



GUI BASICS

Documentation

[Abstract](#)

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.
When you're ready to add your content, just click here and start typing.]

Dhruvil Dobariya
dhruvildobariya21@gmail.com

INDEX

1	BASIC OF HTML.....	1
1.1	WHAT IS HTML?.....	1
1.2	HISTORY OF HTML.....	1
1.3	STRUCTURE OF HTML.....	1
2	BASIC CONTROLS	3
2.1	FORM:	3
2.2	FILE CONTROLS	4
3	CONTROL'S ATTRIBUTES	7
3.1	INTRODUCTION	7
3.2	NAME.....	7
3.3	VALUE:.....	7
3.4	ID.....	7
3.5	CLASS	8
4	BASIC TAG WITH ITS ATTRIBUTE	9
4.1	ACTION TAG.....	9
4.2	IMAGE TAG	11
4.3	META TAG.....	12
4.4	RESPONSIVENESS.....	13
4.5	OTHER TAGS	14
5	BASIC OF CSS	26
5.1	INTRODUCTION	26
5.2	TYPES OF CSS.....	26
5.3	CSS SYNTAX.....	27
5.4	SELECTORS	27
5.5	PROPERTIES.....	32
6	BASIC OF BOOTSTRAP	39
6.1	INTRODUCTION:	39
6.2	HOW TO USE BOOTSTRAP.....	39
6.3	STRUCTURE OF BOOTSTRAP.....	39
7	BASIC OF JAVASCRIPT	42
7.1	INTRODUCTION	42
7.2	USE OF JAVASCRIPT	42
7.3	WAY TO USE JAVASCRIPT	42
7.4	SYNTAX OF JAVASCRIPT	43
7.5	EVENTS IN JAVASCRIPT	43
7.6	EVENTLISTENER.....	45
7.7	VALIDATION	46

BASIC OF HTML

1.1 WHAT IS HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

1.2 HISTORY OF HTML

- The first version of HTML was written by Tim Berners-Lee in 1993.
- Since then, there have been many different versions of HTML.
- The most widely used version throughout the 2000's was HTML 4.01, which became an official standard in December 1999.
- Currently Html 5 is stable version of html.

1.3 STRUCTURE OF HTML

Example:

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

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

</body>
</html>
```

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document.
- The <html> element is the root element of an HTML page.
- The <head> element contains meta information about the HTML page.

- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab).
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading.
- The <p> element defines a paragraph.

BASIC CONTROLS

2.1 FORM:

- The <form> tag is used to create an HTML form for user input.
- The <form> element can contain one or more of the following form elements,
 - <input>
 - <textarea>
 - <button>
 - <select>
 - <option>
 - <optgroup>
 - <fieldset>
 - <label>
 - <output>

2.1.1 ATTRIBUTES:

- The <form> tag also supports the Global Attributes in HTML.
- The <form> tag also supports the Event Attributes in HTML.
- The <form> tag has own attributes, like...

Attribute	Value	Description
accept-charset	character_set	Specifies the character encodings that are to be used for the form submission
action	URL	Specifies where to send the form-data when a form is submitted
autocomplete	on off	Specifies whether a form should have autocomplete on or off
enctype	application/x-www-form-urlencoded	Specifies how the form-data should be encoded when submitting it to the server (only for method="post")

	multipart/form-data text/plain	
method	get post	Specifies the HTTP method to use when sending form-data
name	text	Specifies the name of a form
novalidate	novalidate	Specifies that the form should not be validated when submitted
rel	external help license next nofollow noopener noreferrer opener prev search	Specifies the relationship between a linked resource and the current document
target	_blank _self _parent _top	Specifies where to display the response that is received after submitting the form

2.2 FILE CONTROLS

- The `<input type="file">` defines a file-select field and a "Browse" button for file uploads.
- To define a file-select field that allows multiple files to be selected, add the `multiple` attribute.

Example:

```
<input type="file" name="" id="">
```

2.2.1 ATTRIBUTES:

- It contains all common attribute which have input tag.
- Like, id, name, class...

Value:

- A file input's value attribute contains a string that represents the path to the selected file(s).
- If no file is selected yet, the value is an empty string ("").
- When the user selected multiple files, the value represents the first file in the list of files they selected.

Accept:

- The accept attribute value is a string that defines the file types the file input should accept.
- This string is a comma-separated list of unique file type specifiers.
- Because a given file type may be identified in more than one manner, it's useful to provide a thorough set of type specifiers when you need files of a given format.
- A valid case-insensitive filename extension, starting with a period (".") character.
- For example: .jpg, .pdf, or .doc.
- A valid MIME type string, with no extensions.
- The string audio/* meaning "any audio file".
- The string video/* meaning "any video file".
- The string image/* meaning "any image file".

Example:

```
<input type="file" accept="image/*,.pdf" />
```

Capture:

- The capture attribute value is a string that specifies which camera to use for capture of image or video data.
- if the accept attribute indicates that the input should be of one of those types.
- A value of user indicates that the user-facing camera and/or microphone should be used.
- A value of environment specifies that the outward-facing camera and/or microphone should be used.

GUI Basic

- If this attribute is missing, the user agent is free to decide on its own what to do.
- If the requested facing mode isn't available, the user agent may fall back to its preferred default mode.

Multiple:

- When the multiple Boolean attribute is specified, the file input allows the user to select more than one file.

2.2.2 GETTING INFORMATION ON SELECTED FILES

- The selected files' are returned by the element's `HTMLInputElement.files` property, which is a `FileList` object containing a list of `File` objects.
- The `FileList` behaves like an array, so you can check its `length` property to get the number of selected files.
- Each `File` object contains the following properties,

name:

- The file's name.

lastModified:

- A number specifying the date and time at which the file was last modified, in milliseconds since the UNIX epoch (January 1, 1970 at midnight).

size:

- The size of the file in bytes.

type:

- The file's MIME type.

CONTROL'S ATTRIBUTES

3.1 INTRODUCTION

- HTML have some basic control attributes like,
 - name
 - id
 - class
 - value

3.2 NAME

- The name attribute specifies the name of an <input> element.
- The name attribute is used to reference elements in a JavaScript, or to reference form data after a form is submitted.
- Only form elements with a name attribute will have their values passed when submitting a form.

Example:

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

3.3 VALUE:

- The value attribute specifies the value of an <input> element.

Example:

```
<label for="name">Name : </label>
<input type="text" name="name" id="name" value="Dhruvil Dobariya">
```

3.4 ID

- The HTML id attribute is used to specify a unique id for an HTML element.
- You cannot give more than one element with the same id in an HTML document.
- The id attribute is used to point to a specific style declaration in a style sheet.
- It is also used by JavaScript to access and manipulate the element with the specific id.
- The syntax for id is: write a hash character (#), followed by an id name.
- Id also used for bookmark.

- To use a bookmark, you must first create it, and then add a link to it.
- Then, when the link is clicked, the page will scroll to the location with the bookmark.
- It is a global attribute, means it can be used with all HTML elements.

3.5 CLASS

- The class attribute specifies one or more classnames for an element.
- The class attribute is mostly used to point to a class in a style sheet.
- It can also be used by a JavaScript (via the HTML DOM) to make changes to HTML elements with a specified class.
- It is a global attribute, means it can be used with all HTML elements.

BASIC TAG WITH ITS ATTRIBUTE

4.1 ACTION TAG

- The <a> tag defines a hyperlink, which is used to link from one page to another.
- The most important attribute of the <a> element is the href attribute, which indicates the link's destination.
- 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
- But we can change these color using CSS.
- The <a> tag also supports the Global Attributes and Event Attribute in HTML.

4.1.1 ATTRIBUTES:

Attribute	Value	Description
download	filename	Specifies that the target will be downloaded when a user clicks on the hyperlink
href	URL	Specifies the URL of the page the link goes to
hreflang	language_code	Specifies the language of the linked document
media	media_query	Specifies what media/device the linked document is optimized for
ping	list_of_URLs	Specifies a space-separated list of URLs to which, when the link is followed, post requests with the body ping will be sent by the

		browser (in the background). Typically used for tracking.
referrerpolicy	no-referrer no-referrer-when-downgrade origin origin-when-cross-origin same-origin strict-origin-when-cross-origin unsafe-url	Specifies which referrer information to send with the link
rel	alternate author bookmark external help license next nofollow noreferrer noopener prev search tag	Specifies the relationship between the current document and the linked document
target	_blank _parent _self _top	Specifies where to open the linked document
type	media_type	Specifies the media type of the linked document

4.2 IMAGE TAG

- The tag is used to embed an image in an HTML page.
- Images are not technically inserted into a web page, images are linked to web pages.

4.2.1 ATTRIBUTES:

Attribute	Value	Description
alt	text	Specifies an alternate text for an image
crossorigin	anonymous use-credentials	Allow images from third-party sites that allow cross-origin access to be used with canvas
height	pixels	Specifies the height of an image
ismap	ismap	Specifies an image as a server-side image map
loading	eager lazy	Specifies whether a browser should load an image immediately or to defer loading of images until some conditions are met
longdesc	URL	Specifies a URL to a detailed description of an image
referrerpolicy	no-referrer no-referrer-when-downgrade origin origin-when-cross-origin unsafe-url	Specifies which referrer information to use when fetching an image
sizes	sizes	Specifies image sizes for different page layouts
src	URL	Specifies the path to the image

srcset	URL-list	Specifies a list of image files to use in different situations
usemap	#mapname	Specifies an image as a client-side image map
width	pixels	Specifies the width of an image

4.3 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.
- It is used to specify,
 - specify character set
 - page description
 - keywords
 - author of the document
 - viewport settings
- It will not be displayed on the page, but is machine parsable.
- It is used by browsers (how to display content or reload page), search engines (keywords), and other web services.
- There is a method to let web designers take control over the viewport (the user's visible area of a web page), using the <meta> tag.

4.3.1 ATTRIBUTES:

Attribute	Value	Description
charset	character_set	Specifies the character encoding for the HTML document
content	text	Specifies the value associated with the http-equiv or name attribute

GUI Basic

http-equiv	content-security-policy content-type default-style refresh	Provides an HTTP header for the information/value of the content attribute
name	application-name author description generator keywords viewport	Specifies a name for the metadata

Example:

```
<head>
  <!-- Define keywords for search engines: -->
  <meta name="keywords" content="HTML, CSS, JavaScript">

  <!-- Define a description of your web page: -->
  <meta name="description" content="About meta tag">

  <!-- Define the author of a page: -->
  <meta name="author" content="Dhruvil A. Dobariya">

  <!-- Refresh document every 30 seconds: -->
  <meta http-equiv="refresh" content="30">

  <!-- Setting the viewport to make your website look good on all devices:
-->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>
</head>
```

4.4 RESPONSIVENESS

GUI Basic

- It is a web development approach that creates dynamic changes to the appearance of a website, depending on the screen size and orientation of the device being used to view it.
- For archive responsiveness we have three approach,
 - Media Query
 - Web Browser
 - Website Interface(HTML, CSS, Javascript)

Media Query:

- Media queries are supported by the latest web browsers and work by creating media queries.
- They are called “media queries” and can be used in various ways, such as in connection with the layout of a page or as part of the content of an application.
- They also permit designers to fabricate various formats utilizing similar HTML archives by specifically serving templates dependent on the client specialist’s highlights, like the browser’s window’s size.

Web Browser:

- Since websites can contain many images, it is essential to keep these images fluid too.
- The thought behind liquid pictures is that you convey concepts at the greatest size they will utilize.
- In responsive web pages, designers don’t announce the stature and width in your code instead, they let the browsers resize the pictures on a case by case basis while utilizing CSS to manage their relative size.
- It’s an extraordinary and straightforward procedure to resize images correctly.

Website Interface(HTML, CSS, Javascript):

- Many mobile devices are touchscreen nowadays, which requires mindfulness of the size of the interactive elements within interfaces.
- Aside from the flexibility of images, usage of native controls such as keyboards and drop-out menus should be optimized so it can provide a great experience to its users, whether it’s on mobile or desktop.

4.5 OTHER TAGS

4.5.1 BASIC TAGS:

Tag	Description
<!DOCTYPE>	Defines the document type
<html>	Defines an HTML document
<head>	Contains metadata/information for the document
<title>	Defines a title for the document
<body>	Defines the document's body
<h1> to <h6>	Defines HTML headings
<p>	Defines a paragraph

	Inserts a single line break
<hr>	Defines a thematic change in the content
<!--...-->	Defines a comment

4.5.2 FORMATTING TAGS:

Tag	Description
<acronym>	Not supported in HTML5. Use <abbr> instead. Defines an acronym

<abbr>	Defines an abbreviation or an acronym
<address>	Defines contact information for the author/owner of a document/article
	Defines bold text
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<bdo>	Overrides the current text direction
<big>	Not supported in HTML5. Use CSS instead. Defines big text
<blockquote>	Defines a section that is quoted from another source
<center>	Not supported in HTML5. Use CSS instead. Defines centered text
<cite>	Defines the title of a work
<code>	Defines a piece of computer code
	Defines text that has been deleted from a document
<dfn>	Specifies a term that is going to be defined within the content

	Defines emphasized text
	Not supported in HTML5. Use CSS instead. Defines font, color, and size for text
<i>	Defines a part of text in an alternate voice or mood
<ins>	Defines a text that has been inserted into a document
<kbd>	Defines keyboard input
<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within a known range (a gauge)
<pre>	Defines preformatted text
<progress>	Represents the progress of a task
<q>	Defines a short quotation
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)

GUI Basic

<s>	Defines text that is no longer correct
<samp>	Defines sample output from a computer program
<small>	Defines smaller text
<strike>	Not supported in HTML5. Use or <s> instead. Defines strikethrough text
	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<template>	Defines a container for content that should be hidden when the page loads
<time>	Defines a specific time (or datetime)
<tt>	Not supported in HTML5. Use CSS instead. Defines teletype text
<u>	Defines some text that is unarticulated and styled differently from normal text
<var>	Defines a variable

<wbr>	Defines a possible line-break
--------------------	-------------------------------

4.5.3 FORMS AND INPUTS TAGS:

Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control
<textarea>	Defines a multiline input control (text area)
<button>	Defines a clickable button
<select>	Defines a drop-down list
<optgroup>	Defines a group of related options in a drop-down list
<option>	Defines an option in a drop-down list
<label>	Defines a label for an <input> element
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<datalist>	Specifies a list of pre-defined options for input controls

<output>	Defines the result of a calculation
-----------------------	-------------------------------------

4.5.4 FRAMES TAGS:

Tag	Description
<frame>	Not supported in HTML5. Defines a window (a frame) in a frameset
<frameset>	Not supported in HTML5. Defines a set of frames
<noframes>	Not supported in HTML5. Defines an alternate content for users that do not support frames
<iframe>	Defines an inline frame

4.5.5 IMAGES TAGS:

Tag	Description
	Defines an image
<map>	Defines a client-side image map
<area>	Defines an area inside an image map
<canvas>	Used to draw graphics, on the fly, via scripting (usually JavaScript)
<figcaption>	Defines a caption for a <figure> element

<figure>	Specifies self-contained content
<picture>	Defines a container for multiple image resources
<svg>	Defines a container for SVG graphics

4.5.6 AUDIO / VIDEO TAGS:

Tag	Description
<audio>	Defines sound content
<source>	Defines multiple media resources for media elements (<video>, <audio> and <picture>)
<track>	Defines text tracks for media elements (<video> and <audio>)
<video>	Defines a video or movie

4.5.7 LINKS TAGS:

Tag	Description
<a>	Defines a hyperlink
<link>	Defines the relationship between a document and an external resource (most used to link to style sheets)
<nav>	Defines navigation links

4.5.8 LIST TAGS:

Tag	Description
<code></code>	Defines an unordered list
<code></code>	Defines an ordered list
<code></code>	Defines a list item
<code><dir></code>	Not supported in HTML5. Use <code></code> instead. Defines a directory list
<code><dl></code>	Defines a description list
<code><dt></code>	Defines a term/name in a description list
<code><dd></code>	Defines a description of a term/name in a description list

4.5.9 TABLE TAGS:

Tag	Description
<code><table></code>	Defines a table
<code><caption></code>	Defines a table caption
<code><th></code>	Defines a header cell in a table

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

4.5.10 STYLES AND SEMANTICS:

Tag	Description
<style>	Defines style information for a document
<div>	Defines a section in a document
	Defines a section in a document
<header>	Defines a header for a document or section
<footer>	Defines a footer for a document or section

<main>	Specifies the main content of a document
<section>	Defines a section in a document
<article>	Defines an article
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<summary>	Defines a visible heading for a <details> element
<data>	Adds a machine-readable translation of a given content

4.5.11 META INFO TAGS:

Tag	Description
<head>	Defines information about the document
<meta>	Defines metadata about an HTML document
<base>	Specifies the base URL/target for all relative URLs in a document
<basefont>	Not supported in HTML5. Use CSS instead. Specifies a default color, size, and font for all text in a document

4.5.12 PROGRAMMING TAGS:

Tag	Description
<script>	Defines a client-side script
<noscript>	Defines an alternate content for users that do not support client-side scripts
<applet>	Not supported in HTML5. Use <embed> or <object> instead. Defines an embedded applet
<embed>	Defines a container for an external (non-HTML) application
<object>	Defines an embedded object
<param>	Defines a parameter for an object

BASIC OF CSS

5.1 INTRODUCTION

- CSS stands for Cascading Style Sheets.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- It can control the layout of multiple web pages all at once.

5.2 TYPES OF CSS

- We have three types of CSS,
 - Inline CSS
 - Internal or Embedded CSS
 - External CSS

5.2.1 INLINE CSS:

- Inline CSS contains the CSS property in the body section attached with element is known as inline CSS.
- This kind of style is specified within an HTML tag using the style attribute.

5.2.2 INTERNAL OR EMBEDDED CSS:

- This can be used when a single HTML document must be styled uniquely.
- The CSS rule set should be within the HTML file in the head section.

5.2.3 EXTERNAL CSS:

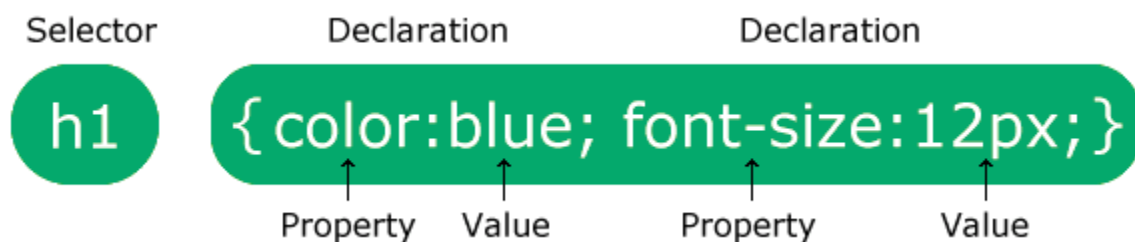
- External CSS contains separate CSS file which contains only style property with the help of tag attributes.
- CSS property written in a separate file with .css extension and should be linked to the HTML document using link tag.
- This means we can use CSS file in multiple HTML documents.

5.2.4 PRIORITY OF CSS:

- Inline CSS has the highest priority, then comes Internal/Embedded followed by External CSS which has the least priority.
- Multiple style sheets can be defined on one page.
- If for an HTML tag, styles are defined in multiple style sheets then the below order will be followed.

5.3 CSS SYNTAX

- 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.



Example:

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

- Here, p is a selector in CSS.
- color is a property, and red is the property value.
- text-align is a property, and center is the property value.

5.4 SELECTORS

- CSS have different types of selectors like,
 - Basic Selector
 - Grouping Selector
 - Combinator Selector
 - Pseudo Selector

5.4.1 BASIC SELECTOR:

- Basic selector contains five different types like,
 - Universal selector
 - Class selector

GUI Basic

- ID selector
- Attribute selector
- Type selector

Universal Selector:

- Selects all elements.
- Optionally, it may be restricted to a specific namespace or to all namespaces.

Syntax: * ns|* *|*

Example:

```
*{  
  background-color: rgb(209, 209, 209);  
  font-size: 25px;  
}
```

Class Selector:

- Selects all elements that have the given class attribute.

Syntax: .classname

Example:

```
.class-selector{  
  color: white;  
}
```

Id Selector:

- Selects an element based on the value of its id attribute.

Syntax: #idname

Example:

```
#Id-Selector{  
  color: red;  
}
```

Type(Tag) Selector:

- Selects all elements that have the given node name.

GUI Basic

Syntax: elementname

Example:

```
span{
  color: green;
}
```

Attribute Selector:

- Selects all elements that have the given attribute.

Syntax: [attr], [attr=value], [attr~=value], [attr|=value], [attr^=value], [attr\$=value], [attr*=value]

Example:

```
[type=text]{
  color: aqua;
}
```

5.4.2 GROUPING SELECTOR:

- Grouping selector contains five different types like,
 - Selector list

Select list Selector:

- Selects all the matching nodes.
- All node separated by “,”.

Syntax: elementname1, elementname2

Example:

```
i, .select-list-with-class{
  color: blue;
}
```

5.4.3 COMBINATOR SELECTOR

- Combinator selector contains five different types like,
 - Descendant combinator
 - Child combinator
 - General sibling combinator

- Adjacent sibling combinator

Descendant Combinator:

- It select elements which is exist inside the particular element.
- First we write element which contain our selector node and give space and we write second element which is our selector node.

Syntax: div span

- So here, “div span” will match all elements that are exist a <div> element.

Example:

```
div b{  
  color: blueviolet;  
}
```

Child Combinator:

- It select elements which is directly children of the particular element.
- First we write element which is **directly parent node** of our selector node and write “>” and we write second element which is our selector node.

Syntax: ui > li

- So here, “ui > li” will match all elements that are directly child of the <ui> element.

Example:

```
p>b{  
  color: brown;  
}
```

General Sibling Combinator:

- It select elements which is sibling of the particular element.
- First we write element which is sibling of our selector node and write “~” and we write second element which is our selector node.

Syntax: p ~ span

- So here, “p ~ span” will match all elements that are sibling of the <p> element.

Example:


```
h3~h4, h3~.sibling-combinator-with-class{
    color: cadetblue;
}
```

Adjacent Combinator:

- It select elements which **first and immediately follows** the particular element.
- First we write element which is followed by our selector node and write “+” and we write second element which is our selector node.

Syntax: h2 + p

- So here, “h2 + p” will match all <p> element that is first and followed the <h2> element.

Example:

```
header+footer{
    color: yellowgreen;
}
```

5.4.4 PSEUDO SELECTOR

- Pseudo selector contains five different types like,
 - Pseudo classes selector
 - Pseudo elements selector

Pseudo Classes Selector:

- It select element based on their state.
- First we write element which is our selector node and write “:” and we write state of that element when we want to select it in that state.

Syntax: a:visited

- So, here <a> is selected when it visited.

Example:

```
a:visited{
    color: coral;
}
input:invalid{
    background-color: red;
}
```

GUI Basic

Pseudo Element Selector:

- It is used to select particular parts of selector.
- First we write element which is our selector node and write “::” and we write part of that element which we want to select it.

Syntax: p::first-line

- So, here first line of <p> is selected.

Example:

```
p::first-line{
  color: rgb(73, 82, 27);
}
```

5.5 PROPERTIES

- CSS have many properties, like background, text, box-model...

5.5.1 BACKGROUND PROPERTIES:

Property	Description
background	Defines a variety of background properties within one declaration.
background-attachment	Specify whether the background image is fixed in the viewport or scrolls.
background-clip	Specifies the painting area of the background.
background-color	Defines an element's background color.
background-image	Defines an element's background image.
background-origin	Specifies the positioning area of the background images.
background-position	Defines the origin of a background image.
background-repeat	Specify whether/how the background image is tiled.
background-size	Specifies the size of the background images.

5.5.2 FONT PROPERTIES:

Property	Description
font	Defines a variety of font properties within one declaration.
font-family	Defines a list of fonts for element.
font-size	Defines the font size for the text.
font-size-adjust	Preserves the readability of text when font fallback occurs.

font-stretch	Selects a normal, condensed, or expanded face from a font.
font-style	Defines the font style for the text.
font-variant	Specify the font variant.
font-weight	Specify the font weight of the text.
color	Specify the color of the text of an element.

5.5.3 TEXT PROPERTIES:

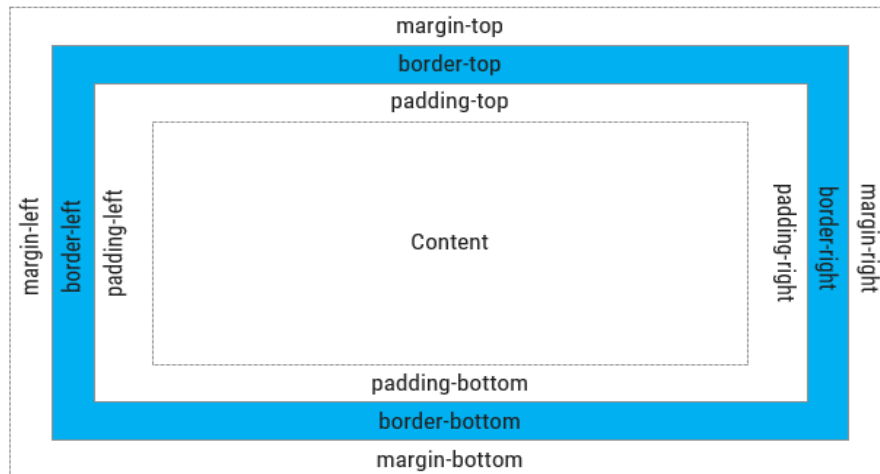
Property	Description
text-align	Sets the horizontal alignment of inline content.
text-align-last	Specifies how the last line of a block or a line right before a forced line break is aligned when text-align is justify.
text-decoration	Specifies the decoration added to text.
text-decoration-color	Specifies the color of the text-decoration-line.
text-decoration-line	Specifies what kind of line decorations are added to the element.
text-decoration-style	Specifies the style of the lines specified by the text-decoration-line property
text-indent	Indent the first line of text.
text-justify	Specifies the justification method to use when the text-align property is set to justify.
text-overflow	Specifies how the text content will be displayed, when it overflows the block containers.
text-shadow	Applies one or more shadows to the text content of an element.
text-transform	Transforms the case of the text.

5.5.4 BOX MODEL PROPERTIES:

- All HTML elements can be considered as boxes.
- 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 HTML elements, and it consists of margins, borders, padding, and the actual content.

GUI Basic

- The box model allows us to place a border around elements and space elements in relation to other elements.



Property	Description
padding-bottom	Sets the padding to the bottom side of an element.
padding-left	Sets the padding to the left side of an element.
padding-right	Sets the padding to the right side of an element.
padding-top	
margin	Sets the margin on all four sides of the element.
margin-bottom	Sets the bottom margin of the element.
margin-left	Sets the left margin of the element.
margin-right	Sets the right margin of the element.
margin-top	Sets the top margin of the element.
border	Sets the width, style, and color for all four sides of an element's border.
border-bottom	Sets the width, style, and color of the bottom border of an element.
border-bottom-color	Sets the color of the bottom border of an element.
border-bottom-left-radius	Defines the shape of the bottom-left border corner of an element.
border-bottom-right-radius	Defines the shape of the bottom-right border corner of an element.
border-bottom-style	Sets the style of the bottom border of an element.
border-bottom-width	Sets the width of the bottom border of an element.
border-collapse	Specifies whether table cell borders are connected or separated.

border-color	Sets the color of the border on all the four sides of an element.
border-image	Specifies how an image is to be used in place of the border styles.
border-image-outset	Specifies the amount by which the border image area extends beyond the border box.
border-image-repeat	Specifies whether the image-border should be repeated, rounded or stretched.
border-image-slice	Specifies the inward offsets of the image-border.
border-image-source	Specifies the location of the image to be used as a border.
border-image-width	Specifies the width of the image-border.
border-left	Sets the width, style, and color of the left border of an element.
border-left-color	Sets the color of the left border of an element.
border-left-style	Sets the style of the left border of an element.
border-left-width	Sets the width of the left border of an element.
border-radius	Defines the shape of the border corners of an element.
border-right	Sets the width, style, and color of the right border of an element.
border-right-color	Sets the color of the right border of an element.
border-right-style	Sets the style of the right border of an element.
border-right-width	Sets the width of the right border of an element.
border-spacing	Sets the spacing between the borders of adjacent table cells.
border-style	Sets the style of the border on all the four sides of an element.
border-top	Sets the width, style, and color of the top border of an element.
border-top-color	Sets the color of the top border of an element.
border-top-left-radius	Defines the shape of the top-left border corner of an element.
border-top-right-radius	Defines the shape of the top-right border corner of an element.
border-top-style	Sets the style of the top border of an element.
border-top-width	Sets the width of the top border of an element.
border-width	Sets the width of the border on all the four sides of an element.

5.5.5 LIST PROPERTIES:

Property	Description
list-style	Defines the display style for a list and list elements.
list-style-image	Specifies the image to be used as a list-item marker.
list-style-position	Specifies the position of the list-item marker.
list-style-type	Specifies the marker style for a list-item.

5.5.6 OVERFLOW PROPERTIES:

Property	Description
overflow	Specifies the treatment of content that overflows the element's box.
overflow-x	Specifies the treatment of content that overflows the element's box horizontally.
overflow-y	Specifies the treatment of content that overflows the element's box vertically.

5.5.7 FLEXBOX PROPERTIES:

Property	Description
flex	Specifies the components of a flexible length.
flex-basis	Specifies the initial main size of the flex item.
flex-direction	Specifies the direction of the flexible items.
flex-flow	A shorthand property for the flex-direction and the flex-wrap properties.
flex-grow	Specifies how the flex item will grow relative to the other items inside the flex container.
flex-shrink	Specifies how the flex item will shrink relative to the other items inside the flex container.
flex-wrap	Specifies whether the flexible items should wrap or not.
order	Specifies the order in which a flex items are displayed and laid out within a flex container.

5.5.8 GRID PROPERTIES:

Property	Description
column-gap	Specifies the gap between the columns
gap	A shorthand property for the row-gap and the column-gap properties
grid	A shorthand property for the grid-template-rows, grid-template-columns, grid-template-areas, grid-auto-rows, grid-auto-columns, and the grid-auto-flow properties
grid-area	Either specifies a name for the grid item, or this property is a shorthand property for the grid-row-start, grid-column-start, grid-row-end, and grid-column-end properties
grid-auto-columns	Specifies a default column size
grid-auto-flow	Specifies how auto-placed items are inserted in the grid
grid-auto-rows	Specifies a default row size
grid-column	A shorthand property for the grid-column-start and the grid-column-end properties
grid-column-end	Specifies where to end the grid item
grid-column-gap	Specifies the size of the gap between columns

grid-column-start	Specifies where to start the grid item
grid-gap	A shorthand property for the grid-row-gap and grid-column-gap properties
grid-row	A shorthand property for the grid-row-start and the grid-row-end properties
grid-row-end	Specifies where to end the grid item
grid-row-gap	Specifies the size of the gap between rows
grid-row-start	Specifies where to start the grid item
grid-template	A shorthand property for the grid-template-rows, grid-template-columns and grid-areas properties
grid-template-areas	Specifies how to display columns and rows, using named grid items
grid-template-columns	Specifies the size of the columns, and how many columns in a grid layout
grid-template-rows	Specifies the size of the rows in a grid layout
row-gap	Specifies the gap between the grid rows

5.5.9 ANIMATION PROPERTIES:

Property	Description
animation	Specifies the keyframe-based animations.
animation-delay	Specifies when the animation will start.
animation-direction	Specifies whether the animation should play in reverse on alternate cycles or not.
animation-duration	Specifies the number of seconds or milliseconds an animation should take to complete one cycle.
animation-fill-mode	Specifies how a CSS animation should apply styles to its target before and after it is executing.
animation-iteration-count	Specifies the number of times an animation cycle should be played before stopping.
animation-name	Specifies the name of @keyframes defined animations that should be applied to the selected element.
animation-play-state	Specifies whether the animation is running or paused.
animation-timing-function	Specifies how a CSS animation should progress over the duration of each cycle.

5.5.10 TRANSITION PROPERTIES:

Property	Description
transition	Defines the transition between two states of an element.

transition-delay	Specifies when the transition effect will start.
transition-duration	Specifies the number of seconds or milliseconds a transition effect should take to complete.
transition-property	Specifies the names of the CSS properties to which a transition effect should be applied.
transition-timing-function	Specifies the speed curve of the transition effect.

5.5.11 TRANSFORM PROPERTIES:

Property	Description
transform	Applies a 2D or 3D transformation to an element.
transform-origin	Defines the origin of transformation for an element.
transform-style	Specifies how nested elements are rendered in 3D space.

5.5.12 OTHER PROPERTIES:

Property	Description
cursor	Specify the type of cursor.
direction	Define the text direction/writing direction.
display	Specifies how an element is displayed onscreen.
position	Specifies how an element is positioned.
opacity	Specifies the transparency of an element.
float	Specifies whether or not a box should float.
height	Specify the height of an element.
width	Specify the width of an element.
left	Specify the location of the left edge of the positioned element.
right	Specify the location of the right edge of the positioned element.
top	Specify the location of the top edge of the positioned element.
bottom	Specify the location of the bottom edge of the positioned element.
z-index	Specifies a layering or stacking order for positioned elements.
max-height	Specify the maximum height of an element.
max-width	Specify the maximum width of an element.
min-height	Specify the minimum height of an element.
min-width	Specify the minimum width of an element.
word-break	Specifies how to break lines within words.
word-spacing	Sets the spacing between words.
word-wrap	Specifies whether to break words when the content overflows the boundaries of its container.

BASIC OF BOOTSTRAP

6.1 INTRODUCTION:

- Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.
- Bootstrap is completely free to download and use!
- It provides faster and easier web development.
- It contains HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins.
- It also gives you the ability to easily create responsive designs.
- Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter, and released as an open source product in August 2011 on GitHub.
- Bootstrap 5 is a latest version of Bootstrap.
- Until Bootstrap 4, it use jQuery with Javascript.
- But in Bootstrap 5, it removed jQuery and it totally use vanilla Javascript for DOM manipulation.

6.2 HOW TO USE BOOTSTRAP

- We have two different way to use Bootstrap,
 - Install Bootstrap
 - Using CDN
- We should include four file in HTML file for use Bootstrap.
 - bootstrap.css or bootstrap.min.css
 - bootstrap.js or bootstrap.min.js
 - popper.js or popper.min.js
 - jQuery (if use Bootstrap 5 then doesn't need to include)
- jQuery must include above the all JS files.
- We can also include the bundle.js or it's minified file.
- bundle.js is a combination of bootstrap.js and popper.js file, So these file don't need to include when we include bundle.js or it's minified file.

6.3 STRUCTURE OF BOOTSTRAP

- A web page using twitter bootstrap has a basic html structure which should start from type of document declaration, navigation and container in body, adding style sheets, java scripts declaration and Meta tags in header.
- For uniform rendering of its components and controls across all browsers, a HTML 5 doctype is needed by twitter bootstrap.
- This is due to the fact that HTML 5 elements are understood by the bootstrap which makes it necessary to include an appropriate doctype tag to the web page.

6.3.1 HTML HEADER STRUCTURE:

- In order for a responsive working of the bootstrap with mobile devices and latest browsers, the minimum requirements in the html header part should be to include a bootstrap style sheet and declare a viewport meta tag, which is crucial for a mobile device and bootstrap to be able to respond in accordance with the zoom level and the width of display.
- Adjusting viewport's width enables browsers to appropriately respond for the display size.
- During the earlier versions of bootstrap, a separate style sheet referred to as bootstrap-responsive existed.
- But latter, an inbuilt responsive in nature base style sheet known as bootstrap.css which is a single style sheet that handles web page responsiveness was developed.
- This bootstrap.css proves to be very useful for debugging during development.

6.3.2 HTML BODY STRUCTURE:

- Before you close the body tag in the html body, the bootstrap java script file and jQuery file must be included.
- The bootstrap java script file should be rendered after the jQuery file.

6.3.3 LAYOUT:

Fluid layout:

- No extra style sheet or additional step is required to obtain a responsive fluid layout in the bootstrap.
- It is designed to implement the responsive layout by default.
- This layout is highly recommended for public websites.

Fixed layout:

- Creating a website with a fixed layout is not generally recommended.

- This is mainly due to the increasing amount of internet website traffic that is arising from small devices such as smart phones and tablets.
- A fixed layout is however needed in some cases such as intranet applications.
- If this is the case, steps followed to disable the fluid responsiveness of the layout like,
 - Removing viewport meta tag
 - Override width property to a fixed size
 - If you are using NavBar, get rid of expanding and collapsing behavior
 - Instead of .col-md-* and .col-lg-*, use col-xs-* class in grid layouts

6.3.4 COMPONENTS:

Menu section:

- In bootstrap, to design a menu is the easiest thing to do in the web designing world.
- This is because it is designed to be responsive by default and even has the ability to get new appearances in smaller devices.

Header area:

- Jumbotron, which can display large headers and contents is a highly usable class offered by bootstrap.
- It is largely utilized by product based websites.

The content area:

- This content should be divided into three equal parts placed side by side.
- With the bootstrap's flex box-based grid, doing this is one easy and fast task.
- A12-column grid system is provided, so dividing the screen into equal parts and all one needs to do is to specify which HTML is occupied by each part.

Footer area:

- Footer area uses the same principle as the content area.

BASIC OF JAVASCRIPT

7.1 INTRODUCTION

- JavaScript (JS) is a lightweight, interpreted(implementations execute instructions directly without earlier compiling a program into machine language), or just-in-time(run time compilation) compiled programming language with first-class functions(function treat like variable).
- is a prototype-based(classes are not explicitly defined), multi-paradigm, single-threaded(run on main thread), dynamic language(interpreter assigns variables a type at runtime based on the variable's value).
- It is used for client-side scripting as well as server side using Node.js.

7.2 USE OF JAVASCRIPT

- JavaScript helps the users to build modern web applications to interact directly without reloading the page every time.
- JavaScript is commonly used to dynamically modify HTML and CSS to update a user interface by the DOM API.
- It is mainly used in web applications.
 - Web Development
 - Mobile Development
 - Game Development
 - Presentation

7.3 WAY TO USE JAVASCRIPT

- We should use JavaScript two different way,
 - Internal scripting
 - External scripting

7.3.1 INTERNAL SCRIPTING:

- We wrap script inside <script>.
- We use <script> inside <head> or <body>.

7.3.2 External SCRIPTING:

- We write script in separate JavaScript file, which have “.js” extension.
- We use <script> inside <head> or <body>.

7.4 SYNTAX OF JAVASCRIPT

Basic Syntax:

- JavaScript is case-sensitive.
- Statements should end in a semicolon (;), but it is not compulsory.

Variable Syntax:

- Must be defined before being used.
- The variable name can contain A – Z, a – z, underscore or digits and must start with a letter or an underscore (“_”).
- The data type does not have to be explicitly defined.

Comment Syntax:

- For single line comment we use “//”.
- For multiline comment we use “*/.../*”

7.5 EVENTS IN JAVASCRIPT

- Events are very important part of DOM.
- Using JavaScript we should manipulate element of DOM based on various events.
- We have four types of events in JavaScript,
 - Mouse events
 - Keyboard events
 - Frame or Object events
 - Form events

7.5.1 MOUSE EVENTS:

- These events base on mouse.
- Here we should manipulate DOM on mouse events.

Event	Attribute	Description
click	onclick	The event occurs when the user clicks on an element
dblclick	ondblclick	The event occurs when the user double-clicks on an element
mousedown	onmousedown	The event occurs when a user presses a mouse button over an element
mousemove	onmousemove	The event occurs when a user moves the mouse pointer over an element

mouseover	onmouseover	The event occurs when a user mouse over an element
mouseout	onmouseout	The event occurs when a user moves the mouse pointer out of an element
mouseup	onmouseup	The event occurs when a user releases a mouse button over an element

7.5.2 KEYBOARD EVENTS:

- These events based on keyboard.
- Here, we should manipulate DOM based on keyboard event.

Event	Attribute	Description
keydown	onkeydown	The event occurs when the user is pressing a key or holding down a key
keypress	onkeypress	The event occurs when the user is pressing a key or holding down a key
keyup	onkeyup	The event occurs when a keyboard key is released

7.5.3 FRAME OR OBJECT EVENTS:

- These events based on object.
- Here, we should manipulate DOM based on object event.

Event	Attribute	Description
abort	onabort	The event occurs when an image is stopped from loading before completely loaded (for <object>)
error	onerror	The event occurs when an image does not load properly (for <object>, <body> and <frameset>)
load	onload	The event occurs when a document, frameset, or <object> has been loaded
resize	onresize	The event occurs when a document view is resized
scroll	onscroll	The event occurs when a document view is scrolled

7.5.4 FORM EVENTS:

- These events based on form.

- Here, we should manipulate DOM based on form event.

Event	Attribute	Description
blur	onblur	The event occurs when a form element loses focus
change	onchange	The event occurs when the content of a form element, the selection, or the checked state have changed (for <input>, <select>, and <textarea>)
focus	onfocus	The event occurs when an element gets focus (for <label>, <input>, <select>, <textarea>, and <button>)
reset	onreset	The event occurs when a form is reset
select	onselect	The event occurs when a user selects some text (for <input> and <textarea>)
submit	onsubmit	The event occurs when a form is submitted

7.6 EVENTLISTENER

- Event listener is a method which used to attached event handler with particular element.
- We can attach many event handler on a element.
- Here, we are attach event handler with the DOM object not HTML element.
- It provide more reliability and flexibility, because we are not deal with HTML markup, we can directly control any element without HTML markup control.
- We have two method in event listener.

Syntax:

```
element.addEventListener(event, function, useCapture); // Add event listener  
element.removeEventListener(); // remove event listener
```

- “event” parameter is use to specify event.
- “function” parameter is used to specify function that execute when event triggered.
- “useCapture” is optional parameter which by default is false.
- “useCapture” is used for specify propagation mode.
- We have two type of propagation,
 - Bubbling
 - Capturing
- Bubbling is use when we want that child element handle event before parent element when event of child element triggered.

GUI Basic

- Let say, we have one <div> and inside that we have <p>, now we bind event listener on both element for same event, So when event of <p> triggered at that time <p> handle event first and after <div> handle event second.
- Bubbling is use when we want that parent element handle event before child element when event of child element triggered.
- Let say, we have one <div> and inside that we have <p>, now we bind event listener on both element for same event, So when event of <p> triggered at that time <div> handle event first and after <p> handle event second.

7.7 VALIDATION

- It is a technique that used to ensure that user fill data correctly.
- In traditional way validation, we perform server-side validation.
- If the data entered by a client was incorrect or was simply missing, the server would have to send all the data back to the client and request that the form be resubmitted with correct information.
- This was really a lengthy process which used to put a lot of burden on the server.
- JavaScript provides a way to validate form's data on the client's computer before sending it to the web server.
- We have two type of validation which we should do for archive better validation,
 - Basic Validation
 - Emptiness
 - Confirm Password
 - Length Validation...
 - Data Format Validation
 - Email Validation
 - Mobile Number Validation
 - Enrollment Number Validation...
- We are use regular expression for validate data.

7.7.1 REGEXP:

- A regular expression is an object that describes a pattern of characters.
- Regular expressions are used to perform pattern-matching and "search-and-replace" functions on text.

Syntax:

```
var pattern = "^[\\w]+$"; // will allow only words in the string
var regex = new RegExp(pattern);
```



```
if (regex.test(testString)){  
    //Valid  
} else {  
    //Invalid  
}
```

Regular Expression Syntax:

- To find word characters in the string we can use `\w`
- We can also use `[a-zA-Z0-9_]` for the same
- To find non-word characters in the string we can use `\W`
- to find digit characters in the string we can use `\d`
- We can also use `[0-9]` for the same
- To find non-digit characters in the string we can use `\D`
- We can use `\n` for new line and `\t` for tab