

Unit -1

HTML

Prepared by:
Dr. Susheela

HTML

- Hyper Text Markup Language
- HTML was first created by Tim Berners-Lee, Robert Cailliau, and others starting in 1989
- HTML is the standard markup language for creating Web pages
- HTML elements tell the browser how to display the content
- Hypertext means that the document contains links that allow the reader to jump to other places in the document or to another document altogether.
- A Markup Language is a way that computers speak to each other to control how text is processed and presented. To do this HTML uses two things: tags and attributes.
- The latest version is known as HTML5.

HTML Tags

- HTML tags are used to **mark up the start of an HTML element** and they are usually enclosed in angle brackets. An example of a tag is: <h1>.
- Most tags must be opened <h1> and closed </h1> in order to function.
- When using multiple tags, the tags must be closed in the order in which they were opened. For example:

```
<body>  
<p>This is really important!  
</p>  
</body>
```

HTML Tag's Attributes

- Attributes contain additional pieces of information. Attributes take the form of an opening tag and additional info is placed inside.
- An example of an attribute is:

```

```

HTML Editors

- Notepad++
 - Auto-completion feature
 - No support for Mac.
- TextEdit (Mac)
- Sublime Text 3
 - Easily customizable
 - Beginner-friendly
 - Pleasant color schemes to choose from
 - Available for Mac, Windows, and Linux
 - Can't print documents or code
 - No toolbar or dashboard available.
- Komodo Edit
 - Available for Mac, Windows, and Linux
 - Impressive language support
 - No autocompletion by default

Basic Construction of an HTML Page

- <!DOCTYPE html> — This tag **specifies the language** you will write on the page (HTML 5).
- <html> — This tag signals that from here on we are going to write in HTML code.
 - There is one tag that is always included: <title>
- <head> — This is where all the **metadata for the page** goes — stuff mostly meant for search engines and other computer programs.
- <body> — This is where the **content of the page** goes.
- The HTML <body> is where we add the content which is designed for viewing by human eyes.

```
<!DOCTYPE html>
<html>
<head>
<title> Title of Page </title>
</head>
<body>

<h1>First Heading</h1>
<p>paragraph</p>

</body>
</html>
```

HTML 5.0

- HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).
- HTML5 introduces a number of new elements and attributes that can help you in building modern websites.
- HTML5 is designed, as much as possible, to be backward compatible with existing web browsers.
- HTML5 comes with a lot of flexibility, and it supports the following features –
 - Uppercase tag names.
 - Quotes are optional for attributes.
 - Attribute values are optional.
 - Closing empty elements are optional.

Dynamic Page Support

Now a days there is need of dynamic as well as interactive websites rather than static websites. Here are enormous features that provide the dynamic touch to the website :

- time – This helps in adding current time as well as date to the webpage.
- meter – It helps in indicating that how much space in the storage disk is still there.
- progress bar – It helps in knowing the progress of the task that has been assigned for its completion.

New Elements in HTML 5

- Navigation Tag <nav>
- Header Tag <header>
- Footer Tag <footer>
- Section Tag <section>
- Article Tag <article>
- Main Tag <main>
- Figure Tag <figure>
- Figure caption Tag <figcaption>
- Mark Tag <mark>
- Progress Tag <progress>
- Sematic Tag <sem>
- Audio Tag <audio>
- Video Tags <video>
- Media Control
- Scalable Vector Graphics Tag <svg>
- Canvas Tag <canvas>

HTML	HTML5
HTML does not provide native audio and video support.	HTML5 provides native audio and video support.
HTML only supports vector graphics if used in conjunction with different technologies like <u>Flash</u> , <u>VML</u> , or <u>Silverlight</u> .	HTML5 supports SVG (Scalable Vector Graphics), Canvas, and other virtual vector graphics.
HTML doesn't allow users to draw shapes such as circles, triangles, and rectangles.	HTML5 allows users to draw shapes such as circles, triangles, and rectangles.
HTML only uses browser cache and cookies to store data temporarily.	HTML5 uses web SQL databases, local storage, and application cache for storing data temporarily.
JavaScript and browser interface run in the same thread.	JavaScript and browser interface run in separate threads.
Longer <u>document type declaration</u> .	Shorter document type declaration.
Longer character encoding declaration. Uses the ASCII <u>character set</u> .	Shorter <u>character encoding</u> declaration. Uses the UTF-8 character set.
Compatible with almost all browsers.	Only compatible with newer browsers, considering there are many new tags and elements which only some browsers support.
Built based on <u>Standard Generalized Markup Language (SGML)</u> .	HTML5 has improved parsing rules providing enhanced compatibility.
Programmers are unable to use features that determine a user's geolocation..	HTML5 has a JavaScript geolocation API, which can be used to identify any user's location when accessing the website.
It cannot handle inaccurate syntax.	It is capable of handling inaccurate syntax.

Semantic Elements in HTML5

- Semantic elements in HTML refers to the tags that provide meaning to an HTML page rather than just presentation.
- It makes HTML more comprehensible by better defining the different sections and layout of web pages.
- The semantic HTML tags help the search engines and other user devices to determine the importance and context of web pages.
- The pages made with semantic elements are much easier to read.
- It has greater accessibility. It offers a better user experience.



Navigation <nav> Tag

```
<!DOCTYPE html>
<html>
<body>
<nav>
  <a href="/html/">HTML</a>
  <a href="/css/">CSS</a>
  <a href="/js/">JavaScript</a>
  <a href="/jquery/">jQuery</a>
</nav>

</body>
</html>
```

[HTML](#) [CSS](#) [JavaScript](#) [jQuery](#)

Article Tag <article>

- The <article> tag specifies independent, self-contained content.
- An article should make sense on its own and it should be possible to distribute it independently from the rest of the site.
- Potential sources for the <article> element:
 - Forum post
 - Blog post
 - News story

The <article> element does not render as anything special in a browser. However, you can use CSS to style the <article> element (see example below).

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

<h1> Article Tag</h1>

<article>
<h2>Google Chrome</h2>
<p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!</p>
</article>

<article>
<h2>Mozilla Firefox</h2>
<p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.</p>
</article>

</body>
</html>
```

Article Tag

Google Chrome

Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!

Mozilla Firefox

Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.

Header Tag <header>

- HTML **<header> tag** is used as a container of introductory content or navigation links. Generally a <header> element contains one or more heading elements, logo or icons or author's information.
- You can use several <header> elements in one document, but a <header> element cannot be placed within a <footer>, <address> or another <header> element.

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

<h1> Header Tag</h1>

<header>
  <h1> Heading..... </h1>
  <p>Use of header tag</p>
  <p>Some additional information here</p>
</header>
|
</body>
</html>
```

Header Tag

Heading.....

Use of header tag

Some additional information here

Footer Tag <footer>

- The <footer> tag defines a footer for a document or section.
- A <footer> element typically contains:
 - authorship information
 - copyright information
 - contact information
 - sitemap
 - back to top links
 - related documents

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

<h1>The Footer Tag</h1>

<footer>
  <address>
    Written by <a href="mailto:susheela.dahiya@gehu.ac.in">
      Dr. Susheela Dahiya</a><br>
    Visit us at:<br>
    Example.com<br>
    Dehradun<br>
    India
  </address>
</footer>

</body>
</html>
```

The Footer Tag

Written by [Dr. Susheela Dahiya](#)
Visit us at:
Example.com
Dehradun
India

Main Tag <main>

- HTML <main> tag is used to represent the main content of the <body> tag.
- The <main> tag is written within <body> tag. It is used to accurately describe the primary content of a page.
- The content of the main tag is directly related to the central topic of the document.
- Author should not include more than one <main> tag within a document.
- The <main> element should not be used as a child of an <article>, <aside>, <header>, <footer>, or <nav> element.

Mark Tag <mark>

- HTML <mark> tag is used to highlight the text

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

<p>
<h2>This is an example of <mark> Mark Tag </mark> </h2>
</p>

</body>
</html>
```

This is an example of **Mark Tag**

Progress Tag

- The <progress> tag represents the completion progress of a task.
- Always add the <label> tag for best accessibility practices

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

<h1>The Progress Tag</h1>

<label for="file">Downloading progress:</label>
<progress id="file" value="30" max="100"> 30% </progress>

</body>
</html>
```

The Progress Tag

Downloading progress:



Figure & Figure Caption Tag

- 1. HTML <figure> tag** is used to mark up a photo in the document on a web page.
- 2. HTML <figcaption> tag** is used to add a caption to a photo.

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

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

</body>
</html>
```



Fig.1 - Trulli, Puglia, Italy.

Figure & Figure Caption Tag

1. **HTML <figure> tag** is used to mark up a photo in the document on a web page.
2. **HTML <figcaption> tag** is used to add a caption to a photo.

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

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

</body>
</html>
```



Fig.1 - Trulli, Puglia, Italy.

<video> tag

- HTML 5 supports only mp4 & ogg audio/video
- The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.
- You can set up different sources, and the first source that fits the preferences is the one being used.
- Attributes:
 - Controls - Adds video controls, like play, pause, and volume.
 - Preload - auto, none
 - width and height - If height and width are not set, the page might flicker while the video loads.
 - Add muted after autoplay to let your video start playing automatically (but muted)

```
<!DOCTYPE html>
<html>
<body>
<h2>Example of video and audio tag</h2>

<video controls preload="none">
  <source src="movie.mp4" type="video/mp4">
    Your browser does not support the video tag.
</video>

</body>
</html>
```

Example of video and audio tag



```
<!DOCTYPE html>
<html>
<body>
<h2>Example of video and audio tag</h2>

<video controls preload="auto">
  <source src="movie.mp4" type="video/mp4">
    Your browser does not support the video tag.
</video>

</body>
</html>
```

Example of video and audio tag



Audio Tag

```
<!DOCTYPE html>
<html>
<body>
<h2>Example of audio tag</h2>

<audio controls autoplay>
  <source src = "/html5/audio.ogg" type = "audio/ogg" />
  <source src = "/html5/audio.wav" type = "audio/wav" />
  Your browser does not support the audio element.
</audio>
</body>
</html>
```

Example of audio tag



```
<!DOCTYPE html>
<html>
<body>
<h2>Example of video and audio tag</h2>

<video controls preload="none">
  <source src="movie.mp4" type="video/mp4">
    Your browser does not support the video tag.
</video>

</body>
</html>
```

Example of video and audio tag



```
<!DOCTYPE html>
<html>
<body>
<h2>Example of video and audio tag</h2>

<video controls preload="auto">
  <source src="movie.mp4" type="video/mp4">
    Your browser does not support the video tag.
</video>

</body>
</html>
```

Example of video and audio tag



Scalable Vector Graphics <svg> Tag

```
<!DOCTYPE html>
<html>
<body>
<h2> Rounded Rectangle </h2>
<svg width="500" height="500">
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)">
    <rect x="50" y="20" rx="20" ry="20" width="150" height="150" style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
  Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

Rounded Rectangle



```
<!DOCTYPE html>
<html>
<body>
<h2> Rectangle </h2>
<svg width="400" height="100">
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />
  Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

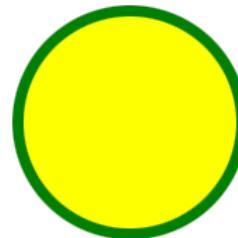
Rectangle



```
<!DOCTYPE html>
<html>
<body>
<h2> Circle </h2>
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"
fill="yellow" />
Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

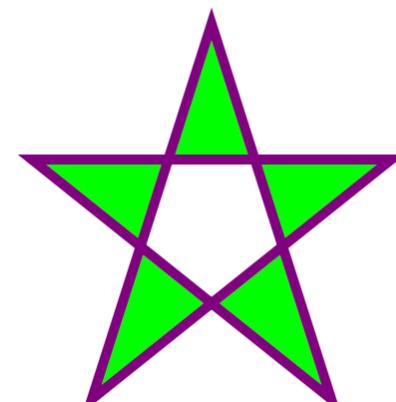
Circle



```
<!DOCTYPE html>
<html>
<body>
<h2> Star </h2>
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

Star



HTML Text Formatting Tags

Element	Meaning	Purpose
	Bold	Highlight important information
	Strong	Similarly to bold, to highlight key text
<i>	Italic	To denote text
	Emphasised Text	Usually used as image captions
<mark>	Marked Text	Highlight the background of the text
<small>	Small Text	To shrink the text
<strike>	Striked Out Text	To place a horizontal line across the text
<u>	Underlined Text	Used for links or text highlights
<ins>	Inserted Text	Displayed with an underline to show an inserted text
<sub>	Subscript Text	Typographical stylistic choice
<sup>	Superscript Text	Another typographical presentation style
		Defines deleted text

```
<!DOCTYPE html>
<html>
<body>
<p>This text is normal.</p>
<p><b>This text is bold</b></p>
<p><strong>This text is strong</strong></p>
<p><i>This text is italic</i></p>
<p><u>This text is underline</u></p>
<p><s>This text is strike through</s></p>
<p><small>This is some smaller text.</small></p>
<p>Do not forget to buy <mark>fruits</mark> today.</p>
<p>My favorite color is <del>blue</del> red.</p>
<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
<p>This is <sub> subscript </sub> and <sup>superscript</sup></p>
<p><em>This text is emphasized.</em></p>
</body>
</html>
```

This text is normal.

This text is bold

This text is strong

This text is italic

This text is underline

This text is strike through

This is some smaller text.

Do not forget to buy fruits today.

My favorite color is blue red.

My favorite color is blue red.

This is subscript and superscript

This text is emphasized.

HTML Elements

The HTML **element** is everything from the start tag to the end tag:

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

Nested HTML Elements

- HTML elements can be nested

```
<!DOCTYPE html>
<html>
<head>
<title> Title of Page </title>
</head>
<body>
    <h1>First Heading</h1>
    <p>paragraph</p>

</body>
</html>
```

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

Style Attribute

- The style attribute is used to add styles to an element, such as color, font, size, etc.
- HTML supports 140 standard color names.

```
<tagname style="property:value;">

<p style="color:red;">This is a red paragraph.</p>
<body style="background-color:powderblue;">
<h1 style="background-color:powderblue;">This is a heading</h1>
<p style="background-color:green;">This is a paragraph.</p>
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
<h1 style="border:2px solid Red;">Hello World</h1>
```

```
<head>
<style> /* This is internal styling */
h1
{
color: indianred;
}
p
{
color: red;
}
</style>
</head>
```

HTML Headings

- HTML headings are defined with the `<h1>` to `<h6>` tags.
- `<h1>` defines the most important heading. `<h6>` defines the least important heading.
- `<h1>Heading 1</h1>`
`<h2>Heading 2</h2>`
`<h3>Heading 3</h3>`
`<h4>Heading 4</h4>`
`<h5>Heading 5</h5>`
`<h6>Heading 6</h6>`

HTML Line Breaks

- The HTML
 element defines a line break.
- Use
 if you want a line break (a new line) without starting a new paragraph:

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

- The
 tag is also an empty tag, means that it has no end tag.

This is
a paragraph
with line breaks.

HTML Paragraphs

- A paragraph always starts on a new line and is usually a block of text.
- <p> tag defines a paragraph.
- <p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>
- <p>
The number of lines in a paragraph depends on the size of the browser window. If you resize the browser window, the number of lines in this paragraph will change.
</p>

HTML Horizontal Rules

- The `<hr>` tag defines a thematic break in an HTML page and is most often displayed as a horizontal rule.
- The `<hr>` tag is an empty tag means that it has no end tag
- The `<hr>` element is used to separate content (or define a change) in an HTML page:

```
<h1>This is heading 1</h1>
```

```
<p>This is some text.</p>
```

```
<hr>
```

```
<h2>This is heading 2</h2>
```

```
<p>This is some other text.</p>
```

```
<hr>
```

```
<h2>This is heading 2</h2>
```

```
<p>This is some other text.</p>
```

This is heading 1

This is some text.

This is heading 2

This is some other text.

This is heading 2

This is some other text.

 Tag

```

```

1. Absolute URL - Links to an external image that is hosted on another website. Example: src="https://www.gehu.ac.in/images/img1.jpg".

- External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.

2. Relative URL - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page. Example: src="img_girl.jpg". If the URL begins with a slash, it will be relative to the domain. Example: src="/images/img_girl.jpg".

- It is almost always best to use relative URLs. They will not break if you change domain.

HTML Comment Tag

- HTML Comment Tag can help document the source code.
- Comments can sometimes also used to hide content.

Syntax:

```
<!-- Write your comments here -->
```

Example:

```
<!DOCTYPE html>
<html>
<body>
<!-- This is a comment -->
<p>This is a paragraph.</p>
<!-- Comments are not displayed in the browser -->
</body>
</html>
```

HTML Links - Hyperlinks

- Links allow users to click their way from page to page.
- [tag defines a hyperlink.](#)
- Syntax:
`Link text`
- By default, links will appear as follows in all browsers:
 - An unvisited link is underlined and blue
 - A visited link is underlined and purple
 - An active link is underlined and red
- The target attribute specifies where to open the linked document.
The target attribute can have one of the following values:
 - `_self` - Default. Opens the document in the same window/tab as it was clicked
 - `_blank` - Opens the document in a new window or tab
 - `_parent` - Opens the document in the parent frame
 - `_top` - Opens the document in the full body of the window
- `Visit GEHU!`

HTML Link Colors

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link {
  color: green;
  background-color: transparent;
}

a:visited {
  color: pink;
  background-color: transparent;
}

a:hover {
  color: red;
  background-color: transparent;
}

a:active {
  color: yellow;
  background-color: transparent;
}
</style>

<h2>Link Colors</h2>
<p>You can change the default colors of links</p>
<a href="html_images.asp" target="_blank">HTML Images</a>

</body>
</html>
```

Link Colors

You can change the default colors of links

HTML Images

Link Colors

You can change the default colors of links

HTML Images

- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked

Lists

- Ordered Lists
- Unordered Lists
- Definition List <dl>

Tag	Description
<u></u>	Defines an unordered list
<u></u>	Defines an ordered list
<u></u>	Defines a list item
<u><dl></u>	Defines a description list
<u><dt></u>	Defines a term in a description list
<u><dd></u>	Describes the term in a description list

Lists

- Ordered Lists
 - Unordered Lists
 - Definition List <dl>
-

```
<h2>An ordered HTML list</h2>
```

```
<ol>
  <li>An item </li>
  <li>Another item </li>
  <li>Another goes here </li>
</ol>
```

```
<h2>An unordered HTML list</h2>
```

```
<ul>
  <li>An item </li>
  <li>Another item </li>
  <li>Another goes here </li>
</ul>
```

An ordered HTML list

1. An item
2. Another item
3. Another goes here

An unordered HTML list

- An item
- Another item
- Another goes here

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

<h2>Definition list</h2>

<dl>
  <dt>Item</dt>
  <dd>The definition goes here
</dl>

</body>
</html>
```

Definition list

Item

The definition goes here

Unordered List

Value	Description	
disc	Sets the list item marker to a bullet (default)	<pre><ul type="circle"> Coffee Tea Milk </pre> <ul style="list-style-type: none">◦ Item 1◦ Item 2◦ Item 2
circle	Sets the list item marker to a circle	
square	Sets the list item marker to a square	
none	The list items will not be marked	<pre>SQUARE ■ Item 1 ■ Item 2 ■ Item 2</pre> <pre>NONE Item 1 Item 2 Item 2</pre>

Unordered List with Disc Bullets

```
<ul style="list-style-type:disc;">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 2</li>
</ul>
```

Unordered List with Disc Bullets

- Item 1
- Item 2
- Item 2

Ordered List

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

<h2>Ordered List with Numbers</h2>

<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```

Ordered List with Numbers

- 1. Coffee
- 2. Tea
- 3. Milk

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

- A. Coffee
- B. Tea
- C. Milk

- I. Coffee
- II. Tea
- III. Milk

```
<ol type="A" start="E">
  <li> Coffee </li>
  <li> Tea </li>
  <li> Milk </li>
</ol>
```

OUTPUT
E. Coffee
F. Tea
G. Milk

Nested List

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

<h2>A Nested List</h2>
<p>Lists can be nested (list inside list):</p>

<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>

</body>
</html>
```

A Nested List

Lists can be nested (list inside list):

- Coffee
- Tea
 - Black tea
 - Green tea
- Milk

Table Tag

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

Table Tag

Cell spacing	<code><table cellspacing="">...</table></code>
Cell padding	<code><table cellpadding="">...</table></code>
Table border	<code><table border="">...</table></code>
Alignment	<code><table align=center/left/right>...</table></code>
colspan in table	<code><table colspan="">...</table></code>
rowspan in table	<code><table rowspan="">...</table></code>
Cell color	<code><table bgcolor="#\$\$\$\$\$\$">...</table></code>
No linebreaks	<code><table nowrap>...</table></code>

```
<!DOCTYPE html>
<html>
<head> <title> Table Tag </title> </head>
<body>
<table>
<tr>
  <th>Name</th>
  <th>Job</th>
  <th>Working Experience</th>
</tr>
<tr>
  <td>John</td>
  <td>Software Engineer</td>
  <td>5 Years</td>
</tr>
<tr>
  <td>Ale</td>
  <td>Senior Web developer</td>
  <td>2 Year</td>
</tr>
<tr>
  <td>Jack</td>
  <td>Junior Tech Writer</td>
  <td>6 Months</td>
</tr>
</table>
</body>
</html>
```

Name	Job	Working Experience
John	Software Engineer	5 Years
Ale	Senior Web developer	2 Years
Jack	Junior Tech Writer	6 Months

Table Tag

```
<!DOCTYPE html>
<html>
<head> <title> Table Tag </title> </head>
<body>
<table border="1">
  <tr>
    <th>Name</th>
    <th>Job</th>
    <th>Working Experience</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Software Engineer</td>
    <td>5 Years</td>
  </tr>
  <tr>
    <td>Ale</td>
    <td>Senior Web developer</td>
    <td>2 Year</td>
  </tr>
  <tr>
    <td>Jack</td>
    <td>Junior Tech Writer</td>
    <td>6 Months</td>
  </tr>
</table>
```

Name	Job	Working Experience
John	Software Engineer	5 Years
Ale	Senior Web developer	2 Year
Jack	Junior Tech Writer	6 Months

cellspacing Attribute

```
<html>
<head> <title> Table Tag </title> </head>
<body>
<table border="1" cellspacing = "10">
  <tr>
    <th>Name</th>
    <th>Job</th>
    <th>Working Experience</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Software Engineer</td>
    <td>5 Years</td>
  </tr>
  <tr>
    <td>Ale</td>
    <td>Senior Web developer</td>
    <td>2 Year</td>
  </tr>
  <tr>
    <td>Jack</td>
    <td>Junior Tech Writer</td>
    <td>6 Months</td>
  </tr>
</table>
</body>
</html>
```

Name	Job	Working Experience
John	Software Engineer	5 Years
Ale	Senior Web developer	2 Year
Jack	Junior Tech Writer	6 Months

```
<html>
<head> <title> Table Tag </title> </head>
<body>
<table border = 5>
  <tr>
    <th>Name</th>
    <th colspan="2">Jobs</th>
    <th>Working Experience</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Software Engineer</td>
    <td>Data Analyst</td>
    <td rowspan="2">5 Years</td>
  </tr>
  <tr>
    <td>Ale</td>
    <td colspan="2">Senior Web developer</td>
  </tr>
  <tr>
    <td>Jack</td>
    <td>Junior Tech Writer</td>
    <td>Blogger</td>
    <td>6 Months</td>
  </tr>
</table>
</body>
</html>
```

Name	Jobs	Working Experience
John	Software Engineer	Data Analyst
Ale	Senior Web developer	5 Years
Jack	Junior Tech Writer	Blogger
		6 Months

Screensh

```
<body>
<table border = 1>
  <tr>
    <th>Name</th>
    <th colspan="2">Jobs</th>
    <th>Working Experience</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Software Engineer</td>
    <td>Data Analyst</td>
    <td>5 Years</td>
  </tr>
  <tr>
    <td>Ale</td>
    <td colspan="2">Senior Web developer</td>
    <td>2 Year</td>
  </tr>
  <tr>
    <td>Jack</td>
    <td>Junior Tech Writer</td>
    <td>Blogger</td>
    <td>6 Months</td>
  </tr>
</table>
</body>
```

Name	Jobs	Working Experience
John	Software Engineer	Data Analyst
Ale	Senior Web developer	2 Year
Jack	Junior Tech Writer	Blogger

```
<body>





```

New Employees Records

Name	Jobs	Working Experience
John	Software Engineer	Data Analyst 5 Years
Ale	Senior Web developer	
Jack	Junior Tech Writer	Blogger 6 Months

```

<body>
<table border = "10">
  <caption>
    <b>New Employees Records</b>
  </caption>
  <tr>
    <th>Name</th>
    <th colspan="2">Jobs</th>
    <th>Working Experience</th>
  </tr>
  <tr>
    <td bgcolor = "Green">John</td>
    <td>Software Engineer</td>
    <td>Data Analyst</td>
    <td rowspan="2">5 Years</td>
  </tr>
  <tr>
    <td>Ale</td>
    <td colspan="2" bgcolor = "Red">Senior Web developer</td>
  </tr>
  <tr>
    <td>Jack</td>
    <td>Junior Tech Writer</td>
    <td>Blogger</td>
    <td>6 Months</td>
  </tr>
</table>
</body>

```

Name	Jobs	Working Experience
John	Software Engineer	Data Analyst
Ale	Senior Web developer	5 Years
Jack	Junior Tech Writer	Blogger
		6 Months

```

<body>
<table border = "10">
  <caption>
    <b>New Employees Records</b>
  </caption>
  <tr>
    <th>Name</th>
    <th colspan="2">Jobs</th>
    <th>Working Experience</th>
  </tr>
  <tr id="eligible" bgcolor="#0bb31e">
    <td>John</td>
    <td>Software Engineer</td>
    <td>Data Analyst</td>
    <td rowspan="2">5 Years</td>
  </tr>
  <tr>
    <td>Ale</td>
    <td colspan="2" bgcolor = "Red">Senior Web developer</td>
  </tr>
  <tr>
    <td>Jack</td>
    <td>Junior Tech Writer</td>
    <td>Blogger</td>
    <td>6 Months</td>
  </tr>
</table>
</body>

```

New Employees Records

Name	Jobs	Working Experience
John	Software Engineer	Data Analyst
Ale	Senior Web developer	5 Years
Jack	Junior Tech Writer	Blogger
		6 Months

Align =
left, right, center

```
<table border = "10">
<caption>
  <b>New Employees Records</b>
</caption>
<tr>
  <th>Name</th>
  <th colspan="2">Jobs</th>
  <th>Working Experience</th>
</tr>
<tr id="eligible" bgcolor="#0bb31e">
  <td>John</td>
  <td>Software Engineer</td>
  <td>Data Analyst</td>
  <td>5 Years</td>
</tr>
<tr>
  <td>Ale</td>
  <td colspan="2" bgcolor = "Red">Senior Web developer</td>
  <td>7 Years</td>
</tr>
<tr>
  <td>Jack</td>
  <td>Junior Tech Writer</td>
  <td>Blogger</td>
  <td align = "center">6 Months</td>
</tr>
</table>
```

New Employees Records			
Name	Jobs		Working Experience
John	Software Engineer	Data Analyst	5 Years
Ale	Senior Web developer		7 Years
Jack	Junior Tech Writer	Blogger	6 Months

```





```

New Employees Records

Name	Jobs		Working Experience
John	Software Engineer	Data Analyst	5 Years
Ale	Senior Web developer		7 Years
Jack	Junior Tech Writer	Blogger	6 Months

New Employees Records

Name	Jobs		Working Experience
John	Software Engineer	Data Analyst	5 Years
Ale	Senior Web developer		7 Years
Jack	Junior Tech Writer	Blogger	6 Months

```
<table>
  <caption>Nested Tables</caption>
  <tr>
    <th>Header of Table 1</th>
    <th>Header of Table 2</th>
  </tr>
  <tr>
    <td>
      <table>
        <tr>
          <th>1st Header of nested table 1</th>
          <th>2nd Header of nested table 1</th>
        </tr>
        <tr>
          <td>1st cell of nested table</td>
          <td>2nd cell of nested table</td>
        </tr>
        <tr>
          <td>3rd cell of nested table</td>
          <td>4th cell of nested table</td>
        </tr>
      </table>
    </td>
    <td>
      <table>
        <tr>
          <th>1st Header of nested table 2</th>
          <th>2nd Header of nested table 2</th>
        </tr>
        <tr>
          <td>1st cell of nested table</td>
          <td>2nd cell of nested table</td>
        </tr>
        <tr>
          <td>3rd cell of nested table</td>
          <td>4th cell of nested table</td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```

Nested Tables

Header of Table 1

1st Header of nested table 1 2nd Header of nested table 1

1st cell of nested table 2nd cell of nested table
3rd cell of nested table 4th cell of nested table

Header of Table 2

1st Header of nested table 2 2nd Header of nested table 2

1st cell of nested table 2nd cell of nested table
3rd cell of nested table 4th cell of nested table

HTML <form>

- <form> is a HTML tag to collect input data with interactive controls.
- It provides facilities to input text, number, values, email, password, and control fields such as checkboxes, radio buttons, submit buttons, etc., or in other words, form is a container that contains input elements like text, email, number, radio buttons, checkboxes, submit buttons, etc.
- Forms are generally used when you want to collect data from the user.
- For example, a user wants to register for an event, so he/she has to first enter name, mail id, contact number etc. in the registration form to get registered.

Form Tags

Tag	Description
<form>	It defines an HTML form to enter inputs by the user side.
<input>	It defines an input control.
<textarea>	It defines a multi-line input control.
<label>	It defines a label for an input element.
<fieldset>	It groups the related elements in a form.
<legend>	It defines a caption for a <fieldset> element.
<select>	It defines a drop-down list.
<optgroup>	It defines a group of related options in a drop-down list.
<option>	It defines an option in a drop-down list.
<button>	It defines a clickable button.
<datalist>	It specifies a list of pre-defined options for input control.
<keygen>	It defines a key-pair generator field for forms.
<output>	It defines the result of a calculation.

<input> tag Attributes

1	type Indicates the type of input control and for text input control it will be set to text .
2	name Used to give a name to the control which is sent to the server to be recognized and get the value.
3	value This can be used to provide an initial value inside the control.
4	size Allows to specify the width of the text-input control in terms of characters.
5	maxlength Allows to specify the maximum number of characters a user can enter into the text box.

<input> tag type Attribute values

- Text

```
<input type="text" name = "sname" size = "50" required>
```

- Radio

```
<input type = "radio" name = "Gender" value = "Male">
```

- Checkbox

```
<input type="checkbox" name="color" value="red" />Red</li>
```

- Textarea

```
<textarea rows = "5" cols = "50" name = "description">
```

Enter description here...

```
</textarea>
```

- Submit

```
<button type="submit">Submit</button>
```

Text Box

- **Single-line text input controls** – This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.
- **Password input controls** – This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML **<input>** tag.
- **Multi-line text input controls** – This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

```
<label>Username : <input type="text" name = "sname" size = "50" required>
</label>
<label>Password : <input type="password" name = "pass" /></label>
```

<input> tag

```
<label>Username :  
    <input type="text" name = "sname" size = "50" required>  
</label>  
  
<label>Password :  
    <input type="password" name = "pass" maxlength = "10" />  
</label>
```

Username :

Password :

textarea

```
<label>Description : </label><br>
<textarea rows = "5" cols = "50" name = "description">
    Enter description here...
</textarea>
```

Output:

Description :

Enter description here...

Sr.No	Attribute & Description
1	name Used to give a name to the control which is sent to the server to be recognized and get the value.
2	rows Indicates the number of rows of text area box.
3	cols Indicates the number of columns of text area box

Radio button

```
<label> Gender: <input type = "radio" name = "Gender" value = "Male"> Male  
<input type = "radio" name = "Gender" value = "Female"> Female
```

Output:

Gender: Male Female

Sr.No	Attribute & Description
1	type Indicates the type of input control and for checkbox input control it will be set to radio.
2	name Used to give a name to the control which is sent to the server to be recognized and get the value.
3	value The value that will be used if the radio box is selected.
4	checked Set to <i>checked</i> if you want to select it by default.

checkbox

Choose Colors:

- <input type="checkbox" name="color" value="red">Red
- <input type="checkbox" name="color" value="blue">Blue
- <input type="checkbox" name="color" value="green" >Green

Output:

Choose Colors:

1. Red
2. Blue
3. Green

Sr.No	Attribute & Description
1	type Indicates the type of input control and for checkbox input control it will be set to checkbox ..
2	name Used to give a name to the control which is sent to the server to be recognized and get the value.
3	value The value that will be used if the checkbox is selected.
4	checked Set to <i>checked</i> if you want to select it by default.

Select Box (Drop Down) Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

```
<select name = "dropdown">  
    <option value = "Maths" selected>Maths</option>  
    <option value = "Physics">Physics</option>  
</select>
```

Output:

Maths 

Following is the list of important attributes of `<option>` tag –

Sr.No	Attribute & Description
1	value The value that will be used if an option in the select box box is selected.
2	selected Specifies that this option should be the initially selected value when the page loads.
3	label An alternative way of labeling options

Sr.No	Attribute & Description
1	name Used to give a name to the control which is sent to the server to be recognized and get the value.
2	size This can be used to present a scrolling list box.
3	multiple If set to "multiple" then allows a user to select multiple items from the menu.

```
<select name = "dropdown" size = "2">  
    <option value = "Maths" selected>Maths</option>  
    <option value = "Physics">Physics</option>  
</select>
```

Output:

Maths
Physics

```
<html>
<head><title> Student Registration </title></head>
<body>
<h1> Student Registration </h1>
<form>
    <label>Username : <input type="text" name = "sname" size = "50" maxlength = "10" placeholder = "Enter your name"></label> <br><br>
    <label>Password : <input type="password" name = "pass" maxlength = "10"></label> <br> <br>
    <label> Gender: <input type = "radio" name = "Gender" value = "Male"> Male
    <input type = "radio" name = "Gender" value = "Female"> Female <br> <br>
    Choose programming languages you know:<br>
    <ol>
        <li> <input type="checkbox" name="C" value="yes" checked>C</li>
        <li> <input type="checkbox" name="C++" value="yes" />C++</li>
        <li> <input type="checkbox" name="Java" value="yes" checked>Java</li>
        <li> <input type="checkbox" name="PHP" value="yes" /> PHP </li>
        <li> <input type="checkbox" name="Python" value="yes" />Python</li>
    </ol> <br> <br>
    Choose your Favourite Subject: <br>
    <select name = "dropdown" size = "2" multiple>
        <option value = "OS">OS</option> <option value = "DBMS" selected>DBMS</option>
        <option value = "CSA">CSA</option> <option value = "Data Structure">Data Structure</option>
    </select> <br><br>
    <label>Description : </label><br> <textarea rows = "5" cols = "50" name = "description"> Enter description here... </textarea> <br><br>
    Do you want to receive notifications: <br>
    <input type="radio" name="agree" value="yes" checked>Yes <br>
    <input type="radio" name="agree" value="no">No <br> <br><br>
    <button type="submit">Submit</button>
</form>
</body>
</html>
```

Student Registration

Username :

Password :

Gender: Male Female

Choose programming languages you know:

1. C
2. C++
3. Java
4. PHP
5. PHP

Choose your Favourite Subject:

OS
DBMS
CSA
Data Structure

Description :

Enter description here...

Do you want to receive notifications?

- Yes
 No

The Method Attribute

- The method attribute specifies the HTTP method to be used when submitting the form data.
- The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").
- The default HTTP method when submitting form data is GET.
- Example
 - <form action="/action_page.php" method="get">
 - <form action="/action_page.php" method="post">
- **GET:**
 - Appends the form data to the URL, in name/value pairs
 - NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
 - The length of a URL is limited (2048 characters)
 - Useful for form submissions where a user wants to bookmark the result
 - GET is good for non-secure data, like query strings in Google
- **POST:**
 - Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
 - POST has no size limitations, and can be used to send large amounts of data.
 - Form submissions with POST cannot be bookmarked

Data List Tag

- The `<datalist>` tag specifies a list of pre-defined options for an `<input>` element.
- The `<datalist>` tag is used to provide autocomplete feature in the HTML files. It can be used with an input tag so that users can easily fill the data in the forms using select the data.
- Syntax: `<datalist> ... </datalist>`
- Once the user starts typing the input element of `<datalist>` tag, the user will see the pre-defined options starting with the letter or word typed by the user.
- To use the `<datalist>` tag, the id of the tag must be the same as of the `<input>` element attribute.
- Both `<datalist>` and `<select>` tags are used for choosing an option from the given list. But the main difference between both is that in the `<datalist>` tag the user can enter its own input and add that as an option with the help of the `<input>` element whereas the `<select>` tag doesn't provide this feature.

```

<!DOCTYPE html>
<html>
<body>

<h1>The datalist element</h1>

<form action="/action_page.php" method="get">
  <label for="browser">Choose your browser from the list:</label>
  <input list="browsers" name="browser" id="browser">
  <datalist id="browsers">
    <option value="Edge">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
  <input type="submit">
</form>

</body>
</html>

```

The datalist element

Choose your browser from the list:

Submit

- Edge
- Firefox
- Chrome
- Opera
- Safari

The datalist element

Choose your browser from the list:

Submit

Submitted Form Data

Your input was received as:

browser=Internet Explorer

The server has processed your input and returned this answer.

Some More Tags:

- <fieldset> tag
 - The <fieldset> tag is used to group related elements in a form.
 - The <fieldset> tag draws a box around the related elements.
 - The <legend> tag is used to define a caption for the <fieldset> element.
- <map> tag
 - The <map> tag is used to define an image map. An image map is an image with clickable areas (hotspots).
 - Required attribute is “name” attribute which is associated with the tag’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.
- <bdi> Bi-Directional Isolation Tag
 - The <bdi> tag isolates a part of text that might be formatted in a different direction from other text outside it.
 - This element is useful when embedding user-generated content with an unknown text direction.
- <embed> Embedded Tag
 - The <embed> tag defines a container for an external resource, such as a web page, a picture, a media player, or a plug-in application.
 - `<embed type="image/jpg" src="pic_trulli.jpg" width="300" height="200">`
 - `<embed type="text/html" src="snippet.html" width="500" height="200">`
 - `<embed type="video/webm" src="video.mp4" width="400" height="300">`
 - Note: It is better to use , <iframe>, <video>, <audio> tag instead of <embed> tag

Some More Tags:

- <output> Tag
 - The <output> tag is used to represent the result of a calculation

Attribute	Value	Description
<u>for</u>	<i>element_id</i>	Specifies the relationship between the result of the calculation, and the elements used in the calculation
<u>form</u>	<i>form_id</i>	Specifies which form the output element belongs to
<u>name</u>	<i>name</i>	Specifies a name for the output element

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

<h1>The output element</h1>

<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">
<input type="range" id="a" value="50">
+<input type="number" id="b" value="25">
=<output name="x" for="a b"></output>
</form>

<p><strong>Note:</strong> The output element is not supported in Edge 12 (or earlier).</p>

</body>
</html>
```

The output element

A screenshot of a web browser displaying a form. It contains a range slider with a value of 50, a plus sign (+), and a number input field with a value of 25. Below the form, the output element displays the calculated sum, which is 74. A note at the bottom states: "Note: The output element is not supported in Edge 12 (or earlier)."

Note: The output element is not supported in Edge 12 (or earlier).

HTML <meta> Tag

- The <meta> tag defines metadata about an HTML document.
- Metadata is data (information) about data.
- <meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.
- Metadata will not be displayed on the page, but is machine parsable.
- Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

Attributes of <meta> tag

Attribute	Value	Description
<u>charset</u>	<i>character_set</i>	Specifies the character encoding for the HTML document
<u>content</u>	<i>text</i>	Specifies the value associated with the http-equiv or name attribute
<u>http-equiv</u>	content-security-policy content-type default-style refresh	Provides an HTTP header for the information/value of the content attribute
<u>name</u>	application-name author description generator keywords viewport	Specifies a name for the metadata

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="refresh" content="30">
</head>
```

```
<body>
```

<p>All meta information goes in the head section...</p>

```
</body>
```

```
</html>
```

1. Keywords - Define keywords for search engines

2. Refresh - Refresh document every 30 seconds:

3. Viewport - the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

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

The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

Character entities in HTML

- Reserved characters or the characters that creates ambiguity in code must be replaced with character entities.
- Ex: less than (<) or greater than (>) signs
- Entity name or Entity number both can be used to display a special character

Copyright Mark - Decimal - ©, Hexadecimal - &#A9

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

<h2>HTML Entity Example</h2>

<h3>The less-than sign: &lt; &#60;</h3>
<h3>The greater-than sign: &gt; &#62;</h3>
<h3>The ampersand (and) sign: &amp;lt;&gt;</h3>
<h3>The check sign: &check; &#9745; &#10004;</h3>
<h3>The registered trade mark sign: &reg; &#174;</h3>
<h3>The copyright mark sign: &copy; &#174;</h3>

</body>
</html>
```

Registered trademark
↑

HTML Entity Example

The less-than sign: <<

The greater-than sign: >>

The ampersand (and) sign: &

The check sign: ✓ ☑ ✓

The registered trade mark sign: ®

The copyright mark sign: © ®

HTML | <frame> Tag

- HTML Frames are used to divide the web browser window into multiple sections where each section can be loaded separately.
- A frameset tag is the collection of frames in the browser window.
- The `<frame>` tag was used in HTML 4 to define one particular window (frame) within a `<frameset>`
- Not Supported in HTML5.
- Use the `<iframe>` tag instead of `<frame>` tag in HTML5

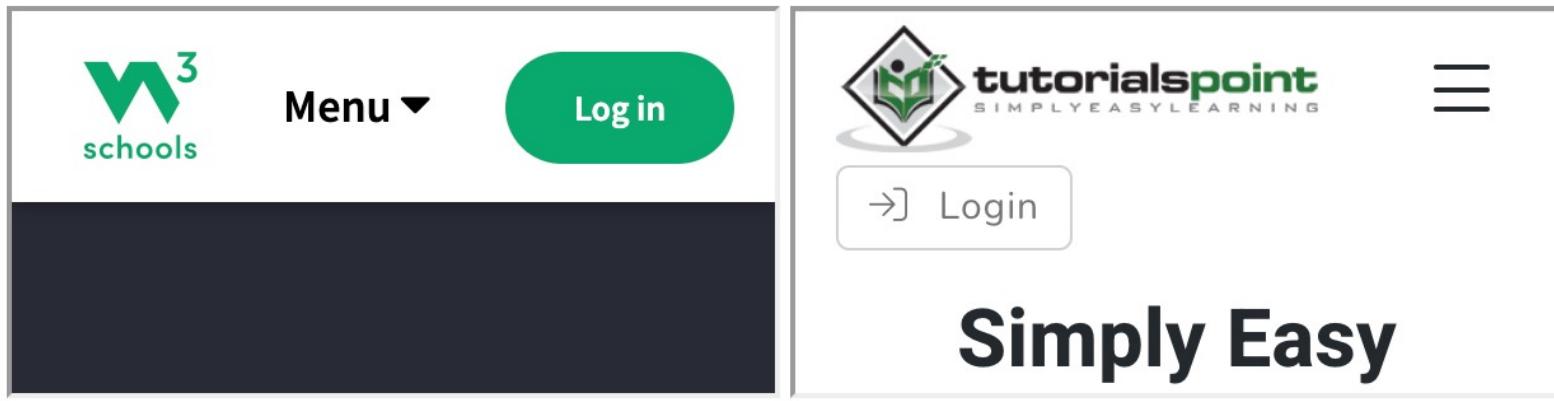
```
<!DOCTYPE html>
<html>
<body>

<h1>The iframe element</h1>

<iframe src="https://www.w3schools.com/" title="W3 School">
</iframe>
<iframe src="https://www.tutorialspoint.com/" title="Tutorialspoint">
</iframe>

</body>
</html>
```

The iframe element



The W3C Markup Validation Service

- Most pages on the World Wide Web are written in computer languages (such as HTML) that allow Web authors to structure text, add multimedia content, and specify what appearance, or style, the result should have.
- As for every language, these have their own grammar, vocabulary and syntax, and every document written with these computer languages are supposed to follow these rules. The (X)HTML languages, for all versions up to XHTML 1.1, are using machine-readable grammars called DTDs, a mechanism inherited from SGML.
- However, Just as texts in a natural language can include spelling or grammar errors, documents using Markup languages may (for various reasons) not be following these rules. The process of verifying whether a document actually follows the rules for the language(s) it uses is called validation, and the tool used for that is a validator. A document that passes this process with success is called valid.
- "Markup Validation" is the process of checking a Web document against the grammar (generally a DTD) it claims to be using.
- The W3C markup validation service is used to checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc.
- The W3C Markup Validator provides Perl/CGI/SGML/XML/DTD-based validation of a variety of document types.

The W3C Markup Validation Service

- The Markup Validator is a free tool and service that validates markup: in other words, it checks the syntax of Web documents, written in formats such as (X)HTML.
- The Validator compares your HTML document to the defined syntax of HTML and reports any discrepancies.
- The Markup Validator is maintained at W3C by W3C staff and benevolent collaborators, who receive a lot of help from contributors (read the full credits).
- Why Web professionals should choose to validate:
 - WYSINWOG - What You See Is Not What Others Get
 - Validation as a debugging tool
 - Validation as a future-proof quality check
 - Validation eases maintenance
 - Validation helps teach good practices
 - Validation is a sign of professionalism



Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

[Validate by URI](#)[Validate by File Upload](#)[Validate by Direct Input](#)

Validate by URI

Validate a document online:

Address:

▼ More Options

Character Encoding Only if missing**Document Type** Only if missing

List Messages Sequentially Group Error Messages by Type

Show Source

Clean up Markup with HTML-Tidy

Show Outline

Validate error pages

Verbose Output

[Check](#)

References

- <https://www.geeksforgeeks.org/>
- <https://www.w3schools.com/>
- <https://www.tutorialspoint.com/>
- <https://data-flair.training/blogs/web-browsers/>
- <https://dev.to/mahmoudessam/the-architecture-of-web-browsers-1o1k>
- <https://xd.adobe.com/ideas/process/information-architecture/different-types-of-website-structures/>
- <https://www.infidigit.com/blog/website-structure/>
- <https://www.uxpin.com/studio/blog/web-structures-explained/>
- <https://www.naukri.com/learning/articles/top-features-of-html5-with-examples/>
- <https://www.admecindia.co.in/web-design/10-new-features-html5-examples/>
- https://validator.w3.org/docs/why.html#why_pros

Unit -1

CSS

Prepared by:
Dr. Susheela

Cascading Style Sheets

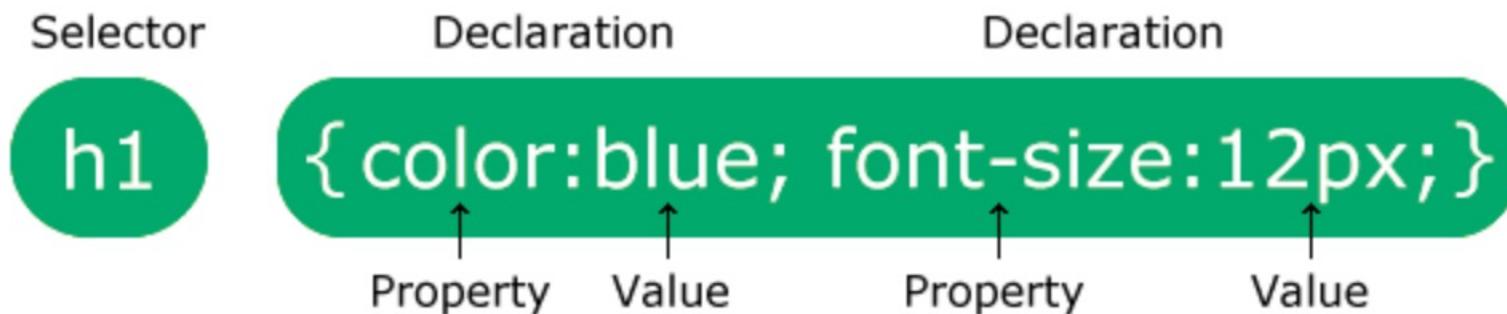
- CSS is the language we use to style an HTML document.
- CSS provides a set of style rules for defining the layout of HTML documents.
- CSS describes how HTML elements should be displayed.
- HTML largely determines textual content, CSS determines visual structure, layout, and aesthetics. HTML is a markup language, and CSS is a style sheet language.
- CSS makes it easier to enhance the look of the different elements on a web page.

Need for CSS

1. Faster Page Speed / Reduce redundancy and increase usability
 - CSS enables you to use less code. By using CSS, you can use one CSS rule and apply it to all occurrences of a certain tag within an HTML document.
2. Better User Experience
 - CSS not only makes web pages easy on the eye, it also allows for user-friendly formatting. When buttons and text are in logical places and well organized, user experience improves.
3. Quicker Development Time
 - With CSS, you can apply specific formatting rules and styles to multiple pages with one string of code. One cascading style sheet can be replicated across several website pages.
4. Easy Formatting Changes /Easier to Maintain the site
 - If you need to change the format of a specific set of pages, there's no need to change for every individual page. Just edit the corresponding CSS stylesheet and the changes will be applied to all the pages that are using that style sheet. site
5. Compatibility Across Devices
 - Whether mobile or tablet, desktop, or even smart TV, CSS combines with HTML to make responsive design possible
6. Supported by all browsers
7. Presentation style of document separated from content

Basic Syntax and Structure

- A CSS rule consists of a selector and a declaration block.



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

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

- p is a selector in CSS (it points to the HTML element you want to style: <p>).
- color is a property, and red is the property value
- text-align is a property, and center is the property value

Types of CSS

- **Inline CSS –**
 - by using the style attribute inside HTML elements
- **Internal CSS –**
 - by using a <style> element in the <head> section
- **External CSS –**
 - by using a <link> element to link to an external CSS file
 - An external style sheet is used to define the style for many HTML pages.
 - To use an external style sheet, add a link to it in the <head> section of each HTML page:

Inline CSS

```
<h1 style="color:blue;">A Blue Heading</h1>  
  
<p style="color:red;">A red paragraph.</p>
```

External CSS

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

styles.css

Internal CSS

```
<!DOCTYPE html>  
  
<html>  
<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>  
</html>
```

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

Types of Elements

- Block Level Elements

CSS Selectors

There are 6 types of selectors

1. Simple Selector
2. Class Selector
3. Id Selector
4. Nested Selector
5. Universal Selector
6. Group Selector
7. Pseudo Classes Selectors

Simple Selector – Select by Tag

- Selector form is a single element to which the property and value is applied
- For example - Select all `` elements

```
span  
{  
background: DodgerBlue;  
color: #ffffff;  
}
```

```
<span>Here's a span with some text.</span>  
<p>Here's a p with some text.</p>  
<span>Here's a span with more text.</span>
```

Here's a span with some text.

Here's a p with some text.

Here's a span with more text.

Class Selector

- Class Selectors are used when you have a bunch of elements that should receive same styling.
- Using Class Selector, we can assign different styles to the same element.
- Example - with class="sky" and Elements with class="code"

```
span.sky
{
    background: DodgerBlue;
}

.code
{
    font-family:Consolas;
}
```

```
<span class="sky">Here's a span with some text.</span>
<span>
    Another <span class="code"><span> &lt; span &gt; </span>
</span>
```

Here's a span with some text.
Another .

Id Selector

- To define a special style for special element we can use “id Selector”.
- Id Selectors are used when you have exactly one element that should receive a certain kind of styling
- Example - Select `` with `id="top"`

```
<span id="top">Here's a span with some text.</span>
```

```
<span>Here's another.</span>
```

```
span#top
```

```
{
```

```
background: DodgerBlue;
```

```
}
```

Here's a span with some text.
Here's another.

Universal Selector

- * selector is commonly known as the universal selector
- This selector selects and styles ALL elements
- Example

```
*  
{  
    font-family: Calibri;  
}
```

Nested Selector: Descendant

- <a> inside <div class="items">

```
div.items a
```

```
{  
    color: green;  
    font-weight: bold;  
}
```

```
<div class="items">
```

```
    <a href="#">Item1</a>
```

```
    <a href="#">Item2</a>
```

```
    <a href="#">Item3</a>
```

```
    <ul>
```

```
        <li><a href="#">Item4</a></li>
```

```
        <li><a href="#">Item5</a></li>
```

```
        <li><a href="#">Item6</a></li>
```

```
    </ul>
```

```
</div>
```

Item1 Item2 Item3

- Item4
- Item5
- Item6

Nested Selector: Adjacent Sibling

- <p> coming after <p>

p + p

```
{  
  font-style: italic;  
  font-weight: bold;  
}
```

```
<div>  
  <p>I'm a paragraph</p>  
  <p>I am selected!</p>  
</div>  
<p>I am selected!</p>  
<h2>Monkey hair</h2>  
<p>I am NOT selected</p>
```

I'm a paragraph

I am selected!

I'm a paragraph

Monkey hair

I am NOT selected

Grouping Selector

- You can apply a style to many selectors if you like.
- Just separate the selectors with a comma, as given in the following example:

```
h1, h2, h3  
{  
    color: #36C;  
    font-weight: normal;  
    font-size: 20px;  
}
```

Pseudo Classes Selectors

- A way of accessing HTML items that are not a part of the document tree.
- One of its main uses is to control the appearance of unvisited and visited links on a Webpage.
- Types of pseudo-class selectors
 - a:link – an unvisited link
 - a:visited – a visited link
 - a:hover – a link you have your mouse over
- CSS example:

```
a: hover  
{  
text-decoration: none;  
}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
a:link {
```

```
    color: green;
```

```
    background-color: transparent;
```

```
}
```

```
a:visited {
```

```
    color: pink;
```

```
    background-color: transparent;
```

```
}
```

```
a:hover {
```

```
    color: red;
```

```
    background-color: transparent;
```

```
}
```

```
a:active {
```

```
    color: yellow;
```

```
    background-color: transparent;
```

```
}
```

```
</style>
```

```
<h2>Link Colors</h2>
```

```
<p>You can change the default colors of links</p>
```

```
<a href="html_images.asp" target="_blank">HTML Images</a>
```

```
</body>
```

```
</html>
```

HTML Link Colors Using CSS

Link Colors

You can change the default colors of links

HTML Images

Link Colors

You can change the default colors of links

HTML Images

- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked

CSS Length Units

- There are two types of length units: absolute and relative.
- Absolute Length Units
 - The absolute length units are fixed, and a length expressed in any of these will appear as exactly that size.
 - Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.
- Relative Length Units
 - Relative length units specify a length relative to another length property.
 - Relative length units scale better between different rendering mediums.

Absolute Length Units

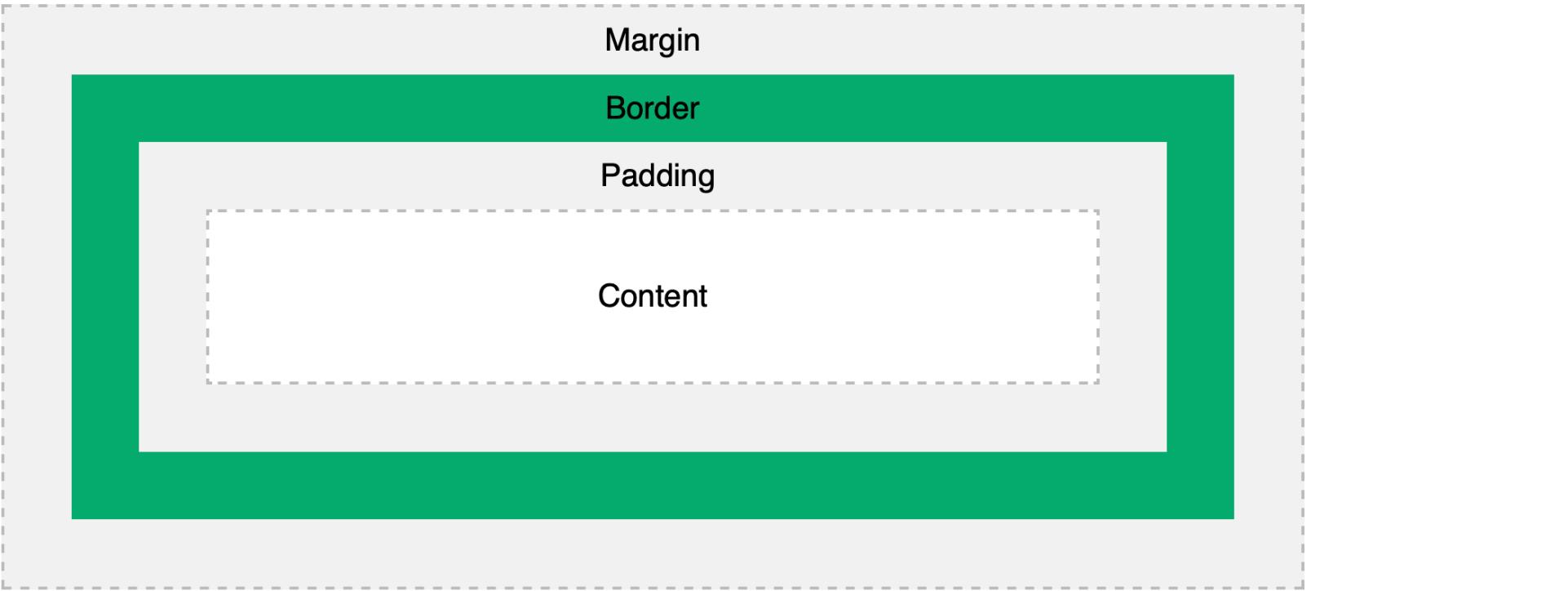
Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

Relative Length Units

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

CSS Box Model

- In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element.
- It consists of margins, borders, padding, and the actual content. The image below illustrates the box model:
- Explanation of the different parts:
- **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



```
div {  
    width: 300px;  
    border: 15px solid green;  
    padding: 50px;  
    margin: 20px;  
}
```

CSS Box Model

- box-sizing property is used to define the size of the Box using width property.
- By default, width property is used to define the width of the content box.
- It consists of margins, borders, padding, and the actual content. The image below illustrates the box model:

The diagram shows a dark gray rectangular container divided into three horizontal sections. The top section is labeled 'border-box' with a light blue background. The middle section is labeled 'content-box' with a white background. The bottom section is labeled 'padding-box' with a light gray background. The 'border-box' section is slightly taller than the others.

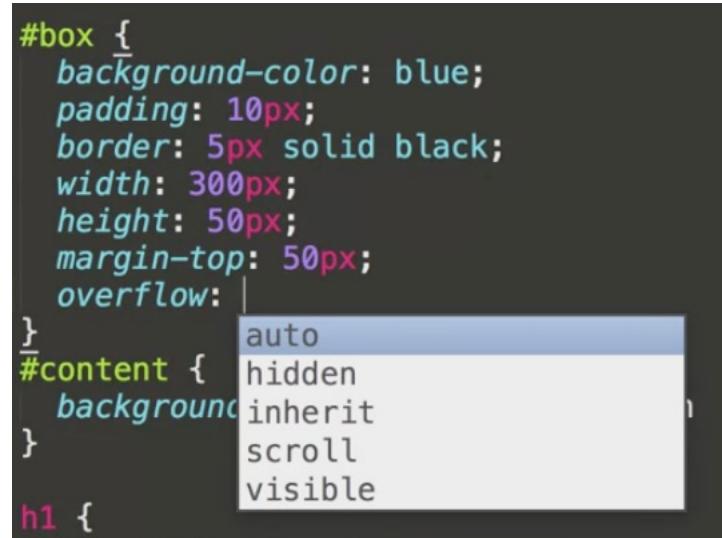
```
#box {
  background-color: blue;
  padding: 10px 30px 10px 30px;
  border: 20px solid black;
  margin: 40px;
  width: 300px;
  box-sizing: border-box;
}
#content {
  background-color: white;
}
```

```
div {
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
```

CSS Box Model

- overflow property is used in-case when the content of the box takes more space than the specified size of the box (height & width of the box).
- Possible values: auto, hidden, inherit, scroll, visible
- Default value is visible.

```
#box {  
    background-color: blue;  
    padding: 10px;  
    border: 5px solid black;  
    width: 300px;  
    height: 50px;  
    margin-top: 50px;  
    overflow: |  
}  
#content {  
    background: |  
}  
h1 {  
    . . .  
}
```



CSS - background property

- The background property is a shorthand property for:
 - background-color
 - background-image
 - background-position
 - background-size
 - background-repeat
 - background-origin
 - background-clip
 - background-attachment

background-color

- With CSS, a color is most often specified by:
 - a valid color name - like "red"
 - a HEX value - like "#ff0000"
 - an RGB value - like "rgb(255,0,0)"
- Opacity / Transparency
 - The opacity property specifies the opacity/transparency of an element. It can take a value from 0.0 - 1.0.
 - The lower value, the more transparency

```
<!DOCTYPE html>
<html>
<head> <style>
div {
    background-color: green;
}
div.first {
    opacity: 0.3;
}
div.second {
    opacity: 0.6;
}
</style> </head>
<body>

<h1>Transparent Boxes</h1>

<div class="first"> <h1>opacity 0.3</h1> </div>

<div class="second"> <h1>opacity 0.6</h1> </div>

<div> <h1>opacity 1 (default)</h1> </div>

</body>
</html>
```

Transparency using RGBA

```
<!DOCTYPE html>
<html>
<head> <style>
div {
    background-color: green;
}
div.first {
    opacity: 0.3;
}
div.second {
    opacity: 0.6;
}
</style> </head>
<body>

<h1>Transparent Boxes</h1>

<div class="first"> <h1>opacity 0.3</h1> </div>
<div class="second"> <h1>opacity 0.6</h1> </div>
<div> <h1>opacity 1 (default)</h1> </div>

</body>
</html>
```

```
<style>
div {
    background: rgb(0, 128, 0);
}

div.first {
    background: rgba(0, 128, 0, 0.3);
}

div.second {
    background: rgba(0, 128, 0, 0.6);
}
```

Transparent Boxes

opacity 0.3

opacity 0.6

opacity 1 (default)

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("bgdesert.jpg");
}

p {
    background-image: url("paper.gif");
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the background!</p>

</body>
</html>
```

Hello World!

This page has an image as the background!

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right top;
    margin-right: 200px;
    background-attachment: fixed;
}
</style>
</head>
<body>
```

background-image

- The background-image property sets one or more background images for an element.
- By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.
- Possible property values:
 - **url** - The URL to the image. To specify more than one image, separate the URLs with a comma
 - **none** - No background image will be displayed. This is default
 - **initial** - Sets this property to its default value.
 - **inherit** - Inherits this property from its parent element.

background-position

- The background-position property sets the starting position of a background image.
- By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.
- The top left corner is 0 0 or 0% 0%. The right bottom corner is 100 100 or 100% 100%.
- Possible values:
 - **left top, left center, left bottom, right top, right center, right bottom, center top, center center, center bottom** - If you only specify one keyword, the other value will be "center"
 - **x% y%** - x is the horizontal position and y is the vertical position.
 - **xpos ypos** – xpos is the horizontal position and ypos is the vertical position.
 - **initial** - Sets this property to its default value.
 - **inherit** - Inherits this property from its parent element.

background-position: 50% 50%;

background-position: bottom right;

background-position: 50px 150px;

background-size

- The background-size property specifies the size of the background images.
- Possible property values:
 - **auto** – Default value (original size)
 - **length** - Sets the width and height of the background image. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto".
 - **percentage** – Sets the width and height of the background image in percent of the parent element. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto"
 - **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
 - **initial** – Sets this property to its default value
 - **inherit** – Inherits this property from its parent element.
- **background-size: auto;**
- **background-size: 75% 50%;**
- **background-size: cover;**

background-repeat

- The background-repeat property sets if/how a background image will be repeated.
- By default, a background-image is repeated both vertically and horizontally.

Value	Description
repeat	The background image is repeated both vertically and horizontally. The last image will be clipped if it does not fit. This is default
repeat-x	The background image is repeated only horizontally
repeat-y	The background image is repeated only vertically
no-repeat	The background-image is not repeated. The image will only be shown once
space	The background-image is repeated as much as possible without clipping. The first and last image is pinned to either side of the element, and whitespace is distributed evenly between the images
round	The background-image is repeated and squished or stretched to fill the space (no gaps)
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

background-attachment

- The background-attachment property sets whether a background image scrolls with the rest of the page, or is fixed.

Value	Description
scroll	The background image will scroll with the page. This is default
fixed	The background image will not scroll with the page
local	The background image will scroll with the element's contents
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

background-origin

- The background-origin property specifies the origin position (the background positioning area) of a background image.

Value	Description
padding-box	Default value. The background image starts from the upper left corner of the padding edge
border-box	The background image starts from the upper left corner of the border
content-box	The background image starts from the upper left corner of the content
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

Note: This property has no effect if background-attachment is "fixed".

background-clip

- The background-clip property defines how far the background (color or image) should extend within an element.

Value	Description
border-box	Default value. The background extends behind the border
padding-box	The background extends to the inside edge of the border
content-box	The background extends to the edge of the content box
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

Manipulating Texts

- Text Color

- 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)"

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  color: blue;
}

h1 {
  color: green;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<p>This is an ordinary paragraph. Notice that this text is blue. The default text color for a page is defined in the body selector.</p>
<p>Another paragraph.</p>

</body>
</html>
```

This is heading 1

This is an ordinary paragraph. Notice that this text is blue. The default text color for a page is defined in the body selector.

Another paragraph.

Text Alignment and Text Direction

- **text-align:**
 - used to set the horizontal alignment of a text
 - center, left, right, justify, initial, inherit
- **text-align-last:**
 - specifies how to align the last line of a text
 - auto, left, right, center, justify, start, end, initial, inherit
- **vertical-align:**
 - sets the vertical alignment of an element.
 - Baseline, text-top, text-bottom, sub, super

CSS Text

- color
- background-color
- text-align
- text-align-last
- direction
- unicode-bidi
- vertical-align

CSS Fonts

1. Font Family
2. Font Style
3. Font Weight
4. Font Variant
5. Font Size
6. Font Color
7. Font Shorthand

CSS Font Families

In CSS there are five generic font families:

- 1. *Serif*** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
- 2. *Sans-serif*** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
- 3. *Monospace*** fonts - here all the letters have the same fixed width. They create a mechanical look.
- 4. *Cursive*** fonts imitate human handwriting.
- 5. *Fantasy*** fonts are decorative/playful fonts.

Generic Font Family

Examples of Font Names

Serif

Times New Roman
Georgia
Garamond

Sans-serif

Arial
Verdana
Helvetica

Monospace

Courier New
Lucida Console
Monaco

Cursive

Brush Script MT
Lucida Handwriting

Fantasy

COPPERPLATE
Papyrus

```
<!DOCTYPE html>
<html>
<head>
<style>
.p1 {
  font-family: "Times New Roman", Times, serif;
}

.p2 {
  font-family: Arial, Helvetica, sans-serif;
}

.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
</style>
</head>
<body>

<h1>CSS font-family</h1>
<p class="p1">This is a paragraph, shown in the Times New Roman font.</p>
<p class="p2">This is a paragraph, shown in the Arial font.</p>
<p class="p3">This is a paragraph, shown in the Lucida Console font.</p>

</body>
</html>
```

CSS font-family

This is a paragraph, shown in the Times New Roman font.

This is a paragraph, shown in the Arial font.

This is a paragraph, shown in the Lucida Console font.

CSS Font Style

- 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)

```
p.normal {  
    font-style: normal;  
}
```

```
p.italic {  
    font-style: italic;  
}
```

```
p.oblique {  
    font-style: oblique;  
}
```

CSS Font Weight

- The font-weight property specifies the weight of a font: normal or bold

```
p.normal {  
    font-weight: normal;  
}
```

```
p.thick {  
    font-weight: bold;  
}
```

CSS Font Variant

- The font-variant property specifies whether or not a text should be displayed in a small-caps font.
- In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

```
p.normal {  
    font-variant: normal;  
}
```

```
p.small {  
    font-variant: small-caps;  
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-variant: normal;
}

p.small {
  font-variant: small-caps;
}
</style>
</head>
<body>
```

<h1>The font-variant property</h1>

```
<p class="normal">My name is Susheela.</p>
<p class="small">My name is Susheela.</p>
```

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

The font-variant property

My name is Susheela.

MY NAME IS SUSHEELA.

CSS Font Variant

- The font-size property sets the size of the text.
- Always use the proper HTML tags, like `<h1>` - `<h6>` for headings and `<p>` for paragraphs.
- The font-size value can be an absolute, or relative size.

```
h1 {  
    font-size: 40px;  
}  
  
h2 {  
    font-size: 1.875em; /* 30px/16=1.875em */  
}  
  
body {  
    font-size: 100%;  
}  
  
p {  
    font-size: 10vw;  
}
```

CSS Font with Shorthand Declaration

- The font property is a shorthand property for:
 - font-style
 - font-variant
 - font-weight
 - font-size/line-height
 - font-family
- The font-size and font-family values are required. If one of the other values is missing, their default value are used.

Note: The line-height property sets the space between lines.

- if you specify background-color specifically and then you go ahead and override it with the background, but you don't specify the color or you don't specify whatever the specific property is, whether it's background-image or background-repeat or background-position and so on.
- If you specify font without any dash subproperty, whatever properties that are inherited with a dash will be overridden unless you actually call them out directly, straight in the font property.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.a {
  font: 15px Arial, sans-serif;
}

p.b {
  font: italic small-caps bold 12px/30px Georgia, serif;
}
</style>
</head>
<body>

<h1>The font shorthand Property</h1>

<p class="a">This is a paragraph. The font size is set to 15 pixels, and the font family is Arial.</p>

<p class="b">This is a paragraph. The font is set to italic and bold, with small-caps (all lowercase letters are converted to uppercase). The font size is set to 12 pixels, the line height is set to 30 pixels, and the font family is Georgia.</p>

</body>
</html>
```

The font shorthand Property

This is a paragraph. The font size is set to 15 pixels, and the font family is Arial.

THIS IS A PARAGRAPH. THE FONT IS SET TO ITALIC AND BOLD, WITH SMALL-CAPS (ALL LOWERCASE LETTERS ARE CONVERTED TO UPPERCASE). THE FONT SIZE IS SET TO 12 PIXELS, THE LINE HEIGHT IS SET TO 30 PIXELS, AND THE FONT FAMILY IS GEORGIA.

CSS Border

- The “border-style” property specifies what kind of border to display.
- **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

```
<!DOCTYPE html>
<html>
<head>
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid}
</style>
</head>
```

```
<body>
```

The border-style Property

This property specifies what kind of border to display:

```
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="none">No border.</p>
<p class="hidden">A hidden border.</p>
<p class="mix">A mixed border.</p>
```

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

The border-style Property

This property specifies what kind of border to display:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

No border.

A hidden border.

A mixed border.

Padding

The CSS padding properties are used to generate space around an element's content, inside of any defined borders. If the padding property has four values:

- **padding: 25px 50px 75px 100px;**

- top padding is 25px
- right padding is 50px
- bottom padding is 75px
- left padding is 100px

Property	Description
<u>padding</u>	A shorthand property for setting all the padding properties in one declaration
<u>padding-bottom</u>	Sets the bottom padding of an element
<u>padding-left</u>	Sets the left padding of an element
<u>padding-right</u>	Sets the right padding of an element
<u>padding-top</u>	Sets the top padding of an element

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
    width: 300px;
    background-color: yellow;
}
div.ex2 {
    width: 300px;
    padding: 25px;
    background-color: lightblue;
}
</style>
</head>
<body>
<h2>Padding and element width</h2>
<div class="ex1">This div is 300px wide.</div>
<br>
<div class="ex2">The width of this div is 350px, even though it is
defined as 300px in the CSS.</div>
</body>
</html>
```

Padding and element width

This div is 300px wide.

The width of this div is 350px, even though it is defined as 300px in the CSS.

```
div {
    padding: 25px 50px 75px 100px;
}
```

CSS Lists

- 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
- CSS Lists Properties
 - **list-style-type** - specifies the type of list item marker ([circle](#), [square](#), [upper-roman](#), [lower-alpha](#)).
 - **list-style-image** - specifies an image as the list item marker.
 - **list-style-position** - specifies the position of the list-item markers (bullet points).
 - "**list-style-position: outside;**" means that the bullet points will be outside the list item.
 - "**list-style-position: inside;**" means that the bullet points will be inside the list item.
 - **list-style-type:none** - can also be used to remove the markers/bullets. Default margin and padding will remains in the list. To remove this, add margin:0 and padding:0 to or

CSS List - Shorthand property

- The list-style property is a shorthand property. It is used to set all the list properties in one declaration:

```
ul {  
    list-style: square inside url("gehu.jpg");  
}
```

- When using the shorthand property, the order of the property values are:

- list-style-type (if a list-style-image is specified, the value of this property will be displayed if the image for some reason cannot be displayed)
- list-style-position (specifies whether the list-item markers should appear inside or outside the content flow)
- list-style-image (specifies an image as the list item marker)
- If one of the property values above is missing, the default value for the missing property will be inserted, if any.

```
<!DOCTYPE html>
<html>
<head>
<style>
ol {
    background: #ff9999;
    padding: 20px;
}

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

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

ul li {
    background: #cce5ff;
    color: darkblue;
    margin: 5px;
}
</style>
</head>
```

```
<body>

<h1>Styling Lists With Colors</h1>

<ol>
    <li>Coffee</li>
    <li>Tea</li>
</ol>

<ul>
    <li>Coffee</li>
    <li>Tea</li>
</ul>

</body>
</html>
```

Styling Lists With Colors

1. Coffee
2. Tea

- Coffee
- Tea

Position Property

- The position property specifies the type of positioning method used for an element.
- There are five different position values:
 - static
 - relative
 - fixed
 - absolute
 - sticky

Property	Description
<u>bottom</u>	Sets the bottom margin edge for a positioned box
<u>clip</u>	Clips an absolutely positioned element
<u>left</u>	Sets the left margin edge for a positioned box
<u>position</u>	Specifies the type of positioning for an element
<u>right</u>	Sets the right margin edge for a positioned box
<u>top</u>	Sets the top margin edge for a positioned box

position: static

- HTML elements are positioned static by default
- Static positioned elements are not affected by the top, bottom, left, and right properties.

position: relative

- HTML element 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

position: fixed

- HTML element 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.
- A fixed element does not leave a gap in the page where it would normally have been located.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.static {
    position: static;
    border: 3px solid #73AD21;
}
div.relative {
    position: relative;
    left: 30px;
    border: 3px solid #73AD21;
}
div.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    border: 3px solid #73AD21;
}
</style>
</head>
```

```
<h2>position: static;</h2>
<div class="static">
This div element has position: static;
</div>
<h2>position: relative;</h2>
<div class="relative">
This div element has position: relative;
</div>
<div class="fixed">
This div element has position: fixed;
</div>
</body>
</html>
```

position: static;

This div element has position: static;

position: relative;

This div element has position: relative;

shot

This div element has position: fixed;

position: absolute

- An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
- However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
    position: relative;
    width: 400px;
    height: 200px;
    border: 3px solid #73AD21;
}

div.absolute {
    position: absolute;
    top: 80px;
    right: 0;
    width: 200px;
    height: 100px;
    border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: absolute;</h2>

<div class="relative">This div element has position: relative;
    <div class="absolute">This div element has position: absolute;</div>
</div>

</body>
</html>
```

position: absolute;

This div element has position: relative;

This div element has position: absolute;

position: sticky

- An element with `position: sticky;` is positioned based on the user's scroll position.
- A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like `position:fixed`).

```
<head>
<style>
div.sticky {
  position: -webkit-sticky;
  position: sticky;
  top: 0;
  padding: 5px;
  background-color: #cae8ca;
  border: 2px solid #4CAF50;
}
</style>
</head>
```

```
<body>
<p>Try to <b>scroll</b> inside this frame to understand how sticky positioning works.</p>
```

```
<div class="sticky">I am sticky!</div>
```

```
<div style="padding-bottom:2000px">
  <p>In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.</p>
  <p>Scroll back up to remove the stickyness.</p>
</div>
```

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

Try to **scroll** inside this frame to understand how sticky positioning works.

I am sticky!

In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.

Scroll back up to remove the stickyness.

Output After Scrolling

I am sticky!

Scroll back up to remove the stickyness.

Bootstrap

- Bootstrap is an open-source and free CSS framework that helps in directing a responsive device-friendly mobile-first front-end webpage development tool.
- Bootstrap mainly includes CSS (Cascading Style Sheets) and an optional JavaScript-supported design template (plug-ins) that deals with typography, buttons, forms, and other user interface components. This Bootstrap framework helps rapid web development and supports developers in creating responsive web pages.
- Twitter Blueprint was the first name for Bootstrap and was developed on Twitter by Mr. Mark Otto and Jacob Thornton. It was released as an open-source product on GitHub in August 2011. The framework is primarily built to encourage design uniformity and reliability of web pages across applications. Before its existence, developers used various external libraries to perform responsive web development, leading to incompatibilities in web development and heavy maintenance burdens.

Benefits of Bootstrap

- Browser supportive: Every browser supports this Bootstrap Framework.
- Mobile-first approach: The Bootstrap framework has a preexisting mobile-first style all through the library and not as separate files.
- Simple and easy to start: If you know HTML and CSS, you can quickly start working with Bootstrap, and its documentation is available on the official site.
- Responsive design and looks: Web pages designed using the Bootstrap framework have responsive CSS that can adjust to the screen size of large desktops, notebooks, tablets, and mobiles.
- Easy customization: It provides some built-in components and functionalities that are easy to customize.
- Clean interface or Developers: The bootstrap framework provides a new and consistent result for building user interfaces on web pages.
- It is an open-source framework with web-based customization.
- **Benefits of Bootstrap Framework**
 - It produces fewer cross-browser bugs.
 - It is a consistent framework supported by all web browsers and CSS-based compatibility improvements.
 - It is a lightweight and hence widely used framework for creating responsive sites.
 - It's easily customizable.
 - It has a simple and effective grid system.
 -

Media Query

- Media Query is used to style webpage.

```
<!doctype HTML>
<html>
  <head>
    <style>
      div
      {
        font-size: 30px;
        border: 5px double blue;
        margin: 10px;
      }
      div:nth-child(2)
      {
        font:40px rgb(0, 33, 17);
        background: blue;
        background-color: powderblue;
      }
      div > h3:nth-child(2)
      {
        background: yellow;
        font-size:20px;
      }
    </style>
  </head>
  <body>
    <div>
      <h1>Hello World</h1>
      <h3>This is a test</h3>
    </div>
  </body>
</html>
```

endly front-end

Program (Media Query)

```
<!doctype HTML>
<html>
<head>
  <style>
    div
    {
      font-size: 30px;
      border: 5px double blue;
      margin: 10px;
    }
    div:nth-child(2)
    {
      font:40px black rgb(0, 33, 17);
      background: blue;
      background-color: powderblue;
    }
    div > h3:nth-child(2)
    {
      background: yellow;
      font-size:20px;
    }
  </style>
</head>
<body>
```

```
  <div>
    This is 1st div element
    <h3> This is First heading </h3>
    <h3> This is Second heading </h3>
    <h3> This is Third heading </h3>
    <h3> This is Forth heading </h3>
  </div>

  <div>
    This is 2nd div element
  </div>

</body>
</html>
```

The code illustrates the use of CSS media queries and various CSS properties like font-size, border, and background-color. It defines two main styles: one for all div elements and another specifically for the second child of a div. A media query is used to target screens between 500px and 1000px wide, changing the font size of the first div's content to 80% and applying a yellow background and 90% font size to its second child h3 element.

This is 1st div element

This is First heading

This is Second heading

This is Third heading

This is Forth heading

This is 2nd div element

OUTPUT (When Page width is
in between 500 – 1000px)

OUTPUT (When Page width
either less than 500 or
greater than 1000px)

This is 1st div element

This is First heading

This is Second heading

This is Third heading

This is Forth heading

This is 2nd div element