

## Introduction to CSS

HTML is just van skeletal layout of a website. We need CSS to design a website, add styles to it and make it look beautiful.

### What is CSS

CSS stands for Cascading style Sheets

CSS is optional but it converts an off looking HTML page into a beautiful & responsive website

### Installing VS Code

We will use Microsoft Visual Studio Code as a tool to edit our code. It is very powerful, free and customizable

### Why Learn CSS?

CSS is a very demanded skill in the world of web development. If you are successfully able to master CSS, you can customize your websites as per your liking.

### Your first line of CSS

Create a .css file inside your directory and add it to your HTML. Add the following line to your CSS

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

This will make your page background as red.

## HTML Refresher

HTML is a bunch of tags used to lay the structure of a page.

Download HTML notes as part of these notes for a detailed deepdive. If you know basic HTML, continue!

## Chapter 1 - Creating our first CSS Website

We will create our first CSS website in this section.

What is DOM?

DOM stands for document object model. When a page is loaded, the browser creates a DOM of the page which is constructed as a tree of objects.

HTML id and class attributes

When an HTML element is given an id, it serves as a unique identifier for that element.

On the other hand, when an HTML element is given a class, it now belongs to that class. More than one elements can belong to a single class but every element must have a unique id (if assigned).

We can add multiple classes to an element like this

`<div id="first" class="c1 c2 c3">  
 ...  
</div>`

*↳ multiple classes followed by spaces*

Three ways to add CSS to HTML

There are 3 ways to add CSS to HTML:

1. <style> tag → Adding <style> ... </style> to HTML
2. Inline CSS → Adding CSS using style attribute
3. External CSS → Adding a stylesheet (.css) to HTML using <link> tag.

## CSS Selectors

A CSS selector is used to select an HTML element(s) for styling

body {  
 color: red; → Declaration (property: value)  
 background: pink; → Declaration  
}

### Element selector

It is used to select an element based off the tagname  
For example:

h2 {  
 color: blue;  
}

### id selector

It is used to select an element with a given id  
For example:

#first { → # is used to target by id  
 color: white;  
 background: black;  
}

### Class selector

It is used to select an element with a given class  
For example:

.red {  
 background: red;  
}

## Important Notes :

→ We can group selectors like this :

`h1, h2, h3, div {`

`color: blue;` → `h1, h2, h3` and `div` will be red  
}

→ We can use element class as a selector like this :

`p.red {`

`color: red;` → all paragraphs of `p.red` will get color of red  
}

→ \* can be used as a universal selector to select all the elements

`* {`

`margin: 0;`

`padding: 0;`  
}

→ An inline style will override external and internal styles

## Comments in CSS

Comments in CSS is text which is not parse and is thus ignored

## Chapter 1 - Practice Set

- 1 Create a website with a class red div which has a background color of red and color white.
- 2 Create an element with id head and verify that background color works on it as inline, external as well as using style tag CSS
- 3 Create a CSS class one and verify that it works on multiple elements.
- 4 Create multiple CSS classes and verify that all of these work on the same element
- 5 Have a look at the MDN CSS reference and try to play around with few key-value CSS rules.

## Chapter 2 - Colors & Backgrounds

CSS rules are simple key-value pairs with a selector  
We can write CSS rules to change color and set backgrounds

The color property

The CSS color property can be used to set the text color inside an element.

↳ {

color : red; → Text color will be changed to red.  
}

Similarly we can set color for different elements

Types of color values

Following are the most commonly used color values in CSS

- 1> RGB → Specify color using Red, green, blue values eg. `rgb(200, 98, 70)`
- 2> HEX Code → Specify color using hex code.  
eg. `# ff7180`
- 3> HSL → Specify the color using hsl values  
eg. `hsl(8, 90%, 63%)` ↳ hue, saturation, lightness

The value of the color or background color is provided as any one of these values

Note : We also have an RGBA and HSLA values for color but they are rarely used by beginners.  
A stands for alpha-

The background-color property

The CSS background-color property specifies the background color of a Container

For eg:

body {  
background-color: brown;  
}

Can be other types of colors as well

The background-image property

Used to set an image as the background.

body {  
background-image: url("harry.jpg");  
}

The image is by default repeated in x & y directions

The background-repeat property

Can be any of :

- repeat-x → repeat in horizontal direction
- repeat-y → repeat in vertical direction
- no-repeat → image not repeat

See more possible values at MDN docs

The background-size property

Can be following :

- cover → fits & no empty space remains
- contain → fits & image is fully visible
- auto → Display in original size
- 50px 30px → Set width & height will be set automatically

→ {{width}} {{height}} → Set width & height

Note: Always check the MDN docs to dissect a given CSS property. Remember, practice will make you perfect

The background-position property sets the starting position of a background image.

```
div1 {  
background-position: left top;  
}
```

The background-attachment property

Defines a scrollable / non-scrollable character of a background image.

```
div2 {  
background-attachment: fixed;  
}
```

The background shorthand

A single property to set multiple background properties.

```
div3 {  
background: red url('img.png') no-repeat fixed right top;  
}  
color      ↓      repeat  
        image
```

One of the properties can be missing given the others are in order.

→ {{width}} {{height}} → Set width & height

Note: Always check the MDN docs to dissect a given CSS property. Remember, practice will make you perfect

The background-position property sets the starting position of a background image.

div 1 {  
background-position: left top;  
}

The background-attachment property defines a scrollable/non-scrollable character of a background image.

div 2 {  
background-attachment: fixed  
}

The background shorthand  
A single property to set multiple background properties.

div 3 {  
background: red url('img.png') no-repeat fixed right top;  
}

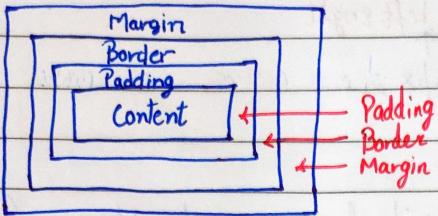
One of the properties can be missing given the others are in order.

## Chapter 2 - Practice Set

- 1 Create a dark blue navigation bar with light color items
- 2 change the color of the main container on your page to dark red.
- 3 Create a div and add a background image with a given width and height
- 4 Create a vertical box and add a fixed non scrolling background to it
- 5 Verify that the background shorthand property works with some of the values Skipped.

## Chapter 3 - CSS Box Model

The CSS box model looks at all the HTML elements as boxes



Setting width & Height

We can set width and height in CSS as follows

```

#box {
    height: 70px;
    width: 70px;
}
    
```

Note that the total width/height is calculated as follows:

Total height = height + top/bottom padding + top/bottom border  
+ top/bottom margin

Setting Margin & Padding

We can set margin and padding as follows:

```

.box {
    margin: 3px;
    padding: 4px;
}
    
```

Sets top, bottom, left & right values

· boxMain {  
 margin : 7px 0px 2px 11px;  
 }

top  
 right  
 bottom  
 left  
 ↗  
 ↘  
 ↙  
 ↚  
 clockwise

· boxLast {  
 margin : 7px 3px;  
 }

We can also set individual margins / paddings like this :

margin - top : 7px  
 margin - bottom : 3px  
 margin - left : 8px  
 margin - right : 9px

} Same goes with padding

## Setting Borders

We can set the border as follows

· bx {  
 border - width : 2px;  
 border - style : solid;  
 border - color : red;  
 }

} or just set border : 2px solid red;  
 (Shorthand)

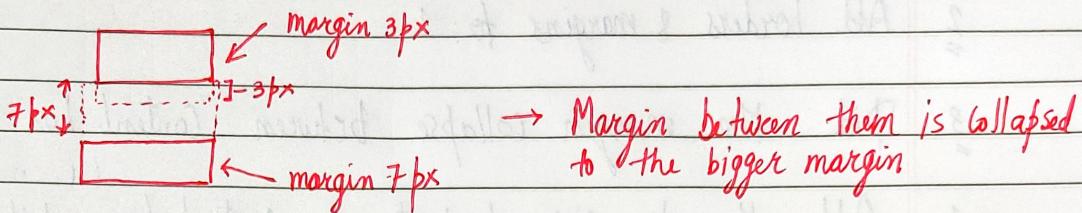
## Border Radius

We can set border radius to create rounded borders

· div2 {  
 border - radius : 7px;  
 }

## Margin Collapse

When two margins from different elements overlap, the equivalent margin is the greater of the two. This is called margin collapse.



## Box Sizing

Determines what out of padding and border is included in elements width and height.

Can be content-box or border-box

• `div {`

box-sizing: border-box;

}

↳ Include only content in width/height

↳ The content width and height includes content + padding + border

## Chapter 3 - Practice Set

- 1 Create a website layout. Add a header box, 1 content box and one footer.
- 2 Add borders & margins to 1
- 3 Did the margin collapse between Content box & footer?
- 4 Add the box-sizing property to Content box. What changes did you notice?

## Chapter 4 - fonts & display

### The display property

The CSS `display` property is used to determine whether an element is treated as a block/inline element & the layout used for its children.

↳ flexbox/grid/etc.

#### `display: inline`

Takes only the space required by the element. No linebreaks before and after. Setting width/height not allowed.  
(or margin/padding)

#### `display: block`

Takes full space available in width and leaves a newline before and after the element

#### `display: inline-block`

Similar to inline but setting height, width, margin and padding is allowed. Elements can sit next to each other

#### `display: none` vs `visibility: hidden`

With `display: none`, the element is removed from the document flow. Its space is not blocked.

With `visibility: hidden`, the element is hidden but its space is reserved.

### text-align property

Used to set the horizontal alignment of a text

• `div {`

`text-align: center;`

`}`

text-decoration property

Used to decorate the text

Can be overline, line-through, underline, none

text-transform property

Used to specify uppercase and lowercase letters in a text.

p. uppercase {

    text-transform: uppercase;  
    }

line-height property

Used to specify the space between lines.

· Small {

    line-height: 0.7;  
    }

Font

Font plays a very important role in the look and feel of a website

font-family

Font family specifies the font of a text.

Can hold multiple values as a "fallback" system

p {

    font-family: "Times new Roman", monospace;



Always do this to ensure the correct font  
of your choice is rendered

## Web Safe Fonts

These fonts are universally installed across browsers.

### How to add Google fonts

In order to use custom google fonts, go to google fonts then select a style and finally paste it to the style.css of your page.

### Other font properties

Some of the other font properties are listed below:

font-size → Sets the size of the font

font-style → Sets the font style

font-variant → Sets whether text is displayed in small-caps

font-weight → Sets the weight of the font

### Generic families

Broad class of similar fonts eg. serif, sans-serif

Just like when we say fruit, it can be any fruit.

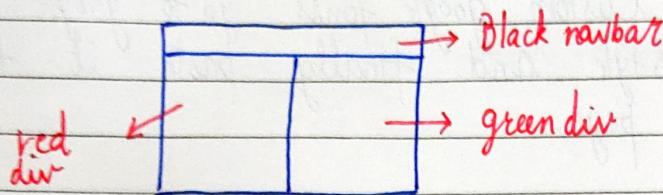
When we say serif it can be any serif font.

font-family → Specific

Generic family → Generic

## Chapter 4 - Practice Set

- 1 Create the following website layout



- 2 Add a footer with Google font "Ballu Bhai" to ①
- 3 Remove the underlines from links in ①
- 4 Demonstrate the difference between display: none and visibility: hidden using a div
- 5 Change the footer to all uppercase in ①

## Chapter 5 - Size, position & Lists

There are more units for describing size other than 'px'  
There are rem, cm, vw, vh, percentages etc.

What's wrong with pixels?

Pixels (px) are relative to the viewing device.

For a device with size 1920x1080, 1px is 1 unit out of 1080/1920.

Relative lengths

These units are relative to the other length property.  
Following are some of the most commonly used relative lengths

1. em → Unit relative to the parent font size  
↳ em means "my parent element's font size"
2. rem → Unit relative to the root font size (<html> tag)
3. vw → Unit relative to 1% Viewport width.
4. vh → Unit relative to 1% Viewport height.
5. % → Unit relative to the parent element

min/max-height/width property

CSS has a min-height, max-height, min-width and max-width property.

If the content is smaller than the minimum height, minimum height will be applied.

Similar is the case with other related properties

The position property

Used to manipulate the location of an element

Following are the possible values:

- static : The default position - top / bottom / left / right / z-index has no effect.
- relative : The top / bottom / left / right / z-index will now work. Otherwise the element is in the flow of document like static.
- absolute : The Element is removed from the flow & is relatively positioned to its first non-static ancestor - top / bottom etc works
- fixed : Just like absolute except the element is positioned relative to the browser window
- sticky : The Element is positioned based on user's scroll position

List-style property

The list style property is a shorthand for type, position & image

ul {

    list-style : square inside url('harry.jpg')

}

    list-style-type

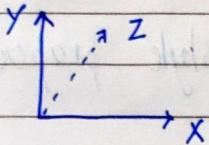
    list-style-position

    list-style-image

z-index property

The z-index property specifies the stack order of an element.

It defines which layer will be above which in case of overlapping elements.



=> Z is the third dimension.

## Chapter 5 - Practice Set

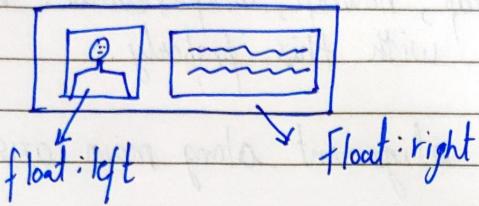
- 1 Create a responsive navbar using relative lengths
- 2 Create a sticky navbar using position property
- 3 Demonstrate the use of list-style property using a UL as example
- 4 Demonstrate the use of z-index using an example

## Chapter 6 - Flexbox

Before we look into the CSS flexbox, we will look into float and clear properties.

The float property

Float property is simple. It just flows the element towards left/right



The clear property

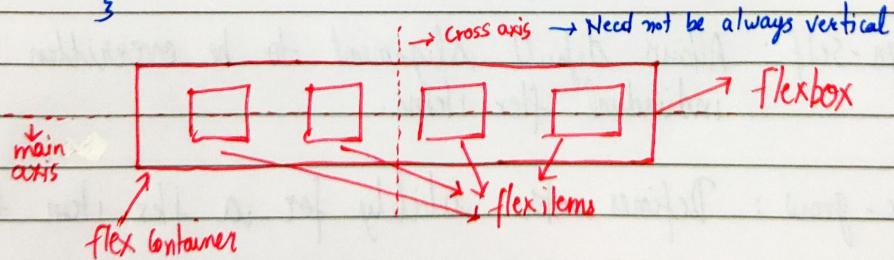
Used to clear the float. It specifies what elements can float beside a given element

The CSS flexbox

Aims at providing a better way to layout, align and distribute space among items in a container.

Container ↗

display: flex; ⇒ Initialize a flexbox  
↗



## flex-direction property

Defines the direction towards which items are laid.  
Can be row, row-reverse, column, column-reverse  
**default**

## Flex properties for parent (flex container)

Following are the properties for the flex parent:

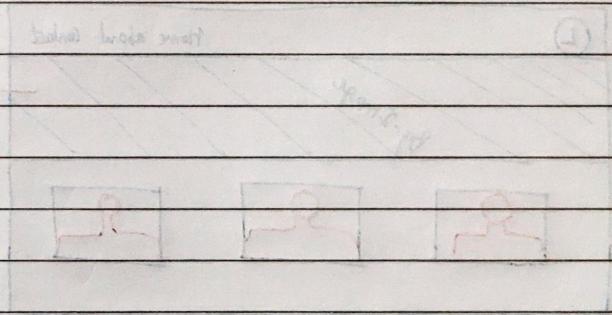
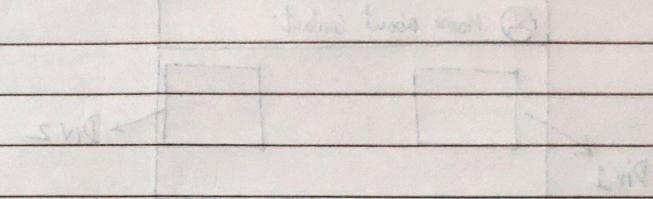
- 1 flex-wrap: Can be wrap, nowrap, wrap-reverse. Wrap items as needed with this property.
- 2 justify-content: Defines alignment along main axis.
- 3 align-items: Defines alignment along cross axis.
- 4 align-content: Aligns a flex container's lines when there is extra space in the cross axis.

## Flex properties for the children (flex items)

Following are the properties for the flex children.

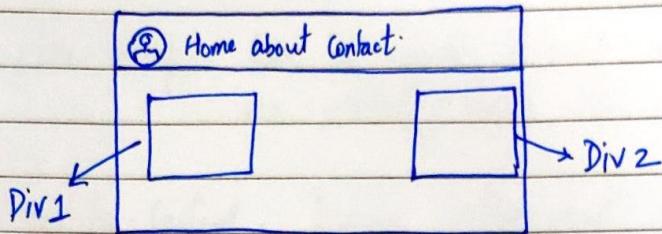
- 1 order: Controls the order in which the items appear in the flex container
- 2 align-self: Allows default alignment to be overridden for the individual flex items.
- 3 flex-grow: Defines the ability for a flex item to grow

4 flex-shrink : Specifies how much a flex item will shrink relative to the rest of the flex items.

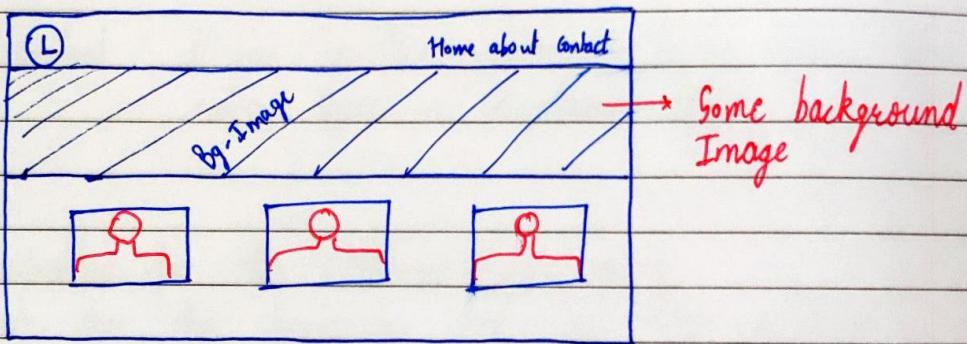


## Chapter 6 - Practice Set

- 1 Create a layout of your choice using float.
- 2 Create the same layout in ① using flexbox.
- 3 Create the following navigation bar Using flexbox



- 4 Create the following layout using flexbox:



## Chapter 7 - CSS Grid & Media Queries

A CSS grid can be initialized using :

```
Container {  
    display: grid;  
}
```

All direct children automatically becomes grid items

The grid-column-gap property

Used to adjust the space between the columns of a CSS grid

The grid-row-gap property

Used to adjust the space between the rows of a CSS grid.

The grid-gap property

Shorthand property for grid-row-gap & grid-column-gap

```
Container {  
    display: grid;  
    grid-gap: 40px 100px;  
}
```

Note : For a single value of grid-gap, both row and column gaps can be set in one value.

Following are the properties for grid container:

- 1, The `grid-template-columns` property can be used to specify the width of columns

Container {

`display: grid;`

`grid-template-columns: 80px 120px auto;`

- 2, The `grid-template-rows` property can be used to specify the height of each row

Container {

`display: grid;`

`grid-template-rows: 70px 150px;`

- 3, The `justify-content` property is used to align the whole grid inside the container.

- 4, The `align-content` property is used to vertically align the whole grid inside the container.

Following are the properties for grid item:

- 1, The `grid-column` property defines how many columns an item will span.

grid-item {

`grid-column: 1/5;`

2 The grid-row property defines how many rows an item will span.

3 We can make an item to start on column 1 and span 3 columns like this :

```
.item {  
    grid-column: 1 / span 3;  
}
```

## CSS Media Queries

Used to apply CSS only when a certain condition is true.

Syntax :

```
@media only screen and (max-width: 800px) {  
    body {  
        background: red;  
    }  
}
```

## Chapter 7 - Practice Set

- 1 Create a header with content using CSS grid.
- 2 Create the layouts created in Chapter 6 - Practice Set using CSS grid.
- 3 Create a webpage which is green on large devices, red on medium & yellow on small devices.

## Chapter 8 - Transforms, Transitions & Animations

Transforms are used to rotate, move, skew or scale elements. They are used to create a 3-D effect

The transform property

Used to apply a 2D or 3D transformation to an element

The transform-origin property

Allows to change the position of transformed elements

2D transforms → can change x & y axis

3D transforms → can change z axis as well

CSS 2D transform methods

You can use the following 2-D transforms in CSS:

- 1> translate()
- 2> rotate()
- 3> scaleX()
- 4> scaleY()
- 5> skew()
- 6> matrix()
- 7> scale()

CSS 3D transform methods

- 1> rotateX()
- 2> rotateY()
- 3> rotateZ()

## CSS Transitions

Used to change property values smoothly, over a given duration.

The transition property

The transition property is used to add transition in CSS.

Following are the properties used for CSS transition.

1. transition-property → The property you want to transition
2. transition-duration → Time for which you want transition to apply
3. transition-timing-function → How you want the property to transition
4. transition-delay → Specifies the delay for the transition

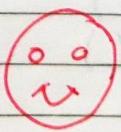
All these properties can be set using a single shorthand property

transition : width 3s ease-in 2s;  
① property      ② duration      ③ timing-function      ④ delay

Transitioning multiple properties

We can transition multiple properties as follows:

transition : opacity 1s ease-out 1s, transform 2s ease-in;



Yes you can  
skip transition  
delay here!

## CSS Animations

Used to animate CSS properties with more control.  
We can use @keyframes rule to change the animation from a given style to a new style.

```
@keyframes harry {  
from { width: 20px; } → Can change multiple properties  
to { width: 31px; }  
}
```

Properties to add Animations

Following are the properties used to set animation in CSS:

- 1, animation-name → name of the animation
- 2, animation-duration → How long does the animation run?
- 3, animation-timing-function → Determines speed curve of the animation
- 4, animation-delay → Delay for the start of an animation
- 5, animation-iteration-count → Number of times an animation should run
- 6, animation-direction → Specifies the direction of the animation

The Animation shorthand

All the animation properties from 1-6 can be applied like this:

animation: harry 6s linear 1s infinite reverse;  
① ② ③ ④ ⑤ ⑥

Using percentage value States with animation

We can use % values to indicate what should happen when a certain percent of animation is completed

@ Keyframes harry {

0% {

width: 20px;

}

⇒ Can add as many intermediate properties as possible

50% {

width: 80px;

}

100% {

width: 200px;

}

}

## Chapter 8 - Practice Set

- 1 Create a thin progress bar animation for a website
- 2 Create the same progress bar using transition
- 3 Create a rotating image animation using CSS
- 4 Create a slider with 3 images using CSS.

## Project 1 - E Commerce Website

Create a homepage for an E-commerce website.  
Use media queries to make it responsive.

# Front End Development - HTML

# Table of Content

- What is HTML?
- Features
- HTML Editor
- HTML Skeleton
- Comments
- HTML Elements
- Basic Tags
- Attributes

# What is HTML?

- Stands for HyperText Markup Language
- HyperText: Link between web pages.
- Markup Language: Text between tags which defines structure.
- It is a language to create web pages
- HTML defines how the web page looks and how to display content with the help of elements
- It forms or defines the structure of our Web Page
- Need to save your file with .html extension

# Features Of HTML

- The learning curve is very easy (easy to modify)
- Create effective presentations
- Add links wherein we can add references
- Can display documents on platforms like Mac , Windows, Linux etc
- Add videos, graphics and audios making it more attractive.
- Case insensitive language

# HTML Editors

- Simple editor: Notepad
- Notepad++
- Atom
- Best editor: Sublime Text.

# HTML Skeleton

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>

</body>
</html>
```

## <!DOCTYPE html>

Instruction to the browser about the HTML version.

## <html>

Root element which acts as a container to hold all the code

Browser should know that this a HTML document

Permitted content: One head tag followed by one body tag

## <head>

Everything written here will never be displayed in the browser

It contains general information about the document

Title, definitions of css and script sheets

Metadata(information about the document)

## <body>

- Everything written here will be displayed in the browser
- Contains text, images, links which can be achieved through tags.
- Examples:
  - <p> This is our first paragraph. </p>
  - <a href="http://www.google.com">Go To Google</a>
  - 

# HTML Comments

- Comments don't render on the browser
- Helps to understand our code better and makes it readable.
- Helps to debug our code
- Three ways to comment:
  - Single line
  - Multiple line
  - Comment tag //Supported by IE

# HTML Element

- Elements are created using tags
- Elements are used to define semantics
- Can be nested and empty

## Basic Structure

```
<p color="red"> This is our first Paragraph </p>
```

- Contains following things:
  - Start tag: <p>
  - Attributes: color ="red"
  - End tag: </p> // optional
  - Content: This is our first Paragraph

# Element Types

- **Block Level :**
  - Takes up full block or width and adds structure in the web page
  - Always starts from new line
  - Always end before the new line
  - Example :
    - <p >
    - <div>
    - <h1>...<h6>
    - <ol>
    - <ul>

# Element Types

- **Inline Level:**
  - Takes up what is requires and adds meaning to the web page
  - Always starts from where the previous element ended
  - Example :
    - <span>
    - <strong>
    - <em>
    - <img>
    - <a>

# Basic Tags:

- Enclosed within <>
- Different tags render different meaning .
- **<title> tag**
  - Whatever is written this tag comes up in the web page's tab
  - Defines the title of the page
  - Syntax: <title>Home </title>
- **<p> tag**
  - Defines the paragraph
  - Syntax:<p > This is our first Paragraph </p>

## List of Self closing tags

- **<hr> tag**
  - Stands for horizontal rule
  - Dividing the web page
- **<br> tag**
  - Stands for break line
  - Moving to next line
- **<img> tag**
  - To add images in the web page

- **<h1> tag .....** **<h6>tag**
  - Stands for heading tag
  - Defines heading of a page
  - h1 represents most important page in the page
  - h6 represents least important page in the page
- **<strong> tag**
  - Defines the text to be bold
  - Replaced <b>tag //HTML5
- **<em> tag**
  - Defines the text to be bold
  - Replaced <i>tag //HTML5

- **<ol> tag**
  - Stands for ordered list
  - To define series of events that take place in some order
  - Example making a tea (like a flow chart)
  - <ol>.....</ol>
- **<ul> tag**
  - Stands for unordered list
  - To define series of events that take place where order is not important.
  - Example your hobbies
  - <ul>.....</ul>

- **<li> tag**
  - Defines the list item
  - Used inside the ‘ol’ and ‘ul’ tag to define the events
  - <li></li>
- **<div> and <span> tags**
  - Both of these are used to group different tags .
  - Acts like a container.
  - Effective while styling.
  - <div>.....</div>
  - <span>....</span>
  - Difference <div> is block level and <span> is inline level.

- **<img> tag**
  - Used to add images in a web page
  - Syntax: 
  - Self closing tag.
- **<a> tags**
  - Used to add links in a web page
  - <a href="url"> Name of the link </a>

- **<table> tag**

- Used to create a table on a web page
- Need other tags for completing the creation of a table
  - <tr> : for marking the table row
  - <th> : for table header
  - <td> : for table column data
- Everything is always enclosed within <tr>
  - <thead> : to keep all header data
  - <tbody> : to keep all body data

- <form> tag

- Action attribute: It specifies the URL to send form data to
- Method attribute: specifies the type of HTTP request(GET or POST)
- Example: <form action="/my-form-submitting-page" method="POST">
- <input>: used to accept data from the user
- Some types of inputs are:

- **Text:** used to store text data. Syntax: type="text"
- **Password:** used to enter a secure password. Syntax: type="password"
- **Placeholder:** temporary text in input fields. It is generally accompanied by "text" and "password" attributes. Syntax: placeholder="insert- text-here"
- **Button:** used to include buttons in the form. Syntax: type="button" value="insert-text-here"
- **Submit button:** For creating a submit button. All the data will get submitted when it is clicked. Syntax: type="submit"
- **Checkbox:** to provide the ability to check multiple options. Syntax: type="checkbox". To check options by default, set it with the checked attribute.

- **Radio Button:** allows one to choose a single option. Syntax: type="radio". Keep the name attribute of all the options the same.
- **<select>:** For every possible option to select, use an <option> tag<option>
- **Text Areas:** multi-line plain-text editing control. Syntax: <textarea>. You can specify how large the text area is by using the "rows" and "cols" attributes
- **Labels:** add captions for individual items in a form. Syntax: <label>. A label can be used by placing the control element inside the <label> element, or by using the "for" and "id" attributes.
- Validations ensure that users fill out forms in the correct format, e.g.:
  - a. **required:** The Boolean attribute which makes a field mandatory:
  - b. **email:** the browser will ensure that the field contains an @ symbol.

# Attributes

- Properties associated with each tag.
- <tag name="value"></tag> is the structure.
- Global Attribute:
  - Title : Add extra information (hover)
  - Style: Add style information(font,background,color,size)
- 
  - src is the attribute used in image tag to define path
  - Width is attribute used to define width in pixels
  - Alt i.e alternate text if image is not loaded
- <a href="url"> Name of the link </a>
  - href used to define path of the link.

# Conclusion

- Introduction to HTML
- Comments
- HTML Elements
- Basic Tags
- Attributes

# Thank You

## Chapter 0 - Introduction

HTML → Hyper Text Mark Up Language

HTML is the language of the web. It is used to create websites

We use HTML tags to define look & feel of a website

With understanding of these tags and how to put them together, we can create beautiful websites easily!

Then why CSS & JavaScript

HTML is used for defining layout of a page - A barebone page structure

CSS is used to add styling to that barebone page created using HTML

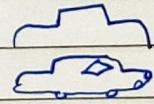
JavaScript is used to program logic for the page layout eg. What happens when a user hovers on a text, When to hide or show elements etc.

A Beautiful analogy

HTML = Car body (only metal)

CSS = Car paint, decoration etc.

JavaScript = Car engine + Interior logic



We will start learning how to build beautiful layouts in this course.

## Installing VS Code

We can use any text editor of our choice. Here I am using VS Code because it is light weight, open source & from Microsoft.

Go to google, type VS Code & install it

Note : You can write HTML even in Notepad. Text editors like VS Code just makes these things easier

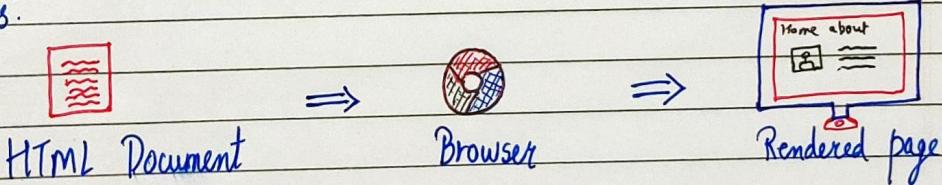
## Chapter 1 - Creating our first website

We start building a website by creating a file named index.html  
 index.html is a special filename which is presented when the website root address is typed.

### A Basic HTML Page

```
<!DOCTYPE html> → Specifies this is an HTML5 doc
<html> → Root of an HTML page
<head> → Contains page metadata
<title> Harry's Website </title> → Contains title
</head>
<body> → The main body of the page (rendered by the browser)
<h1> This is a heading </h1> → heading tag
<p> My paragraph </p> → paragraph tag
</body> → Closing body tag
</html> →
```

A tag is like a container for either content or other HTML tags.



### Imp Notes

- Head & body tags are children of HTML tag.
- HTML is the parent of Head & Body tags
- Most of the HTML elements have opening & closing tag with content in between opening & closing tags.
- Some HTML tags have no content. These are called Empty elements eg <br>

- We can either use .htm or .html extension
- You can use "Inspect Element" or "View Page Source" option from chrome to look into a website's HTML code.

HTML element = Start tag + Content + End tag

### Comments in HTML

Comments in HTML are used to mark text which should not be parsed. They can help document the source code:

<!-- HTML Comment -->

### Case Sensitivity

HTML is a case insensitive language. <H1> and <h1> tags are the same.

## Chapter 1 - Practice Set

- = 1 Inspect your favorite website and change something on the page which is displayed.
- = 2 Go to your favorite website and try to view the page source and write the exact lines of code. Does it clone the website? Why?
- = 3 Write any HTML code inside a text file. Does it work if you write it using notepad?

## Chapter 2 - Basic HTML Tags

We can add elements inside the body tag to define the page layout.

### HTML Element

Everything from starting to the ending tag.

<body> → Opening tag  
→ Content ←  
</body> → Closing tag

### HTML Attributes

Used to add more information corresponding to an HTML tag.

Example : <a href="https://codewithharry.com/"> Harry </a>  
                    ↑  
                    anchor tag  
                    ↓  
                    href attribute

We can either use single or double quotes in attributes

### The Heading Tag

Heading tag is used to mark headings in HTML. From h1 to h6, we have tags for the most important to the least important heading.

<h1> Most Important heading </h1>

Note: We should not use

<h2> Another heading H2 </h2>

HTML headings to make

<h3> Another heading H3 </h3>

text thick or bold.

<h4> Another heading H4 </h4>

<h5> Another heading H5 </h5>

<h6> Another heading H6 </h6>

## The Paragraph Tag

Paragraph tags are used to add paragraphs to an HTML page.

< p > This is a paragraph </ p >

## The Anchor Tag

The Anchor tag is used to add links to an existing content inside an HTML page.

< a href="https://google.com" > Click me </ a >

## The img Tag

img tag is used to add images in an HTML page

< img src="image.jpg" >  
    ↳ relative url of an image

## Bold, italic and underline tags

We can use bold, italic and underline tags to highlight the text as follows:

< b > This is bold </ b >

< i > This is italic </ i >

< u > This is underline </ u >

## br tag

The br tag is used to create line breaks in an HTML document.

big and small tags

We can make the text a bit larger and a bit smaller using big and small tags respectively.

hr tag

<hr> tag in HTML is used to create a horizontal ruler often used to separate the content.

Subscript & superscript

We can add subscript and superscripts in HTML as follows:

<sub> this </sub> is subscript

<sup> this </sup> is superscript

pre tag

HTML always ignores extra spaces and newlines. In order to display a piece of text as is, we use pre tag

<pre>

This is written

using pre  
tag

</pre>

⇒ Rendered as-is

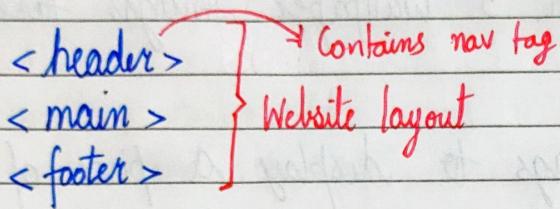
## Chapter 2 - Practice Set

- 1 Create an HTML page with a heading (title heading), a primary heading and a sub-heading.  
Which tags did you use?
- 2 Create a page with 5 wallpaper images taken from the internet
- 3 Use br and hr tags to display a piece of text with linebreaks.
- 4 Try to write the following chemical equation using HTML.  
$$\text{C} + \text{O}_2 \longrightarrow \text{CO}_2$$
- 5 Try to write a wikipedia article using HTML.

## Chapter 3 - Creating a page layout

When we use the right tag in right place, it results in a better page layout, better indexing by search engines and better user experience.

We use the following tag to get the job done



Inside the main tag we insert the following tags:

- <main> → The main opening tag
- <section> → A page section
- <article> → a self contained content
- <aside> → Content aside from the content (eg Ads etc)
- </main> → The main closing tag

Creating a page like this is not necessary but it creates a readable & structured layout.  
Also they are useful for SEO.

Link attributes

- <a href = "/contact"> Contact us </a> → Contact page opens in same tab
- <a href = "/contact" target = "\_blank"> Contact us </a> → opens in a new tab

We can put any content inside an anchor tag (images, headings etc are all allowed)

If the page is inside a directory, we need to make sure that we link to the correct page.

↳ Same applies to img tag as well

We can add links to images like this

```
<a href="/about"><img src='a.jpg' width="120"></a>
```

↳ Height will be set automatically

The Div tag

div tag is often used as a container for other elements  
div is a block level element.

↳ Always takes full width

The Span tag

Span is an inline container.

↳ Takes as much width as necessary

## Chapter 3 - Practice Set

- 1 Create an SEO friendly website using HTML.
- 2 Create an HTML page which opens google when clicked on an image.
- 3 Create a website which has your 5 top used websites bookmarked. The links should open in a new tab.

## Chapter 4 - Lists, tables & forms

### Lists

Lists are used to display content which represents a list.

Unordered list : Used to list unordered items

<ul>

<li> Home </li>

<li> About </li>

⋮

</ul>

Ordered list : used to list ordered items

<ol>

<li> Phone </li>

<li> PC </li>

<li> Laptop </li>

</ol>

### Tables

The <table> tag is used to define tables in HTML.  
It is used to format & display tabular data.

tr tag : used to display table row

td tag : used to display table data

th tag : used in place of table data for displaying table headers

We can define as many table rows as we want.

To add a caption to the table, we use `<caption>` tag inside table.

`thead` tag : Used to wrap table head (Caption & `tr` with `th`)

`tbody` tag : Used to wrap the table body

`Colspan` attribute

This attribute is used to create cells spanning multiple columns.

`<th Colspan = "3"> Harry </th>`

→ Spans 3 columns

HTML forms

An HTML form is used to collect input from the user  
`form` tag is used for the same

`<form>`

-- Element of the form --

`</form>`

There are different form elements for different kinds of user input

→ `input` element : Can be of type text, checkbox, radio, button and submit. We also have a 'file' type

→ `textarea` element : Defines a multi line text input. `cols` and `rows` attributes can be used to size the textarea.

→ `Select` element : Defines a drop down list

Note : you don't have to remember all the tags, you will automatically memorize them with practice

## Embedding Videos

Video tag is used to play videos in HTML

<video src = 'harry.mp4'> Error </video>

Attributes for video

We can use :

- Width : To adjust width of a video (Height automatically adjusts)
- We can use autoplay/loop to autoplay or loop the video.

## Chapter 4- Practice Set

- 1 Create an HTML page with video embedded inside it.
- 2 Replace this video in 1 with a YouTube video.
- 3 Create an HTML form for a travel website to book a vacation
- 4 Create a table displaying score of cricket players in a match using HTML

## Chapter 5 - SEO

We will focus only on HTML standpoint of SEO. We will not be looking into keyword building and content optimization aspect of SEO.

### Types of SEO

- On page SEO → Can be done by HTML developers
- Off page SEO

### HTML SEO

HTML developers can implement SEO using the following techniques :

1> Set the title very nice & to the point

2> Set the meta description

<meta name="description" content="...">>

3> Set a nice URL slug

4> Set the meta keywords tag.

5> Set the meta author tag.

<meta name="author" content="Harry">>

6> Set a favicon

7 Compress images & other resources

8 Remove unused HTML/CSS & JS files + Compress them

9 Add alt text to images