HTML table can be greatly improved with CSS.
- To specify table borders in CSS, use the **border** property.
- Width and height of a table are defined by the **width** and **height** properties.
- The **text-align** property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.
- To control the space between the border and the content in a table, use the **padding** property on <td> and <th> elements.
- Use the **:hover** selector on <tr> to highlight table rows on mouse over.
- For zebra-striped tables, use the **nth-child()** selector and add a **background-color** to all even (or odd) table rows.

# CSS Table

```
<html>
<head>
<style>
#students {
  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
  border-collapse: collapse;
  width: 100%;
}
#students td, #students th {
  border: 1px solid #ddd;
  padding: 8px;
}
#students tr:nth-child(even){background-color: #f2f2f2;}
#students tr:hover {background-color: #ddd;}
#students th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4CAF50;
  color: white;
}
</style>
</head>
<body>
<table id="students">
 <tr>
  <th>Name</th>
  <th>Contact</th>
  <th>Address</th>
 </tr>
 <tr>
  <td>Nischal Basnet</td>
  <td>1234567890</td>
  <td>Baneshwor</td>
 </tr>
 <tr>
  <td>Jitu Nepal</td>
  <td>1234567890</td>
  <td>Lokanthali</td>
 </tr>
 <tr>
  <td>Swostima Khadka</td>
  <td>1234567890</td>
  <td>Mitrapark</td>
 </tr>
 <tr>
  <td>Dayahang Rai</td>
  <td>1234567890</td>
  <td>Chabahil</td>
 </tr>
 <tr>
  <td>Sugam Pokharel</td>
  <td>1234567890</td>
  <td>Lalitpur</td>
 </tr>
</table>
</body>
</html>
```