HTML

- **Hypertext Markup Language (HTML)** is the standard **markup language** for creating web pages and web applications.
- XHTML stands for **EXtensible HyperText Markup Language**. XHTML is almost identical to HTML.
- **HTML elements** are the building blocks of HTML pages.
- HTML elements are represented by **HTML tags**.
- HTML tags are element names surrounded by angle brackets (<tag>)

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

- HTML tags normally come in pairs like and
- The first tag in a pair is the start tag (opening tag), the second tag is the end tag (closing tag).
- The end tag has a forward slash inserted before the tag name.
- Browsers do not display the HTML tags, but use them to interpret the content of the page.
- A few tags are called **non-container tags (empty tags)**, because they don't contain any content.
- Non-container tags end in />. Example:
 ,for line break.
- The basic XHTML/HTML documents contains three parts:
  - DOCTYPE: It is used to declare a DTD (Document Type Definition).
  - head: The head section is used to declare the title and other attributes.
  - body: The body tag contains the content of web pages. It consists many tags.
- A simple html document is:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
        <head>
                <title>Simple html document</title>
        </head>
        <body>
                <h1>heading element</h1>
                <p>paragraph element</p>
        </body>
 </html>
```

  - The text between the <head> tag and the </head> tag is header information. Header information is not displayed in the browser window.
  - The text between the <body> tags is the text that will be displayed in your browser.

[HTML comment: <!-- comment here -->]

Tag attributes

- Attributes can **provide additional information** about the HTML elements on your page.
- Attributes always come in name/value pairs like this: name="value".
- Attributes are always added to the start tag of an HTML element.
- Attribute values should always be enclosed in quotes.
- Example: <body bgcolor="red">

- Common attributes that every html tag can have are:
  - **id**: specifies a unique id for an HTML element. The id value can be used by CSS and JavaScript to perform certain tasks for a unique element with the specified id value.
  - **class:** class attribute is used to define equal styles for elements with the same class name.
  - **name**: specifies the name of html tag which is sent to the server to be recognized and get the value.

Basic HTML tags

1. Head tags

a. The **<title>** tag adds title to a web page.

b. The **<meta>** tag defines information about an XHTML document.

- This information is usually referred to as metadata.
- The metadata information is used by the search engines to determine what the document is about.
- Metadata will not be displayed on the page.
- Metadata is always passed as name/value pairs. Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.
- The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.
- Example:
  - <meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript" /> (Defines keyword for search engines)
  - <meta name="description" content="Free Web tutorials on HTML and CSS"/> (provides a description of your web page)
  - <meta name="author" content="John Doe"/> (defines the author of the web page)
  - <meta name="viewport" content="width=device-width, initial-scale=1.0" /> (to change page width according to device width)

c. **<style>,<link>** etc.

2. Basic text markup

a. **Heading tag** (<hl></hl>,<h2></h2>, <h3></h3>,<h4></h4>, <h5></h5>,<h6></h6>)
The heading tags display text in sizes ranging from the largest, h1, to the smallest, h6.

b. **Paragraph tag** ( ) : Defines a paragraph.

c. **Line breaks** (
): It inserts a line break.

d. **<pre> </pre> tag:** The <pre> tag defines preformatted text. Text in a <pre> element is displayed in a fixed-width font and it preserves both spaces and line breaks.

e. **Block quotation :** All of the text within the <blockquote> and </blockquote> tags is set off from the regular document text, usually with indented left and right margins and sometimes in italicized typeface. **(<blockquote></blockquote>)**

f. **Font style and sizes:**

- The HTML **** tag defines bold text, without any extra importance.
- The HTML **** tag defines strong text, with added semantic "strong" (importance).
- The HTML **** tag defines italic text.

- The HTML **** element defines **emphasize**d text, with added semantic importance. *(<b> and <i> defines bold and italic text, but and means that the text is "important".).*
- The **<sub>** element defines subscripted text.
- The **<sup>** element defines superscripted text.
- The **<del> or <strike>** element is used to strike through the text marking the part as deleted.
- The **<ins> or <u>** element is used to underline a text marking the part as inserted or added.

g. **Horizontal rules (<hr />)**: It creates a line to separate content of a web page.

3. Character Entities

Result	Description	Entity Name	Entity Number
	non-breaking space		
<	less than	<	<
>	greater than	>	>
&	Ampersand	&	&
¢	Cent	¢	¢
£	Pound	£	£
¥	Yen	¥	¥
€	Euro	€	€
©	Copyright	©	©
®	registered trademark	®	®

4. IMAGES

The most common methods of representing images are the Graphic Interchange Format (GIF)and the Joint Photographic Experts Group (JPEG)format and PNG (Portable Network Graphics).

- The <img /> tag defines an image in an HTML page.
- The <img /> tag has a must required attribute: src , which specifies the URL of the image.
- Example: ****
- Commonly used attributes for <img> tag are:

Attributes	Definition
src	specifies the URL of the image.
alt	Specifies an alternate text for an image
height	Adjust height of the image (in pixels)
width	Adjust width of the image (in pixels)

Example:

****

5. Hypertext Links

- The <a> tag defines a hyperlink, which is used to link from one page to another or link to another section on the same page.
- The most important attribute of the <a> element is the **href** (hypertext reference) attribute, which indicates the link's destination.
- The anchor tag that specifies a link is called the **source** of that link. The document whose address is specified in a link is called the **target** of that link.
- **Visit google**
- Uses a 'target' attribute to specify where to open the link (in the same tab or in another). Target=" _blank" will open the page in another tab/window.
- **Visit google**
- To open the linked page when you click on an image (clickable image):
  - ** **

- **Targets within document**
  - The target element can include an id attribute, which can then be used to identify it in an href attribute.
  - Example: <h4 id = "footerhead"> Footer Details </h4>

  If the target is in the same document as the link, the target is specified in the href attribute value by preceding the id value with a pound sign (#).

   Go to footer details

6. LISTS

A. Unordered list

- An unordered list is a collection of items where the order of appearance is not important (e.g., a bulleted list).
- The <ul> tag, which is a block tag, creates an unordered list.
- Each item in a list is specified with an <li> tag (li is an acronym for list item).
- Example:

```
<ul>
  <li>Java</li>
  <li>C</li>
  <li>C++</li>
  <li>WT</li>
</ul>
```

- Attributes used:
  - type="*value*"
    It defines the style of list item marker. The value can be **circle, square** and **disc** .
  - Example:

```
<ul type="square">
  <li>Java</li>
  <li>C</li>
  <li>C++</li>
```

```
  <li>WT</li>
</ul>
```

B. Ordered list

- An ordered list is a collection of items that are numbered(default:1,2,3...).
- Lists in which the order of items is important.
- The <ol> tag, which is a block tag, creates an ordered list.
- Each item in a list is specified with an <li> tag (li is an acronym for list item).
- Example:

```
<ol>
  <li>Java</li>
  <li>C</li>
  <li>C++</li>
  <li>WT</li>
</ol>
```

- Attributes used:
  - type="*value*"
    It defines the numbering style. The value can be **1,A,a,I,i**.
  - start="*value*"
    It defines the start value of numbers/letters.
  - Example:

```
<ol type="A" start="3">
  <li>Java</li>
  <li>C</li>
  <li>C++</li>
  <li>WT</li>
</ol>
```

C. Java
D. C
E. C++
F. WT

Figure 3:output

C. Definition list

- Definition lists are used to specify lists of terms and their definitions.
- The <dl> tag defines the definition list, the <dt> tag defines the definition term, and the <dd> tag describes each term definition.
- Example:

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

Definition list

Coffee
    A black hot drink
Milk
    A white cold drink

Figure 4:output

D. Nested list example

7. Tables

- A table is a matrix of cells composed of rows and columns.
- A table is specified as the content of the block tag <table>.
- The line around the outside of the whole table is called the border.
- A displayed table is preceded by a title, given as the content of a <caption> tag.
- Each row of a table is specified with a row tag, <tr>. Within each row, the row label is specified by the table heading tag, <th> or with a table data tag, <td>.
- The **border** attribute is the most common attribute for the <table> tag.
    - The values can be : border="border" or border="1" or border="2" etc.
- Example:

```
<table border="1">
        <caption>Students Marks</caption>
            <tr>
                <th>S.N</th>
                <th>Name</th>
                <th>Marks</th>
            </tr>
            <tr>
                <td>1</td>
                <td>Ram</td>
                <td>45</td>
            <tr>
                <td>1</td>
                <td>Sita</td>
                <td>46</td>
            </tr>
</table>
```

- **rowspan and colspan attributes** (of <th> or <td> tag)
    - The colspan attribute defines the number of columns a cell should span (or merge) horizontally.
    - The rowspan attribute specifies the number of rows a cell should span vertically.

- **align and valign attributes**
  - The placement of the content within a table cell can be specified with the align and valign attributes in the <tr>, <th>, and <td> tags.
  - The align attribute has the possible values **left, right,** and **center**.
  - The default alignment for th cells is center; for td cells, it is left.
  - The valign attribute of the <th> and <td> tags have the possible values **top** and **bottom**. (*The valign attribute is not supported in HTML5)

- **cellpadding and cellspacing attributes** (of <table> tag)
  - The cellpadding attribute is used to specify the spacing between the content of a cell and the inner walls of the cell.
  - The cellspacing attribute is used to specify the distance between cells in a table.

Example:

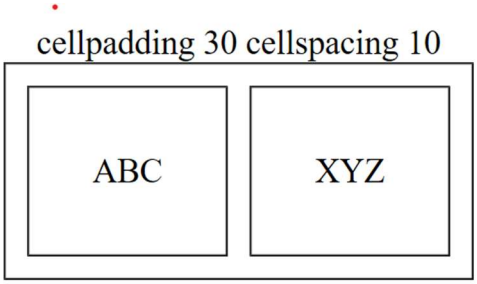
```
<table border="1">
        <tr>
                <th rowspan="2"> Item</th>
                <th colspan="2">Year 2019</th>
        </tr>
        <tr>
                <th>Cost price</th>
                <th>Selling price</th>
        </tr>
        <tr>
                <th> T.V</th>
                <td>30,000</th>
                <td>35,000</th>
        </tr>
        <tr>
                <th> Laptop</th>
                <td align="center">60,000</th>
                <td align="right">65,000</th>
        </tr>
        <tr>
                <th colspan="2">Total</th>
                <td>100000</td>
  </table>
```

Item	Year 2019	
	Cost price	Selling price
T.V	30,000	35,000
Laptop	60,000	65,000
Total		100000

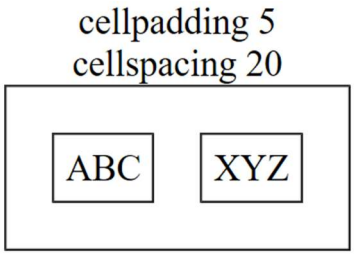
Figure 5:Output

Example:

```
<table border="border" cellspacing="10" cellpadding="30">
        <caption> cellpadding 30 cellspacing 10</caption>
                <tr>
                        <td >ABC</td>
                        <td>XYZ</td>
                 </tr>
   </table>
```




```
<table border="border" cellspacing="20" cellpadding="5">
        <caption> cellpadding 5 cellspacing 20</caption>
                <tr>
                        <td>ABC</td>
                        <td>XYZ</td>
                </tr>
</table>
```



- **Table with {thead, tbody, tfoot, th} Tags**
  - **thead** is used to enclose a group of rows in a table as a header.
  - **tfoot** is used to enclose a group of rows in a table as a footer, such as last row for summary.
  - **tbody** is for main body of the table.

```
<table border="1">
<thead>
<tr>
<th>cats</th>
<th>dogs</th>
</tr>
</thead>

<tbody>
<tr>
<td>7</td>
<td>6</td>
</tr>
</tbody>

<tfoot>
<tr>
<th colspan="2">Cats win!</th>
</tr>
</tfoot>

</table>
```

cats	dogs
7	6
Cats win!	

8. Forms

- The HTML <form> element defines a form that is used to collect user input.
- Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons etc.
- All of the controls of a form appear in the content of a **<form> tag** (a type of block tag)
- The commonly used attributes of <form> tag are:
  - **action**: specifies the URL of the application on the Web server that is to be called when the user clicks the Submit button.
  - **target**: Specifies the target window or frame where the result of the script will be displayed. It takes values like _blank, _self etc.
  - **method**: specifies one of the two techniques, **get** (default value) or **post**, used to pass the form data to the server.
  - **enctype**: to specify how the browser encodes the data before it sends it to the server.
    - **enctype=" mutipart/form-data" –** This is used when you want to upload binary data in the form of files like image, word file etc.
- Basic syntax:

```
<form action = "submitform.php" method = "GET|POST" target="_blank">
        <!--form elements like input, textarea etc.-->
</form>
```

HTML Form controls/ Form elements

1. <input> element

- Many of the commonly used controls are specified with the inline tag <input>.
- The <input> element can be displayed in several ways, depending on the **type** attribute.
- <input> tag is used for text,password,email,checkboxes,radio buttons, submit button, reset button etc.
- Common attributes of <input> tag are:
  - **type:** it specifies the type of input control .
    - example: for text field type="text" & type="email" for email field.
  - **name:** specifies the name of <input> tag which is sent to the server to be recognized and get the value.
  - **id:** to use with CSS or JavaScript.

❖ **Input type text**

- <input type="text" name="username" />
- Other attributes:
  - value: used to provide an initial value inside the control.
  - size: specifies the width of input control in terms of characters.
  - maxlength: specifies the maximum number of characters a user can enter into the text box.
  - placeholder: specifies a short hint that describes the expected value of an input field.

❖ **Input type password**

- <input type="password" name="userpass" />
- Other attributes: same as in input type text.

❖ **Input type email**

- <input type="email" name="useremail" />

- The input value is automatically validated to ensure it is a properly formatted e-mail address.

❖ **Input type submit**
- defines a button for submitting form data to a form-handler.
- <input type="submit" value="submit" />

❖ **Input type reset**
- defines a reset button that will reset all form values to their default values
- <input type="reset" value="Reset" />

❖ **Input type button**
- Defines a clickable button.
- <input type="button" value="click me" />

(**the submit & reset buttons are called action buttons**)

❖ **RADIO Buttons**
- The <input type="radio"> defines a radio button.
- Radio buttons are normally presented in radio groups (a collection of radio buttons describing a set of related options).
- Only one radio button in a group can be selected at a time.
- the value of name attribute must be same within a radio buttons group.
- Example:

```
<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
```

❖ **Check Boxes**
- Checkboxes let a user select ZERO or MORE options of a limited number of choices.
- Example:

```
<input type=”checkbox” name=”subjects” value=”WT” /> Web Technology

<input type=”checkbox” name=”subjects” value=”C” /> C Programming

<input type=”checkbox” name=”subjects” value=”java” /> JAVA
```

- The value of name attribute should be same for a group of checkboxes.
- checked=”checked” ,attribute is used to select a default option.

2. <select> tag

- The <select> element is used to create a drop-down list/menu.
- Each of the items in a menu is specified with an **<option> tag**, nested in the select element.
- The content of an <option> tag is the value of the menu item.
- The <option> tag can include the **selected** attribute, which specifies that the item is preselected. (selected="selected").
- Example:

```
<select name="country">
 <option value="Nep" >Nepal</option>
 <option value="Pak">Pakistan</option>
 <option value="Ind">India</option>
</select>
```

Figure 6: default appearance

Figure 7: after clicking dropdown

❖ Other attributes used:
  - multiple :to select multiple values from dropdown
  - size="*integer value*" : Defines the number of visible options in a drop-down list. (<select size="2">)

3. <textarea> tag

- The <textarea> tag defines a multi-line text input control.
- The attributes **rows** (for height) **& cols** (for width) is used to define the size of text area.
- Example:

```
<textarea rows="4" cols="50">
</textarea>
```

- Other attributes used:

maxlength	*number*	Specifies the maximum number of characters allowed in the text area
name	*text*	Specifies a name for a text area
placeholder	*text*	Specifies a short hint that describes the expected value of a text area
readonly	readonly	Specifies that a text area should be read-only

9. Frames

- HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document.
- A collection of frames in the browser window is known as a frameset.

- The number of frames and their layout in the browser window are specified with the <frameset> tag.
- A document has either a frameset or a body but cannot have both.
- The rows attribute of <frameset> tag defines horizontal frames and cols attribute defines vertical frames.
- Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.
- **rows & cols attribute** (of <frameset> tag)**:**
    - The height of each rows and the width of columns is set in following ways**:**
        - Use absolute value in pixel
            - <frameset cols = "300, 400, 300"> or <frameset rows = "300, 400, 300">
        - Use percentage value
            - <frameset cols = "30%, 40%, 30%"> or <frameset rows = "30%, 40%, 30%">
        - Use wild card values
            - <frameset cols = "30%, *"> or <frameset rows = "30%, *">
- **Attributes of <frame> tag**
    - **src:** to define the source file url that should be loaded into the frame.
    - **name:** used to give names to the frame. It is also used to indicate which frame a document should loaded into.

```
<frameset rows="40%,*">
        <frame src="test.html">
        <frameset cols="500,100,*">
                <frame src="test.html">
                <frame src="test.html">
                <frame src="test.html">
        </frameset>
</frameset>
```

✓ ***Iframes***

- The <iframe> tag specifies an inline frame.
- An inline frame is used to embed another document within the current HTML document.

```
<iframe src="https://www.google.com"></iframe>
```

- Attributes used:
    - src: Specifies the address of the document to embed in the <iframe>
    - height & width : specifies height and width of iframe in pixels.

Difference between HTML and XHTML

- In HTML, we can have the empty or open tags means it is not required to end the tag e.g. . In XHTML, the tags should be closed or self closed, if opened. for e.g.   or
- In HTML, While defining the attributes it is not necessary to mention quotes. For e.g. <option selected>. In XHTML, while defining the attributes it is mandatory to mention quotes. For e.g. <option selected=”Selected”>.
- In HTML, the values of attributes are not so important. For e.g. <input type=”radiobutton” selected>. In XHTML, the values of attributes are important. For e.g. <input type=”radiobutton” selected=”selected”>.

- ❖ In HTML, there are no strict rules on writng the structure of elements for e.g.   Hello world . In XHTML, there are strict rules on writing structureof elements For e.g. Hello world .
- ❖ In HTML, the tags and attributes can be described in lower case or upper case. In XHTML, the tags and attributes can be described in lower case only.
- ❖ In HTML, one root element is not mandatory. In XHTML, the documents should have one root element.
- ❖ In HTML, XML declaration is not necessary. In XHTML, it is based on the set of rules of XML.

- ✓ **Image map**
  - An image-map is an image with clickable areas.
  - The <map> tag is used to define a client-side image-map.
  - The name attribute of the <map> element is associated with the <img>'s usemap attribute and creates a relationship between the image and the map.
  - The <map> element contains a number of <area> elements, that defines the clickable areas in the image map.
    Example:

```
<img src="planets.gif" usemap="#planetmap">

<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" href="sun.htm" >
  <area shape="circle" coords="90,58,3" href="mercur.htm" >
  <area shape="circle" coords="124,58,8" href="venus.htm">
</map>
```

- ✓ **mailto link**
  - Mailto link is a type of HTML link that activates the default mail client on the computer for sending an e-mail.
  - If you have Microsoft Outlook, for example as your default mail client, pressing a mailto link will open a new mail window.
  - Example:

    Send mail

    With subject and body:
    Email Us

 Send mail

- ✓ Block element vs Inline element
  - A block-level element always starts on a new line and takes up the full width of a page, from left to right.
  - It has a line break before and after the element.
  - Example: ,<h1>..<h6>,<table>,<ol>,<ul>, etc.

- An inline element does not cause a line break (start on a new line) and does not take up the full width of a page.

- Example: <a>,<img>,<b>, etc.

✓ ** and **
- By default, a is a block-level-element and a is an inline element.
- Both are used as content wrappers.
- is often used as a container for some text(usually single word/a line of text).
- The element can be used to style parts of the text.
- The element is often used as a container for other HTML elements.
- The element can be used to style blocks of content.

✓ **Logical and Physical tags**
- Logical tags are designed to describe (to the browser) the enclosed text's meaning.
- For example: By placing text in between you are telling the browser that the text has some greater importance.
  Other examples: ,,<cite> etc.

- Physical tags are used to indicate exactly how specific characters are to be formatted.
- Examples: <b>,<i>,<sub>,<sup>,<u> etc.

CSS

- **CSS (Cascading Style Sheet)** is a language that describes the style of an HTML document.
- CSS describes how HTML elements should be displayed (element's presentation).

We can use style sheets in three different ways in out HTML document. (Levels of Style Sheets)

1) **Inline Style Sheet**
   - It is used to apply a unique style to html elements individually.
   - You can insert inline styles anywhere in the middle of your HTML code.
   - To use inline styles, you use the **style attribute** in the relevant tag. The style attribute can contain any CSS property.
2) **Internal Style Sheet (Document level)**
   - It is used for applying styling to the entire body of the document.
   - You define internal styles in the head section of an HTML page, by using the <style> tag.
3) **External Style Sheet**
   - With an external stylesheet file, you can change the look of an entire website by changing just one file.
   - External CSS is a file that contains only CSS code and is saved with a **".css"** file extension.
   - It is linked to html document using **<link>** element.

[**CSS Comments**: Comments are introduced with /* and terminated with */]

Syntax

- 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.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.
- CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

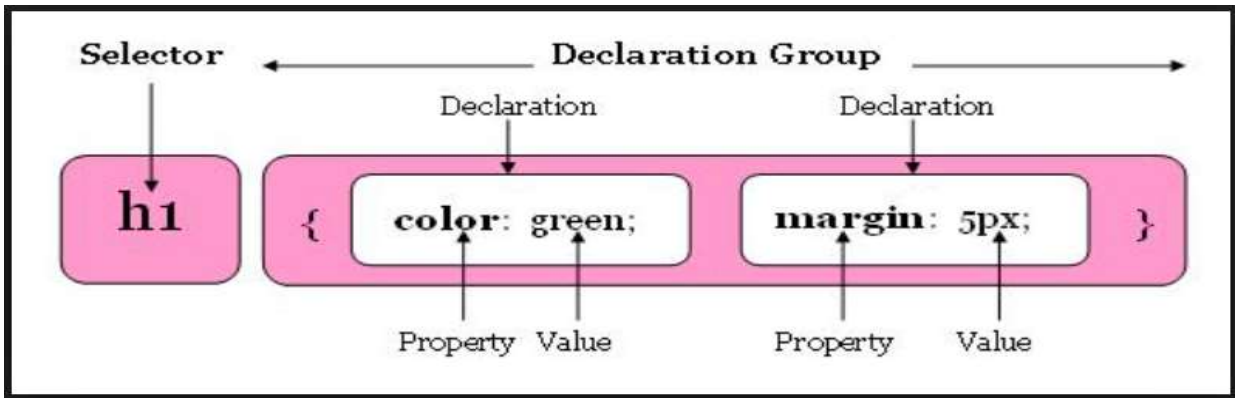


Figure 8: CSS Style Specification Format

CSS Syntax

- Inline
```
<p style="color:red">Content</p>
```
- Internal
```
<head>
<style type="text/css”>
p {color:blue;}
</style>
</head>
```
- External
```
<head>
<link href="style.css”
rel="stylesheet” type="text/css”>
</head>

p {color:green; font:arial, sans-serif;}
```

- The **<link>** tag is used to specify external style sheets.
- The **'href'** attribute is used to specify the URL of the style sheet document.
- Within <link>, the **'rel'** attribute is used to specify the relationship between linked document to the document in which the link appears.

CSS Selectors

- In CSS, selectors are patterns used to select the element(s) you want to style.

Types of Selectors:

1. **Element Selector**
   - The element selector selects elements based on the element name.
   - It selects all elements of the given type within a document.
     Examples:
     h1 { color:red; } →applies red color to all h1 elements
     p { font-size: 24pt;} →applies font size 24pt to all paragraph content.
   - **Group selector:**
     - When several selectors share the same declarations, they may be grouped into a comma-separated list.
     - Example:

       h1, h2, p { color: blue; } →blue color applied to all h1,h2 and p elements.
   - **Descendant selectors:**
     - Used to match an element that is the descendant of another element in the document tree.
       Example: <h1>This headline is very important</h1>
       Then css will be → h1 em { color: blue; }

2. **Class Selector**
   - The CSS class selector matches elements based on the contents of their class attribute.
   - It's declared with a dot preceding a class name.

   Syntax:

```
.class_name { style properties }
```

   Example:

   HTML:

   CSS: .introduction{
   color:red;
   }

   - **Generic Classes :** multiple tags with same class name.

   Example:

```
<h1 class="intro">Introduction heading </h1>
<p class="intro">introduction paragraph.</p>
```

   Then CSS can be applied to each element in the following way:

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

3. **ID Selector**
   - The CSS id selector matches elements based on the contents of their id attribute.
   - An ID selector is declared using a hash, or pound symbol (#) preceding the value of id attribute of html element.
     - Syntax: **#id_value{ style properties}**
     - Example: HTML:

       CSS: #summary { color:green; }

4. **Universal Selector**
   - The CSS universal selector (*) matches elements of any type.

```
/* Selects all elements */
* {
  color: green;
}
```

Pseudo Classes

- A pseudo-class is used to define a special state of an element.
- Pseudo classes are styles that apply when something happens, rather than because the target element simply exists.
- Syntax:

```
selector:pseudo-class {
  property:value;
}
```

- It can be used to:
  - Style an element when a user mouse over it.
  - Style visited and unvisited links differently.
  - Style an element when it gets focus.

:link Use this class to add special style to an unvisited link.	`a:link { color: #FF0000; }`
:visited Use this class to add special style to a visited link.	`/* visited link */ a:visited { color: #00FF00; }`
:hover Use this class to add special style to an element when you mouse over it.	`/* mouse over link */ a:hover { color: #FF00FF; }`
:active Use this class to add special style to an active element.	`/* selected link */ a:active { color: #0000FF; }`
:focus Use this class to add special style to an element while the element has focus.	`input:focus { background-color: yellow; }`

CSS PROPERTY VALUE FORMS

Colors

Colors in CSS can be specified by the following methods:

```
Selector { color: red; }                    /* color keyword   */
Selector { color: rgb(0 255 0); }           /* RGB range 0-255 */
Selector { color: rgb(0% 100% 0%); }  /* RGB range 0%-100% */
Selector { color: #ff6347; }          /* hexadecimal code (#rrggbb) All values must be
                                              between 00 and FF. */
```

- style="background-color:Blue;“ ->> background color
- style="color:Tomato;“ ->> font color

Background properties

background-color:

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

background-image:

The background-image property specifies an image to use as the background of an element.

```
body {
    background-image: url("paper.gif");
}
```

By default, the background-image property repeats an image both horizontally and vertically.

Other properties:

- background-repeat
  - body {
    background-image: url("gradient_bg.png");
    background-repeat: repeat-x; (repeat-y,no- repeat)
    }
- background-position
  - The position of the image is specified by the background-position property:
    - background-position: right top;
  - Background-size
    - :width height (in pixel or %)
    - Cover/contain/auto/initial

cover	Resize the background image to cover the entire container, even if it has to stretch the image or cut a little bit off one of the edges
contain	Resize the background image to make sure the image is fully visible

left top
left center
left bottom
right top
right center
right bottom
center top
center center
center bottom

Font Properties

Property	Description
font	Sets all the font properties in one declaration
font-family	Specifies the font family for text
font-size	Specifies the font size of text
font-style	Specifies the font style for text
font-variant	Specifies whether or not a text should be displayed in a small-caps font
font-weight	Specifies the weight of a font

Examples:

font-family: 'Times New Roman', Times, serif;

font-size: 2px / font-size: 50% / font-size:large / font-size: 2em (1 em=16px) / font-size:24pt etc.

font-style:italic; [Other 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)

font-variant:small-caps; [or font-variant:normal;]

font-weight:bold [Other values : normal/lighter/lightest/bolder or from 100-900 in multiple of 100]

Font shorthand:

Syntax: font: font-style font-variant font-weight font-size font-family;

Example → font: bold 14pt 'Times New Roman' ;

Text properties

text-align	Specifies the horizontal alignment of text
text-decoration	Specifies the decoration added to text
text-indent	Specifies the indentation of the first line in a text-block

Examples:

text-align: left; [Other values : right/center/justify]

text-decoration:overline; [Other values: line-through/underline/none]

text-indent:2%; [Other units: em/ px]

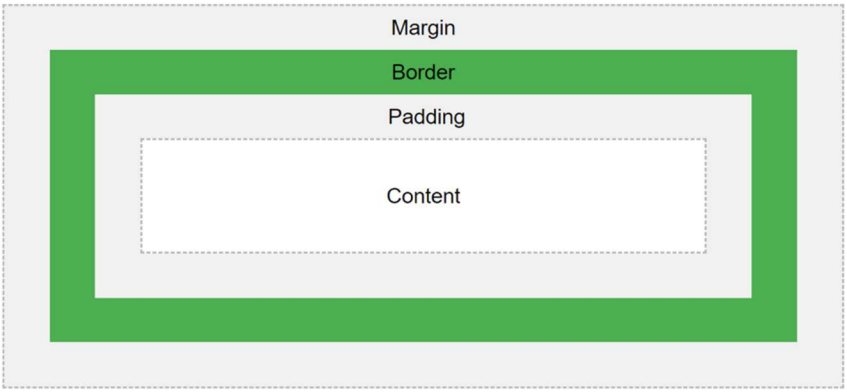
Height and width

- The height and width properties are used **to set the height and width of an element**.
- The height and width can be set to **auto** (this is default. Means that the browser calculates the height and width), or be specified in **length values, like px, cm, etc**., or in percent **(%)** of the containing block.

```
div {
    height: 100px;
    width: 500px;
    background-color: powderblue;
}
```

CSS BOX MODEL

- All HTML elements can be considered as boxes.
- The CSS box model is essentially a box that wraps around every HTML element.
- It consists of: margins, borders, padding, and the actual content.
- It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model.



- **Content** - The content of the box, where text and images appear.
- **Padding** - Clears an area around the content. The padding is transparent.
- **Border** - A border that goes around the padding and content.
- **Margin** - Clears an area outside the border. The margin is transparent.

CSS Border

The CSS border properties allow you to specify the style, width, and color of an element's border.

a. Border style

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

border-style:none; defines no border

b. Border width

The `border-width` property specifies the width of the four borders.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

The `border-width` property can have from one to four values (for the top border, right border, bottom border, and the left border).

```
p.one {
    border-style: solid;
    border-width: 5px;
}

p.two {
    border-style: solid;
    border-width: medium;
}

p.three {
    border-style: solid;
    border-width: 2px 10px 4px 20px;
}
```

c. **Border color**

The border-color property is used to set the color of the four borders.
Example: border-color:red;

d. **Border shorthand**

To specify all the individual border properties together.

```
p {
    border: 5px solid red;
}
```

- ✓ You can also specify all the individual border properties for just one side:
- ✓ Example→ border-left: 6px solid red;

CSS Margin

- ➢ The CSS margin properties are used to create space around elements, outside of any defined borders.

CSS has properties for specifying the margin for each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

- Values: **auto** - the browser calculates the margin || **length** - specifies a margin in px, pt, cm, etc. or

specifies a margin in % of the width of the containing element.

CSS Padding

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
  - `padding-top`
  - `padding-right`
  - `padding-bottom`
  - `padding-left`

- Example: padding: 25px 50px 75px;
  top padding is 25px
  right and left paddings are 50px
  bottom padding is 75px
  padding: 25px 50px;
  top and bottom paddings are 25px
  right and left paddings are 50px

List properties

- The CSS list properties allow you to:
  - Set different list item markers for ordered lists.
  - Set different list item markers for unordered lists.
  - Set an image as the list item marker.
  - Add background colors to lists and list items.

  - For Unordered list
    - list-style-type: circle/square/disc/none;
    - list-style-image: url('image path');
  - For unordered list
    - list-style-type: decimal/upper-alpha/lower-alpha/upper-roman/lower-roman
  - ✓ Other properties that can be applied: color,background-color,padding etc.
  - ✓ Examples:

```
ol {
  background: #ff9999;
  padding: 20px;
}

ul {
  background: #3399ff;
  padding: 20px;
}

ol li {
  background: #ffe5e5;
  padding: 5px;
  margin-left: 35px;
}

ul li {
  background: #cce5ff;
  margin: 5px;
}
```

Table properties

- With CSS table properties, you can set how tables and table columns should be displayed.

1. Table border
   - To specify table borders in CSS, use the **border** property.
     Example:
     ```
     table, th, td {
           border: 1px solid black;
     }
     ```
   - The **border-collapse** property sets whether the table borders should be collapsed into a single border or not.

     Example:

     ```
     table {
       border-collapse: collapse;
     }
     ```

2. Table height & width
   - Width and height of a table are defined by the width and height properties.
     Example:
     ```
     table {
       width: 100%;
     }

     th {
     ```

```
  height: 50px;
}
```

3. Text align
   - For the alignment of text either horizontally or vertically.
   - The **text-align** property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.
   - The **vertical-align** property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>.

   Examples:

```
th {
  text-align: left;
}
td {
  vertical-align: bottom;
}
```

4. Table padding and spacing
   - To control the space between the border and the content in a table, use the **padding** property on <td> and <th> elements
   - The **border-spacing** property sets the distance between the borders of adjacent cells.

   Example:

```
th, td {
  padding: 15px;
}
#table2 {
    border-spacing: 15px 50px;
}
```

   //Using two values (the first sets the horizontal spacing and the second sets the vertical spacing)
   //If single value it sets both the horizontal spacing and the vertical spacing

5. Table color

   Example:

```
th {
   background-color: #4CAF50;
   color: white;
}
```

6. Table caption side
   - Specify the placement of table captions

   Example:

```
#table1 {
caption-side: top;

}
```

```
#table2 {
      caption-side: bottom;

}
```

CSS POSITION

- The position property specifies the type of positioning method used for an element.
- There are five different position values:
  - static
  - relative
  - fixed
  - absolute
  - sticky
  - ✓ Elements are then positioned using the top, bottom, left, and right properties.

Static position

- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- An element with position: static; is not positioned in any special way, it is always positioned according to the normal flow of the page.

Relative position

- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.
- Other content will not be adjusted to fit into any gap left by the element.

Fixed position

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.
- The top, right, bottom, and left properties are used to position the element.

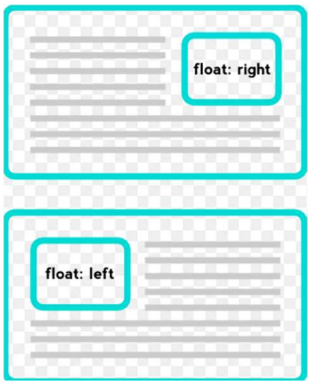
Absolute position

- An element with position: absolute; is positioned relative to the nearest positioned ancestor.

CSS FLOAT

- The float property is used for positioning and formatting html content e.g. let an image float left to the text in a container.
- The float property can have one of the following values:
  - left - The element floats to the left of its container
  - right- The element floats to the right of its container
  - none - The element does not float (will be displayed just where it occurs in the text).

  - ✓ the float property can be used to wrap text around images.



- The clear property specifies what elements can float beside the cleared element and on which side.

- The clear property can have one of the following values:

  none - Allows floating elements on both sides. This is default

  left - No floating elements allowed on the left side

  right- No floating elements allowed on the right side

  both - No floating elements allowed on either the left or the right side