

HTML

(Hyper Text Markup Language)

What is Hyper Text?

- Hyper means “beyond”
- Hyper text resembles text that contains content beyond what we see.

What is a Markup Language?

- The markup is derived from common computer terminology, where “marking up” is the process of preparing for presentation.
- Markup language is a presentation language.

Evolution of Markup Languages

- GML [Generic Markup Language] at “CERN” Labs
- SGML [Standard Generic Markup Language]
- Early 1990’s “**Tim Berners Lee**” introduced “HTML” for “Mosaic” browser.
- IETF (Internet Engineering Task Force) **1993**
- **1995 HTML 2.0**
- **1997 HTML 3.2**
- **1997 HTML 4.0**

- **2014 HTML 5.0**
- **2017 HTML 5.2 [W3C and WHATWG]**
Web Hypertext Application Technology Work
Group
World Wide Web Consortium

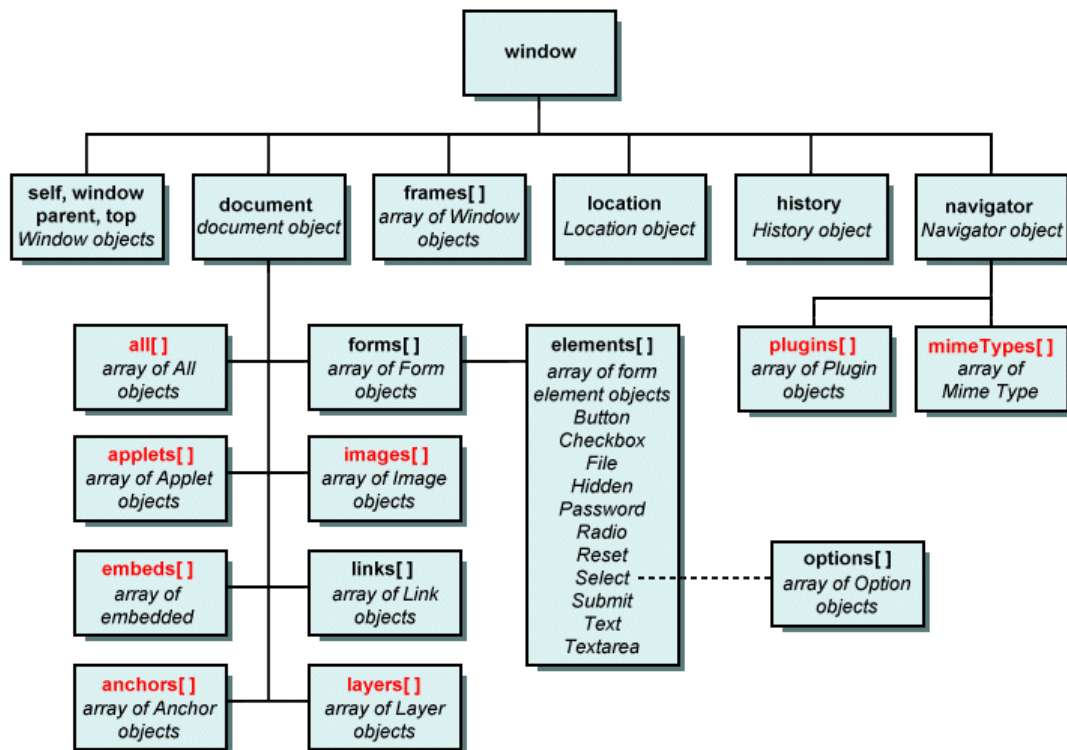
What is DHTML?

- Dynamic HTML. [Obsolete – No Longer in use]

What is HTML?

- It is a markup language.
- It is used for presentation.
- HTML is used to present DOM.

What is DOM?



- Browser presents content in a hierarchy called DOM.
- Document Object Model
- It is a hierarchy used to present contents in HTML.

What is Static DOM and Dynamic DOM?

- The DOM which is initially loaded by HTML is static.
- HTML can present static DOM.
- Static DOM can't handle interactions.
- JavaScript, jQuery, Angular JS, React etc. are used to convert the static DOM into dynamic DOM.

How HTML Presents the DOM?

- By using Elements

What are the elements used for presenting DOM?

- Normal Elements
- Void Elements
- RC Data Elements
- Raw Text Elements
- Foreign Elements

Element Type	Description
Normal Elements	<ul style="list-style-type: none">- Elements which return a presentation directly on call back [without any additional attributes].- Elements in HTML are built by using tags.- Normal Elements require a start tag and end tag.- Normal Elements will start returning presentation but can't stop implicitly.- They require explicit end tag.- Usually require start and end tags.

	<p>Ex:</p> <p> Bold </p>
Void Elements	<ul style="list-style-type: none"> - The term void refers to element that doesn't return any presentation directly on call back. - Void means no return type. - They can return only the specific content and stop implicitly. - Void elements doesn't require "End Tag". <p>Ex:</p> <p> tag Image – Element</p>
RC Data Elements	<ul style="list-style-type: none"> - Rich Content Elements - These elements will not allow any another element with in the context. <p>Ex:</p> <p><textarea> </textarea></p>
Raw Text Elements	<ul style="list-style-type: none"> - These elements are presented without a tag. <p>Ex:</p> <p>&copy; ©</p> <p>&#8377;</p>
Foreign Element	<ul style="list-style-type: none"> - These are HTML elements used in HTML but requires additional library.

	<ul style="list-style-type: none"> - Every browser can't understand these elements. - You have to import a library that makes the browser compatible with element. <p>Ex: SVG, MathML, Canvas</p>
--	--

What is Element and What is Tag?

- HTML presents using Elements.
- Elements are built by using Tags.

Image - **Element**
**** - **Tag**

Anchor - **Element**
<a> - **Tag**

Bold - **Element**
**** - **Tag**

What is difference between Attribute and Property?

- Attributes are used statically in Tags.
- Properties are use dynamically in Programming.



Join @Viralveda on Telegram. Tap on this.

```
var img = new Image();  
img.src = "shoe.jpg";      //src is property
```

- Every attribute of HTML tag doesn't have relative property.

Structure of HTML Page

- Every HTML page comprises of 2 sections at high level
 - Document Declaration
 - Document Scope

Document Declaration:

- It comprises of information about HTML version.
- It informs the parser that we are using HTML 5 to design web page.
- The document declaration in HTML 5 is defined by using the following Entity

<!DOCTYPE html>

Note: Comments in HTML are written with in "<!-- your comments -->"

Document Scope:

- It specifies the boundary of HTML document.
- It defines the start and end of every document in browser.
- Document scope in HTML is defined by using
<html>
</html>
- Every document scope must specify which language content it is presenting.
- Language is defined by using the attribute “**lang**”
<html lang="en-in">
</html>

Sections in HTML document scope

- Every HTML document scope comprises of 2 major sections
 - Head Section
 - Body Section

Head Section:

- Head section comprises of content, which is ***intended to load into memory*** when page is requested by client.

- The content from memory of browser can be accessed by page or browser when ever required.
- Typically head section is defined with **<head>**
</head>
- It comprises of contents like
 - Title
 - Link
 - Meta
 - Script
 - Style

Title Element:

- It comprises of title, which is displayed in the browser title bar.
- The title is also used in bookmarking the page.

<html lang="en-IN">

<head>

<title>Amazon Shopping</title>

</head>

</html>

Link Element:

- It is used to link external files to your web page.
- External files include short cut icons, stylesheets etc.

Ex: Link Shortcut Icon to Web Page [Favicon]

- Favicon must be an “icon” file with extension “.ico”
- Size of icon is between 16x16 pixels and 32x32 pixels
- Create a new folder by name “Icons” in your project.
- Add a new file into “Icons” folder by name “favicon.ico”
- Right Click on Icons folder and “Reveal in Explorer”
- Right Click on “icon file” and select “Open with Paint”
- Set the page size 32 x 32 pixels
- Design your icon
- Save
- Go to your web page and link the icon file

```
<html lang="en-IN">
```

```
<head>
```

```
<title>Amazon Shopping</title>
```

```
<link rel="shortcut icon"
```

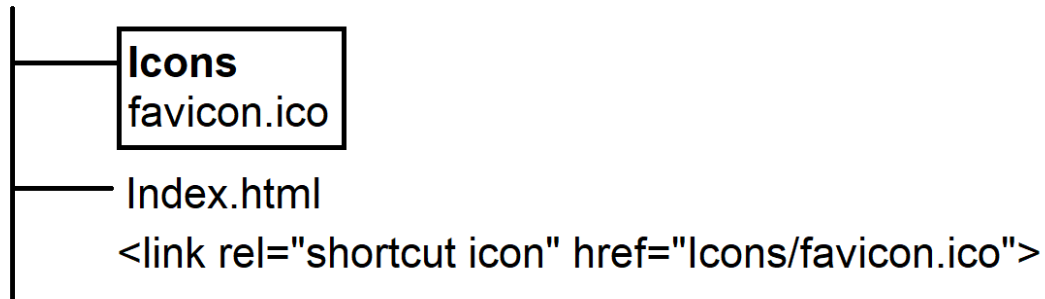
```
href="Icons/favicon.ico">
```

```
</head>
```

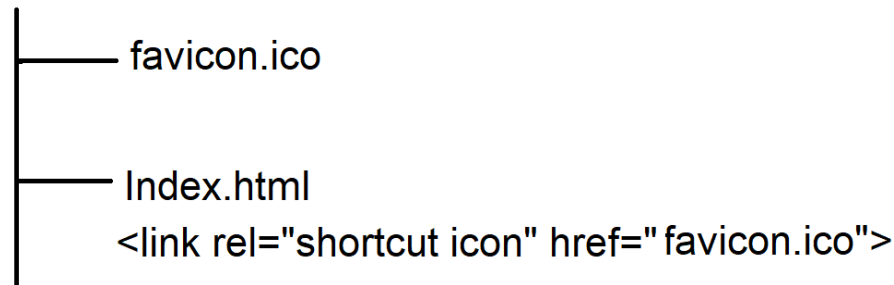
```
</html>
```

Note: Never use physical path directly in web development.

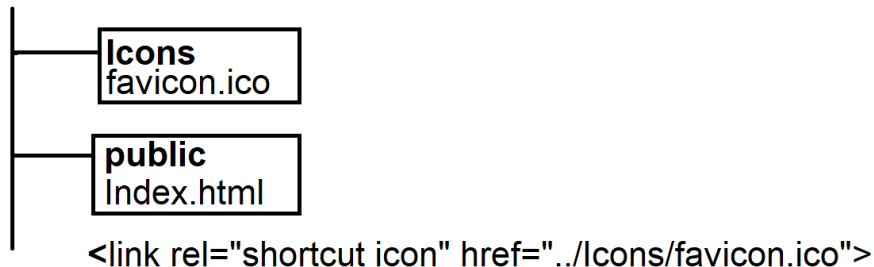
FullStack-8PM



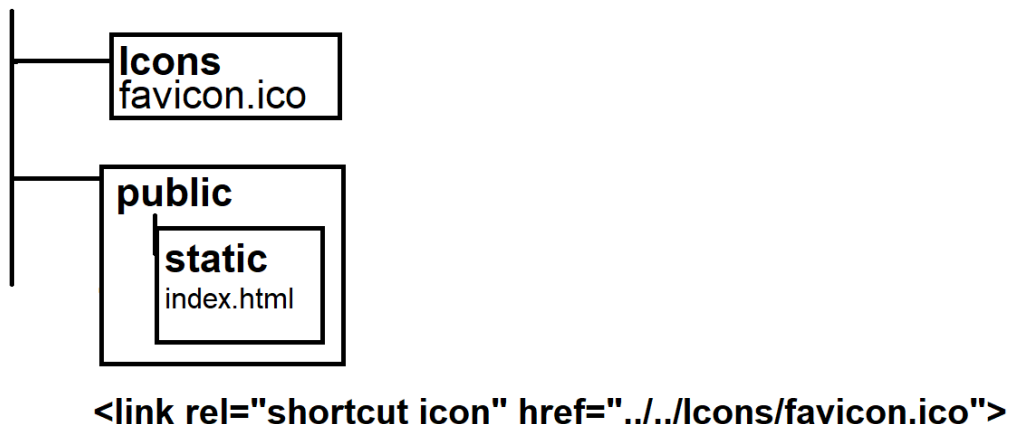
FullStack-8PM



FullStack-8PM



FullStack-8PM



FAQ: When to use “\” back slash “/” forward slash?

- Virtual Path: “/” forward slash
“Icons/favicon.ico”
- Physical Path: “\” back slash
“C:\Icons\Favicon.ico”

Meta in Head Section

- The terms meta refers to “Meta Data”.
- Meta Data contains information about your page, which is provided to SEO [Search Engine Optimization] to make the page more SEO friendly.
- **Meta is one of the options used in SEO. It is not only the option for SEO.**
- Meta is also used for Responsive Pages. **It is not only the options for Responsive.**

<!--Document declaration-->

<!DOCTYPE html>

<!--Document Scope-->

<html lang="en-IN">

```
<head>

  <title>Amazon Shopping</title>

  <link rel="shortcut icon"
href=" ../Icons/favicon.ico">

  <meta charset="utf-8">

  <meta name="keywords" content="Best Software
Traning, Best IT Training, in Hyd, Chennai, US">

  <meta name="description" content="something
about your website..">

  <meta name="author" content="Author Name for
Blog">

  <meta http-equiv="refresh" content="4">

</head>

</html>
```

Body Section

- It comprises of content to display in browser workspace.
- The body section is configured by using “<body>” element.

Syntax:

<body>

Some content

</body>

- Body tag comprises following attributes

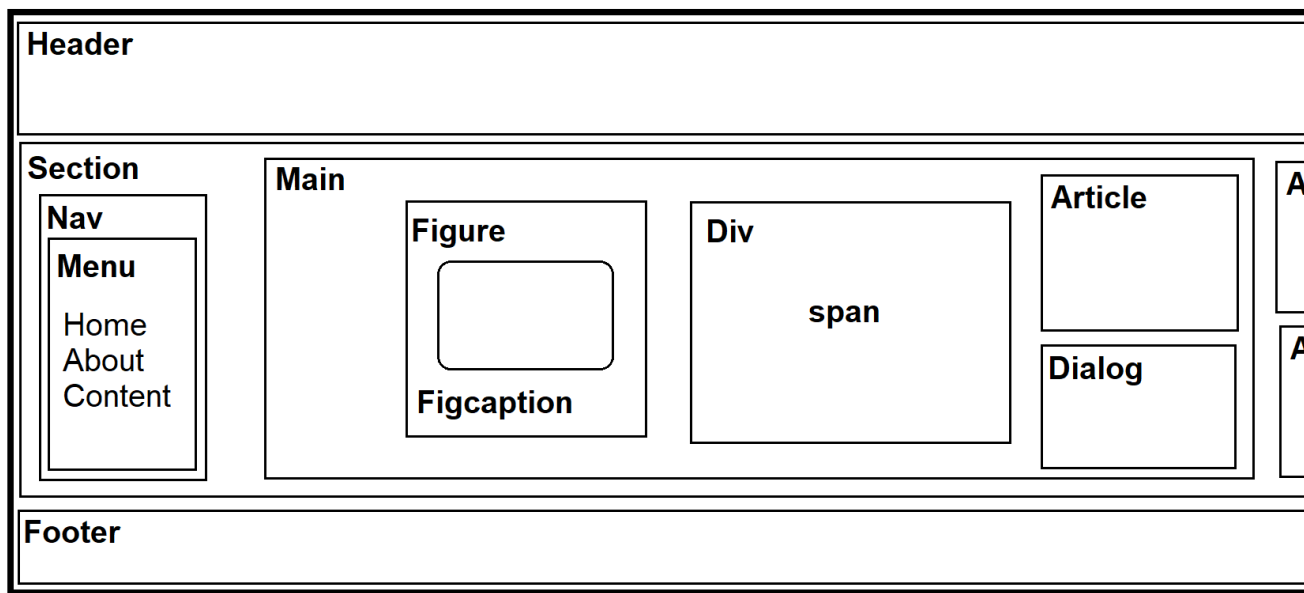
Attribute	Description
bgcolor	It sets a background color for page.
text	<p>It sets color for text in page. [Foreground color] Ex:</p> <pre><body bgcolor="red" text="yellow"> Welcome to Amazon Shopping </body></pre>
background	<p>It sets a background image for body section. Ex:</p> <pre><body background="../Images/banner3.jpg" text="white"> Welcome to Amazon Shopping </body></pre>
alink	It defines color for active link.
vlink	<p>It defined color for visited link. Ex:</p> <pre><body vlink="green" alink="red"> Home <a</pre>

	href="http://www.amazon.in">Amazon </body>
leftmargin	Sets the space between the content and browser window. Ex: <body leftmargin="50" rightmargin="50" topmargin="50" bottommargin="20"> Some paragraph.. </body>
rightmargin	
topmargin	
bottommargin	
align	It aligns the entire body content to left, center, right or justify Ex: <body align="justify"> </body>

Semantics of HTML Body

- Semantic elements are the elements used for a generic purpose.
- HTML 5 body section introduced new semantics that can make body content more SEO friendly.
- HTML 5 body section related semantic elements are container elements.
- A container comprises of content like text, headings, tables, pictures etc.

aside	It is a container used to define content, which is not relative to current website.
article	It is a container used to define content, which is relative to the current website and which summarizes the topics.
dialog	It is a container, which can popup and allow interactions with page.
figure	It is used to encapsulate any image with caption.
figcaption	It is used to define a caption for image.
header	It defines the content to display at the top margin of page.
footer	It defines the content to display at the bottom margin of page.
section	It defines the content between header and footer.
main	It defines the main content in section.
nav	It defines the navigation area.
menu	It defines the items used for navigation.
div	It is a container used for division of content in page.
span	It is a container used to span with existing content.



Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Layout Design</title>
```

```
<style>
```

```
  header
```

```
{
```

```
  background-color: maroon;
```

```
  color:white;
```

```
  text-align: center;
```

```
  font-size: 24px;
```

Join @Viralveda on Telegram. Tap on this.

```
border-radius: 30px;
}
footer
{
background-color: maroon;
color:white;
text-align: center;
font-style: italic;
border-radius: 30px;
}
section {
height: 500px;
}
main {
text-align: center;
}
menu div
{
width: 200px;
background-color:maroon;
```

```
    color:white;
    text-align: center;
    font-size: 23px;
    margin-top: 20px;
    padding: 10px;
    border-radius: 30px;
}

menu div:hover {
    background-color: black;
    color:white;
    cursor: grab;
}

main {
    font-size: 23px;
}

main span {
    background-color: yellow;
    color:red;
    font-size: 32px;
    font-weight: bold;
```

```
}  
aside {  
    float:right;  
    height: 40px;  
    width: 100px;  
    border:2px solid maroon;  
    font-size: 20px;  
}  
article {  
    text-align: center;  
    margin-top: 100px;  
    margin-left: 600px;  
    position: absolute;  
    border:2px dotted maroon;  
    padding: 20px;  
    border-radius: 20px;  
    background-color: lightgreen;  
}  
</style>  
</head>
```

```
<body>
```

```
  <header>
```

```
    Amazon Shopping
```

```
  </header>
```

```
  <section>
```

```
    <aside>
```

```
      ads here..
```

```
    </aside>
```

```
  <article>
```

```
    <div>DEC Sale 70%</div>
```

```
    <div>Latest Updates</div>
```

```
    <div>News..</div>
```

```
  </article>
```

```
  <nav>
```

```
    <menu>
```

```
      <div>Home</div>
```

```
      <div>Electronics</div>
```

```
      <div>Footwear</div>
```

```
      <div>Fashion</div>
```

```
    </menu>
```

```
</nav>

<main>

    Year End Sale. <span>Amazon offers 50%
    OFF</span>. Ends on DEC - 30.

    <figure>

        your image

        <figcaption>Fig:1</figcaption>

    </figure>

</main>

<dialog open>

    Chat with HR

</dialog>

</section>

<footer>

    <div>&copy; copyright 2020 Amazon</div>

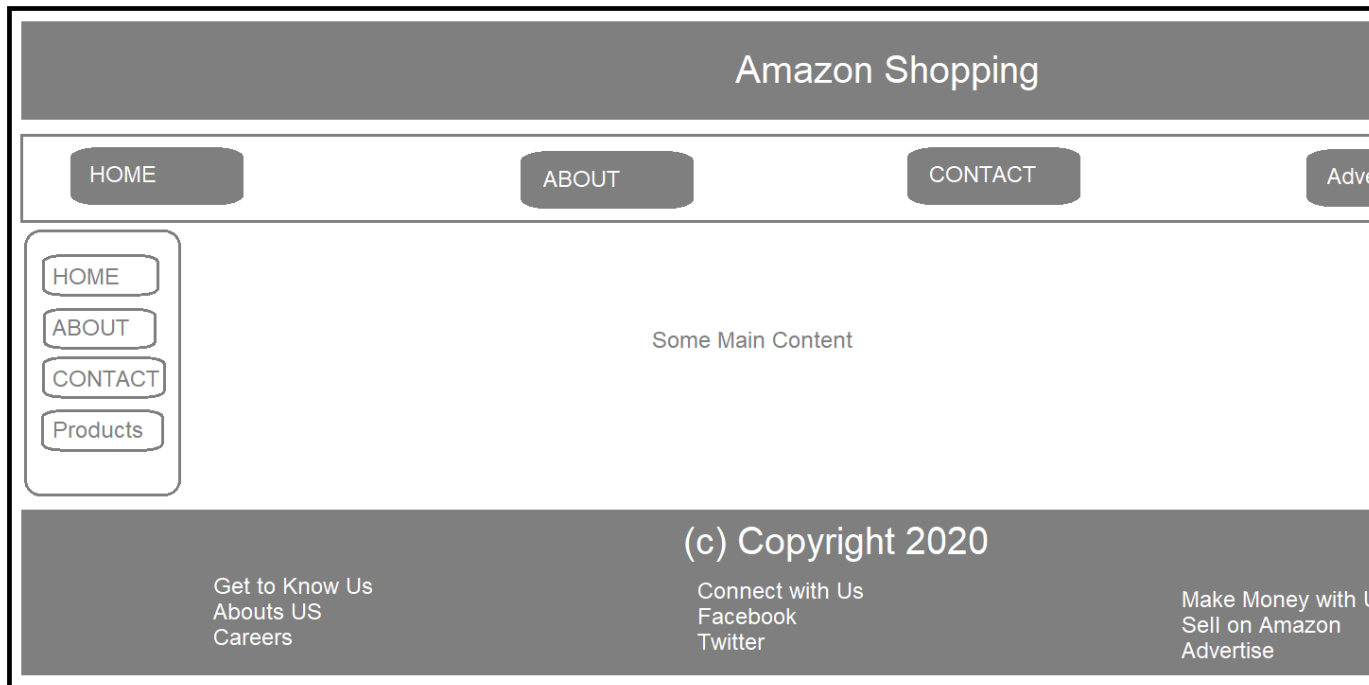
    <div>Electronics | Footwear | Fashion |
    Support</div>

</footer>

</body>

</html>
```

Task:



How to arrange content into columns?

- By using style attributes
 - column-count
 - column-rule
 - column-gap

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>News Online</title>
```

```
<style>
```

```
header {
```

```
font-size: 30px;
font-weight: bold;
text-align: center;
background-color: gray;
color:white;
padding: 10px;
}

section {
    margin-top: 20px;
    column-count: 5;
    column-gap: 30px;
    column-rule: dotted;
    text-align: justify;
}

footer {
    margin-top: 20px;
    column-count: 4;
    text-align: left;
    background-color:gray;
    color:white;
```



```
        padding: 20px;
    }
</style>
</head>
<body>
    <header>
        THE HINDU
    </header>
    <section>
        <div>
```

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this

software is subject to your volume license agreement rather than this agreement.

This agreement describes your rights and the conditions upon which you may use the Windows software. You should review the entire agreement, including any supplemental license terms that accompany the software and any linked terms, because all of the terms are important and together create this agreement that applies to you. You can review linked terms by pasting the (aka.ms/) link into a browser window.

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

This agreement describes your rights and the conditions upon which you may use the Windows software. You should review the entire agreement, including any supplemental license terms that accompany the software and any linked terms, because all of the terms are important and together create this agreement that applies to you. You can review linked terms by pasting the (aka.ms/) link into a browser window.

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

This agreement describes your rights and the conditions upon which you may use the Windows

software. You should review the entire agreement, including any supplemental license terms that accompany the software and any linked terms, because all of the terms are important and together create this agreement that applies to you. You can review linked terms by pasting the ([aka.ms/](#)) link into a browser window.

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

This agreement describes your rights and the conditions upon which you may use the Windows software. You should review the entire agreement, including any supplemental license terms that

accompany the software and any linked terms, because all of the terms are important and together create this agreement that applies to you. You can review linked terms by pasting the (aka.ms/) link into a browser window.

</div>

</section>

<footer>

<div>

<div>Movie News</div>

<div>Sports</div>

<div>Stock</div>

</div>

<div>

<div>Facebook</div>

<div>Twitter</div>

<div>YouTube</div>

</div>

<div>

<div>editor@hindu.com</div>

<div>www.epaper.com/hindu</div>

```
        <div>www.thehindu.com</div>
    </div>
    <div>
        <div>editor@hindu.com</div>
        <div>www.epaper.com/hindu</div>
        <div>www.thehindu.com</div>
    </div>
</footer>
</body>
</html>
```

Ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Menu</title>
        <style>
            menu div {
                width: 200px;
                height: 30px;
                padding: 10px;
```

```
        background-color: darkcyan;
        color:white;
        text-align: center;
        border-radius: 10px;
        display: inline;
        margin-top: 50px;
    }
</style>
</head>
<body>
    <header>
        <nav>
            <menu>
                <div>Home</div>
                <div>About</div>
                <div>Contact</div>
                <div>Login</div>
            </menu>
        </nav>
    </header>
```

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

Basic HTML elements and Entities for presenting content in body section

- Line Breaks

- Browser ignores the line breaks given in editor.
- You have to manually add line break by using “
” element.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Demo</title>
  </head>
  <body>
    Welcome to HTML.<br>
    This is our static Page.<br> <br>
    Contents in body section <br> without semantics.
```


`</body>`

`</html>`

**FAQ: What is difference between `
` and `
`?**

- There is not such element called “`
`” in HTML.
- Always use only `
`
- `
` is used to indicate that it is a self-ending or void element.

- Blank Spaces

- Browser ignores additional spaces between words and characters.
- Browser allows only one characters space between words or chars.
- To add manual space, you have to use “` `” [non-breakable space].
- It is “Raw Text Element”

Ex:

`<!DOCTYPE html>`

`<html>`

`<head>`

`<title>Demo</title>`

`</head>`

```
<body>
  H &nbsp;T &nbsp;M &nbsp;L
</body>
</html>
```

- **Pre-formatted Text**

- It is used to present the content exactly as defined in source code.
- It will keep the spaces and line breaks as defined in editor.
- The content must be defined within “<pre>” element.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Demo</title>
    <style>
      pre {
        background-color: lightgrey;
        border:2px solid black;
        padding:10px;
```

```
        width: 400px;
    }
</style>
</head>
<body>
    <div>Sample C Program</div>
    <div>
        <pre>
            #include <stdio.h>

            main
            {
                printf("Welcome to HTML");
            }
        </pre>
    </div>
</body>
</html>
```

- **Code Element**

- `<code>` is an element used to define code snippet in web page.

- It allows to browser and SEO to understand that the content enclosed in some computer code.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Demo</title>
```

```
    <style>
```

```
      pre {
```

```
        background-color: lightgrey;
```

```
        border:2px solid black;
```

```
        padding:10px;
```

```
        width: 400px;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <div>Sample C Program</div>
```

```
    <div>
```

```
      <pre>
```

```
<code>
#include <stdio.h>
main
{
    printf("Welcome to HTML");
}
</code>
</pre>
</div>
</body>
</html>
```

Variables in HTML Code Presentation:

- You can <var> for variable in HTML.
- Variables are designated with <var> for code snippet.

Ex:

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

```
</head>
<body>
  <code>
    <var>x</var> = 10; <br>
    <var>y</var> = 20; <br>
    <var>z</var> = <var>x</var> + <var>y</var>;
  </code>
</body>
</html>
```

Address in HTML body

- The <address> element is used to define your contact details in web page.
- SEO uses the address and can display in search results.

Ex:

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

```
<body>
  <address>
    Naresh-I-Technologies <br>
    hr@nareshit.in <br>
    phone: 040-59595842
  </address>
</body>
</html>
```

Headings in HTML

- Headings need a different appearance in page.
- HTML provides pre-defined heading levels.
- HTML headings levels are defined by using <hn>.
- “n” refers to level number from 1 to 6.

Syntax:

<h1> </h1>

<h2> </h2>

.....

<h6> </h6>

- Heading levels from 1 to 6 will decrease the size.
- Heading can be defined with “align” attribute, which can align left, center, right or justify.

FAQ: Why to use heading element for headings?

- Heading elements will make your topics SEO friendly.
- SEO can identify the topics on your page if they are defined in heading.

Note:

- Don't use headings for highlighting any word or sentence in a paragraph.
- Heading by default will have line break above and below.
- Don't use too many headings in a page. SEO can SPAM your page.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Demo</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1 align="center">HTML</h1>
```

```
    <h2>Web Terminology</h2>
```

```
    <h3>Web Site</h3>
```


It is a virtual directory on web server.

`<h3>Web Page</h3>`

It is a Hypertext document.

`</body>`

`</html>`

Data List with Terms and Definitions:

- Data List is defined with `<dl>`
- Data list is a collection of terms defined with `<dt>` and definition defined with `<dd>`

Syntax:

`<dl>`

`<dt>Term</dt>`

`<dd> Definition </dd>`

`<dt> Term </dt>`

`<dd> Definition </dd>`

`<dd> Definition </dd>`

`</dl>`

Ex:

`<!DOCTYPE html>`

`<html>`

`<head>`

```
<title>Demo</title>
<style>
  dt {
    font-weight: bold;
    background-color: lightgrey;
  }
</style>
</head>
<body>
  <h1 align="center">HTML</h1>
  <h2>Web Terminology</h2>
  <dl>
    <dt>Web Server</dt>
    <dd>It resembles software and hardware.</dd>
    <dd>It handle the request and response.</dd>
    <dt>Web Site</dt>
    <dd>It is a virtual directory.</dd>
    <dt>Web Page</dt>
    <dd>It is an hyper text document.</dd>
  </dl>
```

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

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Demo</title>
    <style>
      dt {
        font-weight: bold;
        background-color: lightgrey;
      }
    </style>
  </head>
  <body>
    <h2>Product Details</h2>
    <dl>
      <dt>Product Name</dt>
      <dd>Samsung TV</dd>
      <dt>Product Price</dt>
```

```
<dd>45600.54</dd>
<dt>Stock Status</dt>
<dd>Available</dd>
</dl>
</body>
</html>
```

Details and Summary

- It is used to display the content in detail only when required.
- It allows to expand and collapse your content.
- It saves the screen space.

Syntax:

```
<details>
<summary> Your Title </summary>
some text...
</details>
```

- You can specify “open” attribute to show the details.

Ex:

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

```
<head>
  <title>Demo</title>
  <style>
    dt {
      font-weight: bold;
      background-color: lightgrey;
    }
  </style>
</head>
<body>
  <h2 align="center">Web Development</h2>
  <details open>
    <summary>HTML</summary>
```

HTML is a markup language. Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device

manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</details>

<details>

<summary>CSS</summary>

CSS defines style for HTML elements. Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</details>

<details>

<summary>JavaScript</summary>

It is used to handling client side interactions. Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</details>

</body>

</html>

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Demo</title>
```

```
  </head>
```

```
  <body>
```

```
    <h2>Amazon Shopping</h2>
```

```
    <details>
```

```
      <summary>Electronics</summary>
```

```
      <dl>
```

```
        <dd>Televisions</dd>
```

```
        <dd>Cameras</dd>
```

```
        <dd>Watches</dd>
```

```
      </dl>
```

```
    </details>
```

```
    <details>
```

```
      <summary>Footwear</summary>
```

```
      <dl>
```

```
        <dd>Sports</dd>
```



```
        <dd>Sneakers</dd>
        <dd>Casuals</dd>
    </dl>
</details>
</body>
</html>
```

Field Set and Legends

- Field set is container with frame.
- Legend puts a caption for field set.

Syntax:

```
<fieldset>
    <legend> Title </legend>
    Your conten...
</fieldset>
```

Ex:

```
<!DOCTYPE html>

<html>

    <head>

        <title>Demo</title>

        <style>

            legend {
```

```
        background-color:black;
        color:white;
    }
    fieldset {
        margin-top: 20px;
    }
</style>
</head>
<body>
    <fieldset>
        <legend align="center">Annexure-1</legend>
        HTML is a markup language.
```

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you

acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</fieldset>

<fieldset>

<legend align="center">Annexure-2</legend>

JavaScript is used for client side interactions.

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</fieldset>

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

Paragraphs and Blockquotes

- HTML ignores the line breaks and paragraph marks.
- You have to manually add paragraphs by using <p> element.
- It supports “align” attribute, which can set text left, center, right or justified.
- Paragraphs will have a line break before and after.

Syntax:

```
<p>
```

Your Content

```
</p>
```

- **Blockquote** is similar to paragraph but have left and right indentation for text.
[Indentation is space between margin and text]
- It is defined by using “<blockquote>” element.
- In SEO blockquote is used a summary for content in page.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Paragraph</title>
<style>
    blockquote {
        padding-left: 150px;
        padding-right: 150px;
        font-style: italic;
        border-top: 2px double darkcyan;
        border-bottom: 2px double darkcyan;
        width: 400px;
    }
</style>
</head>
<body>
    <blockquote align="justify">
        Depending on how you obtained the Windows
        software, this is a license agreement between (i) you
        and the device manufacturer or software installer
        that distributes the software with your device; or (ii)
        you and.
    </blockquote>
    <p align="justify">
```

This agreement describes your rights and the conditions upon which you may use the Windows software. You should review the entire agreement, including any supplemental license terms that accompany the software and any linked terms, because all of the terms are important and together create this agreement that applies to you. You can review linked terms by pasting the (aka.ms/) link into a browser window.

</p>

<p align="justify">

Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</p>

</body>

</html>

Ordered and Unordered List

- **Order List** will add auto numbering for a list of options, which can update automatically when you add or delete items.
- **Orders List** is defined by using Element.
- **Items in list** are defined by using .

Syntax:

```
<ol>
```

```
<li> Item-1 </li>
```

```
<li> Item-2 </li>
```

```
</ol>
```

Ex:

```
<body>
```

```
<h2>Web Terminology</h2>
```

```
<ol>
```

```
<li>Internet</li>
```

```
<li>Web Server</li>
```

```
<li>Web Site</li>
```

```
<li>Web Page</li>
```

```
</ol>
```

</body>

- can be defined with following attributes

Attribute	Description
Type	<p>If specifies the numbering type, which can be:</p> <ul style="list-style-type: none">- a- A- i- I- 1 <p>Ex:</p> <pre><ol type="a"> Internet Web Server Web Site Web Page </pre>
start [number]	<p>It defines the numbering level to start with.</p> <p>Ex:</p> <pre><ol type="A" start="5"> Internet Web Server Web Site Web Page </pre>
reversed	<p>It arranges the numbering in</p>

	<p>reverse order. It is not support on old version browser.</p> <p>Ex:</p> <pre> <ol reversed start="5"> Internet Web Server Web Site Web Page </pre>
lang	<p>It specifies the numbering language according to specified region. It is supported only on modern browsers.</p> <p>Ex:</p> <pre> <ol lang="hi" > Internet Web Server Web Site Web Page </pre>

Numbering can be defined in multi-level:

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
  <title>Paragraph</title>
</head>
<body>
  <h2>Web Terminology</h2>
  <ol type="I">
    <li>
      HTML
      <ol type="1">
        <li>Normal Elements
          <ol type="a">
            <li>Bold</li>
            <li>Italic</li>
          </ol>
        </li>
        <li>Void Elements
          <ol type="a">
            <li>Line Break</li>
            <li>Image</li>
          </ol>
        </li>
      </ol>
    </li>
  </ol>
</body>
```

```
        </li>
    </ol>
</li>
<li>
    JavaScript
    <ol type="1">
        <li>Variables</li>
        <li>Data Types</li>
    </ol>
</li>
</ol>
</body>
</html>
```

- **Numbering will not continue the previous list once you end it.**

Ex:

```
<body>
    <ol>
        <h2>Web Terminology</h2>
        <li>Web Site</li>
```

```
<li>Web Page</li>
<li>URL</li>
<h2>Terms continued..</h2>
<li>Internet</li>
<li>Web</li>
</ol>
</body>
```

FAQ: How can we arrange list items side by side?

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Paragraph</title>
    <style>
      ol {
        display: flex;
        list-style: none;
      }
      li {
```

```
margin-left: 40px;
border:2px solid darkcyan;
background-color: darkcyan;
color:white;
padding:5px;
text-align: center;
border-radius: 10px;
width: 200px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ol>
```

```
<li>Home</li>
```

```
<li>About</li>
```

```
<li>Contact</li>
```

```
<li>Signin</li>
```

```
<li>Help</li>
```

```
</ol>
```

```
</body>
```

</html>

Unordered List

- It is used to define bulleted list.
- Bulleted list comprises symbol instead of numbering.
- Unordered list is defined by using ""
- List items are defined by using ""
- The attribute "type" defines the symbol type, which can be
 - disc
 - circle
 - square

Ex:

<body>

```
<ul type="square">  
  <li>Web Sever</li>  
  <li>Web Site</li>  
  <li>Web Page</li>  
</ul>
```

</body>

FAQ: How to define custom symbol as bullet symbol?

- By using style attribute "list-style-image"

- It uses “url()” that specifies the name and location of image to display.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>List</title>
```

```
    <style>
```

```
      ul {
```

```
        list-style-image: url("../icons/bullet.png");
```

```
      }
```

```
      li {
```

```
        font-size: 34px;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <ul>
```

```
      <li>Web Sever</li>
```

```
      <li>Web Site</li>
```

```
        <li>Web Page</li>
    </ul>
</body>
</html>
```

Unordered list can be nested:

Ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>List</title>
        <style>

        </style>
    </head>
    <body>
        <ul>
            <li>Web Sever
                <ul>
                    <li>IIS</li>
                    <li>Apache Tomcat</li>
```



```
        </ul>
    </li>
    <li>Web Site</li>
    <li>Web Page
        <ul type="square">
            <li>Static Page</li>
            <li>Dynamic Page</li>
        </ul>
    </li>
</ul>
</body>
</html>
```

List can be “Assorted” with combination of Ordered and Unordered:

Ex:

```
<body>
    <ol type="I">
        <li>HTML
    </ol>
    <li>Normal Elements
```

```
<ul type="circle">
  <li>Bold</li>
  <li>Italic</li>
</ul>
</li>
<li>Void Elements
  <ul type="circle">
    <li>Image</li>
    <li>Break</li>
  </ul>
</li>
</ol>
</li>
<li>JavaScript</li>
</ol>
</body>
```

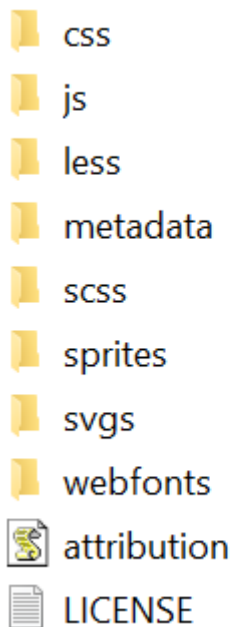
**Library for Icons and Symbols used in real-world
application development**

“Fontawesome”

- Visit the following URL

<https://fontawesome.com/how-to-use/on-the-web/setup/hosting-font-awesome-yourself>

- Click on
“Download Fontawesome Free for Web”
- It will download “fontawesome ZIP” folder
- Extract the ZIP folder and copy all its sub folders



- Go to your project and create a new folder by name “Fonts”
- Paste all the copied sub folders into “Fonts” folder.
- Now your project is ready to use “Fontawesome”

Ex:

<!DOCTYPE html>

<html>

```
<head>
  <title>Menu</title>
  <link rel="stylesheet" href="../fonts/css/all.css">
  <style>
    ul {
      list-style: none;
    }
    li {
      margin-top: 20px;
      border: 1px solid darkcyan;
      padding: 5px;
      width: 200px;
      background-color: lightcyan;
      color: blue;
    }
  </style>
</head>
<body>
  <ul>
```

```
<li> <span class="fa fa-home"></span>
<span>Home</span></li>

<li> <span class="fa fa-tv"></span>
Electronics</li>

<li> <span class="fa fa-tshirt"></span>
Fashion</li>

<li> <span class="fa fa-shoe-prints"></span>
Footwear</li>

<li> <span class="fa fa-bell"></span>
Notifications</li>

<li> <span class="fa fa-user"></span> Login</li>
</ul>
</body>
</html>
```

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Fontawesome</title>
    <link rel="stylesheet" href="../fonts/css/all.css">
    <style>
```

```
ul {
    list-style: none;
    width: 140px;
}
li{
    margin-top: 10px;
    border-bottom: 1px solid gray;
    padding: 10px;
}
li:hover {
    background-color: black;
    color:white;
    cursor:grab;
}
</style>
</head>
<body>
    <ul>
        <li><span class="fa fa-home"></span>
<span>Home</span> </li>
```

```
<li><span class="fa fa-user-circle"></span>
<span>Profile</span> </li>

<li><span class="fa fa-bell"></span>
<span>Notifications</span> </li>

</ul>

</body>

</html>
```

Text Formatting in HTML

- Text formatting includes changing the font, font style, size, color and effects etc.
- HTML provides the following elements for formatting the text.

Element	Description
	It is used to configure the following options for text <ul style="list-style-type: none">- Face: It defines font family like, Arial, Time New Roman, Sans-Serif etc. Ex: Welcome to HTML

	<ul style="list-style-type: none">- Size: It defines the font size, which can be from 1 to 7 levels. [Default size is 3]- Color: It defines color for text. <p>Note: is deprecated [Obsolete] from HTML 5. It is recommended to define font only with CSS.</p>
--	--

Ex:

```
<body>
```

```
    <font face="Arial" color="red" size="5">Welcome  
to HTML</font>
```

```
</body>
```

Ex:

```
<body>
```

```
    <font size="4" color="red">W</font>
```

```
    <font size="5" color="blue">E</font>
```

```
    <font size="6" color="green">L</font>
```

```
    <font size="7" color="yellow">C</font>
```

```
    <font size="6" color="green">O</font>
```

```
    <font size="5" color="blue">M</font>
```



```
<font size="4" color="red">E</font>
</body>
```

FAQ: How to define colors in HTML?

- HTML allows 2 techniques for colors
 - Color Name / Color Shade Name
 - Hexadecimal Code

Colors in HTML

- **Color Names:**
 - HTML allows you to define 16 Million colors.
 - Only 17 colors are known by their name.
 - Aqua
 - Blue
 - Gray
 - Lime
 - Navy
 - Orange
 - Red
 - Teal
 - Yellow
 - Black
 - Fuchsia
 - Green

- Maroon
- Olive
- Purple
- Silver
- White
- Transparent
- HTML also support hundreds of color shade names
 - Darkcyan
 - Lightcyan
 - Lightgreen etc..

Ex:

<body bgcolor="lightgreen"> </body>

- Hexadecimal Color Code

- Hexadecimals colors are basically “RGB” color
- R-Red, G-Green, B-Blue
- Hexadecimal number system is 16 base number system, where we use 16 different values ranging from “0 to F”.
- Hexadecimal color can be defined in 3 or 6 chars followed by hash “#”
 - 3 Chars Code
#RGB
 - 6 Chars Code

#RRGGBB

- Red, Green and Blue Value can be “0 to F”

Hexa: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
[0 is minimum, f is maximum]

- First place is for Red
- Second Place is for Green
- Third place is for Blue

Ex:

#FF0000	- Red	#F00
#00FF00	- Green	#0F0
#0000FF	- Blue	#00F
#FFFF00	- Yellow	#FF0
#000000	- Black	#000
#FFFFFF	- White	#FFF

Ex:

```
<body bgcolor="#ff00ff"> </body>
```

Font Styles and Font Effects

Font Style	Tag
Bold	

Italic	<i> [Emphasized]
Font Effects	Tag
Underline	<u> <ins>
Strikeout	<strike>
Super Script	<sup> Character position raised from base line
Sub Script	<sub> Character position lowered from base line

- HTML document is designed and reviewed by developer.
- Documents have 2 modes
 - Design Mode
 - Review Mode
- In review mode the design can be modified.
- Modifications are indicated using few review tags like <ins> etc.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Font</title>
```

```
    <link rel="stylesheet" href="../fonts/css/all.css">
```

```
<style>
```

```
  #rating {
```

```
    background-color: green;
```

```
    color:white;
```

```
    font-weight: bold;
```

```
    padding: 4px;
```

```
    border-radius: 5px;
```

```
  }
```

```
  li {
```

```
    margin-bottom: 5px;
```

```
  }
```

```
  ul {
```

```
    margin-left: -20px;
```

```
  }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
  <table width="1000">
```

```
    <tr valign="top">
```

<td>

<div>

</div>

</td>

<td>

<div>

<div>

Realme 7 (Mist Blue, 64 GB)

4.3

1,73,000 Ratings & 18,000 Reviews

6 GB RAM | 64 GB ROM |
Expandable Upto 256 GB

16.51 cm (6.5 inch) Full HD+
Display

64MP + 8MP + 2MP + 2MP |
16MP Front Camera

5000 mAh Lithium-ion
Battery

MediaTek Helio G95
Processor

</div>

</div>

</td>

<td>

<div>

₹ 14,999


```
<font color="gray" face="arial">
  <strike>₹ 17,000</strike>
</font>
<font color="green" face="arial">
  16% Off
</font>
<br>
  Upto <font face="arial"><b>₹
13,200 </b></font> off on Exchange.
</div>
</td>
</tr>
</table>
</body>
</html>
```

Images in HTML

- The HTML tag is used to embed images into page.
- Browser can't support all image types.

- Whenever you are configuring images, browser verifies its “MIME” type.

- **MIME [Multipurpose Internet Mail Extension]**

Ex:

Flower.jpg image/jpeg [MIME Type]

Flower.jpeg

Flower.jfif

Flower.pjpeg

- You have to use only supported image types.
- HTML standard Image types

Abbreviation	File Format	MIME Type	File Extension
APNG	Animated Portable Network Graphics	Image/apng	.apng
BMP	Bitmap File	Image/bmp	.bmp
GIF	Graphics Interchange Format	Image/gif	.gif
ICO	Microsoft Icon	Image/x-icon	.ico, .cur
JPEG	Joint Photographic Expert	Image/jpeg	.jpg, .jpeg, .jfif,

	Group		.jpeg, .jpg
PNG	Portable Network Graphics	Image/png	.png
SVG	Scalar Vector Graphics	Image/svg+xml	.svg
TIFF	Tagged Image File Format	Image/tiff	.tiff, .tif
WebP	Web Picture Format	Image/webp	.webp

FAQ: Tell me about XML format Image in HTML?

- SVG is XML format image.
- This image format can't be blocked by firewall.
- XML format images can be available offline.
- Cross platform [Every Device]

A developer must know the type of image suitable for situation.

Image Type	Purpose
PNG	<ul style="list-style-type: none"> - High resolution - High definition

	<ul style="list-style-type: none"> - Occupy more space in memory - Hight pixel depth. - Good for providing image which user can download and use for later.
JPG	<ul style="list-style-type: none"> - Compressed Image format - Occupies less space. - Less pixel depth than PNG - High resolution - High definition - Loose the quality on ZOOM. - Good for on screen presentation.
GIF	<ul style="list-style-type: none"> - Low resolution - Low definition - Less pixel depth than JPG - Only 256 colors - Can have animation. - Good for logos, bullets, buttons.
SVG	<ul style="list-style-type: none"> - SVG is XML format image. - This image format can't be blocked by firewall. - XML format images can be available offline. - Cross platform [Every Device]

	<ul style="list-style-type: none"> - Vector Graphics [2D] - Not pixel based. - Zoom image – Quality increase
WebP	<ul style="list-style-type: none"> - It embeds image into page. - When you try to save image, it saves entire page. - It will not allow to capture the image separately.

- **If you are using images for advertisements then you have to configure the size also as per standards**

Leaderboard	728 x 90	[pixels in width and height]
Half Page	300 x 600	
Medium Rectangle	300 x 250	
Large Rectangle	336 x 280	

- **Photoshop** is the tool used for publishing and editing images.
- **The tag ** is used to embed image into page.
- It is a void element.
- is configure with following attributes

Attribute	Description
-----------	-------------

alt	It defines the alternative text to display when image is unable to load.
src	It specifies the name path of image to display.
Width & height	It specifies the width and height in pixels or in percentage.
border	It specifies border for image in pixels.
title	It specifies the screen tip to display for image when mouse is over the image.
align	Align image left or right. Image will not support center. Align will wrap text around image.
Hspace	Horizontal space between text and image.
Vspace	Vertical space between text and image. Note: In latest versions of HTML “hspace and vspace” are obsolete.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Images in HTML</title>
```

</head>

<body>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you

and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</p>

</body>

</html>

- Image attributes which are used with integration of server-side technologies.

Attribute	Description
crossorigin	<p>It specifies the accessibility of resources from cross-origin. [CORS – Cross Origin Resource Sharing]</p> <ul style="list-style-type: none">- Allow Anonymous [Allow All to access]- Use Credentials [Need Authentication] <p>Ex:</p> <pre></pre>
decoding	<p>It specifies how to load the image along with other content in page.</p> <ul style="list-style-type: none">- Sync: Synchronized, it will block other content while loading image.- Async: Allows to load image along with other contents. <p>Ex:</p>

	<pre></pre>
Importance	<ul style="list-style-type: none"> - It defines the priority for image. - It can be low, high or auto. - Images with high priority are loaded first. <p>Ex:</p> <pre></pre>

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Images in HTML</title>
```

```
<style>
```

```
.card {
```

```
width: 250px;
```

```
padding: 10px;
```

```
border: 2px solid darkcyan;
```

```
border-radius: 10px;
```

```
margin: 30px;
```

```
}
```

```
.card-header {
```

```
        text-align: center;
        background-color:lightgoldenrodyellow;
        font-size: 26px;
        padding:3px;
    }
    .card-footer {
        text-align: center;
        background-color:lightgoldenrodyellow;
        padding: 3px;
    }
    .card-img {
        border:2px solid darkcyan;
        border-radius: 100px;
    }
    .card-body {
        text-align: center;
    }
    .card-deck {
        display: flex;
    }
</style>
</head>
<body>
    <div class="card-deck">
        <div class="card">
            <div class="card-body">
```

```
        
```

```
    </div>
```

```
    <div class="card-header">
```

```
        <h3>Nike Casuals</h3>
```

```
    </div>
```

```
    <div class="card-footer">
```

```
        <p>₹ 6700.66</p>
```

```
        <p>more..</p>
```

```
    </div>
```

```
</div>
```

```
<div class="card">
```

```
    <div class="card-header">
```

```
        <h3>Lee Boot</h3>
```

```
    </div>
```

```
    <div class="card-body">
```

```
        
```

```
    </div>
```

```
    <div class="card-footer">
```

```
        <p>₹ 7000.66</p>
```

```
        <p>more..</p>
```

```
</div>
</div>
```

```
<div class="card">
  <div class="card-header">
    <h3>Shirt</h3>
  </div>
```

```
  <div class="card-footer">
    <p>₹ 4000.66</p>
    <p>more..</p>
  </div>
```

```
  <div class="card-body">
    
  </div>
</div>
```

```
<div class="card">
  <div class="card-header">
    <h3>Jeans</h3>
  </div>
  <div class="card-body">
```

```

        
    </div>
    <div class="card-footer">
        <p>₹ 3000.66</p>
        <p>more..</p>
    </div>
</div>

```

```

</div>
</body>
</html>

```



Ex:

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <title>Card</title>
    <style>
      .card {
        width: 300px;
        border:2px solid black;
        border-top-left-radius:20px;
        border-top-right-radius: 20px;
        margin:30px;
      }
      .card-body {
        padding: 10px;
        text-align: justify;
        font-family: Arial;
      }
      .card-img {
        border-top-left-radius:20px;
        border-top-right-radius: 20px;
      }
```

```
.card-deck {  
    display: flex;  
}  
</style>  
</head>  
<body>  
    <div class="card-deck">  
        <div class="card">  
            <div>  
                  
            </div>  
            <div class="card-body">  
                <h3>Wild Life</h3>  
                <p>Depending on how you obtained the  
Windows software, this is a license agreement  
between (i) you and the device manufacturer or  
software installer that distributes the software with  
your device; or (ii) you and Microsoft Corporation (or,  
based on where you live or, if a business, where your  
principal place of business is located, one of its  
affiliates) if you acquired the software from a retailer.
```

Microsoft is the device manufacturer for devices
produced by Microsoft or one of its </p>

</div>

</div>

<div class="card">

<div>

</div>

<div class="card-body">

<h3>Wild Life</h3>

<p>Depending on how you obtained the
Windows software, this is a license agreement
between (i) you and the device manufacturer or
software installer that distributes the software with
your device; or (ii) you and Microsoft Corporation (or,
based on where you live or, if a business, where your
principal place of business is located, one of its
affiliates) if you acquired the software from a retailer.
Microsoft is the device manufacturer for devices
produced by Microsoft or one of its </p>

</div>

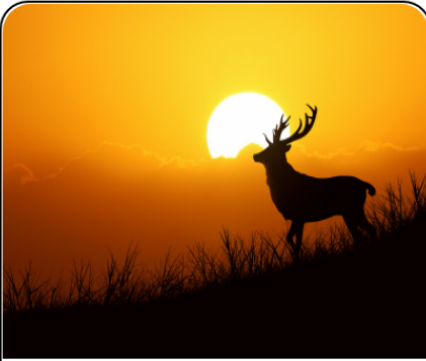
</div>

</div>

</body>

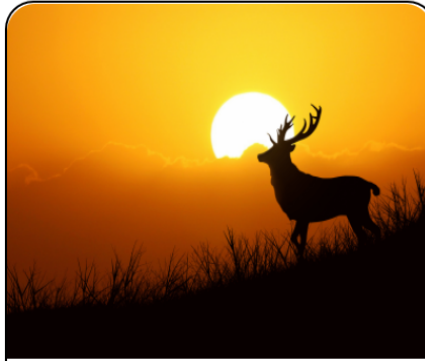
</html>

Ex:



Wild Life

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its



Wild Life

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its

<!DOCTYPE html>

<html>

<head>

<title>Card</title>

<style>

```
.card {  
    width: 300px;  
    border: 2px solid black;  
    border-top-left-radius: 20px;  
    border-top-right-radius: 20px;  
    margin: 30px;  
}  
  
.card-body {  
    padding: 10px;  
    text-align: justify;  
    font-family: Arial;  
}  
  
.card-img {  
    border-top-left-radius: 20px;  
    border-top-right-radius: 20px;  
}  
  
.card-deck {  
    display: flex;  
}
```

```
</style>
</head>
<body>
  <div class="card-deck">
    <div class="card">
      <div>
        
      </div>
      <div class="card-body">
        <h3>Wild Life</h3>
        <p>Depending on how you obtained the
Windows software, this is a license agreement
between (i) you and the device manufacturer or
software installer that distributes the software with
your device; or (ii) you and Microsoft Corporation (or,
based on where you live or, if a business, where your
principal place of business is located, one of its
affiliates) if you acquired the software from a retailer.
Microsoft is the device manufacturer for devices
produced by Microsoft or one of its </p>
      </div>
```

</div>

<div class="card">

<div>

</div>

<div class="card-body">

<h3>Wild Life</h3>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its </p>

</div>

</div>

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

Hyperlinks in HTML

- Link is clickable text, picture or graphic that navigates the user to any specified location when clicked.
- Hyperlink is clickable text, picture or graphics that navigates over “Http”.
- Hyperlinks are required to design navigation in website.

Note: According the Web Designing standards every website must have a proper navigation mechanism. User must able to reach to any topic just with one click.

- HTML Hyperlinks are created by using “Anchor” Element.
- Anchor is designed by using “<a>” tag.
- Hyperlinks in Website are classified into 2 types
 - Intra document links
 - Inter document links

Intra document link

- It is a hyperlink that handles navigation from one location to another within the page.
- User can navigate to any location within the page.
- In order to navigate to any location within page, you have to define unique reference ID for the target element.

Ex:

```
<h2 id="electronics"> </h2>
```

```
<div id="footwear"> </div>
```

```
<img id="pic">
```

- The anchor element uses "href" attribute that specifies the target location.
- "href" will set the specified path in "URL".
- You can refer any ID by using "#"

Syntax:

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

FAQ: How to change color for Visited, Active and Normal Links?

- You can use <body> tag attributes
 - alink
 - vlink
- You can define with styles
 - a:link { }
 - a:visited { }

- a:active { }

FAQ: How to remove underline for Hyperlink?

- You can use “style”
- text-decoration:none

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Intra Document Navigation</title>
```

```
    <link rel="stylesheet" href="../fonts/css/all.css">
```

```
    <style>
```

```
      .container {
```

```
        display: flex;
```

```
        column-count: 2;
```

```
        column-gap: 60px;
```

```
      }
```

```
      .content {
```

```
        height: 400px;
```

```
        overflow: scroll;
    }
    ul {
        list-style: none;
    }
    li {
        width: 100px;
        margin-top: 20px;
        border: 2px solid darkcyan;
        padding: 5px;
        border-radius: 5px;
        background-color: darkcyan;
        color: white;
    }
    a {
        color: white;
        text-decoration: none;
    }
    a: hover {
        text-decoration: underline;
```



```
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="navbar">
      <ul>
        <li> <a href="#electronics"><span class="fa fa-
tv"></span> Electronics</a> </li>
        <li> <a href="#footwear"><span class="fa fa-
shoe-prints"></span> Footwear</a></li>
        <li> <a href="#fashion"><span class="fa fa-
tshirt"></span> Fashion</a></li>
      </ul>
    </div>
    <div class="content">
      <h2 id="electronics">Electronics</h2>
      
      
```

<p>Depending on See:Wild Life how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and

Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<h2 id="footwear">Footwear</h2>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.

Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<h2 id="fashion">Fashion</h2>

```

```

```

```

```
<p>Depending on how you obtained the
Windows software, this is a license agreement
between (i) you and the device manufacturer or
software installer that distributes the software with
your device; or (ii) you and Microsoft Corporation (or,
based on where you live or, if a business, where your
principal place of business is located, one of its
affiliates) if you acquired the software from a retailer.
Microsoft is the device manufacturer for devices
produced by Microsoft or one of its affiliates, and
Microsoft is the retailer if you acquired the software
directly from Microsoft. Note that if you are a volume
license customer, use of this software is subject to your
volume license agreement rather than this
agreement.</p>
```

```
<div id="wild">
```

```

```

```
</div>
```

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software

directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

</div>

</div>

</body>

</html>

- **Intra document links can use target selector of “css” in order to highlight the target content if it is not scrollable.**

Ex:

<!DOCTYPE html>

<html>

<head>

<title>Intra Document Links</title>

<style>

ul {

list-style: none;

display: flex;

margin-left: -60px;


```
}  
li {  
    margin-left: 30px;  
    border: 2px solid darkcyan;  
    background-color: darkcyan;  
    width: 200px;  
    padding: 5px;  
    text-align: center;  
}  
a {  
    color: white;  
    text-decoration: none;  
}  
.topic {  
    border: 2px solid darkcyan;  
    padding: 10px;  
    background-color: darkcyan;  
    color: white;  
    margin-top: 20px;  
}
```

```
.topic:target {
    background-color: black;
    color:white;
}
</style>
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="#html">HTML</a></li>
                <li><a href="#css">CSS</a></li>
                <li><a href="#js">JavaScript</a></li>
            </ul>
        </nav>
    </header>
    <div id="html" class="topic">
        <h2>HTML</h2>
        <p>It is a markup language..</p>
    </div>
```

```
<div id="css" class="topic">
  <h2>CSS</h2>
  <p>It defines styles for HTML</p>
</div>
<div id="js" class="topic">
  <h2>JavaScript</h2>
  <p>It is a language.</p>
</div>
</body>
</html>
```

- **You can handle Scrolling of content in a page with a “Sticky Position”.**

Ex:

```
<!DOCTYPE html>
<html>
  <header>
    <title>Accordion</title>
    <style>
      dt {
```

```
        background-color: black;
        padding:10px;
        color:white;
        font-weight: bold;
        font-size: 24px;
        top:0px;
        position: sticky;
    }
</style>
</header>
<body>
    <h1 align="center">Tutorial Online</h1>
    <dl>
        <dt>HTML</dt>
        <dd>Void Elements</dd>
        <dd>Normal Elements</dd>
        <dd>RC Data Elements</dd>
        <dd>Raw Text Elements</dd>
        <dd>Void Elements</dd>
        <dd>Normal Elements</dd>
```

<dd>RC Data Elements</dd>
<dd>Raw Text Elements</dd>
<dd>Void Elements</dd>
<dd>Normal Elements</dd>
<dd>RC Data Elements</dd>
<dd>Raw Text Elements</dd>
<dd>Void Elements</dd>
<dd>Normal Elements</dd>
<dd>RC Data Elements</dd>
<dd>Raw Text Elements</dd>
<dd>Void Elements</dd>
<dd>Normal Elements</dd>
<dd>RC Data Elements</dd>
<dd>Raw Text Elements</dd>
<dd>Void Elements</dd>
<dd>Normal Elements</dd>
<dd>RC Data Elements</dd>
<dd>Raw Text Elements</dd>
<dd>Void Elements</dd>
<dd>Normal Elements</dd>

<dd>RC Data Elements</dd>

`<dd>Raw Text Elements</dd>`

	<dt>CSS</dt>
--	--------------

<dd>Selectors</dd>

<dd>Transitions</dd>

<dd>Responsive Design</dd>

<dd>Selectors</dd>

<dd>Transitions</dd>

<dd>Responsive Design</dd>

<dd>Selectors</dd>

<dd>Transitions</dd>

<dd>Responsive Design</dd>

<dd>Selectors</dd>

<dd>Transitions</dd>

<dd>Responsive Design</dd>

<dd>Selectors</dd>

<dd>Transitions</dd>

<dd>Responsive Design</dd>

<dd>Selectors</dd>

<dd>Transitions</dd>

<dd>Responsive Design</dd>

<dd>Selectors</dd>

<dd>Transitions</dd>

<dd>Responsive Design</dd>

<dd>Selectors</dd>

<dd>Transitions</dd>

<dd>Responsive Design</dd>

<dd>Selectors</dd>

<dd>Transitions</dd>

<dd>Responsive Design</dd>

<dt>JavaScript</dt>

<dd>Variables</dd>

<dd>Data TypeScript</dd>

<dd>Operators</dd>

<dd>Statements</dd>

<dd>Variables</dd>

<dd>Data TypeScript</dd>

<dd>Operators</dd>

<dd>Statements</dd>

<dd>Variables</dd>

```
<dd>Data TypeScript</dd>
<dd>Operators</dd>
<dd>Statements</dd>
<dd>Variables</dd>
<dd>Data TypeScript</dd>
<dd>Operators</dd>
<dd>Statements</dd>
<dd>Variables</dd>
<dd>Data TypeScript</dd>
<dd>Operators</dd>
<dd>Statements</dd>
</dl>
</body>
</html>
```

Inter Document Links

- Hyperlinks that handle navigation to any URL or any another page in website.
- It also refers Hyperlink which can invoke email and call clients.
- Navigation can be to
 - File [word document, presentation, pdf]
 - URL

- Email Client
- Mobile Dial application
- HTML Page

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Inter Document Links</title>
```

```
  </head>
```

```
  <body>
```

```
    <ol>
```

```
      <li><a href="newsonline.html">News  
Online</a></li>
```

```
      <li><a href=" ../Images/shoe.jpg">Nike  
Casuals</a></li>
```

```
      <li><a href=" ../docs/cssdemo.pdf">CSS Tutorial  
PDF</a></li>
```

```
      <li><a href="http://www.amazon.in">Amazon  
Shopping</a></li>
```

```
      <li>Contact : <a  
href="mailto:hr@nareshit.in">hr@nareshit.in</a></li>
```

```
<li>Call : <a  
href="tel:+919876543210">+919876543210</a></li>  
  
</ol>  
  
</body>  
  
</html>
```

Note:

- Every file that you configured as target in Hyperlink can't open in browser. Files require special plugins to open the content and show in browser.
- If plugin is missing or not available then the linked file will be downloaded.

FAQ: How to download any file through a link without opening it in browser?

- We can use "**download**" attribute for Hyperlink.

Ex: CSS Tutorial PDF

```
<a href="../docs/cssdemo.pdf" download="css  
tutorial" >CSS Tutorial PDF</a>
```

FAQ: How to open link target in a new Tab?

- Usually, the link target opens in the same browsing tab.
- You can define link with “target” attribute with value “_blank” to open in a new tab.

Syntax:

```
<a href="../Images/shoe.jpg" target="_blank" >Nike  
Casuals</a>
```

FAQ: How to open link target in a new Window?

- You have to use JavaScript “window.open()” method for hyperlink.

Syntax:

```
<a  
href="javascript:window.open('../Images/shoe.jpg','Nik  
e','width=500 height=400')"> Nike Casuals</a>
```

FAQ: How to open link target in the same window along with existing content?

- You can embed the link target into same window along with existing content by using “iFrame”
- Iframe allows to embed any external linked document into the page at specified location.

Ex:

```
<iframe src="https://www.youtube.com/embed/si-KFFOW2gw" width="600" height="500"></iframe>
```

Ex:

- **Add “Pages” folder in your project**
- **Add following files into folder**
 - Home.html
 - Electronics.html
 - Footwear.html
 - Fashion.html
- **Add another file in “public” folder**
 - Index.html

“Code you can find in Pages attached to notes”

Frames in HTML

[Obsolete – No Longer in use]

- HTML provides <frameset> that can split the browser window into multiple windows.
- <frameset> is a collection of “frames”.
- Every frame is designed by using <frame>.

Note: It is not embedding the content into page. It is splitting the window into horizontal or vertical panes.

- The document that is designed with frame can't contain body section.

Syntax:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Shopping | Online</title>
```

```
  </head>
```

```
  <frameset rows="15%, 70%, 15%">
```

```
    <frame></frame>
```

```
    <frame></frame>
```

```
    <frame></frame>
```

```
  </frameset>
```

```
</html>
```

- You can't add any content directly inside frame.
- Frame can handle only a document [Picture, PDF, text, HTML] by using "src" attribute.

Syntax:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
<title>Shopping | Online</title>
</head>
<frameset rows="15%, 70%, 15%">
  <frame></frame>
  <frame src="../Pages/home.html"></frame>
  <frame></frame>
</frameset>
</html>
```

Ex:

1. Add following files into "Pages"

- a. Menu.html
- b. Footwer.html
- c. Home.html
- d. Electronics.html
- e. Footwear.html
- f. Fashion.html

2. Menu.html

```
<head>
  <link rel="stylesheet" href="../fonts/css/all.css">
  <style>
    ul {
      list-style: none;
```

```

    }
    li {
        width: 100px;
        padding:10px;
        margin-top: 30px;
        background-color: darkcyan;
        color:white;
    }
    a {
        text-decoration: none;
        color:white;
    }
</style>
</head>
<body>
    <ul>
        <li><a href="../Pages/home.html"
target="frameBody"> <span class="fa fa-
home"></span> Home</a></li>
        <li><a href="../Pages/electronics.html"
target="frameBody"> <span class="fa fa-
tv"></span> Electronics</a></li>
        <li><a href="../Pages/footwear.html"
target="frameBody"> <span class="fa fa-shoe-
prints"></span> Footwear</a></li>

```

```
    <li><a href="../Pages/fashion.html"
target="frameBody"> <span class="fa fa-
tshirt"></span> Fashion</a></li>
</ul>
</body>
```

3. Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Shopping | Online</title>
  </head>
  <frameset cols="15%, 70%, 15%" noresize>
    <frame src="../Pages/menu.html"></frame>
    <frame name="frameBody"
src="../Pages/home.html"></frame>
    <frame src="../Pages/footer.html"></frame>
  </frameset>
</html>
```

Ex: Image Link

```
<!DOCTYPE html>
<html>
  <head>
    <title>Image Links</title>
```



```
<style>
  .container {
    display: flex;
    column-count: 2;
    column-gap: 40px;
  }
  iframe {
    border:none;
  }
  img {
    border:2px solid;
  }
  .rightPanel {
    width: 1000px;
  }
</style>
</head>
<body>
  <div class="container">
    <div class="leftPanel">
```

```
<h3>Preview</h3>
```

```
<div>
```

```
    <a href="../Images/earpod1.png"
target="frameBody">
```

```
        
```

```
    </a>
```

```
</div>
```

```
<div>
```

```
    <a href="../Images/shoanimated.gif"
target="frameBody">
```

```
        
```

```
    </a>
```

```
</div>
```

```
</div>
```

```
<div class="rightPanel">
```

```
    <iframe name="frameBody" width="100%"
height="500">
```

```
</iframe>
```

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

Tables in HTML

- Tables are used to organize the data in rows and columns.
- HTML uses tables to present content in rows and columns.
- Early version of HTML [4] used tables for designing entire document.
- HTML 5 introduced new elements and attributes for designing tables.
- HTML 5 table elements are

Element	Description
<table>	It is a collection of rows and columns.
<caption>	It sets caption for table.
<thead>	It defines the table header section.
<tbody>	It defines the table body section.
<tfoot>	It defines the table footer section.
<th>	It defines column heading content.
<tr>	It defines the table row.
<td>	It defines the table cell content.

<code><colgroup></code>	It groups a set of columns to define effects.
-------------------------------	---

	caption				
<table>	<div><div><thead></div><div><div><tr></div><div><th></div><div></th></div><div><th></div><div></th></div><div><th></div><div></th></div><div><</div></div></div>				
	<div><div><tbody></div><div><div><tr></div><div><td></div><div></td></div><div><td></div><div></td></div><div><td></div><div></td></div><div></tr></div></div></div>				
<colgroup>	<div><div><tr></div><div><td></div><div></td></div><div><td></div><div></td></div><div><td></div><div></td></div><div></tr></div></div>				
	<div><div><tr></div><div><td></div><div></td></div><div><td></div><div></td></div><div><td></div><div></td></div><div></tr></div></div>				
	<div><div><tfoot></div><div><tr></div><div><td></div><div></td></div><div><td></div><div></td></div><div><td></div><div></td></div><div></tr></div></div>				
</table>					

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Tables</title>
```

```
  </head>
```

```
  <body>
```

```
    <table width="400" border="1">
```

```
      <caption>Products Table</caption>
```

```
      <colgroup span="2" style="background-  
color: yellow;" ></colgroup>
```

```
      <thead>
```

```
<tr>
  <th>Product Id</th>
  <th>Name</th>
  <th>Price</th>
  <th>Preview</th>
</tr>
</thead>
<tbody>
  <tr>
    <td>1</td>
    <td>JBL Speaker</td>
    <td> ₹ 5600.55</td>
    <td>
      
    </td>
  </tr>
  <tr>
    <td>2</td>
    <td>Nike Casuals</td>
    <td> ₹ 3000.44</td>
    <td>
      
    </td>
  </tr>
```

```

        </tbody>
        <tfoot>
            <tr>
                <td></td>
                <td>&copy; copyright 2021</td>
                <td></td>
                <td></td>
            </tr>
        </tfoot>
    </table>
</body>
</html>

```

Formatting Tables in HTML

- Table element provides several attributes that are used to configure and format the table appearance.

Border, Frame and Rules:

Frame	<p>Frame is defined for entire table.</p> <p>Frame uses the values:</p> <ul style="list-style-type: none"> - Box - Void - Above - Below - Rhs
-------	--

	<ul style="list-style-type: none"> - Lhs <p>Syntax:</p> <pre><table frame="box"></pre>
Rules	<p>It is defined for rows and columns.</p> <p>Rules can use following values:</p> <ul style="list-style-type: none"> - All - None - Groups - Rows - Cols <p>Syntax:</p> <pre><table frame="box" rules="groups"></pre>
Border	<p>It is defined for table cell.</p> <p>Border can use only 0 or 1.</p> <p>0 – without border</p> <p>1- with border</p> <p>Note: Border can't be applied if table is defined with rules.</p> <p>Syntax:</p> <pre><table border="1" frame="void"> <table border="1" rules="all"></pre> <p>border will not be applied to table.</p>

Cell Spacing and Padding

cellspacing	It sets space between cells.
cellpadding	It sets space between border and cell content.

Syntax:

```
<table width="600" cellpadding="10" cellspacing="10"
border="1" frame="void">
```

Alignments

align	<p>It aligns the content horizontally left, center, right or justify.</p> <p>Syntax:</p> <pre><table align="center"> <tbody align="center"> <tr align="center"> <td align="center"></pre>
valign	<p>It aligns the content vertically top, center, bottom.</p> <p>Syntax:</p> <pre><tbody valign="top"> <tr valign="bottom"> <td valign="center"></pre>

Syntax:

```
<tbody align="center" valign="center">
```


Height and width

Height	It defines height for row. Syntax: <tr height="200"> <th height="200"> <td height="200">
Width	It defines cell width. <td width="200"> <th width="200">

Syntax:

<table width="300" height="100">

Merging or Rows and Columns

colspan	It can merge specified number of columns into single column.
rowspan	It can merge specifies number of rows into single row.

Ex:

<!DOCTYPE html>

<html>

```
<head>
  <title>Tables</title>
</head>
<body>
  <table border="1" width="400">
    <thead>
      <tr>
        <th rowspan="3">Head Section</th>
      </tr>
      <tr>
        <th colspan="2">Name</th>
        <th colspan="3">Address</th>
      </tr>
      <tr>
        <th>First Name</th>
        <th>Last Name</th>
        <th>City</th>
        <th>State</th>
        <th>Postal Code</th>
      </tr>
```

```
</thead>
<tbody>
  <tr>
    <th rowspan="5">Body Section</th>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td colspan="5" align="center">Table
Split</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
```

```

        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</tbody>
<tfoot>
    <tr>
        <th rowspan="2">Footer Section</th>
    </tr>
    <tr height="50">
        <td align="center" colspan="5"><i>&copy;
copyright 2021</i></td>
    </tr>
</tfoot>

```

```

    </table>
</body>
</html>

```

Head Section	Name		Address		
	First Name	Last Name	City	State	Postal Code
Body Section					
	Table Split				
Footer Section	© <i>copyright 2021</i>				

Background color and Image

bgcolor	It sets a background color for table, body, header, footer, cell, row.
Background d	It sets background image.

Syntax:

```
<table background="../Images/shoeanimated.gif"
border="1" width="500">
```

```
<table bgcolor="red">
```

```
<tr bgcolor="red">
```

Summary

Tags

- table
- thead
- tbody
- tfoot
- th
- tr
- td
- colgroup
- caption

Attributes

- frame
- border
- rules
- width
- height
- align
- valign
- bgcolor
- background
- cellspacing
- cellpadding
- colspan
- rowspan

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Products Catalog</title>
  </head>
  <body>
    <table align="center" width="100%"
cellspacing="20" cellpadding="10" border="0">
      <tr bgcolor="lightgreen">
        <td colspan="2" align="center">
          <h1>Amazon Shopping</h1>
        </td>
      </tr>
      <tr>
        <td width="200">
          
        </td>
        <td>
          <table width="100%" height="100"
border="1">
            <colgroup span="1" style="background-
color: yellow;"></colgroup>
```

```

        <tr>
            <td>Name</td>
            <td>JBL Speaker</td>
        </tr>
        <tr>
            <td>Price</td>
            <td>4600.55</td>
        </tr>
    </table>
</td>
</tr>
<tr>
    <td width="200">
        
    </td>
    <td>
        <table width="100%" height="100"
border="1">
            <colgroup span="1" style="background-
color: yellow;"></colgroup>
            <tr>
                <td>Name</td>
                <td>Nike Casuals</td>
            </tr>
            <tr>

```



```

        <td>Price</td>
        <td>4600.55</td>
    </tr>
</table>
</td>
</tr>
<tr>
    <td width="200">
        
    </td>
    <td>
        <table width="100%" height="100"
border="1">
            <colgroup span="1" style="background-
color: yellow;"></colgroup>
            <tr>
                <td>Name</td>
                <td>Shirt</td>
            </tr>
            <tr>
                <td>Price</td>
                <td>4600.55</td>
            </tr>
        </table>
    </td>

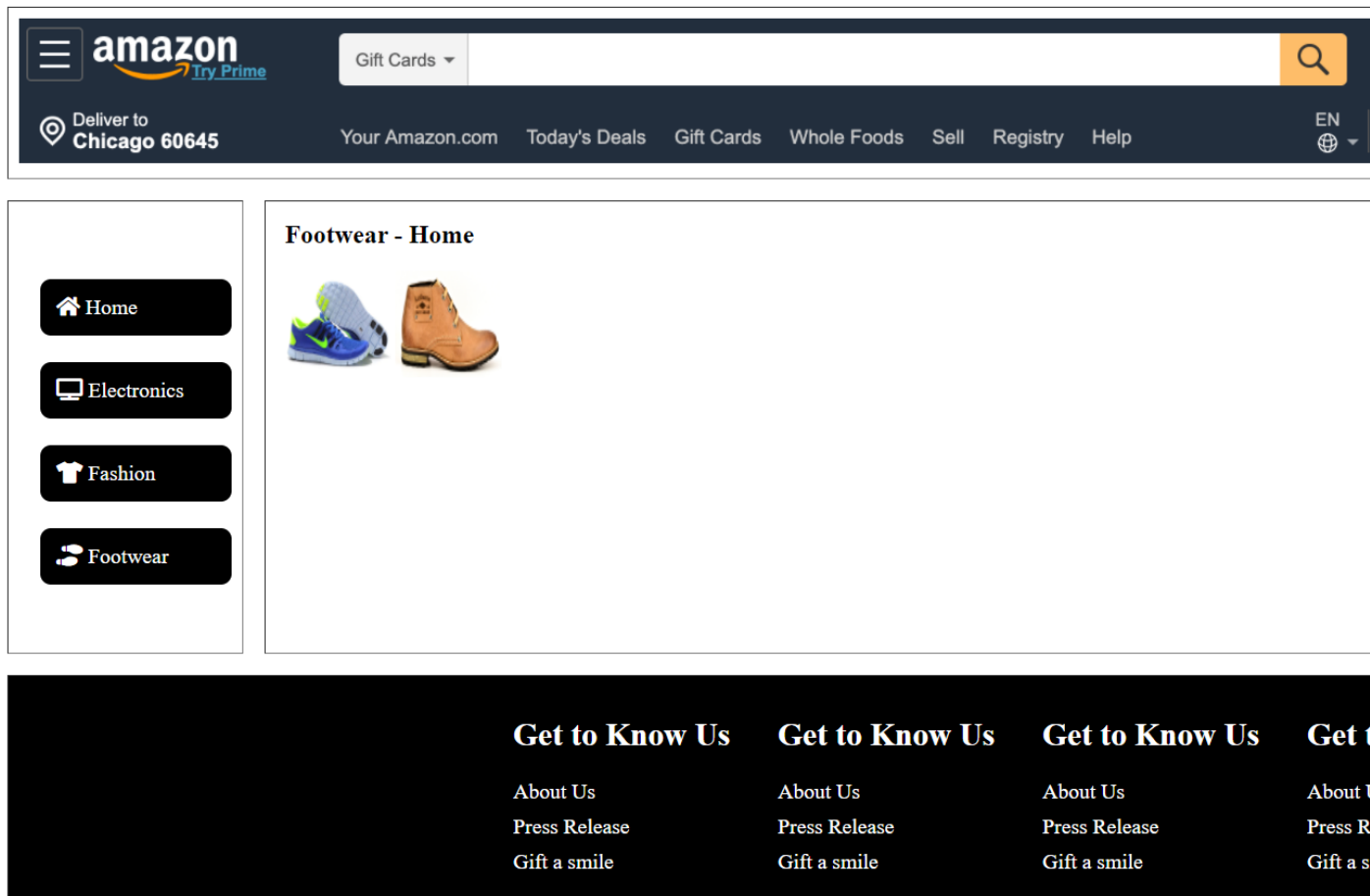
```

```

</tr>
<tr bgcolor="lightgreen">
  <td colspan="2" align="center">
    <i>&copy; copyright 2021</i>
  </td>
</tr>
</table>
</body>
</html>

```

Ex:



<!DOCTYPE html>

```
<html>
  <head>
    <title>Table Layout</title>
    <link rel="stylesheet" href="../../fonts/css/all.css">
    <style>
      ul {
        list-style: none;
      }
      ul > li {
        font-size: 20px;
        color:white;
        margin-top: 10px;
      }
      ol {
        list-style: none;
        margin-left: -20px;
      }
      ol > li {
        width: 150px;
        border:2px solid black;
```

```
padding:13px;
margin-top: 25px;
font-size: 20px;
background-color: black;
color:white;
border-radius: 10px;
}
iframe {
border:none;
}
a {
color:white;
text-decoration: none;
}
</style>
</head>
<body>
<table frame="void" border="1" width="100%"
cellspacing="20" cellpadding="10">
<tr height="100">
```

```
<td colspan="3">
    
</td>
</tr>
<tr height="400">
    <td width="200">
        <ol>
            <li>
                <a href="../Pages/home.html"
target="frameBody"> <span class="fa fa-
home"></span> Home</a>
            </li>
            <li>
                <a href="../Pages/electronics.html"
target="frameBody"> <span class="fa fa-tv"></span>
Electronics</a>
            </li>
            <li>
                <a href="../Pages/fashion.html"
target="frameBody"> <span class="fa fa-
tshirt"></span> Fashion</a>
```

```

        </li>
        <li>
            <a href="../Pages/footwear.html"
target="frameBody"> <span class="fa fa-shoe-
prints"></span> Footwear</a>
        </li>
    </ol>
</td>
<td>
    <iframe src="../Pages/home.html"
name="frameBody" width="100%"
height="400"></iframe>
</td>
<td width="160">
    
</td>
</tr>
<tr height="50">
    <td bgcolor="black" colspan="3">
        <table align="center">

```

```
<tr>
  <td>
    <ul>
      <li>
        <h2>Get to Know Us</h2>
      </li>
      <li>About Us</li>
      <li>Press Release</li>
      <li>Gift a smile</li>
    </ul>
  </td>
  <td>
    <ul>
      <li>
        <h2>Get to Know Us</h2>
      </li>
      <li>About Us</li>
      <li>Press Release</li>
      <li>Gift a smile</li>
    </ul>
```

```
</td>
<td>
  <ul>
    <li>
      <h2>Get to Know Us</h2>
    </li>
    <li>About Us</li>
    <li>Press Release</li>
    <li>Gift a smile</li>
  </ul>
</td>
<td>
  <ul>
    <li>
      <h2>Get to Know Us</h2>
    </li>
    <li>About Us</li>
    <li>Press Release</li>
    <li>Gift a smile</li>
  </ul>
```



```
        </td>
      </tr>
    </table>
  </td>
</tr>
</table>
</body>
</html>
```

Task:



Indian players pose with the winning trophy after defeating Australia by three wickets on the final day of the fourth cricket test match at the Gabba, Brisbane in Australia on Tuesday.

— AP

Improbable win for India

MOSES KONDETY | DC
HYDERABAD, JAN. 19

Brisboon! That more or less sums up the Indian cricket team's ecstatic moment in Brisbane which earned them a 2-1 Test series win on Tuesday that will be talked about for years to come. For it took many years coming. Not since 1988 had any team defeated Australia at this venue. And this Indian team was depleted, inexperienced, weak and written off.

However, young India changed the script, and how! Going into the final

"We are all overjoyed at the success of the Indian Cricket Team in Australia. Their remarkable energy and passion was visible throughout. So was their stellar intent, remarkable grit and determination. Congratulations to the team! Best wishes for your future endeavours

NARENDRA MODI,
Prime Minister



Every time we got hit, we stayed put & stood taller. We pushed boundaries of belief to play fearless but not careless cricket. Injuries & uncertainties were countered with poise & confidence. One of the greatest series wins! Congrats India. Every session we discovered a new hero...

SACHIN TENDULKAR

day of the four-match series needing an improbable 324 for victory — the

highest target chased here was 236, seven decades ago — a charged-up India

set out to take the Brisbane bull by its horns.

Opener Shubman Gill (91) and Cheteshwar Pujara (56) did the spade-work, before Rishabh Pant strode on to the green Gabba ground and bashed the ball about to drive the side home (329 for 7) in top gear with a stroke-filled 89 not out.

Fuel for the purring Pant engine though was provided by high octane efforts from debutant Washington Sundar (62) and Shardul Thakur (67).

■ Page 4: Siraj halted Australia's second innings

THE SUPER SEVEN



MOHAMMED SIRAJ | Son of an auto driver from Hyderabad. He lost his father during the tour but decided to stay on. He got a maiden five-for, and picked 13 wickets in 3 Rests. Faced racist abuse.



SHUBMAN GILL | Born in a Fazilka, Punjab, to an affluent farmer family. His grandfather had prepared a pitch in the farm before his father shifted to Mohali to back his ambitions.



SHARDUL THAKUR | From Palghar, Maharashtra, had hit six sixes in an over as a 13-year-old. Travelled to Mumbai daily. Coached by Dinesh Lad, who also coached Rohit Sharma.



WASHINGTON SUNDAR | The name Washington was his father's tribute to his own mentor P.D.

Washington, who had sponsored him. His talent as an off-break bowler came to the fore when Rahul Dravid and Paras Mhambrey told him to focus on his bowling.



NAVDEEP SAINI | Son of a bus driver from Karnal, used to play tennis ball matches for ₹1,000. Delhi first-class player Sumit Narwal brought him for the Ranji Trophy nets where Gautam Gambhir picked him up. He was in Australia as a net bowler.



THANGARASU NATARAJAN | From the remote village of Chinnappampatti in Tamil Nadu, the son of a daily wage labourer couldn't buy bowling spikes, till IPL riches came along. He has built a cricket academy at his native place and helps talented but needy cricketers. He was in Australia as a net bowler.

Forms in HTML

- Form provides an UI from where user can interact with our application.
- Interaction includes insert, update, delete, query etc.
- Form is a container with collection of elements like button, textbox, checkbox, radio, list box etc.

- HTML form is created by using following elements
 - `<form>`
 - `<input>`
 - `<select>`
 - `<option>`
 - `<datalist>`
 - `<textarea>`
 - `<meter>`
 - `<progress>` etc.

Form Container

- Form container is defined by using `<form>` tag.
- A web page can contain multiple forms. Hence you have to define a start and end tag.

Syntax:

`<form>`

`</form>`

- A page can submit the details only when they are enclosed a form.
- You have to make sure that the form elements are present inside `<form>` container.

Syntax:

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

Not good

<form>

Name : <input type="text" name="Name"> // OK

</form>

- **HTML form element provides following attributes**

Attribute	Description
Id	It defines a unique ID for form.
name	It defines a reference for form.
class	It defines a CSS class for form.
method	It indicates the request method for form, which can be: <ul style="list-style-type: none">- GET- POST The default form method is GET
action	It specifies the target where the data is submitted. It refers to the server-side page to submit form data. Ex: page.jsp, page.aspx, page.php, page.asp Syntax: <form method="post" action="page.aspx">
target	It can open the target page in a new tab or in a frame.

	<p>Syntax:</p> <pre><form method="post" action="page.aspx" target="_blank"></pre>
novalidate	<ul style="list-style-type: none"> - HTML 5 introduced validations. - Validation is the process of verifying the user input and restrict to specific type or range. - Some of the validations provided by HTML 5 are not suitable for our requirements. Hence we have to by-pass [Ignore] the validations by using "novalidate". <p>Ex:</p> <pre><form novalidate> Email: <input type="email" name="Email"> <button>Submit</button> </form></pre>

Note: Always use "Camel Case" for naming.
[camelCase]

Prefix must speak about object type

Suffix must speak about its purpose.

btnDelete, frmRegister

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Forms in HTML</title>
```

```
    <style>
```

```
      .form-login {
```

```
        width: 300px;
```

```
        height: 100px;
```

```
        border: 2px solid;
```

```
        padding: 20px;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <form id="frmLogin" class="form-login"
    name="frmLogin">
```

```

    </form>

</body>

</html>

```

GET vs POST

GET	POST
<p>GET method is used to fetch [Access] data from server.</p> <ul style="list-style-type: none"> - If you are designing a form to get data from server and display in page, then use GET method. 	<p>POST method is used to submit data to server.</p> <ul style="list-style-type: none"> - If you are designing a form to submit the client data to server, then use POST method.
If submit form data on GET request it will be added into URL as Query String .	Data will be in Form Message Body
Any one can view your data.	It is not appended into URL, it is not visible to users.
It is not safe.	It is safe
It is easy to hack data.	It is hard to hack data.
It allows to bookmark the data.	It can't be bookmarked.

It is stored in browser logs. [history]	It is not stored in browser history.
You can't submit complex data, like binary data. It allows only ASCII type.	It can submit complex data.
You can submit limited data. Query string allows 2048 chars.	There is no limit for submitting data.
The data can be cached. It will store data in cache and saves round trip.	Data can't be cached.

FAQ: Can we submit form data on GET request?

A.Yes. But not recommended.

Form Input Elements

- HTML <input> element allows to input any type of value.
- HTML <label> element is used to set a label for fields.

Text Input / String Input:

HTML `<input>` can use the attribute “type” to define the type of value to accept as input. You can use “type=text” if you want to allow string type.

HTML provides several attributes for input element. The attributes suitable for text input are:

Attribute	Description
Id	It defines a unique ID
name	It defines a reference name.
class	It specifies the classes to implement.
value	It specifies the default value to display in side textbox. Syntax: <code><input type="text" value="David" name="txtName"></code>
placeholder	It defines the water mark text for input element. The water mark can't display if there is a value defined.
autofocus	It can set focus to the text box automatically on page load. Only one element in “Form” can be set with autofocus. Syntax: <code><input type="text" placeholder="Last</code>

	Name" autofocus>
readonly	<p>It will not allow to change value. It will not allow to input value. If you want to display a value but should not allow to modify the value, then you can mark it as readonly.</p> <p>Syntax: <input type="text" readonly name="txtId" value="101"></p>
disabled	<p>It will not allow to input or change value. It will not submit value.</p> <p>Syntax: <input type="text" disabled name="txtId" value="101"></p> <p>Disabled vs Readonly</p> <ul style="list-style-type: none"> - Readonly can submit the value - Can't submit the value
required	<ul style="list-style-type: none"> - It will not allow to submit empty. - It is used for mandatory fields. - If textbox is not defined with value, then it will not allow to submit the form. - However if form is defined with "novalidate" then you can

	<p>submit empty.</p> <p>Syntax: <input name="txtId" required="" type="text"/></p>
minlength	<p>It ensures that the input value minimum length matches the lower bound value.</p>
maxlength	<p>It restricts the input value to specified maximum number of chars.</p> <p>You can restrict the length of string in textbox by using minlength and maxlength.</p> <p>Syntax: <input maxlength="10" minlength="4" name="txtName" required="" type="text"/></p>
size	<p>It is used to define the width of textbox. The default width is 20.</p> <p>Syntax: <input maxlength="10" name="txtPin" size="2" type="text"/></p>
list	<ul style="list-style-type: none"> - It uses a data list that contains options to display as autocomplete text.

	<ul style="list-style-type: none"> - Data list is designed by using <datalist> element - Data list contains a set of options defined by using <option> element.
pattern	<ul style="list-style-type: none"> - It is used to verify the format of input value and restrict the value to specific format. - Pattern uses “Regular Expression” to verify the input value. - Regular Expression is built by using meta character and quantifiers.

Note: Form element can't submit its value if not defined with “name” attribute. Hence name is mandatory to submit the element value.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Forms in HTML</title>
```

```
    <style>
```

```
      form {
```

```
        display: flex;
        padding: 5px;
        border: 2px solid darkcyan;
        border-radius: 10px;
    }
    form > div {
        margin-left: 5px;
    }
</style>
</head>
<body>
    <form>
        <div>
            Home | About | Contact
        </div>
        <div>
            <input type="text" placeholder="First Name">
        </div>
        <div>
            <input type="text" placeholder="Last Name">
        </div>
    </form>
</body>
</html>
```

```
</div>
<div>
    <button>Submit</button>
</div>
</form>
</body>
</html>
```

FAQ: Can we change the color of placeholder?

A.Yes. By using “CSS – Placeholder Selector”

Syntax:

```
input::placeholder {
    color:red;
    font-weight: bold;
    font-family: Arial;
}
```

FAQ: Can we change the effect for element in focus?

A.Yes. By using CSS – Focus selector.

Syntax:

```
input:focus {
    border:none;
```

```
    box-shadow: 2px 3px 4px green;
}
```

2px - horizontal

3px - vertical

4px - blur

Ex: **Textbox with Data list**

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Google</title>
```

```
  </head>
```

```
  <body>
```

```
    <form>
```

```
      <div>
```

```
        <h1 align="center">Google</h1>
```

```
      </div>
```

```
      <div align="center">
```

```
        <input type="text" size="40" list="terms" >
```

```
        <datalist id="terms">
```

```
          <option>JavaScript Examples</option>
```

```
<option>JavaScript Tutorial</option>
<option>CSS Examples</option>
<option>CSS Tutorial</option>
</datalist>
</div>
</form>
</body>
</html>
```

Pattern for Input Element

[Regular Expression]

- Regular Expression is used to verify the format of input value.
- Regular Expression is built by using Meta Characters and Quantifiers.
- Pattern uses a Regular Expression to verify format of input value.

Syntax:

```
<input type="text" pattern="regExp">
```

Meta Character	Description
?	It defines zero or one occurrence of a character.

	<p>Ex:</p> <pre><input pattern="colou?r" placeholder="color / colour" type="text" name="txtTest"></pre>
*	<p>It defines zero or more occurrences of a character.</p> <p>Ex:</p> <pre><input pattern="colou*r" type="text" name="txtTest"></pre> <p>[color, colour, colour, coloruuur etc..]</p>
+	<p>It defines one or more occurrences of a character.</p> <p>Ex:</p> <pre><input pattern="colou+r" type="text" name="txtTest"></pre> <p>[colour, colour...]</p>
. [dot]	<p>Matches any single character in the specified string.</p> <p>Ex:</p> <pre><input pattern="b.y" type="text" name="txtTest"></pre> <p>[buy, boy]</p> <p>Ex:</p> <pre><input pattern=".y" type="text" name="txtTest"></pre> <p>[by, my]</p>

	<p>Ex: cow, toy, boy [pattern=".o."]</p> <p>Ex: cat, bat, mat, rat [pattern=".at"]</p>
\	<p>It is used as escape sequence character for special chars. [Chars used by parser]</p> <p>\+91 = +91</p>
\d	<p>It allows only numeric value, any single digit [0-9].</p> <p>Ex:</p> <p><input pattern="\d\d" type="text" name="txtTest"> [any 2 digits number]</p> <p>Ex:</p> <p><input pattern="\d?\d" type="text" name="txtTest"> [one or 2 digits]</p> <p>Ex:</p> <p><input pattern="[0-9]" type="text" name="txtTest"></p>
\D	<p>Not a digit [^0-9] or \D only non-digit.</p> <p>[^0-9] Any value other than number.</p>

	<p>^ Exclude</p> <p>Ex:</p> <pre><input pattern="\D\D" type="text" name="txtTest"></pre>
\w	<p>Matches alpha numeric characters with underscore.</p> <p>Equivalent to [A-Za-z0-9_]</p> <p>Ex:</p> <pre><input pattern="\w\w" type="text" name="txtTest"></pre>
\W	<p>Matches any character that is not a word character.</p> <p>[^A-Za-z0-9_]</p> <p>Ex:</p> <pre><input pattern="\W" type="text" name="txtTest"></pre> <p>[%\$#@&]</p>
^	Exclude specified
\s	<p>Match a single white space character. Including space, tab, form feed, line feed, and other Unicode chars.</p> <p>\n – line feed</p> <p>\f – form feed</p> <p>\t – horizontal tab</p>

	<p>\v – vertical tab</p> <p>Ex:</p> <pre><input pattern="\w\s\w" type="text" name="txtTest"></pre>
left right	<p>The “ ” is used as OR. We can configure multiple by using OR. It can match with any of the specified.</p> <p>Ex:</p> <pre><input pattern="green red blue" type="text" name="txtTest"></pre>
[A-Z]	Only uppercase letters allowed
[a-z]	Only lowercase letters allowed
[0-9]	Only numeric allowed.
[a-zA-Z]	Both upper and lowercase allowed.
[a-Z]	Both upper and lowercase allowed.
[a-zA-Z0-9]	Alpha numeric allowed
[a,d,s]	Only specified chars allowed.
[^a,d,s]	Excluding the specified all others allowed.
[a-zA-M4-9]	Only chars in specified range allowed.
[^a-zA-M4-9]	Excluding specified chars range all others allowed.
(abc)	Exactly the specified group allowed in specified order.
\$	Matches the end of input. “Ends with”.
^	Matches the beginning of input.

	"Starts with".
--	----------------

Quantifiers are used to define the range of value

Quantifier	Description
{n}	Exactly specified number of chars. n – refers to number. {8}
{n,m}	Minimum-n and Maximum-m {4,10} - 4 to 10
{n, }	Minimum-n and Maximum-any {4, }

Write a pattern to validate Name?

[Name can alpha numeric with lower, upper and underscore between 4 to 15 chars]

Ex: \w{4,15}

Write a pattern to validate "XX-0000-XX"?

<input pattern="\^[A-Z]{2}-[0-9]{4}-[A-Z]{2}\$"
type="text" name="txtTest">

Write a pattern to validate UK mobile number?

+447890-6970-69

\^\\+44\\d{4}-\\d{4}-\\d{2}\$

\^\\+44[0-9]{4}-[0-9]{4}-[0-9]{2}\$

Write a pattern to allow only alpha numeric with _ 4 to 15 chars but

Atleast one or many characters can be Uppercase.

John_123N

(?=.*[A-Z])\w{4,15}

Ex:

```
<input pattern="(?=.*[A-Z])\w{4,15}" type="text"
name="txtTest">
```

Number Input:

- HTML can use input type as “number”.
- It allows only numeric values.
- All basic attributes are similar to text input.
- Instead of max length and min length number uses “Min and Max”.
- Min: specifies the minimum range of value.
- Max: specifies the maximum range of value.
- Step: specifies the increment and decrement step value. [By default it is 1]

Syntax:

```
<input type="number" min="15" max="35" step="5" value="17" name="txtAge">
```

```
<input type="number" min="1.5" max="5.5" step="0.1" value="1.5" name="txtAge">
```

Date Input:

- HTML allows to input date value by using the “type=date”.
- It is not supported on all browsers. hence we have to implement using jQuery.
 - **date**
 - **datetime-local**
 - **month**
 - **week**

Syntax:

```
<input type="date" min="2021-01-28" max="2021-03-20" name="txtDob">
```

```
<input type="date" min="2021-01-28" value="2021-01-29" max="2021-03-20" name="txtDob">
```

- You can also use “type=datetime-local” which allows both date and time input.

```
<input type="datetime-local" name="txtDob">
```

- You can also use “type=month” for month input.
- You can use “type=time” for time input.

Range Input:

- It will display a slider that allows to select a range of values.
- Its attributes are similar to number.
- Slider value you can show using JavaScript.
- **Slide value can be submitted but can't display on page in every browser.**

```
<input type="range" min="1000" value="1000"  
max="10000" name="txtDob">
```

Password Input:

- It is similar to text in all attributes.
- It will mask the text with password char “*”
- You can configure by using “type=password”

Syntax:

```
<input required pattern="[A-Z]{4,10}" type="password"  
name="txtPwd">
```


URL Input:

- It validates URL format.
- If you want the user to input website address [URL] and restrict to URL format the you can define “type=url”

Syntax:

```
<input type="url" name="txtUrl">
```

Email Input:

- It validates email address format.
- You can define “type=email”

Syntax:

```
<input type="email" name="txtEmail">
```

Color Input:

- HTML provides a color picker by using “type=color”.
- It allows to select a color for color panel.
- The color code/name is submitted.

Syntax:

```
<input type="color" name="txtColor">
```

File Input:

- It allows the client to browse and select any file from computer to upload into server.
- However it is just a control for selecting file, upload logic we have to write by using “JavaScript/JQuery”.
- You can define by using “type=file”.
- You can use “multiple” attribute to enable selection for multiple files.

Syntax

```
<input type="file" multiple name="txtPhoto">
```

- You can configure specific type dynamically using MIME type or you can also use attribute “accept” for filtering the file type.

Syntax:

```
<input type="file" accept=".jpg" multiple  
name="txtPhoto">
```

Radio Buttons:

- Radio buttons are used to exhibit “Mutex” [Mutual Exclusion] mechanism.
- Radio buttons allow to select one or multiple.

- Once radio button is checked, it can't be unchecked.
- To uncheck the radio button, you have to reload the page or configure "Mutex"
- "Mutex" is a multi-threading mechanism that loads all tasks into memory but allow only one task in process.
- To configure mutex all radios must run on same thread, which you can define by using "common name".
- "Checked" attribute is used to select radio.
- By default, radio submits "ON" as value.
- You can define a value to submit.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Regular Expression</title>
```

```
  </head>
```

```
  <body>
```

```
    <form>
```

```
      <div>
```

```
<label>Upload Photo</label>
<div>
    <input type="radio" value="Male"
name="gender" checked> Male
    <input type="radio" value="Female"
name="gender"> Female
    <button>Submit</button>
</div>
</div>

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

CheckBox

- Checkbox allows user to select and deselect any option.
- You can use checkbox for selecting one or multiple options from a group of choices.
- The attributes are same as radio buttons
 - Id
 - Name

- Class
- Value
- Checked etc.
- It is defined by using input attribute "type=checkbox".

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Regular Expression</title>
```

```
  </head>
```

```
  <body>
```

```
    <form>
```

```
      <div>
```

```
        <label>Select Courses</label>
```

```
        <div>
```

```
          <input type="checkbox" name="course" value="Java" checked> Java
```

```
          <input type="checkbox" name="course" value=".NET"> .NET
```

```
        <button>Submit</button>
```

```
        </div>
    </div>
</form>
</body>
</html>
```

Ex: Apply effects checkbox checked property with CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>CheckBox</title>
    <style>
      .container {
        width: 500px;
        height: 200px;
        padding: 5px;
        background-color: lightgray;
      }
      .terms+span {
        color: red;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="checkbox">
        <input type="checkbox"/>
      </div>
      <div class="terms">
        Terms
      </div>
    </div>
  </body>
</html>
```

```
.terms:checked+span {
    color:green;
}
</style>
</head>
<body>
    <form>
        <div>
            <label>Terms of Service</label>
            <div>
                <div class="container">
                    <p>Depending on how you obtained the
Windows software, this is a license agreement
between (i) you and the device manufacturer or
software installer that distributes the software with
your device; or (ii) you and Microsoft Corporation (or,
based on where you live or, if a business, where your
principal place of business is located, one of its
affiliates) if you acquired the software from a retailer.
Microsoft is the device manufacturer for devices
produced by Microsoft or one of its affiliates, and
Microsoft is the retailer if you acquired the software
directly from Microsoft. Note that if you are a volume
```

license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

</div>

<div>

<input class="terms" type="checkbox">

I Accept

</div>

</div>

</div>

</form>

</body>

</html>

FAQ: How to design a checkbox list?

- HTML is not provided with checkbox list control, you have to design with CSS overflow technique.

<!DOCTYPE html>

<html>

<head>

<title>Checkbox list</title>


```
<style>
  ul {
    list-style: none;
    margin-left: -40px;
  }
  .container {
    border: 2px solid;
    width: 120px;
    height: 60px;
    overflow:auto;
  }
</style>
</head>
<body>
  <h2>Select Courses</h2>
  <div class="container">
    <ul>
      <li> <input type="checkbox"> Java </li>
      <li> <input type="checkbox"> .NET </li>
      <li> <input type="checkbox"> PHP </li>
```

```
<li> <input type="checkbox"> UI </li>
<li> <input type="checkbox"> Fullstack </li>
<li> <input type="checkbox"> Oracle </li>
</ul>
</div>
</body>
</html>
```

Dropdown list:

- It allows the user to select any one option from a group of choices.
- It shows the list of items using a dropdown menu.
- HTML provides the following elements for designing dropdown list:

<select>	It is used to configure a list.
<option>	It is used to configure a list item.
<optgroup>	It is used to group a set of options into one category.

- Every <option> comprises of following attributes
 - value
 - selected

- disabled
- Every <option> comprises of following properties
 - value
 - text
 - selected
 - disabled
- Every option is RC data type

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Dropdown</title>
```

```
  </head>
```

```
  <body>
```

```
    <h3>Select a Category</h3>
```

```
    <form>
```

```
      <select name="lstCategories">
```

```
        <option>Select a Category</option>
```

```
        <option value="Ele101">Electronics</option>
```

```
        <option selected>Footwear</option>
```

```
        <option disabled>Fashion</option>
```

```
      </select>
```

```
        <button>Submit</button>
    </form>
</body>
</html>
```

Ex: Option Group

```
<!DOCTYPE html>
<html>
    <head>
        <title>Dropdown</title>
    </head>
    <body>
        <h3>Select a Product</h3>
        <form>
            <select name="lstProducts">
                <option>Select a Product</option>
                <optgroup label="Electronics" >
                    <option>Samsung TV</option>
                    <option>LG Mobile</option>
                </optgroup>
```

```
<optgroup label="Footwear">
  <option>Nike Casuals</option>
  <option>Lee Boot</option>
</optgroup>
</select>
<button>Submit</button>
</form>
</body>
</html>
```

ListBox

- It everything same as dropdown list in elements and attribute.
- It allows to select one or multiple.
- To change the dropdown into listbox you have use the attribute “size or multiple”

Syntax:

```
<select size="3" multiple>
</select>
```

```
<!DOCTYPE html>
```

```
<html>

  <head>
    <title>Dropdown</title>
  </head>
  <body>
    <h3>Select a Product</h3>
    <form>
      <select size="3" multiple name="lstProducts">
        <optgroup label="Electronics" >
          <option>Samsung TV</option>
          <option>LG Mobile</option>
        </optgroup>
        <optgroup label="Footwear">
          <option>Nike Casuals</option>
          <option>Lee Boot</option>
        </optgroup>
      </select>
      <button>Submit</button>
    </form>
  </body>
```

</html>

Textarea

- It is used to handle multiline input.
- It is similar to textbox but allows multiple lines.
- It is RC data element used to present plain text.
- Attributes are similar to text input, additionally uses
 - Rows
 - Cols

Syntax:

```
<textarea rows="4" cols="40">
```

Your text... / optional

```
</textarea>
```

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Text Area</title>
```

```
<style>
```

```
dt {
```

```
font-weight: bold;
```

```
}
```

```
</style>
</head>
<body>
  <dl>
    <dt>User Name</dt>
    <dd><input type="text"></dd>
    <dt>Address</dt>
    <dd>
      <textarea disabled rows="4" cols="40">
        <b>Your Address.. here..</b>
      </textarea>
    </dd>
    <dt>Comments</dt>
    <dd>
      <textarea rows="4" cols="40">

      </textarea>
    </dd>
  </dl>
</body>
```


</html>

Meter Element

- It is used to display the grade meter.
- It can display a value range with
 - min
 - max
 - value
 - low
 - high

Syntax:

```
<meter min="" max="" value="" low="" high="" id=""  
class="" name="">
```

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Meter</title>
```

```
<style>
```

```
meter {
```

```
width: 300px;
```

```
height: 40px;
```

```
    }
  </style>
</head>
<body>
  <fieldset>
    <legend>Meter</legend>
    <dl>
      <dt>Normal Style</dt>
      <dd>
        <meter></meter>
      </dd>
      <dt>Meter 40%</dt>
      <dd>
        <meter min="1" max="100" value="40">

        </meter>
      </dd>
      <dt>Meter 100%</dt>
      <dd>
        <meter min="1" max="100" value="100">
```

```
        </meter>
    </dd>
    <dt>Meter Poor</dt>
    <dd>
        <meter min="1" max="100" value="100"
low="60" high="80">
```

```
        </meter>
    </dd>
    <dt>Meter Weak</dt>
    <dd>
        <meter min="1" max="100" value="100"
low="20" high="80"></meter>
```

```
    </dd>
</dl>
```

```
</fieldset>
```

```
</body>
```

```
</html>
```

Progress Element:

- It is used to show the status of any task performed in the page.
- The task includes actions like
 - Copying
 - Downloading
 - Uploading
 - Connecting etc.
- It is used to display the time elapsed or the remaining to complete the task.
- Specially progress is used in pages that use “Ajax” [Asynchronous JavaScript And XML]
- Attributes used for progress
 - Min
 - Max
 - Value

Syntax:

```
<progress min="" max="" value="" name="">  
</progress>
```

Ex:

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

```
</head>
<body>
  <dl>
    <dt>Preparing for Download</dt>
    <dd>
      <progress></progress>
    </dd>
    <dt>80% Downloaded</dt>
    <dd>
      <progress min="1" value="80"
max="100"></progress>
    </dd>
    <dt>Download Completed</dt>
    <dd>
      <progress min="1" max="100"
value="100"></progress>
    </dd>
  </dl>
</body>
</html>
```

Form Output Element:

- It is a container element into which application can inject the result of a calculation or the outcome of user input.
- It is defined by using <output> tag.
- It requires a form event “oninput”

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Output Element</title>
```

```
  </head>
```

```
  <body>
```

```
    <form
```

```
      oninput="total.value=qty.value*price.value">
```

```
      <dl>
```

```
        <dt>Name</dt>
```

```
        <dd><input type="text"></dd>
```

```
        <dt>Quantity</dt>
```

```
        <dd>
```

```
        <input type="number" id="qty" name="qty"
value="1">
    </dd>
    <dt>Price</dt>
    <dd>
        <input type="text" id="price"
name="price">
    </dd>
    <dt>Total</dt>
    <dd>
        <output name="total" for="qty price"
></output>
    </dd>
</dl>
</form>
</body>
</html>
```

Buttons:

- Buttons are used to confirm user actions.
 - Record Actions
 - Insert, Update, Delete, Edit
 - Record Navigation

- Next, Previous, First, Last
- Miscellaneous Actions
 - Submit, Register, Login, Reset, Save, Open, Print, Cancel etc.
- **HTML 5** provides “button” element for designing button.
- Buttons in HTML are classified into 2 groups
 - Generic Buttons
 - Non-Generic Buttons
- **Generic Buttons have pre-defined functionality**
 - Submit: Have the ability to submit “form”.
 - `<input type="submit">`
 - `<button type="submit"> Text </button>`
 - `<button> Text </button>`
 - Reset: Have the ability to reset the form.
 - `<input type="reset">`
 - `<button type="reset"> Text </button>`

Note: HTML <button> Allows images and symbols.

HTML <input type="submit"> Will not allow symbols or images.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```



```
<title>Button</title>

<link rel="stylesheet" href="/fonts/css/all.css">

</head>

<body>

  <form>

    <dl>

      <dt>Name</dt>

      <dd><input type="text" value="TV"
name="txtName"></dd>

      <dt>Price</dt>

      <dd><input type="text"
name="txtPrice"></dd>

      <dt>Stock</dt>

      <dd><input type="checkbox"> Available</dd>

    </dl>

    <button>

      <span class="fa fa-user"></span> Login

    </button>

    <button type="reset">

      
```

```
        </button>
    </form>
</body>
</html>
```

- **Non-Generic Button:** It is a simple button without any pre-defined functionality.

```
    <input type="button">
    <button type="button">
```

Ex:

```
<button type="button">
    <span class="fa fa-print"></span> Print
</button>
<input type="button" value="Save">
```

Note: Generic button can have the default functionality only when defined in “<form>” element.

Button size must be minimum 44x44 pixels.

Ex: Change Button Border Radius

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

```
<head>

  <title>Button</title>

  <link rel="stylesheet" href="/fonts/css/all.css">

  <style>
    button {
      border:2px solid red;
      border-radius: 25px;
      height: 50px;
      background-color: lightpink;
    }
  </style>
</head>

<body>
  <form>
    <dl>
      <dt>Name</dt>
      <dd><input type="text"
name="txtName"></dd>
      <dt>Price</dt>
      <dd><input type="text"
name="txtPrice"></dd>
```

```
        <dt>Stock</dt>
        <dd><input type="checkbox"> Available</dd>
    </dl>
    <button type="reset">Reset</button>
</form>
</body>
</html>
```

Ex: Amazon Login

```
<!DOCTYPE html>
<html>
    <head>
        <title>Amazon Login</title>
        <style>
            .container {
                border:1px solid gray;
                border-radius: 5px;
                width: 400px;
                height: 400px;
                justify-content: center;
```

```
    margin:auto;
    padding:40px;
}
h1 {
    font-family: sans-serif;
    font-weight:lighter;
}
label {
    font-weight: bold;
    font-family: Arial;
}
.form-control {
    width: 100%;
    height: 30px;
    border-radius: 5px;
    border:2px solid orange;
    margin-top: 5px;
}
.btn {
    width: 100%;
```

```
        height: 40px;
        border-radius: 5px;
        background-color: lightyellow;
        margin-top: 10px;
    }
    .form-group {
        margin-bottom: 15px;
    }
    a {
        text-decoration: none;
        font-size: 18px;
    }
</style>
</head>
<body>
    <div align="center">
        
    </div>
    <div class="container">
```

```
<h1 align="left">Sign-in</h1>
```

```
<div class="form-group">
```

```
  <label>Email or mobile phone number</label>
```

```
  <div>
```

```
    <input class="form-control" type="text">
```

```
  </div>
```

```
</div>
```

```
<div class="form-group">
```

```
  <button class="btn">Continue</button>
```

```
</div>
```

```
<div class="form-group">
```

```
  <p>By continuing, you agree to Amazon's <a  
href="#">Conditions of Use</a> and Privacy  
Notice.</p>
```

```
</div>
```

```
<div class="form-group">
```

```
  <details>
```

```
    <summary><a href="#">Need  
Help?</a></summary>
```

```
    <p><a href="#">Recover Password</a></p>
```

```
        <p><a href="#">Edit Account
Details</a></p>
    </details>
</div>
</div>
</body>
</html>
```

Multimedia in HTML

- Multimedia is all about animations, audio and video.
- HTML provides various elements for presenting animations, audio and video in page.

Marquee:

- It is used to display sliding and scrolling content in page.
- <marquee> is a container that can contain scrolling and sliding content.

Syntax:

```
<marquee>
```

```
    Your content
```

```
</marquee>
```


- You can use the following attributes to handle marquee

Attribute	Description
Scrollamount	<p>Control the marquee scrolling speed [1 to 100].</p> <p>Syntax: <marquee scrollamount="15"> Flash.. Flash.. 70% OFF on electornics. </marquee></p>
direction	<p>It specifies the scrolling directions, which can be left, right, up or down.</p> <p>Syntax: <marquee direction="up"> </marquee></p>
Width and Height	<p>Sets the width and height for marquee content.</p> <p>Syntax: <marquee width="100" height="300"> </marquee></p>
behavior	<p>It changes the marquee behaviour from scrolling to</p>

	<p>sliding. It can be defined as “alternate, scroll and slide”.</p> <p>Ex:</p> <pre><marquee behavior="alternate" scrollamount="15"> Flash.. Flash.. 70% OFF on electornics. </marquee></pre>
Loop	<p>It specifies the number of time marquee content need to display.</p> <p>Ex:</p> <pre><marquee loop="3" scrollamount="15"> Flash.. Flash.. 70% OFF on electornics. </marquee></pre>
Scrolldelay	<p>It delays the start of marquee.</p> <pre><marquee scrolldelay="100"> </marquee></pre>
Bgcolor	<p>It sets background color for marquee area.</p> <p>Syntax:</p> <pre><marquee bgcolor="yellow"> </marquee></pre>

Marquee Events are “onmouseover” and “onmouseout”.

The methods “start()” is to start marquee and “stop()” to stop the marquee()”.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Marquee</title>
```

```
    <style>
```

```
      </style>
```

```
  </head>
```

```
  <body>
```

```
    <div>
```

```
      <marquee scrollDelay="10">
```

```
        Flash.. Flash.. 70% OFF on electornics.
```

```
      </marquee>
```

```
      <marquee scrollDelay="25">
```

```
        Flash.. Flash.. 70% OFF on electornics.
```

```
      </marquee>
```

```
      <marquee scrollDelay="50">
```

Flash.. Flash.. 70% OFF on electornics.

</marquee>

<marquee scrollldelay="100">

Flash.. Flash.. 70% OFF on electornics.

</marquee>

<marquee scrollldelay="300">

Flash.. Flash.. 70% OFF on electornics.

</marquee>

</div>

<div>

<marquee onmouseout="this.start()" onmouseover="this.stop()" bgcolor="yellow" width="100" scrollamount="10" direction="up">

<div>

</div>

<div>

</div>

```
<div>
    
</div>
<div>
    
</div>
<div>
    
</div>
</marquee>
</div>
</body>
</html>
```

Embed Audio and Video Content

- HTML can use “<embed>” element to embed any multi media content into page.
- HTML 5 introduces <video> and <audio> elements for embedding multi media.

Video Element:

- HTML Video Element <video> embeds a media player into web page.
- The media player will support video playback in document.
- You can embed “audio” content with <video> tag. But recommended to use <audio> tag for audio content, as it provides better attributes for audio.
- The commonly used attributes for “<video>” element

Attribute	Description
autoplay	It uses “boolean” value [true/false]. It enables auto play when set to true.
controls	It is used to define controls for media player. Controls can be like seekbar, and play/resume etc.
crossorigin	It indicates whether to use CORS.
Height /width	To define width and height of media player.
Loop	It uses a boolean value, which allows to loop the video continuously.
Muted	It uses boolean value to mute the video on loading.
Poster	It is used as thumbnail for video before playback.

Src	It defines the name and location of video content.
-----	--

Syntax:

```
<video muted src="" poster="">
```

Message – Is displayed when browser is unable to use video element.

```
</video>
```

Note: You can also use <source> tag for configuring video source. This technique is used to embed multiple videos.

Syntax:

```
<video attributes>
```

```
<source src="name.avi"> </source>
```

```
<source src="name.ogg"> </source>
```

```
</video>
```

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Video</title>
</head>
<body>
  <h2>Video Element</h2>
  <video controls width="300" height="200"
src="../docs/class.mp4">
    <p>Video is not supported on your browser</p>
  </video>
</body>
</html>
```

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Video</title>
  </head>
  <body>
    <h2>Video Element</h2>
    <video poster="../Images/amazon.PNG"
width="300" height="200">
```



```
<source src="../docs/class.mp4" >
<p>Video is not supported on your browser</p>
</video>
<h2>Video Embed</h2>
<embed src="../docs/class.mp4" >
</body>
</html>
```

Note: Audio element is also similar to video. [mp3, MIDI]

Syntax:

```
<audio controls>
  <source src="music.mp3">
</audio>
```

Poster can't be defined for audio on various browsers.

CSS Colors

- Colors in CSS can be defined by using 3 techniques
 - Color Name
 - Hexadecimal color code
 - RGB Color methods

Color Names

- CSS supports 16 million colors.
- Standard color names are few.
- You can directly define the color name or color shade name.

Syntax:

```
h2{
```

```
color:green;
```

```
}
```

```
h2{
```

```
color:lightgreen;
```

```
}
```

Hexadecimal color code

- The primary colors are “Red, Green & Blue”.

- All colors are derived from primary colors.
- Hexadecimal allows to define multiple color shades.
- Hexadecimal colors have a combination of 3 or 6 chars with prefix “#”
- Hexadecimal number system is 16 base number system.
- We use 16 different values to mix and create a color shade.
- Hexadecimal color code uses the values **0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f**
- **0** is minimum and **f** is maximum.
- **Color code have 3 or 6 places other than #**
- If it is 3 then “#RGB”
 - R - Red value
 - G - Green value
 - B - Blue value
- Red, Green and Blue will have 1 place representation

Syntax:

```
<style>
```

```
input {
```

```
    background-color: #af0;
```

```
}  
</style>
```

- If it is 6 places then “#RRGGBB”.
- Red, green and blue values will have 2 places.

Syntax:

```
<style>  
input {  
    background-color: #dddddd;  
}  
</style>
```

RGB color code

- The colors are defined with “Red, Green and Blue” combination.
- The method “rgb()” is used for RGB colors.
- It takes 3 values “rgb(redValue, greenValue, blueValue)”
- The value range for red, green and blue will be between 0 to 255.

Syntax:

```
rgb(0,0,255)  blue color  
rgb(255,0,0)  Red color
```

Ex:

```
<style>
  input {
    background-color:rgb(205,123,215);
  }
</style>
```

RGBA (Red, Green, Blue, Alpha)

- Alpha is for opaque 1.0 is full opaque
- Range between 0.0 to 1.0

```
<style>
  input {
    background-color:rgba(255, 0, 0, 0.3);
  }
</style>
```

CSS Units

- Units define size and position.
- You can configure size with: height and width

- You can configure position with: x-axis, y-axis and z-axis
- The CSS units are categorized into 2 groups
 - Absolute Length Units
 - Relative Length Units

Absolute Length Units

- They are not relative to anything else and are generally considered as normal units.
- These are not affected by other relative elements and their units.

Unit	Name	Equivalent to
cm	Centimetres	$1\text{cm} = 96\text{px}/2.54 = 37\text{px}$
mm	Millimetres	$1\text{mm} = 1/10^{\text{th}}$ of 1 cm
Q	Quarter Millimetres	$1\text{Q} = 1/40^{\text{th}}$ of 1 cm
in	Inches	$1\text{in} = 2.54\text{cm} = 96\text{px}$
pc	Picas	$1\text{pc} = 1/6^{\text{th}}$ of 1in
pt	Point	$1\text{pt} = 1/72^{\text{th}}$ of 1in
Px	Pixels	$1\text{px} = 1/96^{\text{th}}$ of 1in

Relative length units:

- These are related to other contents in the page.
- The size of any element can be determined based on its parent, child or adjacent.
- The advantages are when parent element size is changed will relatively affects the child element also.

Unit	Relative To
em	<p>It uses the font size of parent element and applies to current element.</p> <p>Ex:</p> <pre><!DOCTYPE html> <html> <head> <title>Negation</title> <style> body { font-size: 20px; } .effects { font-size: 2em; } </style> </head> </html></pre>

	<pre> </style> </head> <body> Amazon <ul class="effects"> Item-1 Item-2 Item-3 </body> </html> </pre>
ex	X- height of elements' font [width]
ch	Defined for width, with regard to the root element.
rem	<p>Font size to the root element size.</p> <p>Ex:</p> <pre> <!DOCTYPE html> <html> <head> <title>Negation</title> <style> body { font-size: 20px; } .effects { </pre>

	<pre> font-size: 2em; } .effectsInner { font-size: 1rem; } </style> </head> <body> Amazon <ul class="effects"> Item-1 <ol class="effectsInner"> child-1 child-2 Item-2 Item-3 </body> </html> </pre>
ln	Line height of the element.
vw	1% of the viewport's width.
vh	1% of the viewport's height.
vmin	1% of the viewport's smaller

	dimension.
vmax	1% of the viewport's larger dimension.

Cascading Rules

- ***If a set of effects are re-defined for same element with same type of selector then according to CSS rule the last set of effects are applied to element.***

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Negation</title>
```

```
    <style>
```

```
      h2 {
```

```
        color:red;
```

```
      }
```

```
      h2 {
```

```
        color:blue;
```

```
    }
  </style>
</head>
<body>
  <h2>Amazon Shopping</h2>  // color -
blue
</body>
</html>
```

- ***If element have to choose between type selector and class selector then it will always choose the “class selector”.***

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Negation</title>
    <style>
      .heading {
        color:blue;
```

```
    }  
    h2 {  
        color:red;  
    }  
</style>  
</head>  
<body>  
    <h2 class="heading">Amazon  
Shopping</h2>  
</body>  
</html>
```

- ***If element is configured with both ID and class selector then always ID selector related effects are applied to element.***

Ex:

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

```
<style>
    .heading {
        color:blue;
    }
    h2 {
        color:red;
    }
    #effects {
        color:green;
    }
</style>
</head>
<body>
    <h2 id="effects" class="heading" >Amazon
Shopping</h2>
</body>
</html>
```

- ***If different style attributes are defined in ID, Class and Type selectors and applied to any***

specific element. Then then all styles are applied to element. Only same name style attributes are overridden.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Negation</title>
```

```
    <style>
```

```
      .heading {
```

```
        color:blue;
```

```
        text-align: center;
```

```
      }
```

```
      h2 {
```

```
        color:red;
```

```
        border:2px solid red;
```

```
      }
```

```
      #effects {
```

```
        color:green;
```

```
        box-shadow: 2px 3px 3px red;
    }
</style>
</head>
<body>
    <h2 id="effects" class="heading" >Amazon
Shopping</h2>
</body>
</html>
```

1st priority is for ID

2nd is for Class

3rd is for Type

CSS 2D Transforms

- CSS 2D Transform feature allows the element to be transformed in 2-Dimensional space.
- 2D is along X-Axis and Y-Axis
- CSS provides “transform” property to defined transformations.
- CSS transforms are not supported on various browsers.
- You have to use special plug-in for different browsers.

Browser	Plugin
Chrome, Safari, Opera	-webkit
Firefox	-moz
Internet Explorer 9	-ms

Syntax:

`transform: someMethod();` `//standard`

`-webkit-transform: someMethod();`

`-moz-transform:someMethod();`

`-ms-transform:someMethod()`

-o-transform:someMethod()

Various transforms provided by CSS are:

translate()

- It is used to move the element from its current position to a new position along x and y axis.

Syntax:

transform:translate(xPixels, yPixels);

transform:translateX(pixels);

transform:translateY(pixels);

Note: Transformation happens in just one second, you can set timing for transformation by using “transition” attribute.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Translate</title>
```

```
<style>
  img{
    margin-left: 10px;
    transition: 2s;
  }
  img:hover{
    transform: translate(500px,200px);
    -webkit-transform: translate(500px,
200px);
    -moz-transform: translate(500px, 200px);
    -ms-transform: translate(500px, 200px);
    transition: 2s;
  }
</style>
</head>
<body>
  
</body>
```

</html>

scale()

- It is used to increase or decrease the size of element along x and y axis.
- You can control the height and width of element.

Syntax:

transform:scale(scaleX, scaleY);

transform:scaleX()

transform:scaleY()

- Scale units are defined in fractions
 - 1 = 100
 - 1.5 = 150
 - 2 = 200

Syntax:

transform:scale(2) // both x and y will be 200

transform:scale(2,1) // x will be 200 and y 100

transform:scaleX(2) // x will be 200

X = width

Y = height

Ex:

<!DOCTYPE html>

```
<html>
  <head>
    <title>Translate</title>
    <style>
      img{
        margin-left: 10px;
        transition: 2s;
      }
      img:hover{
        transform: scale(3) ;
        transition: 2s;
      }
      .container div {
        align-items: center;
        justify-content: center;
        margin: auto;
        width: 300px;
        margin-top: 200px;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div>
```

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

Skew()

- Skew is used to tilt the image by specified angle.

Syntax:

transform:skew(sx, sy)

transform:skewX(angle)

transform:skewY(angle)

angle clockwise: 0 to 360

angle counter clockwise: -360

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <title>Translate</title>
```

```
        <style>
```

```
            img{
```

```
        margin-left: 10px;
        transition: 2s;
        height: 300px;
        width: 300px;
    }
    img:hover{
        transform: skew(-20deg) ;
        transition: 2s;
    }
    .container div {
        align-items: center;
        justify-content: center;
        margin: auto;
        width: 300px;
        margin-top: 200px;
    }
</style>
</head>
<body>
    <div class="container">
        <div>
            
        </div>
```

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

Rotate()

- It rotates the element around its origin.
- It uses value defined in deg.
- It uses 0 to 360 deg angle
- You can use -ve value for counter clockwise.

Syntax:

```
transform:rotate('angle');
```

```
transform:rotateX()
```

```
transform:rotateY()
```

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Translate</title>
```

```
    <style>
```

```
      img{
```

```
        margin-left: 10px;
```

```
        transition: 2s;
```

```
        height: 300px;
```

```
        width: 300px;
```

```
      }
```

```
img:hover{
    transform: rotate(-180deg) ;
    transition: 2s;
}
.container div {
    align-items: center;
    justify-content: center;
    margin: auto;
    width: 300px;
    margin-top: 200px;
}
</style>
</head>
<body>
    <div class="container">
        <div>
            
        </div>
    </div>
</body>
</html>
```


Transform Matrix

- Matrix allows to define all 2D transforms to single element.
- The transform methods
 - `translate(x,y)`
 - `translateX()`
 - `translateY()`
 - `scale(x,y)`
 - `scaleX()`
 - `scaleY()`
 - `rotate()`
 - `rotateX()`
 - `rotateY()`
 - `skew()`
 - `skewX()`
 - `skewY()`

Syntax:

```
matrix(scaleX(), skewY(), skewX(), scaleY(),  
translateX(), translateY())
```

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
  <title>Translate</title>
  <style>
    img{
      margin-left: 10px;
      transition: 2s;
    }
    img:hover{
      transition: 2s;
      transform: matrix(1.5,20,0.5,0,0,0);
    }

  </style>
</head>
<body>
  <div class="container">
    <div>
      
```

</div>

</div>

</body>

</html>

CSS 3D Transforms

- CSS 3D transforms allows 3-Dimensional design.
- It will have configuration for X, Y and Z axis.
- The methods used for 3D are same as 2D but comprises of 3rd Dimension.

Translate3d()

- It sets the position of element along x, y and z axis.

Syntax:

translate3d(tx, ty, tz)

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Translate</title>
    <style>
      .container {
        width: 100px;
        height: 100px;
        border:5px solid darkcyan;
```

```
        background-image:
url("../Images/jblspeaker.jpg");
        background-size: 100px 100px;
        background-repeat: no-repeat;
        margin:20px;
    }
    img {
        width: 100px;
        height: 100px;
        border:1px dotted darkcyan;
        opacity: 0.8;
        box-shadow: -25px -25px 3px
darkcyan;
    }
    .transform-effect{
        transform: translate3d(0px, 0px, 0px);
        transition: 2s;
    }
    .transform-effect:hover {
        transform: translate3d(25px, 25px,
100px);
        transition: 2s;
    }
```

```
        </style>
    </head>
    <body>
        <div class="container">

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

Scale3d

- It changes the image size in 3 dimensions.

Syntax:

scale3d(x, y, z)

Ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Translate</title>
        <style>
```

```
.container {
    width: 200px;
    height: 100px;
    border:5px solid darkcyan;
    background-color: darkgray;
    margin:20px;
}
img {
    width: 100px;
    height: 100px;
    border:1px dotted darkcyan;
    opacity: 0.8;
}
.transform-effect{
    transform: scale3d(1,1,1);
    transition: 2s;
}
.transform-effect:hover {
    transform: scale3d(1.5,1.5,1.5);
    transition: 2s;
}

</style>
</head>
```

```
<body>
  <div class="container">

    <marquee onmouseover="this.stop()"
onmouseout="this.start()"
scrollamount="10">
      
      
      
      
    </marquee>

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

CSS Transition

CSS Text Styling

- Font Styling

It provides a set of attributes and values that are used to format the text.

Attribute	Description
<i>color</i>	Defines the foreground color.
<i>font-family</i>	It defines the font family name like: Arial
<i>font-size</i>	It defines the text size.
<i>font-weight</i>	Defines normal, bold, lighter, bolder.
<i>font-style</i>	Defines italic effect.
<i>text-transform</i>	Defines uppercase, lowercase, capitalize, full-width. [affects the font size]
<i>text-decoration</i>	Defines underline, overline, line-through. It is short had for various text decoration options like style of line, color of line etc. <ul style="list-style-type: none">- text-decoration-color- text-decoration-style- text-decoration-line
<i>text-shadow</i>	It defines a shadow for text.
<i>text-align</i>	Left, center, right and justify
<i>line-height</i>	Space between lines.
<i>letter-spacing</i>	Space between letters.
<i>word-spacing</i>	Space between words.
<i>font-variant</i>	Caps, lowercase. [Without effecting the height of text it will change to

	caps]
<i>text-indent</i>	First line spacing.
<i>text-overflow</i>	Control the wrapping of text.
<i>white-space</i>	Control the blank space.
<i>direction</i>	Text direction
<i>text-orientation</i>	Changes the orientation of text.

Ex: Font, Font Style, Color, Text Decoration

```
<!DOCTYPE html>
<html>
  <head>
    <title>Text Effects</title>
    <style>
      div{
        color:red;
        font-family: Arial;
        font-size: 4em;
        font-weight: bold;
        font-style: italic;
        text-transform: uppercase;
        text-decoration: underline;
        text-decoration-color: blue;
        text-decoration-style: wavy;
      }
    </style>
  </head>
  <body>
    <div>sample text</div>
  </body>
```

</html>

FAQ: What are “Web Safe” fonts?

A. These are the font available across all browsers and systems.

FAQ: Name some web safe fonts?

Font Name	Generic Type (define alternative)
Arial	Sans-serif
Courier New	Monospace
Georgia	Serif
Time New Roman	Serif
Trebuchet MS	Sans-serif
Verdana	Sans-serif

FAQ: What are the default fonts?

A. Serif, sans-serif, monospace, cursive, fantasy

Text Shadow

- It defines shadow for text.
- Text shadow includes the following options to define
 - Horizontal Offset [left to right]
 - Vertical offset [front to back]
 - Blur radius
 - Color

Syntax:

text-shadow: hOffset vOffset blurRadius color;

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Text Effects</title>
    <style>
      div{
        font-size: 50px;
        text-transform: uppercase;
        font-weight: bold;
        text-shadow: 2px 3px 2px red;
      }
    </style>
  </head>
  <body>
    <div>sample text</div>
  </body>
</html>
```

Line, character, word spacing

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Text Effects</title>
    <style>
      div{
        text-align: justify;
```

```
    line-height: 45px;
    text-indent: 100px;
    word-spacing: 20px;
    letter-spacing: 5px;
}
</style>
</head>
<body>
    <div>Depending on how you obtained the Windows
software, this is a license agreement between (i) you and the
device manufacturer or software installer that distributes the
software with your device; or (ii) you and Microsoft
Corporation (or, based on where you live or, if a business,
where your principal place of business is located, one of its
affiliates) if you acquired the software from a retailer.
Microsoft is the device manufacturer for devices produced by
Microsoft or one of its affiliates, and Microsoft is the retailer
if you acquired the software directly from Microsoft. Note
that if you are a volume license customer, use of this
software is subject to your volume license agreement rather
than this agreement.
    </div>
</body>
</html>
```

Ex: Text Transform and Font variant

```
<!DOCTYPE html>
<html>
  <head>
    <title>Text Effects</title>
    <style>
      #p1 {
        text-transform: uppercase;
      }
      #p2 {
        font-variant: small-caps;
      }
    </style>
  </head>
  <body>
    <div id="p1">Welcome to CSS</div>
    <div id="p2">Welcome to CSS</div>
  </body>
</html>
```

Ex: **Text Overflow**

```
<!DOCTYPE html>
<html>
  <head>
```

```
<title>Text Effects</title>
<style>
  #p1 {
    width: 80px;
    border: 2px solid;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
  }

</style>
</head>
<body>
  <div title="Welcome to CSS" id="p1">Welcome to
CSS</div>

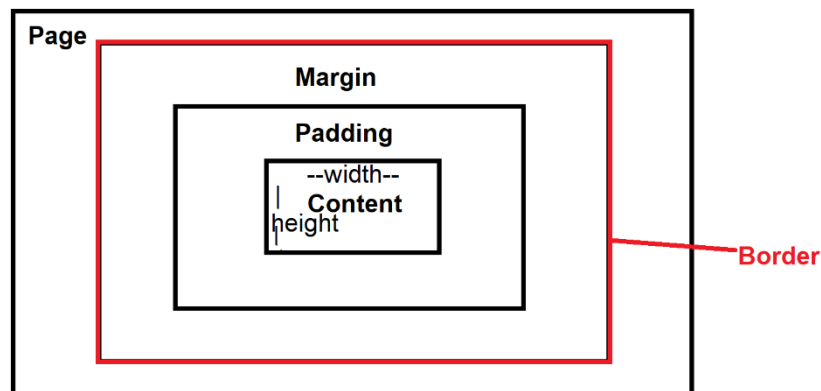
  

</body>
</html>
```

CSS Box Model

- Everything in CSS has a box around it.
- Understanding boxes is important for designing layout.
- It is often called as “Box Model”.
- Box model comprises of

Content Box	The area where your content is displayed. It can be defined by using “height and width” attributes.
Padding Box	The padding is around the content as white space. Its size can be controlled by using “padding”.
Border Box	The border box warps around the content or padding. It is controlled by using “border” property.
Margin Box	The margin is the outmost layer wrapping the content, padding and border. It configures a white space around border. It is defined by using “margin”



Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: gray;
      }
      .box {
        background-color: white;
        padding: 50px;
        text-align: justify;
```



```
border:4px solid;  
margin: 50px;  
}
```

```
</style>  
</head>  
<body>  
<div class="container">  
  <div class="box">
```

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

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

Margin

Padding Box

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

Set Margins, Padding and Border

Margin

- It specifies the space around border.
- You define by using following properties
 - margin [short hand – sets in all directions]
 - margin-top
 - margin-bottom
 - margin-left
 - margin-right

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
    background-color: gray;
```

```
}
```

```
.box {
```

```
    background-color: white;
```

```
    padding: 50px;
```

```
    text-align: justify;
```

```
    border: 4px solid;
```

```
    margin-left: 40px;
```

```
    margin-top: 20px;
```

```
    margin-right: 30px;
```

```
    margin-bottom: 60px;
```

```
}
```

</style>

</head>

<body>

<div class="container">

<div class="box">

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</div>

<div class="box">

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</div>

</div>

</body>

</html>

Padding

- It specifies the space around content
- You can define
 - Padding-left
 - Padding-right
 - Padding-top
 - Padding-bottom
 - Padding
- All options are similar like margin

CSS Transition

- CSS transition is a module of CSS.
- Transition is used for configuring animation for initial and final state.
- Initial state is before transformation.
- Final State is after transformation.
- You can control the behaviour using various CSS effects
 - timing
 - duration
 - other attributes
- CSS Transition comprises of following properties
 - transition
 - transition-delay
 - transition-duration
 - transition-property
 - transition-timing-function

transition-delay	Delay is time taken to start animation. Syntax: transition-delay: 2s;
------------------	---

transition-duration	<p>Time take to complete the transformation.</p> <p>Specifies the duration from initial to final and vice versa.</p> <p>Syntax:</p> <p>transition-duration: 5s;</p>
transition-property	<p>-You can define several effects for transformation.</p> <p>-All effects are transformed from initial to final.</p> <p>-You can define only the effect that you want to transform by using transform-property.</p> <p>Syntax:</p> <p>transition-property: width, height;</p> <p>transition-property:all;</p>
transition-timing-function	<p>-CSS provides a set of pre-defined animations functions which you can apply to content.</p> <p>-These effects have pre-defined timings and behaviour.</p>

	<ul style="list-style-type: none">-The transition timing functions are<ul style="list-style-type: none">- linear: speed will be even from begin to end.- ease-in: starts slowly and increases towards end.- ease-out: starts quickly and slow down.- ease-in-out- steps- jump-start- jump-end- jump-none- jump-both- start- end- step-start- step-end
--	---

Ex: **Duration and Delay**

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Timing</title>
```

```
    <style>
```

```
img {
    margin-left: 10px;
    transition-duration: 2s;
    transition-delay: 5s;
}
img:hover {
    transform: translate(400px,0);
    transition-duration: 2s;
}
</style>
</head>
<body>
    
    </body>
</html>
```

Ex: **Transition Property**

```
<!DOCTYPE html>
<html>
    <head>
        <title>Timing</title>
        <style>
            div{
```



```
        width: 100px;
        height: 100px;
        background-color: red;
        transition-duration: 2s;
        transition-property: width, height;
    }
    div:hover {
        width: 300px;
        height: 300px;
        background-color: yellow;
    }
</style>
</head>
<body>
    <div>

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

Ex: CSS Timing functions

```
<!DOCTYPE html>
<html>
    <head>
```

```
<title>Timing</title>
<style>
  div{
    width: 100px;
    height: 100px;
    transition-duration: 2s;
    background-color: red;
  }
  div:hover {
    width: 1000px;
  }
  #easein {
    transition-timing-function: step-start;
    transition-timing-function: steps(10);
  }
</style>
</head>
<body>
  <div id="easein">
```

Hello ! Welcome to CSS Effects and Animations

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

CSS Animations

- CSS animations are used to animate CSS transitions.
- You can reduce the use of JavaScript and Flash for animation.
- Animation can easily run on low bandwidth devices like mobiles, tabs.
- Animation will work even when JavaScript is disabled or flash plugin is not available.
- To create CSS animation, we need
 - @keyframes
 - animation
 - animation-name
 - animation-duration
 - animation-timing-function
 - animation-delay

- animation-iteration-count
- animation-direction
- animation-fill-mode
- animation-play-state

@Keyframes

- It is used to configure every frame in animation.
- At each frame we can define effects for object.
- Keyframes are defines with initial, final, and intermediate steps.

Syntax:

```
@keyframes identifierName
{
    from { effects for initial state }
    to { effects for final state }
}
```

Syntax:

```
@keyframes identifier
{
    0% { effects }
    40% { effects }
```

```
80% { effects }  
100% {effects}  
}
```

- Keyframes are applied to any element by using “animation-name” property

Syntax:

```
P {  
    animation-name:  
    keyframeIdentifierName;  
}
```

EX:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Timing</title>
```

```
<style>
```

```
.container {
```

```
    overflow: hidden;
```

```
}
```

```
@keyframes slideIn
```

```
{
```

```
    from
    {
        margin-left: 100%;
        transform: scale(3);
    }
    to
    {
        margin-left: 0%;
        transform: scale(1);
    }
}
h2{
    animation-name: slideIn;
    animation-duration: 5s;
}
</style>
</head>
<body class="container">
    <h2>CSS Animations</h2>
```

```
</body>
```

```
</html>
```

- You can add frames in between the start and end by using step percentage.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Timing</title>
```

```
<style>
```

```
.container {
```

```
    overflow: hidden;
```

```
}
```

```
@keyframes slideIn
```

```
{
```

```
    from
```

```
{
```

```
    margin-left: 100%;
```

```
    transform: scale(1);
```

```
    }
    50% {
        font-size: 300%;
    }
    to
    {
        margin-left: 0%;
        transform: scale(1);
    }
}
h2{
    animation-name: slideIn;
    animation-duration: 5s;
}
</style>
</head>
<body class="container">
    <h2>CSS Animations</h2>
</body>
```


</html>

- You can control the iteration count, the number of times to display by using **“animation-iteration-count”**

Ex:

<!DOCTYPE html>

<html>

<head>

<title>Timing</title>

<style>

.container {

overflow: hidden;

}

@keyframes slideIn

{

from

{

margin-left: 100%;

transform: scale(1);

```
    }
    50% {
        font-size: 300%;
    }
    to
    {
        margin-left: 0%;
        transform: scale(1);
    }
}
h2{
    animation-name: slideIn;
    animation-duration: 5s;
    animation-iteration-count: infinite;
}
</style>
</head>
<body class="container">
    <h2>CSS Animations</h2>
```

</body>

</html>

- You can control animation direction by using “animation-direction”

Ex

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Timing</title>
```

```
    <style>
```

```
      .container {
```

```
        overflow: hidden;
```

```
      }
```

```
      @keyframes slideIn
```

```
      {
```

```
        from
```

```
        {
```

```
          margin-left: 100%;
```

```
          transform: rotate(360deg);
```

```
        }
```

```
        to
```

```
        {
```

```
        margin-left: 0%;
        transform: rotate(0deg);
    }
}

img {
    width: 20%;
    height: 70%;
    animation-name: slideIn;
    animation-duration: 5s;
    animation-iteration-count: infinite;
    animation-direction: alternate;
    animation-delay: 5s;
    animation-timing-function: ease;
}
</style>
</head>
<body class="container">
    
    </body>
</html>
```


CSS Position

- Generally, elements are placed according to the document layout flow.
- Position allows to make element behave differently and take them out of the document flow.
- CSS Positions are defined by using “position” style attribute.

Static Position:

- It is according to the normal flow of document.
- It is the default position for every element.
- It will not have any effect on the element position.
- Static position is not affected by top, right, bottom, left and z-index attributes.

Ex:

```
p {  
    position: static;  
    right: 10px;  
}
```

Sticky Position:

- Element is positioned according to the normal flow of document.
- It can make the content sticky after scrolling to the specified margin.
- It required the margins to define and position by using left, right, top or bottom.

Ex:

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <title>Static Position</title>
    <style>
      h2 {
        position: sticky;
        top:0px;
        background-color: yellow;
      }
    </style>
  </head>
  <body>
```

```
    <p>Depending on how you obtained the Windows software,
this is a license agreement between (i) you and the device
manufacturer or software installer that distributes the software
with your device; or (ii) you and Microsoft Corporation (or, based
on where you live or, if a business, where your principal place of
business is located, one of its affiliates) if you acquired the
software from a retailer.</p>
```

```
    <h2>HTML Tutorial</h2>
```

```
    <p>Depending on how you obtained the Windows software,
this is a license agreement between (i) you and the device
manufacturer or software installer that distributes the software
with your device; or (ii) you and Microsoft Corporation (or, based
on where you live or, if a business, where your principal place of
business is located, one of its affiliates) if you acquired the
software from a retailer.</p>
```

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software

with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of

business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

</body>

</html>

Ex: Accordion Sticky

<!DOCTYPE html>

<html>

<head>

<title>Accordion Stick</title>

<style>

dt {

background-color: black;

color:white;

padding: 10px;

position: sticky;

top:0px;

left:0px;

margin:1em 0;

}

</style>

</head>

<body>

<h1>Web Technologies</h1>

<dl>

<dt>HTML</dt>

<dd>Normal Elements</dd>

<dd>Void Elements</dd>

<dd>RC Data Elements</dd>

<dd>Raw Text Elements</dd>

<dd>Normal Elements</dd>

<dd>Void Elements</dd>

<dd>RC Data Elements</dd>

<dd>Raw Text Elements</dd>

<dd>Normal Elements</dd>

<dd>Void Elements</dd>

<dd>RC Data Elements</dd>

<dd>Raw Text Elements</dd>

<dd>Normal Elements</dd>

<dd>Void Elements</dd>

<dd>RC Data Elements</dd>

<dd>Raw Text Elements</dd>

<dd>Normal Elements</dd>

<dd>Void Elements</dd>

<dd>RC Data Elements</dd>
<dd>Raw Text Elements</dd>
<dd>Normal Elements</dd>
<dd>Void Elements</dd>
<dd>RC Data Elements</dd>
<dd>Raw Text Elements</dd>
<dt>JavaScript</dt>
<dd>Variable</dd>
<dd>DataTypes</dd>
<dd>Operators</dd>
<dd>Statements</dd>
<dd>Variable</dd>
<dd>DataTypes</dd>
<dd>Operators</dd>
<dd>Statements</dd>
<dd>Variable</dd>
<dd>DataTypes</dd>
<dd>Operators</dd>
<dd>Statements</dd>
<dd>Variable</dd>
<dd>DataTypes</dd>
<dd>Operators</dd>
<dd>Statements</dd>
<dd>Variable</dd>

<dd>DataTypes</dd>
<dd>Operators</dd>
<dd>Statements</dd>
<dt>CSS </dt>
<dd>Selectors</dd>
<dd>Units</dd>
<dd>Positions</dd>
<dd>Box Layout</dd>
<dd>Selectors</dd>
<dd>Units</dd>
<dd>Positions</dd>
<dd>Box Layout</dd>
<dd>Selectors</dd>
<dd>Units</dd>
<dd>Positions</dd>
<dd>Box Layout</dd>
<dd>Selectors</dd>
<dd>Units</dd>
<dd>Positions</dd>
<dd>Box Layout</dd>
<dd>Selectors</dd>
<dd>Units</dd>
<dd>Positions</dd>
<dd>Box Layout</dd>

```
<dd>Selectors</dd>
<dd>Units</dd>
<dd>Positions</dd>
<dd>Box Layout</dd>
</dl>
</body>
</html>
```

Fixed Position:

- The element is removed from normal document flow.
- Element is no longer a part of DOM flow.
- Its position is fixed with left, right, top and bottom attributes.
- It will not move from the position defined.
- It locks the component at fixed position on page.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Fixed Position</title>
    <link rel="stylesheet" href="../Fonts/css/all.css">
    <style>
      div {
        border: 2px solid;
        padding: 10px;
      }
      aside {
```

```
width: 200px;
padding: 10px;
height: 30px;
border: 3px solid darkred;
background-color: darkolivegreen;
color: white;
text-align: center;
position: fixed;
top: 100px;
right: 100px;
}
</style>
</head>
<body>
<div>
```

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of

business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer.</p>

<aside>

 Your Cart Items

</aside>

</div>

</body>

</html>

Relative Position

- It according to the normal document flow.
- Its position will be relative to adjacent or parent element.
- It is from the parent or from adjacent.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Fixed Position</title>
```

```
    <link rel="stylesheet" href="../Fonts/css/all.css">
```

```
    <style>
```

```
      .box1, .box2, .box3 {
```

```
        width: 200px ;
```

```
        height: 100px;
```

```
        background-color: yellow;
```

```
        border:1px solid black;
```

```
      }
```

```
      .box2 {
```

```
        position: relative;
```

```
        margin-left: 30px;
```

```
      }
```

```
      .boxinner {
```

```
        position: relative;
```

```
        margin-left: 30px;
```

```
        border:2px solid;
        height: 100px;
    }
    .container {
        margin-left: 60px;
        border:1px solid;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="box1">

            </div>
            <div class="box2">
                <div class="boxinner">

                    </div>
                </div>
                <div class="box3">

                    </div>
                </div>
            </div>
        </div>
    </body>
```

</html>

Position Absolute

- The element is removed from the normal document flow.
- It is positioned relative to its closest element.
- It is affected by using left, right, top and bottom values.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Absolute Position</title>
```

```
<style>
```

```
#container {
```

```
    border:2px solid;
```

```
    background-color: yellow;
```

```
}
```

```
aside {
```

```
    position: absolute;
```

```
    border:1px dotted;
```

```
    width: 100px;
```

```
    top:100px;
```

```
    right: 10px;
```

```
}
```

```
#ads {
```

```
    width: 200px;
```

```

        height: 300px;
        position: absolute;
        border: 1px double;
    }
    #adsInner {
        position: absolute;
        width: 150px;
        height: 150px;
        border: 2px solid;
    }
</style>
</head>
<body>
    <div id="container">
        <p>some text..</p>
        <div id="ads">
            Parent
            <div id="adsInner">
                Child
                <aside>Ads here..</aside>
            </div>
        </div>
    </div>
</div>
</body>

```

</html>

Styles and CSS

- Styles are set of attributes defined for HTML elements to make the presentation more interactive and responsive.
- HTML elements have attributes but limited in functionality, style attributes make HTML more effective. They extend HTML element.
- Styles can be defined in 3 ways
 - o Inline Styles
 - o Embedded Styles
 - o External Style Sheet

Inline Styles:

- The styles are defined for elements by using “style” attribute.
- Every element has its own individual styles.
- The styles defined for one element can't be re-used for other elements.
- These styles are faster in rendering as they are local to element.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Styles</title>
  </head>
  <body>
    <h2 style="background-color: green; color:white; text-align: center;">HTML</h2>
    <h2>CSS</h2>
    <h2>JavaScript</h2>
  </body>
</html>
```

Embedded Styles:

- Styles are defined in page by using <style> element.
- You configure in head or body section.
- You can keep all your styles at one location and use across various elements.
- It is good for reusing styles.
- Slower than inline.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Styles</title>
    <style>
      h2 {
        background-color: green;
        color: white;
        text-align: center;
      }
    </style>
  </head>
  <body>
    <h2>HTML</h2>
    <h2>CSS</h2>
    <h2>JavaScript</h2>
  </body>
</html>
```

FAQ: Where to embed the styles, in head or in body?

- A.** If you want to configure a set of style, which are loaded into browser memory, and used later by the elements according to the requirement then keep in <head> section.
- If you want to configure a set of styles, which are applied directly on body load then better define them in <body>.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Styles</title>
    <style>
```

```

        .heading {
            background-color: green;
            color: white;
            text-align: center;
        }
    </style>
</head>
<body>
    <style>
        body {
            background-color: lightgreen;
        }
    </style>
    <h2 class="heading">HTML</h2>
    <h2>CSS</h2>
    <h2>JavaScript</h2>
</body>
</html>

```

FAQ: What is MIME type for Styles?

- MIME type defines the type content present in element.
- The MIME type is used by browser to understand the type of content.
- Styles MIME type is "text/css"

Syntax:

```
<style type="text/css"> </style>
```

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Styles</title>
```



```

<style type="text/css">
    .heading {
        background-color: green;
        color: white;
        text-align: center;
    }
</style>
</head>
<body>
    <style type="text/css">
        body {
            background-color: lightgreen;
        }
    </style>
    <h2 class="heading">HTML</h2>
    <h2>CSS</h2>
    <h2>JavaScript</h2>
</body>
</html>

```

Styles from External File:

- The styles are maintained in a separate style sheet that have the extension “.css”
- You can link the style sheet to any HTML page.
- Styles are accessible across several pages.
- If you use external style sheet then number of requests for page will increase and also the page load time.

Ex:

- Add a new folder “Styles”
 - Add a new style sheet into folder “effects.css”
 - Add effects into style sheet
- ```

h2 {
 background-color:blue;

```

- ```
    color:white;
    text-align: center;
}
```
- Link the stylesheet to your web page.
- ```
<!DOCTYPE html>
<html>
 <head>
 <title>Styles</title>
 <link type="text/css" rel="stylesheet" href="../Styles/effects.css">
 </head>
 <body>
 <h2>HTML</h2>
 <h2>CSS</h2>
 <h2>JavaScript</h2>
 </body>
</html>
```

### **FAQ: What is CDN?**

- CDN is Content Distribution Network
- We can maintain all style sheets in a repository server [SandBox]
- We can directly connect and access the style sheet from repository server instead of download into project.

Ex:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Styles</title>
 <link type="text/css" rel="stylesheet" href="http://127.0.0.1/cdn/styles/effects.css">
 </head>
 <body>
 <h2>HTML</h2>
 <h2>CSS</h2>
 <h2>JavaScript</h2>
 </body>
</html>
```

## Minification of CSS

- Minification is the process of compressing CSS.
- It is always recommended to Minify and use the CSS for production. [Live]
- CSS original file will occupy more space, we have to use them for development but not for production.

### FAQ: What is “media” type for styles?

- Media specifies the styles target, which can be for Print, Screen, Speech.
- “media” is an attribute used to configure styles targeting different sources like printer, screen, audio out etc.
- You can configure “media” attribute for <style> or <link> element.
- Media values can be
  - o All
  - o Print
  - o Screen
  - o Speech
- You can configure effects which will work for specific media.

Ex:

```
<!DOCTYPE html>

<html>

 <head>

 <style type="text/css" media="screen">

 body {

 border:2px solid darkcyan;

 padding:20px;

 }

 h1{

 border:1px solid red;

 box-shadow: 2px 3px 4px red;

 text-align: center;

 padding: 10px;

 }

 </style>
```

```

<style type="text/css" media="print">
 h1{
 border:1px solid red;
 text-align: center;
 padding: 10px;
 }
</style>
</head>
<body>
 <h1>Amazon Shopping</h1>
</body>
</html>

```

## Writing Styles for Elements

- If you are writing inline style for any element  
`<div style="stylePropertyName:value; stylePropertyName:value"> </div>`
- If you are writing styles embedded or in external file  
`<style>`  
   selector  
   {  
     stylePropertyName: value;  
     stylePropertyName: value;  
   }  
`</style>`
- Selector is used to define the target where the given styles need to apply.
- CSS can use various types of selectors
- The primary selectors used in styles are:
  - Type Selector
  - ID Selector
  - Class Selector

## Type Selector

- Type selector refers to HTML element tag name [Image <img>, Bold <b>].
- The given styles will be applied to specified tag where ever it is used in page.

- It will apply effects to every occurrence of the tag in page. You can't disable for any specific.

Ex:

```
<!DOCTYPE html>

<html>

 <head>

 <style>

 h2

 {

 background-color:red;

 color:white;

 text-align:center;

 }

 </style>

 </head>

 <body>

 <h2>HTML</h2>

 <p>It is a markup language.</p>

 <h2>CSS</h2>

 <p>Defines styles of HTML.</p>

 <h2>JavaScript</h2>

 <p>Handles client side interactions.</p>

 </body>

</html>
```

### **ID Selector**

- Every element can be defined with ID.
- You can use ID to access the element and apply effects.
- You can choose to which element you want the effects.
- Element is defined with ID  

```
<div id="effects"> </div>
```
- You can access the ID in styles by using "#" reference  

```
<style>
```

```
#effects
```

```
{
}
```

```
</style>
```

- Every tag can use only one ID reference.
- If you have configured multiple categories of styles with ID selector and want to use for specific tag, then it is not possible to define all effects to one element.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#textEffects {
```

```
 text-align: center;
```

```
 color:yellow;
```

```
}
```

```
#bgEffects {
```

```
 background-color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2 id="textEffects">HTML</h2>
```

```
<p>It is a markup language.</p>
```

```
<h2 id="bgEffects">CSS</h2>
```

```
<p>Defines styles of HTML.</p>
```

```
<h2>JavaScript</h2>
```

```
<p>Handles client side interactions.</p>
```

```
</body>
```

```
</html>
```

## Class Selector

- A class selector is defined by using “.”
- Class is accessed and applied to element by using “class” attribute.
- Every tag can implement multiple classes.
- Multiple classes are specified with space.

```
<style>
.cssClassName
{
}
</style>
<div class="cssClassName1 cssClassName2"> </div>
```

- The CSS selectors are further classified into various groups based on behaviour
  - **Combinators / Rational Selectors**
  - **Attribute Selectors**
  - **Pseudo Selectors**
  - **Structural Pseudo Selectors**

## Rational or Combinators

- These selector default with parent and child elements as well as with elements that have relation.
- Relation like adjacent, below, above, before, after, first, last etc..

Selector	Description
Descendent Selector	<p>Targets all tags under specified parent. It includes any level hierarchy.</p> <p>It defines the parent element and the child element by using space.</p> <p>Syntax:</p> <pre>parentElement childElement {</pre> <pre>}</pre> <p>Ex:</p> <pre>&lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       ol li {         color: red;       }       div p {         color:green;</pre>

	<pre>     }     &lt;/style&gt; &lt;/head&gt; &lt;body&gt;   &lt;h2&gt;Web Technologies&lt;/h2&gt;   &lt;ol&gt;     &lt;li&gt;HTML       &lt;ol&gt;         &lt;li&gt;Void Elements&lt;/li&gt;         &lt;li&gt;Normal Elements&lt;/li&gt;       &lt;/ol&gt;     &lt;/li&gt;     &lt;li&gt;CSS&lt;/li&gt;     &lt;li&gt;JavaScript&lt;/li&gt;   &lt;/ol&gt;   &lt;div&gt;     &lt;blockquote&gt;Blockquote...&lt;/blockquote&gt;     &lt;p&gt;Para-1&lt;/p&gt;     &lt;div&gt;       &lt;p&gt;Para-2&lt;/p&gt;     &lt;/div&gt;   &lt;/div&gt;   &lt;p&gt;Para-3&lt;/p&gt; &lt;/body&gt; &lt;/html&gt; </pre>
Child Selector	<p>It applies effects only to the direct child of parent element.</p> <p>Syntax:</p> <pre>Parent &gt; child {</pre> <pre>}</pre> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       div&gt;p {         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;div&gt;       &lt;p&gt;Para-1&lt;/p&gt;     &lt;/div&gt;     &lt;div&gt;       &lt;span&gt;         &lt;p&gt;Para-2&lt;/p&gt;       &lt;/span&gt;     &lt;/div&gt;   &lt;/body&gt; &lt;/html&gt; </pre>



	<pre>         &lt;/span&gt;       &lt;/div&gt;     &lt;/body&gt;   &lt;/html&gt; </pre>
Adjacent Sibling	<p>It defines effects to an element which is specified immediately after current element. It is not parent and child, it is one below another. It will apply only to the first adjacent element.</p> <p>Syntax: FirstElement + adjacentElement { }</p> <p>Ex: &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       h2+p {         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;h2&gt;HTML Elements&lt;/h2&gt;     &lt;p&gt;Para-1&lt;/p&gt;     &lt;p&gt;Para-2&lt;/p&gt;     &lt;p&gt;Para-3&lt;/p&gt;     &lt;p&gt;Para-4&lt;/p&gt;   &lt;/body&gt; &lt;/html&gt;</p>
General Sibling	<p>It defines effects to all elements which are specified after the current element.</p> <p>Syntax: FirstElement ~ AdjacentElements { }</p> <p>Ex: &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       h2~p {         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;h2&gt;HTML Elements&lt;/h2&gt;     &lt;p&gt;Para-1&lt;/p&gt;     &lt;p&gt;Para-2&lt;/p&gt;     &lt;p&gt;Para-3&lt;/p&gt;     &lt;p&gt;Para-4&lt;/p&gt;   &lt;/body&gt; &lt;/html&gt;</p>

	<pre>         &lt;/style&gt;       &lt;/head&gt;       &lt;body&gt;         &lt;h2&gt;HTML Elements&lt;/h2&gt;         &lt;p&gt;Para-1&lt;/p&gt;         &lt;p&gt;Para-2&lt;/p&gt;         &lt;p&gt;Para-3&lt;/p&gt;         &lt;p&gt;Para-4&lt;/p&gt;       &lt;/body&gt;     &lt;/html&gt; </pre>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Attribute Selectors

- Several elements in HTML are presented by using attribute of tag.  

☐
- "type" is attribute.
- We have to apply effects based on attribute and value.

Syntax:

```
tagName["attribute"] { }
```

```
tagName["attribute=value"] { }
```

### Ex: Attribute and Value

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
 input[type="button"] {
```

```
 background-color: lightgreen;
```

```
 }
```

```
 input[type="password"] {
```

```
 background-color: lightpink;
```

```
 }
```

```
</style>
```

```
</head>
<body>
 <form>
 <dl>
 <dt>Name</dt>
 <dd><input type="text"></dd>
 <dt>Password</dt>
 <dd><input type="password"></dd>
 </dl>
 <input type="button" value="Register">
 </form>
</body>
</html>
```

### **Ex: Only Attribute**

```
<!DOCTYPE html>
<html>
 <head>
 <style>
 p[id] {
 color: red;
 }
 </style>
 </head>
 <body>
 <p>Para-1</p>
 <p id="p2">Para-2</p>
```

<p>Para-3</p>

<p id="p4">Para-4</p>

</body>

</html>

- Attribute selectors can be defined with conditions.
- Effects are applied only to attribute that match the given condition.

Condition	Purpose
[attribute="val"]	<p>Equal specifies that it should be exact match.</p> <p>Ex:</p> <pre>&lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       p[class="Effect"] {         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;p class="paraEffect"&gt;Para-1&lt;/p&gt;     &lt;p class="para"&gt;Para-2&lt;/p&gt;     &lt;p class="Effectpara"&gt;Para-3&lt;/p&gt;     &lt;p class="Effect"&gt;Para-4&lt;/p&gt;   &lt;/body&gt; &lt;/html&gt;</pre>
[attribute^="val"]	<p>It refers the value starting with specified term.</p> <p>Ex:</p> <pre>&lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       p[class^="Effect"] {         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;p class="Effectpara"&gt;Para-3&lt;/p&gt;     &lt;p class="Effect"&gt;Para-4&lt;/p&gt;   &lt;/body&gt; &lt;/html&gt;</pre>

	<pre> &lt;/style&gt; &lt;/head&gt; &lt;body&gt;   &lt;p class="paraEffect"&gt;Para-1&lt;/p&gt;   &lt;p class="para"&gt;Para-2&lt;/p&gt;   &lt;p class="Effectpara"&gt;Para-3&lt;/p&gt;   &lt;p class="Effect"&gt;Para-4&lt;/p&gt; &lt;/body&gt; &lt;/html&gt; </pre>
[attribute\$="val"]	<p>It specifies that the value ending with given term.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       p[class\$="Effect"] {         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;p class="paraEffect"&gt;Para-1&lt;/p&gt;     &lt;p class="para"&gt;Para-2&lt;/p&gt;     &lt;p class="Effectpara"&gt;Para-3&lt;/p&gt;     &lt;p class="Effect"&gt;Para-4&lt;/p&gt;   &lt;/body&gt; &lt;/html&gt; </pre>
[attribute*="val"]	<p>It matches the term at any location.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       p[class*="Effect"] {         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt; </pre>

	<pre> &lt;p class="paraEffect"&gt;Para-1&lt;/p&gt; &lt;p class="para"&gt;Para-2&lt;/p&gt; &lt;p class="Effectpara"&gt;Para-3&lt;/p&gt; &lt;p class="Effect"&gt;Para-4&lt;/p&gt; &lt;/body&gt; &lt;/html&gt; </pre>
[attribute ="val"]	<p>Name starts with specified term and separated with "-".</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       p[class "Effect"] {         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;p class="para-Effect"&gt;Para-1&lt;/p&gt;     &lt;p class="para"&gt;Para-2&lt;/p&gt;     &lt;p class="Effect-para"&gt;Para-3&lt;/p&gt;     &lt;p class="Effect"&gt;Para-4&lt;/p&gt;   &lt;/body&gt; &lt;/html&gt; </pre>
[attribute~="val"]	<p>Name start with specified term and contain blank space.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       p[class~ "Effect"] {         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;p class="para-Effect"&gt;Para-1&lt;/p&gt;     &lt;p class="Effect para"&gt;Para-2&lt;/p&gt; </pre>

	<pre>         &lt;p class="Effect-para"&gt;Para-3&lt;/p&gt;         &lt;p class="Effect"&gt;Para-4&lt;/p&gt;       &lt;/body&gt; &lt;/html&gt; </pre>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------

## Dynamic Pseudo-Classes

- Dynamic indicates that the effect can change according to state and situation.
- Pseudo indicates that it is not referring to exactly the element which is having the same name as selector name.
- The selector name and the element if effects may differ.

Syntax:

link - not <link> element, it refers to <a>

```

class/Id/type: pseudoClass {
}

```

Selector	Description
:link	Specifies effect for Hyperlink.
:visited	It defines effects for visited links.
:hover	It defines effects when mouse pointer is over element.
:active	It defines effects when link is in active state.
:focus	<p>It defines effects when element get focus.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;style&gt;       .txtName+span {         display: none;       }       .txtName:focus+span {         display: inline;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;div&gt;       &lt;label&gt;Name&lt;/label&gt; </pre>

	<pre>&lt;div&gt;   &lt;input class="txtName" type="text"&gt;   &lt;span&gt;Name 4 Chars&lt;/span&gt; &lt;/div&gt; &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------

Syntax:

Element:Link { }

#heading:hover { }

.txtName:focus { }

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.txtName+span {
```

```
 display: none;
```

```
}
```

```
.txtName:focus+span {
```

```
 display: inline;
```

```
}
```

```
input:focus {
```

```
 border:2px solid darkcyan;
```

```
 box-shadow: 2px 2px 3px darkcyan;
```

```
}
```

```
a{
```

```
 text-decoration: none;
```

```
}
```



```
a:hover {
 text-decoration: underline;
}
a:active {
 color:red;
}
a:visited {
 color:green
}
a:link {
 color: gray;
}
</style>
</head>
<body>
<div>
 <label>Name</label>
 <div>
 <input class="txtName" type="text">
 Name 4 Chars
 </div>
</div>
<div>
 Home
 |
 Flipkart
</div>
</body>
</html>
```

## Target pseudo class

Selector	Description
:target	<ul style="list-style-type: none"><li>- It defines effects to any element when it becomes target of a link.</li><li>- You can implement in intra document navigation.</li></ul>

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Target</title>
```

```
<style>
```

```
ul {
```

```
list-style: none;
```

```
display: flex;
```

```
}
```

```
li {
```

```
margin-left: 50px;
```

```
border:2px solid darkblue;
```

```
padding: 10px;
```

```
width: 200px;
```

```
text-align: center;
```

```
border-radius: 10px;
```

```
}
```

```
.group {
```

```
border:2px solid darkgreen;
```

```
background-color: lightgreen;
```

```
color:black;
```

```
margin-top: 20px;
```

```
padding: 10px;
```

```
}
```

```
.group:target {
 background-color: black;
 color:white;
}
</style>
</head>
<body>
<header>
<nav>

HTML
CSS
JavaScript

</nav>
</header>
<section>
<div id="html" class="group">
<h3>HTML</h3>
<p>It is a markup language.</p>
</div>
<div id="css" class="group">
<h3>CSS</h3>
<p>It is to define styles</p>
</div>
<div id="js" class="group">
<h3>JavaScript</h3>
<p>It is a language.</p>
</div>
```

</section>  
</body>  
</html>

## The UI element state pseudo-classes

- Element state indicates the state of element like enabled, disables, readonly, checked.

Selector	Description
:enabled	It defines effects when element is enabled.
:disabled	<p>It defines effects when element is disabled.</p> <p>Ex:</p> <pre>&lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;State&lt;/title&gt;     &lt;style&gt;       input:read-only {         background-color: gainsboro;         color: gray;       }       button:disabled {         cursor: not-allowed;       }       button:enabled {         cursor: grab;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;fieldset&gt;       &lt;legend&gt;User Name&lt;/legend&gt;       &lt;div&gt;         &lt;input readonly type="text" value="John"&gt;         &lt;button disabled&gt;Submit&lt;/button&gt;       &lt;/div&gt;     &lt;/fieldset&gt;   &lt;/body&gt; &lt;/html&gt;</pre>
:read-only	It defines effects when element is set to read-only.

	<p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;State&lt;/title&gt;     &lt;style&gt;       input:read-only {         background-color: gainsboro;         color: gray;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;fieldset&gt;       &lt;legend&gt;User Name&lt;/legend&gt;       &lt;div&gt;         &lt;input readonly type="text" value="John"&gt;         &lt;button&gt;Submit&lt;/button&gt;       &lt;/div&gt;     &lt;/fieldset&gt;   &lt;/body&gt; &lt;/html&gt; </pre>
:checked	<p>It defines effects when element is checked.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;State&lt;/title&gt;     &lt;style&gt;       input[type="checkbox"]+span {         color:red;       }       input[type="checkbox"]:checked+span {         color: green;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;fieldset&gt;       &lt;legend&gt;Terms of Service&lt;/legend&gt;       &lt;textarea rows="4" cols="40"&gt;         Read our terms and conditions..       &lt;/textarea&gt;       &lt;div&gt;         &lt;input type="checkbox"&gt; &lt;span&gt;I </pre>

	Accept</span> </div> </fieldset> </body> </html>
--	--------------------------------------------------------------

### The UI element validation state pseudo classes:

- HTML 5 provides pre-defined form validations like require, email, url, pattern etc.
- CSS can use HTML 5 validations to verify the state valid or not and can apply effects.

Selector	Description
:valid	It defines effects for element if its value is valid against the validation defined. Validation can be verified by using: <ul style="list-style-type: none"> <li>- Minlength</li> <li>- Maxlength</li> <li>- Required</li> <li>- Pattern</li> <li>- Email</li> <li>- URL etc.</li> </ul>
:invalid	It defines effect for element when it is invalid.  Ex: <!DOCTYPE html> <html> <head> <title>State</title> <style> #txtName:valid+span { display: none; } #txtName:invalid+span { display: inline; } #txtName:valid { border:2px solid green; box-shadow: 2px 2px 3px green; } #txtName:invalid { border:2px solid red; box-shadow: 2px 2px 3px red; } </style> </head> <body>



Ex:

```
<!DOCTYPE html>

<html>

 <head>

 <title>State</title>

 <style>

 input:in-range {

 border: 2px solid green;

 box-shadow: 2px 3px 4px green;

 }

 input:out-of-range {

 border: 2px solid red;

 box-shadow: 2px 3px 4px red;

 }

 .form-group {

 margin-top: 20px;

 }

 #txtName:optional+div {

 display: none;

 }

 #txtName:required+div {

 display: block;

 color:red;

 }

 #txtName:valid+div {

 display: none;

 }

 </style>
```



```

</head>

<body>

 <div class="form-group">

 <label>Age</label>

 <div>

 <input type="number" min="16" max="35">

 </div>

 </div>

 <div class="form-group">

 <label>Name</label>

 <div>

 <input required id="txtName" type="text">

 <div>Name required</div>

 </div>

 </div>

</body>

</html>

```

### Structural Pseudo Selector:

- You can target your effects based on the position of element in parent and child hierarchy.

Selector	Description
:first-child	It defines effects only for first child element.
:last-child	It defines effects only for last child element.
:nth-child(LevelNumber)	<p>It defines effects only to specific child element that occurs at given level.</p> <p>Level number starts with 1.</p> <p>Index number starts with 0.</p> <p>You can also define the pre-set values like 'even &amp; odd' to apply effects based on even and odd occurrences.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt; </pre>

```

<title>Structure</title>
<style>
 ol > li:first-child {
 color:red;
 }
 ol > li:last-child {
 color:blue;
 }
 ol > li:nth-child(3){
 color: green;
 font-size: 30px;
 }
</style>
</head>
<body>

 Item-1
 Item-2
 Item-3
 Item-4
 Item-5

</body>
</html>

```

Ex: Even and Odd occurrence

```

<!DOCTYPE html>
<html>
 <head>
 <title>Odd Even</title>
 <style>
 thead > tr {
 background-color: darkcyan;
 color:white;
 }
 tbody > tr:nth-child(even) {
 background-color: lightcyan;
 }
 tbody > tr:nth-child(odd){
 background-color: lightgreen;
 }
 </style>
 </head>
 <body>
 <table border="1" width="400">
 <thead>
 <tr>

```

	<pre>         &lt;th&gt;Name&lt;/th&gt;         &lt;th&gt;Price&lt;/th&gt;       &lt;/tr&gt;     &lt;/thead&gt;     &lt;tbody&gt;       &lt;tr&gt;         &lt;td&gt;TV&lt;/td&gt;         &lt;td&gt;45000.55&lt;/td&gt;       &lt;/tr&gt;       &lt;tr&gt;         &lt;td&gt;Mobile&lt;/td&gt;         &lt;td&gt;41000.22&lt;/td&gt;       &lt;/tr&gt;       &lt;tr&gt;         &lt;td&gt;Nike&lt;/td&gt;         &lt;td&gt;5200.33&lt;/td&gt;       &lt;/tr&gt;       &lt;tr&gt;         &lt;td&gt;Shirt&lt;/td&gt;         &lt;td&gt;4100.33&lt;/td&gt;       &lt;/tr&gt;     &lt;/tbody&gt;   &lt;/table&gt; &lt;/body&gt; &lt;/html&gt; </pre>
<p>:nth-of-type(LevelNumber[n])</p> <p>:nth-of-type(2n)</p> <p>:nth-of-type(2n+startNumber)</p>	<p>It will repeat the effect for every nth occurrence.</p> <p>It will repeat the effect for every 2<sup>nd</sup> occurrence.</p> <p>It will start with specific level.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;Structure&lt;/title&gt;     &lt;style&gt;       ol &gt; li:nth-of-type(2n+1){         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;ol&gt;       &lt;li&gt;Item-1&lt;/li&gt;       &lt;li&gt;Item-2&lt;/li&gt;       &lt;li&gt;Item-3&lt;/li&gt;       &lt;li&gt;Item-4&lt;/li&gt;     &lt;/ol&gt;   &lt;/body&gt; &lt;/html&gt; </pre>

	<pre>         &lt;li&gt;Item-5&lt;/li&gt;       &lt;/ol&gt;     &lt;/body&gt;   &lt;/html&gt; </pre>
:nth-last-of-type(n)	<p>It will apply effect for every nth occurrence from bottom.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;Structure&lt;/title&gt;     &lt;style&gt;       ol &gt; li:nth-last-of-type(2n+1){         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;ol&gt;       &lt;li&gt;Item-1&lt;/li&gt;       &lt;li&gt;Item-2&lt;/li&gt;       &lt;li&gt;Item-3&lt;/li&gt;       &lt;li&gt;Item-4&lt;/li&gt;       &lt;li&gt;Item-5&lt;/li&gt;       &lt;li&gt;Item-6&lt;/li&gt;       &lt;li&gt;Item-7&lt;/li&gt;       &lt;li&gt;Item-8&lt;/li&gt;     &lt;/ol&gt;   &lt;/body&gt; &lt;/html&gt; </pre>
:nth-last-child(n)	<p>It will apply from bottom without repeating.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;Structure&lt;/title&gt;     &lt;style&gt;       ol &gt; li:nth-last-child(2){         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;ol&gt;       &lt;li&gt;Item-1&lt;/li&gt;       &lt;li&gt;Item-2&lt;/li&gt; </pre>

	<pre> &lt;li&gt;Item-3&lt;/li&gt; &lt;li&gt;Item-4&lt;/li&gt; &lt;li&gt;Item-5&lt;/li&gt; &lt;li&gt;Item-6&lt;/li&gt; &lt;li&gt;Item-7&lt;/li&gt; &lt;li&gt;Item-8&lt;/li&gt; &lt;/ol&gt; &lt;/body&gt; &lt;/html&gt; </pre>
:root	<p>It refers to root of document, which is 'body'</p> <p>Ex:</p> <pre> :root {   font-family:Arial; } </pre>
:empty	<p>If any element is empty, without any content then its will define the given effects.</p> <p>You can configure for containers like &lt;div&gt;, &lt;span&gt;, &lt;td&gt;, &lt;dd&gt;, &lt;p&gt; etc.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;Odd Even&lt;/title&gt;     &lt;style&gt;       thead &gt; tr {         background-color: darkcyan;         color:white;       }       tbody &gt; tr:nth-child(even) {         background-color: lightcyan;       }       tbody &gt; tr:nth-child(odd){         background-color: lightgreen;       }       tbody &gt; tr &gt; td:empty {         background-color: red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;table border="1" width="400"&gt;       &lt;thead&gt;         &lt;tr&gt;           &lt;th&gt;Name&lt;/th&gt;           &lt;th&gt;Price&lt;/th&gt;         &lt;/tr&gt; </pre>

	<pre> &lt;/thead&gt; &lt;tbody&gt;   &lt;tr&gt;     &lt;td&gt;TV&lt;/td&gt;     &lt;td&gt;45000.55&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;Mobile&lt;/td&gt;     &lt;td&gt;41000.22&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;Nike&lt;/td&gt;     &lt;td&gt;&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;Shirt&lt;/td&gt;     &lt;td&gt;4100.33&lt;/td&gt;   &lt;/tr&gt; &lt;/tbody&gt; &lt;/table&gt; &lt;/body&gt; &lt;/html&gt; </pre>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Pseudo-Element Selectors:

Selector	Description
::first-line	Effects for first line in paragraph.
::first-letter	<p>Effects for first character.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;Element Selectors&lt;/title&gt;     &lt;style&gt;       p::first-letter {         font-family: Arial;         font-size: 30px;         color:red;       }       p::first-line {         color:red;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;p&gt;Depending on how you obtained the Windows software, this     is a license agreement between (i) you and the device manufacturer </pre>

	<p>or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p> <pre>         &lt;/p&gt;       &lt;/body&gt; &lt;/html&gt; </pre>
::before	Effect or <b>content</b> to add before the current element.
::after	<p>Effect or <b>content</b> to add after the current element.</p> <p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;Before After&lt;/title&gt;     &lt;style&gt;       ul {         display: flex;         list-style: none;       }       li::before {         content: "--&gt;";       }       li:first-child::before {         content: "";       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     Site Map     &lt;nav&gt;       &lt;ul&gt;         &lt;li&gt;Home&lt;/li&gt;         &lt;li&gt;About&lt;/li&gt;         &lt;li&gt;Contact&lt;/li&gt;         &lt;li&gt;Login&lt;/li&gt;       &lt;/ul&gt;     &lt;/nav&gt;   &lt;/body&gt; &lt;/html&gt; </pre>
::placeholder	It will apply effects for placeholder.
::selection	It will apply effects for selection.

	<p>Ex:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;Languages&lt;/title&gt;     &lt;style&gt;       * {         font-style: italic;       }       input::placeholder {         color:lightgreen;       }       p::selection {         background-color: yellow;       }     &lt;/style&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;p&gt;Some content.. select and see..&lt;/p&gt;     &lt;dl&gt;       &lt;dt&gt;Name&lt;/dt&gt;       &lt;dd&gt;&lt;input placeholder="Name 4 chars" type="text"&gt;&lt;/dd&gt;       &lt;dt&gt;Password&lt;/dt&gt;       &lt;dd&gt;&lt;input disabled type="password"&gt;&lt;/dd&gt;       &lt;dt&gt;Mobile&lt;/dt&gt;       &lt;dd&gt;&lt;input required type="text"&gt;&lt;/dd&gt;     &lt;/dl&gt;   &lt;/body&gt; &lt;/html&gt; </pre>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Language Selector:

- It defines effects based on lang configured for element.
- If you page is multi lingual then you can define effects to content based on specific language.  
“:lang()”

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Languages</title>
```

```
<style>
```



```

 p:lang(en){
 font-style: italic;
 }
 </style>
</head>
<body>
 <h2>Language Selector</h2>
 <p>Some Text</p>
 <p lang="en">English US</p>
</body>
</html>

```

### Negation Selector

- It is used to define effects for the elements which are not matching with specified criteria.
- The negation selector is defined using “:not()”
- It will ignore effects for specific element and apply for other.

Ex:

```

<!DOCTYPE html>
<html>
 <head>
 <title>Languages</title>
 <style>
 p:not(#effects){
 color:red;
 }
 </style>
 </head>
 <body>

```

```
<p>Para-1</p>
<p id="effects">Para-2</p>
<p>Para-3</p>
<p id="styles">Para-4</p>
<p id="effects">Para-5</p>
</body>
</html>
```

- You can also configure for properties.

Ex:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Languages</title>
 <style>
 input:not([disabled]) {
 background-color: lightgreen;
 }
 </style>
 </head>
 <body>
 <dl>
 <dt>Name</dt>
 <dd><input type="text"></dd>
 <dt>Password</dt>
 <dd><input disabled type="password"></dd>
 <dt>Mobile</dt>
 <dd><input required type="text"></dd>
 </dl>
```

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

### **Universal Selector:**

- It is defined by using “\*” that represents all.
- It apply effects to all elements.

Ex:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Languages</title>
 <style>
 * {
 font-style: italic;
 }
 </style>
 </head>
 <body>
 <dl>
 <dt>Name</dt>
 <dd><input type="text"></dd>
 <dt>Password</dt>
 <dd><input disabled type="password"></dd>
 <dt>Mobile</dt>
 <dd><input required type="text"></dd>
 </dl>
 </body>
</html>
```

## JavaScript

- JavaScript is a language.
- It is an open source, cross platform, interpreted and just-in-time compiled programming language.
- Interpreter is a translated that translates line by line.
- Just-in-time is a compiler that compiles in browser when requested by client.
- V8 is JavaScript compiler.
- JavaScript is used
  - o Client Side with HTML
  - o Server Side with Node.js
  - o Database with MongoDB
  - o Animations with Flash
- JavaScript supports various types of programming like, imperative, functional programming, structural programming and object-oriented programming.
- In early 1994 **Brendan Eich** introduces a script called “Mocha” for Netscape browser.
- After that “Moca” was renamed into “Live Script”
- 1995 Sun Microsystems took responsibility of maintaining Live Script and re-named as “JavaScript”.
- JavaScript Designed by “Brendan Eich”
- JavaScript initially belong to “Netscape communications”.
- JavaScript follows the standards of “ECMA” [European Computer Manufacturers Association].
- JavaScript versions are ECMAScript 2015, ES6, ES8 ES2020
- ES5, ES6 are most commonly used versions in various web technologies.
- ES6

### Issues with JavaScript

- Browser incompatibility: Every browser has its own extensions to JavaScript and every browser have its own parser [translator].
- JavaScript is not secured: It is client side. Everyone can view.
- JavaScript can be disabled by browser. [Browser can block JavaScript].
- JavaScript is not strongly typed language.

```
var x = 10; // number
x = "John"; // string – valid
```

## JavaScript with HTML

- JavaScript is used to manipulate the HTML DOM.
- It converts the static DOM elements into dynamic DOM elements.
- JavaScript can add elements, remove elements, modify the data, handle validations, handle plugins, browser location, history etc.
- JavaScript can reduce burden on server by managing several interactions client-side.

## Using JavaScript in HTML Page:

- JavaScript can be integrated and used with HTML page by using following techniques
  - o JavaScript can be inline
  - o JavaScript can be embedded.
  - o JavaScript can be from external file.

## JavaScript Inline:

- JavaScript functions are defined within the element.
- They are faster as they are local to element.
- They can't be re-used.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
 <head>
```

```
 <title>Inline</title>
```

```
 </head>
```

```
 <body>
```

```
 <h2>Click Print Button to Print Page</h2>
```

```
 <button onclick="window.print()">Print Page</button>
```

```
 </body>
```

</html>

### JavaScript Embedded:

- You can write the JavaScript functions and embed into page by using <script> element.
- You can embed in head section or body section.
- You can re-use the functions across the page from any element.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
 <head>
```

```
 <title>Inline</title>
```

```
 <script>
```

```
 function PrintPage(){
```

```
 window.print();
```

```
 }
```

```
 </script>
```

```
 </head>
```

```
 <body>
```

```
 <h2>Click Print Button to Print Page</h2>
```

```
 <button onclick="PrintPage()">Print Page</button>
```

```
 <button onclick="PrintPage()">Print</button>
```

```
 </body>
```

```
</html>
```

- Embedded scripts require **MIME type** to define.
- The JavaScript MIME type is "text/javascript" or "language=JavaScript"

Syntax:

```
<script type="text/javascript" language="javascript">

 function PrintPage(){

 window.print();

 }

</script>
```

### JavaScript Strict Mode:

- JavaScript is recommended to write in “Strict” mode.
- It reduces the code inconsistency.
- You can turn ON strict mode by using “use strict” in the code.

Ex:

```
<script>
 "use strict";
 function f1()
 {
 x = 10; // x is not declared as variable
 document.write("x=" + x);
 }
 f1();
</script>
```

Note: remove “use strict” from <script> element, the above code will work normally. In strict mode you have to declare a variable “var x”.

### Download ESLint for Visual Studio Code

#### JavaScript from External File

- JavaScript functions are maintained in a separate script file with extension “.js”.
- You can link the script file to any HTML page by using <script> element.
- You can re-use the function across multiple pages.
- It will increase the page load time.

Ex:

- Create a new folder “scripts”

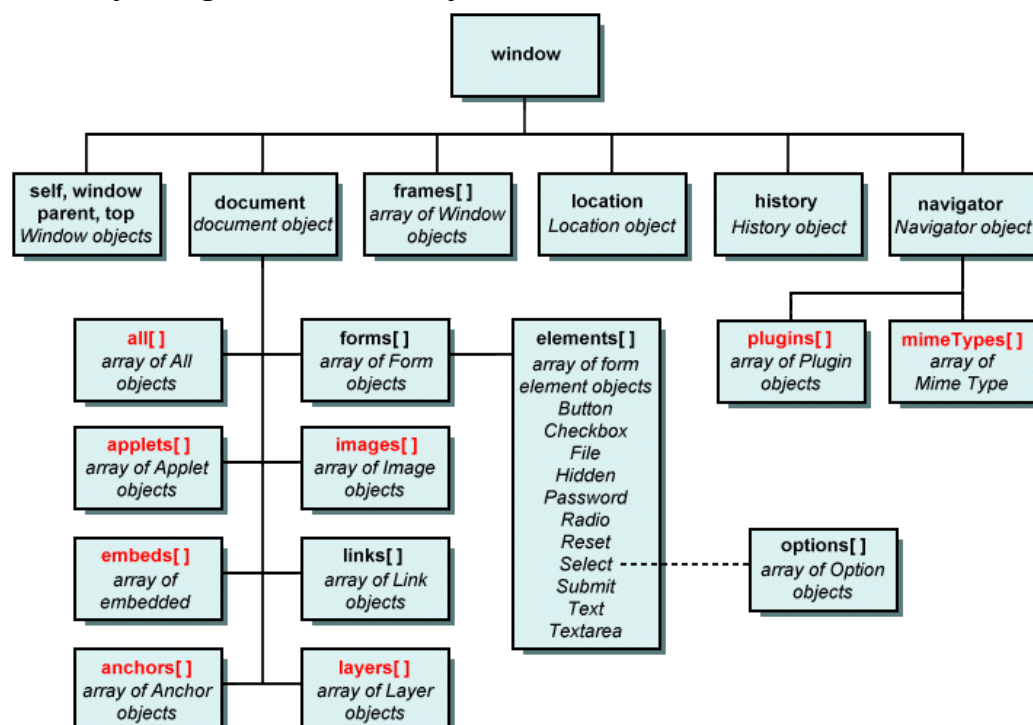
- Add a new file into folder by name "printing.js"
 

```
"use strict";
function PrintPage()
{
 window.print();
}
```
- Link to HTML page
 

```
<!DOCTYPE html>
<html>
 <head>
 <title>External File</title>
 <script src="../Scripts/printing.js"></script>
 </head>
 <body>
 <h2>Print Page by clicking the print button</h2>
 <button onclick="PrintPage()">Print Page</button>
 </body>
</html>
```

## How JavaScript Refers HTML DOM Elements?

### 1. Refer by using DOM hierarchy





Ex: window.document.images[index]  
window.document.forms[index].elements[index]

- Every time when you change the position of any DOM element, you have to update the index in logic.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Reference</title>
```

```
<script type="text/javascript">
```

```
function bodyload(){
```

```
 window.document.images[0].src="../Images/shoe.jpg";
```

```
 window.document.forms[0].elements[1].value = "Register";
```

```
}
```

```
</script>
```

```
</head>
```

```
<body onload="bodyload()">
```

```
<div>
```

```

```

```
</div>
```

```
<div>
```

```
<h2>Register</h2>
```

```
<form>
```

```
User Name:
```

```
<input type="text">
```

```
<input type="button">
```

```
</form>
```

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

## 2. Refer by using “name”

- Every element can have a reference name.
- Name can be common for multiple elements.
- If you are referring a child element, it is mandatory that you have to refer the parent element.

Ex:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Reference</title>
 <script type="text/javascript">
 function bodyload(){
 pic.src="../Images/shoe.jpg";
 frmRegister.btnRegister.value = "Register";
 }
 </script>
 </head>
 <body onload="bodyload()">
 <div>

 </div>
 <div>
 <h2>Register</h2>
 <form name="frmRegister">
```

User Name:

```
<input name="txtName" type="text">
```

```
<input name="btnRegister" type="button">
```

```
</form>
```

```
</div>
```

```
</body>
```

```
</html>
```

### 3. Refer by using ID

- Every element can be defined with unique ID.
- You can access element by using the following document method

**document.getElementById()**

- You can access any element directly with ID.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Reference</title>
```

```
<script type="text/javascript">
```

```
function bodyload(){
```

```
 document.getElementById("pic").src="../Images/shoe.jpg";
```

```
 document.getElementById("btnRegister").value = "Register";
```

```
}
```

```
</script>
```

```
</head>
```

```
<body onload="bodyload()">
```

```
<div>
```

```

```

```
</div>
<div>
 <h2>Register</h2>
 <form id="frmRegister">
 User Name:
 <input id="txtName" type="text">
 <input id="btnRegister" type="button">
 </form>
</div>
</body>
</html>
```

#### 4. Access elements by Tag Name

getElementsByTagName()

**Ex:**

```
<!DOCTYPE html>
<html>
 <head>
 <title>Reference</title>
 <script type="text/javascript">
 function bodyload(){
 x = document.getElementsByTagName("img")
 alert("Total Number of Images: " + x.length);
 }
 </script>
 </head>
 <body onload="bodyload()">
```

```
<div>

</div>

<div>

 <h2>Register</h2>

 <form id="frmRegister">

 User Name:

 <input id="txtName" type="text">

 <input id="btnRegister" type="button">

 </form>

</div>

</body>

</html>
```

## 5. Access elements by class Name

Every element can have multiple classes.

You can access element by using their class name.

Ex:

```
<!DOCTYPE html>

<html>

<head>

 <style>

 .effects {

 background-color: yellow;

 }

 </style>

 <title>Reference</title>

 <script type="text/javascript">
```

```

function bodyload(){
 x = document.getElementsByClassName("effects");
 alert("Total Count: " + x.length);
}
</script>
</head>
<body onload="bodyload()">
 <div>

 </div>
 <div>
 <h2>Register</h2>
 <form id="frmRegister">
 User Name:
 <input class="effects" id="txtName" type="text">
 <input class="effects" id="btnRegister" type="button">
 </form>
 </div>
</body>
</html>

```

## 6. Access elements that have common name

Ex:

```

<!DOCTYPE html>
<html>

```

```
<head>
 <style>
 .effects {
 background-color: yellow;
 }
 </style>
 <title>Reference</title>
 <script type="text/javascript">
 function bodyload(){
 x = document.getElementsByName("pay");
 alert("Total Payment Methods: " + x.length);
 }
 </script>
</head>
<body onload="bodyload()">
 <fieldset>
 <legend>Payment</legend>
 <input type="radio" name="pay" value="Cash"> Cash
 <input type="radio" name="pay" value="UPI"> UPI
 <input type="radio" name="pay" value="Credit Card"> Credit Card

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

## JavaScript Output and Input Methods

## JavaScript Output Method and Properties:

- alert()
- confirm()
- document.write()
- innerHTML
- innerText
- outerHTML
- console.log()
- console.error()
- console.warn()
- console.debug()
- console.info()

### alert():

It is used to display output in a message box.

User have to confirm the message in order to continue.

### Syntax:

```
alert("message/ expression");
```

You can't cancel alert. "esc" key is similar to OK.

EX:

```
<!DOCTYPE html>
```

```
<html>
```

```
 <head>
```

```
 <style>
```

```
 .effects {
```

```
 background-color: yellow;
```

```
 }
```

```
 </style>
```



```
<title>Reference</title>
<script type="text/javascript">
 function DeleteClick(){
 alert("Record Deleted");
 }
</script>
</head>
<body>
 <button onclick="DeleteClick()">Delete</button>
</body>
</html>
```

### **confirm():**

It is similar to alert, but provides option to cancel the message.

It returns **true** on OK and **false** on Cancel.

Ex:

```
<!DOCTYPE html>
<html>
 <head>
 <style>
 .effects {
 background-color: yellow;
 }
 </style>
 <title>Reference</title>
```

```
<script type="text/javascript">
function DeleteClick(){
 x = confirm("Are you sure want to Delete?");
 if(x==true){
 alert("Record Deleted");
 } else {
 alert("Canceled..");
 }
}
</script>
</head>
<body>
 <button onclick="DeleteClick()">Delete</button>
</body>
</html>
```

### **document.write()**

It is used to print output in a new screen of same page.

The output is erased when you refresh the page.

EX:

```
<!DOCTYPE html>
<html>
 <head>
 <style>
 .effects {
 background-color: yellow;
```

```
 }
</style>
<title>Reference</title>
<script type="text/javascript">
 function DeleteClick(){
 x = confirm("Are you sure want to Delete?");
 if(x==true){
 document.write("Record Deleted");
 } else {
 alert("Canceled..");
 }
 }
</script>
</head>
<body>
 <button onclick="DeleteClick()">Delete</button>
</body>
</html>
```

### **innerHTML & innerText**

It is used to define output in containers like div, span, dd, dt, td, p, blockquote.

**innerText** is an RC data element. [Text without formats]

**innerHTML** supports formats.

Ex:

```
<!DOCTYPE html>
```

```

<html>
 <head>
 <style>
 .effects {
 background-color: yellow;
 }
 </style>
 <title>Reference</title>
 <script type="text/javascript">
 function DeleteClick(){
 x = confirm("Are you sure want to Delete?");
 if(x==true){
 document.getElementById("msg").innerText="Record Deleted";
 } else {
 alert("Canceled..");
 }
 }
 </script>
 </head>
 <body>
 <button onclick="DeleteClick()">Delete</button>
 <p align="center" id="msg"></p>
 </body>
</html>

```

Ex:

```

<!DOCTYPE html>
<html>
 <head>
 <style>
 .effects {
 background-color: yellow;
 }
 </style>
 <title>Reference</title>
 <script type="text/javascript">
 function DeleteClick(){
 x = confirm("Are you sure want to Delete?");
 if(x==true){
 document.getElementById("msg").innerHTML="Record Deleted";
 } else {
 alert("Canceled..");
 }
 }
 </script>
 </head>
 <body>
 <button onclick="DeleteClick()">Delete</button>
 <p align="center" id="msg"></p>
 </body>
</html>

```

**innerHTML** will add inside existing element.

**outerHTML** will add by replacing existing element.

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
 <head>
```

```
 <style>
```

```
 .effects {
```

```
 background-color: yellow;
```

```
 }
```

```
 </style>
```

```
 <title>Reference</title>
```

```
 <script type="text/javascript">
```

```
 function DeleteClick(){
```

```
 x = confirm("Are you sure want to Delete?");
```

```
 if(x==true){
```

```
 document.getElementById("msg").outerHTML="<h2><font
color='red'>Record Deleted</h2>";
```

```
 } else {
```

```
 alert("Canceled..");
```

```
 }
```

```
 }
```

```
 </script>
```

```
 </head>
```

```
 <body>
```

```
 <button onclick="DeleteClick()">Delete</button>
```

```
 <p align="center" id="msg"></p>
```

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

### Console options

```
<!DOCTYPE html>
<html>
 <head>
 <style>
 .effects {
 background-color: yellow;
 }
 </style>
 <title>Reference</title>
 <script type="text/javascript">
 function DeleteClick(){
 x = confirm("Are you sure want to Delete?");
 console.info("Delete Initiated");
 if(x==true){
 console.error("Delete Completed");
 document.getElementById("msg").outerHTML="<h2>Record Deleted</h2>";
 } else {
 alert("Canceled..");
 console.warn("Canceled action");
 }
 }
 </script>
```

```
</head>

<body>

 <button onclick="DeleteClick()">Delete</button>

 <p align="center" id="msg"></p>

</body>

</html>
```

## JavaScript Input Properties

- JavaScript can handle input from user by using
  - prompt()
  - Form Input Elements

### Prompt():

- It opens a dialog in browser and allows user to input value.
- User can input any type of value like text, number, boolean etc. but not complex values like images, binary type data etc.
- Prompt is used when input is not regular.

Syntax:

```
prompt("Message", "Default Value");
```

- Prompt returns following
  - null: When it is cancelled with or without value.
  - Empty: When you click OK without value.
  - Value: When you click OK with value.

### Ex:

```
<!DOCTYPE html>

<html>

 <head>

 <title>Input</title>

 <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
```



```

<script>
 function CreateClick(){
 folderName = prompt("Enter Folder Name:", "New_Folder");
 if(folderName==null) {
 document.write("You cancelled");
 } else if (folderName=="") {
 document.write("Name can't be empty");
 } else {
 document.getElementById("msg").innerHTML+= "Folder Created : " +
folderName + "
";
 }
 }
 function bodyload(){
 username = prompt("Your Name:");
 document.getElementById("heading").innerHTML="Hello ! " +
username;
 }
</script>
</head>
<body class="container-fluid" onload="bodyload()">
 <h2 id="heading"></h2>
 <div class="form-group">
 <button onclick="CreateClick()" class="btn btn-primary">Create
Folder</button>
 </div>
 <div class="form-group">
 <p id="msg"></p>

```

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

### **Input using Form Elements:**

- HTML form provide several elements that allow the user to input a value.
- The commonly used elements for input
  - Textbox
  - Checkbox
  - RadioButton
  - ListBox
  - Dropdown etc.

### **Ex:**

```
<!DOCTYPE html>
<html>
 <head>
 <title>Form input</title>
 <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
 <script>
 function RegisterClick(){
 document.getElementById("lblName").innerHTML =
document.getElementById("txtName").value;
 document.getElementById("lblPrice").innerHTML =
document.getElementById("txtPrice").value;
 document.getElementById("lblMfd").innerHTML =
document.getElementById("txtMfd").value;
 document.getElementById("lblShippedTo").innerHTML =
document.getElementById("lstShippedTo").value;
```

```
stock = document.getElementById("optStock");
status = "";
if(stock.checked) {
 status = "Available";
} else {
 status = "Out of Stock";
}
document.getElementById("lblStock").innerHTML = status;
}
</script>
</head>
<body class="container-fluid">
 <div class="row">
 <div class="col-3">
 <h3>Register Product</h3>
 <div class="form-group">
 <label>Name</label>
 <div>
 <input type="text" id="txtName" class="form-control">
 </div>
 </div>
 <div class="form-group">
 <label>Price</label>
 <div>
 <input type="text" id="txtPrice" class="form-control">
 </div>
 </div>
 </div>
 </div>
</body>
</html>
```

```
</div>
<div class="form-group">
 <label>Manufactured</label>
 <div>
 <input type="date" id="txtMfd" class="form-control">
 </div>
</div>
<div class="form-group">
 <label>Shipped To</label>
 <div>
 <select class="form-control" id="lstShippedTo">
 <option>Delhi</option>
 <option>Hyderabad</option>
 </select>
 </div>
</div>
<div class="form-group">
 <label>In Stock</label>
 <div>
 <input type="checkbox" id="optStock"> Yes
 </div>
</div>
<div class="form-group">
 <button onclick="RegisterClick()" class="btn btn-primary btn-block">Register</button>
</div>
</div>
```

```
<div class="col-9">

 <h3>Product Details</h3>

 <table class="table table-hover">

 <colgroup span="1" style="font-weight: bold; background-color:
lightpink; color:white"></colgroup>

 <tr>

 <td>Name</td>

 <td id="lblName"></td>

 </tr>

 <tr>

 <td>Price</td>

 <td id="lblPrice"></td>

 </tr>

 <tr>

 <td>Manufactured</td>

 <td id="lblMfd"></td>

 </tr>

 <tr>

 <td>Shipped To</td>

 <td id="lblShippedTo"></td>

 </tr>

 <tr>

 <td>Stock </td>

 <td id="lblStock"></td>

 </tr>

 </table>

</div>
```

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

Ex:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Input</title>
 <style>
 label {
 font-weight: bold;
 }
 .box{
 width: 300px;
 justify-content: center;
 align-items: center;
 margin: auto;
 }
 </style>
 <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
 <script src="../node_modules/jquery/dist/jquery.js"></script>
 <script
src="../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
 <script>
 function RegisterClick(){
```

```
 document.getElementById("lblName").innerText =
document.getElementById("txtName").value;
```

```
 document.getElementById("lblPrice").innerText =
document.getElementById("txtPrice").value;
```

```
 document.getElementById("lblCity").innerText =
document.getElementById("lstCities").value;
```

```
 stock = document.getElementById("optStock");
 status = "";
 if(stock.checked){
 status = "Available";
 } else {
 status = "Out of Stock";
 }
 document.getElementById("lblStock").innerText = status;
 document.getElementById("lblMfd").innerText =
document.getElementById("txtMfd").value;
 }
}
```

```
</script>
```

```
</head>
```

```
<body class="container-fluid">
```

```
<div class="box">
```

```
<form>
```

```
<h2 class="text-primary text-center">Register Product</h2>
```

```
<div class="form-group">
```

```
<label>Name</label>
```

```
<div>
```

```
 <input type="text" id="txtName" class="form-control">
 </div>
</div>
<div class="form-group">
 <label>Price</label>
 <div>
 <input type="text" id="txtPrice" class="form-control">
 </div>
</div>
<div class="form-group">
 <label>Shipped To</label>
 <div>
 <select id="lstCities" class="form-control">
 <option value="Delhi">Delhi</option>
 <option value="Hyderabad">Hyderabad</option>
 </select>
 </div>
</div>
<div class="form-group">
 <label>Manufactured</label>
 <div>
 <input type="date" id="txtMfd" class="form-control">
 </div>
</div>
<div class="form-group">
 <label>In Stock</label>
```



```
<div>
 <input type="checkbox" id="optStock"> Yes
</div>
</div>
<div class="form-group">
 <button onclick="RegisterClick()" type="button" data-
target="#summary" data-toggle="modal" class="btn btn-primary btn-
block">Register</button>
</div>
</form>
</div>
<div class="modal fade" id="summary">
 <div class="modal-dialog">
 <div class="modal-content">
 <div class="modal-header">
 <h2>Product Details</h2>
 <button class="close" data-dismiss="modal" >x</button>
 </div>
 <div class="modal-body">
 <table class="table table-hover">
 <tr>
 <td>Name</td>
 <td id="lblName"></td>
 </tr>
 <tr>
 <td>Price</td>
 <td id="lblPrice"></td>
 </tr>
 </table>
 </div>
 </div>
 </div>
</div>
```

```

 </tr>
 <tr>
 <td>Shipped To</td>
 <td id="lblCity"></td>
 </tr>
 <tr>
 <td>Stock</td>
 <td id="lblStock"></td>
 </tr>
 <tr>
 <td>Manufactured</td>
 <td id="lblMfd"></td>
 </tr>
 </table>
</div>
<div class="modal-footer">
 <button data-dismiss="modal" class="btn btn-
primary">OK</button>
</div>
</div>
</div>
</div>
</body>
</html>

```

## JavaScript Language Basics

- Variables
- Data Types

- Operators
- Statements
- Functions

## Variables in JavaScript

- Variables are storage locations in memory where you can store a value and use it as a part of any expression.
- Variables configuration contains 3 stages
  - Declaration
  - Assigning or Rendering
  - Initialization

**var x = 10;**  
 └─── Declaring    Initialization  
 └───  
**var y;**  
 └───  
**y = 20;**  
 └─── Rendering  
      Assigning

- Declaring variables in JavaScript is not mandatory if JavaScript is not in strict mode.
- You can directly use rendering without declaring.

Ex:

```

<script>
 function f1(){
 x = 10; // valid
 document.write(`X=${x}`);
 }
 f1();
</script>

```

- Declaring variables mandatory if JavaScript is in strict mode.

Ex:

```
<script>
 "use strict";
 function f1(){
 x = 10; // invalid
 document.write(`X=${x}`);
 }
 f1();
</script>
```

- JavaScript variables can be declared by using following keywords
  - var
  - let
  - const

### Var:

- Var is used to defined a variable with function scope.
- You can declare variable in any block of a function and access anywhere in the function.
- It allows declaring, rendering and initialization.

Ex:

```
<script>
 "use strict";
 function f1(){
 var x = 10; // Initialization
 if(x==10) {
 var y; // Declaration
 y = 20; // Rendering
 }
 document.write("X=" + x + "
" + "Y=" + y);
 }
 f1();
</script>
```

- **Var supports shadowing.**
- Shadowing is the process of re-declaring same name identifier in the block.

Ex:

```
<script>
 "use strict";
```

```

function f1(){
 var x = 10;
 if(x==10) {
 var x; // shadowing
 x = 30;
 var y;
 y = 20;
 }
 document.write("X=" + x + "
" + "Y=" + y);
}
f1();
</script>

```

- **Var also supports “Hoisting”**
- Hoisting is a mechanism allowed for compiler or interpreter, so that you use a variable before declaring.

Ex:

```

<script>
 "use strict";
 function f1(){
 x = 10;
 document.write("X=" + x);
 var x; //Hoisting
 }
 f1();
</script>

```

**LET:**

- It used to define block scoped variable.
- You can access only within the declared block and child block.
- You can't access outside the block.

Ex:

```

<script>
 "use strict";
 function f1(){
 let x = 10;
 if(x==10) {
 let y;
 y = 20;
 }
 }

```

```

 }
 document.write("X=" + x + "
" + "Y=" + y); // Error: y not
defined.

```

```

 }
 f1();
</script>

```

Ex:

```

<script>
 "use strict";
 function f1(){
 let x = 10;
 if(x==10) {
 let y;
 y = 20;
 document.write("X=" + x + "
" + "Y=" + y); // valid
 }
 }

```

```

 }
 f1();
</script>

```

- **Let allows declaration, rendering, initialization.**

Ex:

```

<script>
 "use strict";
 function f1(){
 let x = 10; // intialization
 if(x==10) {
 let y; // declaration
 y = 20; // rendering
 document.write("X=" + x + "
" + "Y=" + y);
 }
 }

```

```

 }
 f1();
</script>

```

- Let will not allow shadowing.
- There can't be same name identifier with in the scope.

```

<script>
 "use strict";
 function f1(){
 let x = 10; // initialization
 if(x==10) {
 let x = 20;
 let x = 30; // invalid – shadowing not
allowed
 let y; // declaration
 y = 20; // rendering
 document.write("X=" + x + "
" + "Y=" + y);
 }

 }
 f1();
</script>

```

- **Let will not allow hoisting.**

Ex:

```

<script>
 "use strict";
 function f1(){
 x = 10;
 document.write("X=" + x);
 let x; // Not Allowed
 }
 f1();
</script>

```

### **const:**

- It is also a block scope variable.
- It will allow only initialization, no declaration, no rendering.
- It will not allow shadowing.
- It will not allow hoisting.

Ex:

```

<script>

```

```

"use strict";

function f1(){

 const x; // not allowed

 x= 10;

 document.write("x=" + x);

}

f1();

</script>

```

### Global Scope for Variables:

- You can declare any variable outside the function to make it global in access.
- You can use var, let or const.
- If you declare variables outside the function the it is referred as module scope.
- You can import a module into another module and access its variables.

### Ex:

```

<script>

"use strict";

var x = 10;

let y = 20;

const z = 30;

function f1(){

 document.write(`f1 values:
 x=${x}
y=${y}
z=${z}
`);

}

function f2(){

 document.write(`f2 values:
 x=${x}
y=${y}
z=${z}`);

}

```



```
f1();
f2();
</script>
```

**FAQ: Can we declare a variable inside function and make it global in access?**

**A. Yes. You can declare variable by using “window” object to make it global.**

**Ex:**

```
<script>
"use strict";
var x = 10;
let y = 20;
const z = 30;
function f1(){
 window.a = 40; //window refers to browser
 document.write(`f1 values:

x=${x}
y=${y}
z=${z}
a=${a}
`);
}
function f2(){
 document.write(`f2 values:
 x=${x}
y=${y}
z=${z}
a=${a}`);
}
f1();
f2();
</script>
```

### **Variables Naming**

- Variable name must start with an alphabet or underscore \_
- It can be alpha numeric, but can't start with number.

Syntax:

```
var jan2020 = 12000;
var 2020jan = 23000; invalid
var _2020jan = 34000; valid
var jan_2020 = 34000; valid
```

### **FAQ: Why underscore is supported? What underscore means?**

- A. Underscore is special symbol that doesn't have any compile time issues. underscore is used by developers to indicate that the variable requires further implementation.

“Marked for Implementation”

- ECMA standards prefer variable name length maximum 255 chars.
- Variable name must speak what it is.
- Always use camel case for naming references.

```
class EmployeeSalary
```

```
{

}
```

```
var employeeSalary = new Employee();
```

```
var txtName;
```

```
var txtPassword;
```

```
var btnSubmit;
```

### **Data Types**

- Data type determines the data structure.
- Data structure specifies the size and behaviour of value in memory.
- Data Type uses a data structure to define the type of value that can be stored in memory.
- JavaScript is implicitly typed; the data type is determined according to the value assigned. There is no specific built-in type.
- JavaScript is not strongly typed, you can store any contradictory values.

Syntax:

```
var x = "John"; //string
x = 10; // valid and x changes to number
```

What type of values JavaScript can handle?

The JavaScript data types are classified into 2 groups

- Primitive Types
- Non-Primitive Types

### **Primitive Types**

- Primitive types are stored in memory stack. [LIFO]
- They have a fixed range for values.
- They are Immutable types.
- Their values can't change according to state and situation.
- JavaScript primitive types are
  - o Number
  - o String
  - o Boolean
  - o Null
  - o Undefined

### **Number Type:**

- Number type is used to handle a numeric value.
- JavaScript number type can allow
  - o Signed Integer  
var x = -10;  
var x = +10;
  - o Unsigned Integer  
var x = 10;
  - o Floating point  
var x = 4.5;  
var x = 33.55;
  - o Double  
var x = 553.558
  - o Decimal  
var x = 45600.6669594; [29 decimal places]
  - o Hexa

0xf00d

- Binary

0b1010

- Octa

0o744

- Exponent

Var x = 2e3;  $[2 \times 10^3] = 2000$

BigInt = 100n; [Binary Data – images - complex]

Ex:

```
<script>
```

```
"use strict";
```

```
function f1(){
```

```
 document.write(`Min Integer: ${Number.MIN_SAFE_INTEGER}
 Max
Integer: ${Number.MAX_SAFE_INTEGER}`);
```

```
}
```

```
f1();
```

```
</script>
```

### Validating Numbers

- The operator “typeof” is used to verify and return the data type of variable.
- isNaN() is a JavaScript that verifies whether the give value is a number or any other type.
- Every value you entered in form element will be “string” type.
- You have to convert the string into number by using the functions
  - parseInt()
  - parseFloat()

Ex:

```
<script>
```

```
var x = 10;
```

```
var y = "4";
```

```
if(isNaN(y)){
```

```
 document.write("Invalid Number");
```

```

} else {
var z = x * y;
document.write("z=" + z);
}
</script>

```

**Ex:**

```

<script>
var x = prompt("Enter X Value");
var y = prompt("Enter Y Value");
if(isNaN(y)){
 document.write("Invalid Number");
} else {
var z = x * y;
document.write("z=" + z);
}

</script>

```

## String Type

- String is a literal with group of characters enclosed in Quotes.
- JavaScript string can be enclosed in
  - Single Quote      ‘      ‘
  - Double Quote      “      “
  - Back Tick      `      `
- Single and double quotes are used to swap between inner and outer string.

**Ex:**

```

<script>

"use strict";

```

```
function f1(){
 var link = "Home";
 document.write(link);
}
f1();
</script>
```

Ex:

```
<script>
 "use strict";
 function f1(){
 var link = 'Home';
 document.write(link);
 }
 f1();
</script>
```

### **Back Tick:**

- Back Tick [ ` ` ] is available from ES5
- It is used to define a string with embedded expression.
- Expression can be embedded with “\${}”
- Expression can't be embedded into string with single or double quote.

Ex:

```
<script>
 "use strict";
 function f1(){
 var age = 20;
```

```

var year = 2020;

document.write("You will be" + " " + (age+1) + " " + "Next Year" + " " +
(year+1) + "
");

document.write(`You will be ${age+1} Next Year ${year+1}`);
}

f1();
</script>

```

Ex:

```

<script>

"use strict";

function f1(){

var title = "Admin Login";

var login = `

<h2>${title}</h2>

<dl>

<dt>Name</dt>

<dd><input type="text"></dd>

<dt>Price</dt>

<dd><input type="text"></dd>

</dl>

<button>Login</button>

`;

document.write(login);

}

f1();
</script>

```

- Several special characters defined in a string will escape printing.
- To print the non-printable characters, we have to use “\”.

Ex:

```
<script>
 "use strict";

 function f1(){
 var path = "\"D:\\Images\\Pics\\mobile.jpg\"";
 document.write(path);
 }

 f1();
</script>
```

Ex:

```
<script>
 "use strict";

 function f1(){
 alert("Hello \n Welcome \n to \n JavaScript");
 document.write("Hello !
 Welcome");
 }

 f1();
</script>
```

Note: The numbers or the values that you access from any element are string type. You have to convert the string into number to handle expressions. JavaScript functions to convert string into number.

- parseInt()
- parseFloat()



Ex:

```
<!DOCTYPE html>

<html>

<head>

 <title>String</title>

 <script>

 function Calculate(){

 var txt1 = document.getElementById("txt1").value;

 var txt2 = document.getElementById("txt2").value;

 document.getElementById("result").innerHTML = parseFloat(txt1) +
parseFloat(txt2);

 }

 </script>

</head>

<body>

 <dl>

 <dt>Number-1</dt>

 <dd><input id="txt1" type="text"></dd>

 <dt>Number-2</dt>

 <dd><input id="txt2" type="text"></dd>

 </dl>

 <button onclick="Calculate()">Calculate</button>

 <h2 id="result"></h2>

</body>

</html>
```

String Manipulation Functions

- JavaScript string object provides a set of properties and methods that are used to manipulate and format string.
- Manipulation methods

Method	Description
charAt()	<p>It returns the character as specified index.</p> <p>Syntax: string.charAt();</p> <p>Ex:</p> <pre>&lt;script&gt; function f1(){     var str = "Welcome to JavaScript";     var char1 = str.charAt(0);     var char2 = str[1]; //New in ES5     document.write(`Char1=\${char1}&lt;br&gt;Char2=\${char2}`) } f1(); &lt;/script&gt;</pre>
charCodeAt()	<p>It returns the character code of character at specified index. ASCII code of characters are accessed. A=65, Z=90</p>
indexOf()	Returns the first occurrence index number of specified characters.
lastIndexOf()	<p>Returns the last occurrence index number of specified char.</p> <p>If the character not found then both methods return “-1”.</p>
trim()	It is used to remove the leading spaces in a string.
substring()	<p>It can extract a portion of string based on specified index. It is similar to slice but will not allow negative values. You can access right to left by using positive value.</p> <p>Ex:</p> <pre>&lt;script&gt; function f1(){     var str = "Welcome to JavaScript";     document.write(str.substring(7,0)); } f1(); &lt;/script&gt;</pre>
substr()	It is a legacy method, will not allow the values right to left.

slice()	<p>It is used to extract a part of string and return a new string.</p> <p>Syntax:</p> <p>slice(startIndex, endIndex)</p> <p>slice(startIndex); slice upto end.</p> <p>Slice(-1); It returns the last character.</p> <p>Slice(-4); It returns the last 4 chars.</p>
split()	<p>It splits string at specific delimiter and returns an array of substrings.</p> <p>You can also restrict the number of items to split.</p> <p>Syntax:</p> <p>String.split('delimiter', count)</p> <p>Ex:</p> <pre>&lt;script&gt; function f1(){     var mobiles = "9876543210,9988776655,9008833113";     var numbers = mobiles.split(',', 2);     for(var number of numbers) {         document.write(number + "&lt;br&gt;");     } } f1(); &lt;/script&gt;</pre>
startsWith()	It returns true if string starts with specified chars.
endsWith()	<p>It returns true if string ends with specified chars.</p> <p>Ex:</p> <pre>&lt;!DOCTYPE html&gt; &lt;html&gt;     &lt;head&gt;         &lt;title&gt;String&lt;/title&gt;         &lt;script&gt;             function Verify(){                 var txtEmail = document.getElementById("txtEmail").value;                 if(txtEmail.endsWith("gmail.com")) {                     document.write("Your Gmail Verified..");                 } else {                     document.write("Only Gmail allowed");                 }             }         &lt;/script&gt;     &lt;/head&gt; &lt;/html&gt;</pre>

	<pre> &lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;fieldset&gt;     &lt;legend&gt;Your Email&lt;/legend&gt;     &lt;input type="text" id="txtEmail" placeholder="Only Gmail Allowed"&gt;     &lt;button onclick="Verify()"&gt;Submit&lt;/button&gt;   &lt;/fieldset&gt; &lt;/body&gt; &lt;/html&gt; </pre>
search()	<p>You can search for any char using a regular expression. It returns the index number of searched string. If character not found then it returns -1</p> <p>Ex:</p> <pre> &lt;script&gt;   function f1(){     var str = "Welcome to JavaScript";     document.write(str.search(/javascript/i));   }   f1(); &lt;/script&gt; </pre>

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>String Demo</title>
```

```
<link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
<script>
```

```
function VerifyName(){
```

```
 var txtName = document.getElementById("txtName").value;
```

```

var msg = document.getElementById("msg");
var firstCharCode = txtName.charCodeAt(0);
if(firstCharCode>=65 && firstCharCode<=90) {
 msg.innerHTML = "";
} else {
 msg.innerHTML = "Name must start with Uppercase Letter";
}
}
function VerifyCard(){
 var txtCard = document.getElementById("txtCard").value;
 var firstChar = txtCard.charAt(0);
 var cardLogo = document.getElementById("cardLogo");
 if(firstChar=="4") {
 cardLogo.src="../Images/visa.png";
 } else if(firstChar=="5"){
 cardLogo.src="../Images/master.png";
 } else {
 cardLogo.src="../Images/invalid.png";
 }
}
</script>
</head>
<body class="container-fluid">
 <div class="form-group">
 <label>User Name</label>
 <div>

```

```

 <input onblur="VerifyName()" class="form-control"
placeholder="Name must start with Uppercase Letter" type="text"
id="txtName">

 <div id="msg" class="text-danger"></div>

 </div>

</div>

<div class="form-group">

 <label>Card Number</label>

 <div class="input-group">

 <input onkeyup="VerifyCard()" type="text" id="txtCard" class="form-
control">

 <div class="input-group-append">

 </div>

 </div>

</div>

</div>

</body>

</html>

```

Ex:

```

<!DOCTYPE html>

<html>

 <head>

 <title>String Demo</title>

```

```
<link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
<script>
```

```
function VerifyName(){
```

```
 var txtName = document.getElementById("txtName").value;
```

```
 var msg = document.getElementById("msg");
```

```
 var firstCharCode = txtName.charCodeAt(0);
```

```
 if(firstCharCode>=65 && firstCharCode<=90) {
```

```
 msg.innerHTML = "";
```

```
 } else {
```

```
 msg.innerHTML = "Name must start with Uppercase Letter";
```

```
 }
```

```
}
```

```
function VerifyCard(){
```

```
 var txtCard = document.getElementById("txtCard").value;
```

```
 var firstChar = txtCard.charAt(0);
```

```
 var cardLogo = document.getElementById("cardLogo");
```

```
 if(firstChar=="4") {
```

```
 cardLogo.src="../Images/visa.png";
```

```
 } else if(firstChar=="5"){
```

```
 cardLogo.src="../Images/master.png";
```

```
 } else {
```

```
 cardLogo.src="../Images/invalid.png";
```

```
 }
```

```
}
```

```
function VerifyEmail(){
```

```
 var txtEmail = document.getElementById("txtEmail").value;
```

```

 var emailError = document.getElementById("emailError");
 var atPos = txtEmail.indexOf("@");
 var dotPos = txtEmail.lastIndexOf(".");
 if(atPos<=2 && (dotPos-atPos)<=2) {
 emailError.innerHTML = "Error: @ missing or not at valid position in
email";
 } else {
 emailError.innerHTML = "Email Verified";
 }

 }
</script>
</head>
<body class="container-fluid">
 <div class="form-group">
 <label>User Name</label>
 <div>
 <input onblur="VerifyName()" class="form-control"
placeholder="Name must start with Uppercase Letter" type="text"
id="txtName">
 <div id="msg" class="text-danger"></div>
 </div>
 </div>
 <div class="form-group">
 <label>Card Number</label>
 <div class="input-group">

```



```

 <input onkeyup="VerifyCard()" type="text" id="txtCard" class="form-
control">
 <div class="input-group-append">

 </div>
 </div>
</div>
<div class="form-group">
 <label>Email</label>
 <div>
 <input onblur="VerifyEmail()" id="txtEmail" type="text" class="form-
control">
 <div id="emailError">

 </div>
 </div>
 </div>
</div>
</body>
</html>

```

Ex:

```

<!DOCTYPE html>
<html>
 <head>
 <title>String Demo</title>

```

```
<link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
<script>
```

```
function VerifyName(){
```

```
 var txtName = document.getElementById("txtName").value;
```

```
 var msg = document.getElementById("msg");
```

```
 var firstCharCode = txtName.charCodeAt(0);
```

```
 if(firstCharCode>=65 && firstCharCode<=90) {
```

```
 msg.innerHTML = "";
```

```
 } else {
```

```
 msg.innerHTML = "Name must start with Uppercase Letter";
```

```
 }
```

```
}
```

```
function VerifyCard(){
```

```
 var txtCard = document.getElementById("txtCard").value;
```

```
 var firstChar = txtCard.charAt(0);
```

```
 var cardLogo = document.getElementById("cardLogo");
```

```
 if(firstChar=="4") {
```

```
 cardLogo.src="../Images/visa.png";
```

```
 } else if(firstChar=="5"){
```

```
 cardLogo.src="../Images/master.png";
```

```
 } else {
```

```
 cardLogo.src="../Images/invalid.png";
```

```
 }
```

```
}
```

```
function VerifyEmail(){
```

```
 var txtEmail = document.getElementById("txtEmail").value;
```

```

 var emailError = document.getElementById("emailError");
 var atPos = txtEmail.indexOf("@");
 var dotPos = txtEmail.lastIndexOf(".");
 if(atPos<=2 && (dotPos-atPos)<=2) {
 emailError.innerHTML = "Error: @ missing or not at valid position in
email";
 } else {
 emailError.innerHTML = "Email Verified";
 }
}
function VerifyPassword(){
 var txtPwd = document.getElementById("txtPwd").value;
 var pwdError = document.getElementById("pwdError");
 if(txtPwd.trim()=="john") {
 pwdError.innerHTML = "Verified";
 } else {
 pwdError.innerHTML = "Invalid Password";
 }
}
</script>
</head>
<body class="container-fluid">
 <div class="form-group">
 <label>User Name</label>
 <div>

```

```
 <input onblur="VerifyName()" class="form-control"
placeholder="Name must start with Uppercase Letter" type="text"
id="txtName">
```

```
 <div id="msg" class="text-danger"></div>
```

```
 </div>
```

```
</div>
```

```
<div class="form-group">
```

```
 <label>Password</label>
```

```
 <div>
```

```
 <input onblur="VerifyPassword()" type="password" id="txtPwd"
class="form-control">
```

```
 <div id="pwdError">
```

```
 </div>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
 <label>Card Number</label>
```

```
 <div class="input-group">
```

```
 <input onkeyup="VerifyCard()" type="text" id="txtCard" class="form-
control">
```

```
 <div class="input-group-append">
```

```

```

```

```

```

```

```
 </div>
```

```
</div>
```

```

</div>
<div class="form-group">
 <label>Email</label>
 <div>
 <input onblur="VerifyEmail()" id="txtEmail" type="text" class="form-
control">
 <div id="emailError">

 </div>
</div>
</div>
</body>
</html>

```

### Compare String and Verify Equality:

- To compare string in JavaScript you can use operators like
  - Greater Than >, >=
  - Less than <, <=
  - Equal =
  - Not-Equal !=
- JavaScript can also compare the string by using “localeCompare()” method.
- “localeCompare()” method returns 0 when matching and 1 on mismatch.  
Syntax:  
sourceString.localeCompare(targetString, 'lang', {options})

Ex:

```

<!DOCTYPE html>
<html>
 <head>

```

```
<title>Compare</title>

<script>

 function VerifyPassword(){

 var pwd = document.getElementById("txtPwd").value;

 var cPwd = document.getElementById("txtConfirm").value;

 var result = pwd.localeCompare(cPwd);

 var lblError = document.getElementById("lblError");

 lblError.innerHTML = (result==0)?"Password
Verified":"Password Mismatch";

 }

</script>

</head>

<body>

 <div>

 <label>Password</label>

 <div>

 <input type="password" id="txtPwd">

 </div>

 </div>

 <div>

 <label>Confirm Password</label>

 <div>

 <input onblur="VerifyPassword()" type="password" id="txtConfirm">

 <div id="lblError">

 </div>

 </div>

 </div>

</body>

</div>
```

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

### **Compare with Regular Expression:**

- The function “match()” is used to verify the value with regular expression.
- It returns true if value is matching with regular expression.

Syntax:

```
String.match(regularExpression); // true-false
```

Ex:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Verify Password</title>
 <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
 <style>
 progress {
 height: 50px;
 width: 100%;
 }
 </style>
 <script>
 function VerifyPassword(){
 var regExp = /(?!.*[A-Z])\w{4,10}/;
 var txtPwd = document.getElementById("txtPwd").value;
```

```
var lblMsg = document.getElementById("lblMsg");
var grade = document.getElementById("grade");
grade.style.display = "inline";

function GradeDisplay(min, max, value){
 grade.min = min;
 grade.max = max;
 grade.value = value;

}

if(txtPwd.match(regExp)){
 lblMsg.innerHTML="Strong Password";
 GradeDisplay(1,100,100);
} else {
 if(txtPwd.length<4) {
 lblMsg.innerHTML="Poor Password";
 GradeDisplay(1,100,20)
 } else {
 lblMsg.innerHTML="Weak Password";
 GradeDisplay(1,100,60);
 }
}

}

</script>
</head>
```



```

<body class="container-fluid">
 <h2>Regular Expression</h2>
 <div class="form-group">
 <label>Password</label>
 <div>
 <input onkeyup="VerifyPassword()" id="txtPwd" type="password"
class="form-control">
 <div>
 <progress min="1" max="100" style="display: none;"
id="grade"></progress>

 </div>
 </div>
</div>
</body>
</html>

```

### String Formatting Functions

- You can dynamically format a string by using following methods
  - bold()
  - italic()
  - fontsize()
  - fontcolor()
  - sup()
  - sub()
  - toUpperCase()
  - toLowerCase()

**Ex:**

```
<!DOCTYPE html>
```

```
<html>

<head>

 <title>Verify Password</title>

 <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <style>

 progress {

 height: 50px;

 width: 100%;

 }

 </style>

 <script>

 function VerifyPassword(){

 var regExp = /(?!.*[A-Z])\w{4,10}/;

 var txtPwd = document.getElementById("txtPwd").value;

 var lblMsg = document.getElementById("lblMsg");

 var grade = document.getElementById("grade");

 grade.style.display = "inline";

 function GradeDisplay(min, max, value){

 grade.min = min;

 grade.max = max;

 grade.value = value;

 }

 if(txtPwd.match(regExp)){
```

```

 lblMsg.innerHTML="Strong Password".fontcolor('green').bold();
 GradeDisplay(1,100,100);
 } else {
 if(txtPwd.length<4) {
 lblMsg.innerHTML="Poor Password".fontcolor('red').italics();
 GradeDisplay(1,100,20)
 } else {
 lblMsg.innerHTML="Weak
Password".fontcolor('yellow').bold().italics();
 GradeDisplay(1,100,60);
 }
 }
}
</script>
</head>
<body class="container-fluid">
 <h2>Regular Expression</h2>
 <div class="form-group">
 <label>Password</label>
 <div>
 <input onkeyup="VerifyPassword()" id="txtPwd" type="password"
class="form-control">
 <div>
 <progress min="1" max="100" style="display: none;"
id="grade"></progress>

 </div>

```

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

## Boolean Types

- Boolean types are defined by using “true or false”.
- Boolean type is used to handle decision making in programming.
- JavaScript boolean types refer to numeric value
  - o 0 : false
  - o 1 : true
- The boolean conditions in JavaScript can be configure with 0 or 1.
- JavaScript can control several HTML properties by using boolean type, which includes checked, selected, disabled etc.

### Syntax:

```
var x = true;

if(x==1) {
 statement on true;
} else {
 statement on false;
}
```

### Ex:

```
<!DOCTYPE html>

<html>

 <head>

 <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <script src="../node_modules/jquery/dist/jquery.js"></script>
```

```
<script
src="../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
```

```
<style>
```

```
body {
 background-color: darkred;
 width: 90%;
}
```

```
.card {
 padding: 20px;
}
```

```
</style>
```

```
<script>
```

```
function OrderClick(){
 document.getElementById("lblName").innerHTML =
document.getElementById("txtName").value;
 document.getElementById("lblMobile").innerHTML =
document.getElementById("txtMobile").value;
```

```
var mealName = "";
```

```
var mealCost = 0;
```

```
var adonName = "";
```

```
var adonCost = 0;
```

```
var total = 0;
```

```
var optBurger = document.getElementById("optBurger");
```

```
var optRoller = document.getElementById("optRoller");
```

```
var optFries = document.getElementById("optFries");
```

```
var optKrusher = document.getElementById("optKrusher");
```

```
if(optBurger.checked) {
 mealCost = 130;
 mealName = optBurger.value;
}
```

```
if(optRoller.checked) {
 mealCost = 100;
 mealName = optRoller.value;
}
```

```
if(optFries.checked) {
 adonCost = 80;
 mealCost = mealCost + adonCost;
 adonName += optFries.value + "
";
}
```

```
if(optKrusher.checked) {
 adonCost = 40;
 mealCost = mealCost + adonCost;
 adonName += optKrusher.value + "
";
}
```

```

 total = mealCost;

 document.getElementById("lblMeal").innerHTML = mealName;
 document.getElementById("lblAdon").innerHTML = adonName;
 document.getElementById("lblTotal").innerHTML = "₹" + total;
 }
</script>
</head>
<body class="container-fluid">
 <header>

 </header>
 <section>
 <div class="accordion" id="orderForm">
 <div class="card">
 <div class="card-header">
 <button data-target="#customerInfo" data-toggle="collapse"
class="btn btn-danger btn-block">Customer Info</button>
 </div>
 <div class="collapse show" id="customerInfo" data-
parent="#orderForm">
 <div class="form-group">
 <label>Customer Name</label>
 <div>
 <input type="text" id="txtName" class="form-control">
 </div>
 </div>
 <div class="form-group">

```

```

 <label>Mobile Number</label>

 <div>

 <input type="text" id="txtMobile" class="form-control">

 </div>

 </div>

</div>

</div>

<div class="card">

 <div class="card-header">

 <button data-target="#mealInfo" data-toggle="collapse" class="btn btn-danger btn-block">Select Meal</button>

 </div>

 <div class="collapse" id="mealInfo" data-parent="#orderForm">

 <div class="row">

 <div class="col">

 <div class="card">

 <div class="card-header">

 <h3>OMG Burger</h3>

 </div>

 <div class="card-body">

 </div>

 <div class="card-footer">

 <h4>

 <input name="meal" id="optBurger" value="OMG Burger" type="radio"> ₹8377; 130/-

 </h4>

```



```

 </div>
 </div>
</div>
<div class="col">
 <div class="card">
 <div class="card-header">
 <h3>OMG Roller</h3>
 </div>
 <div class="card-body">

 </div>
 <div class="card-footer">
 <h4>
 <input name="meal" id="optRoller" value="OMG Roller"
type="radio"> ₹ 100/-
 </h4>
 </div>
 </div>
</div>
</div>
</div>
<div class="card">
 <div class="card-header">
 <button data-target="#adonInfo" data-toggle="collapse" class="btn
btn-danger btn-block">Select Ad-ON</button>
 </div>

```

```

<div class="collapse" id="adonInfo" data-parent="#orderForm">
 <div class="row">
 <div class="col">
 <div class="card">
 <div class="card-header">
 <h3>Large Fries</h3>
 </div>
 <div class="card-body">

 </div>
 <div class="card-footer">
 <h4>
 <input id="optFries" value="Large Fries"
type="checkbox"> ₹ 80/-
 </h4>
 </div>
 </div>
 </div>
 <div class="col">
 <div class="card">
 <div class="card-header">
 <h3>Krusher Brownie</h3>
 </div>
 <div class="card-body">

 </div>
 <div class="card-footer">

```

```

 <h4>
 <input type="checkbox" id="optKrusher" value="Krusher
Browser"> ₹377; 40/-
 </h4>

 </div>
</div>

 <button onclick="OrderClick()" data-target="#billSummary" data-
toggle="modal" class="btn btn-danger btn-block">Place Order</button>
</div>
</div>
</div>
</div>
</section>
<div class="modal fade" id="billSummary">
 <div class="modal-dialog">
 <div class="modal-content">
 <div class="modal-header">
 <h3>Bill Summary</h3>
 <button data-dismiss="modal" class="close">x</button>
 </div>
 <div class="modal-body">
 <table class="table table-hover">
 <tbody>
 <tr>

```

```
 <td>Customer Name</td>
 <td id="lblName"></td>
 </tr>
 <tr>
 <td>Mobile</td>
 <td id="lblMobile"></td>
 </tr>
 <tr>
 <td>Meal Name</td>
 <td id="lblMeal"></td>
 </tr>
 <tr>
 <td>Ad-ONs</td>
 <td id="lblAdon"></td>
 </tr>
 <tr>
 <td>Total Amount</td>
 <td id="lblTotal"></td>
 </tr>
</tbody>
</table>
</div>
<div class="modal-footer">
 <button data-dismiss="modal" class="btn btn-
primary">OK</button>
</div>
</div>
```

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

## Undefined Type

- Undefined type is configured for variables that are not defined with value.
- Variable is defined but value is not assigned or initialized then the compile will configure as “undefined”.
- You can verify whether value defined or not by using undefined.

**Ex:**

```
<script>

function f1(){
 var x;

 if(x==undefined) {
 document.write("there is No value in x");
 } else {
 document.write(`x=${x}`);
 }
}

f1();
</script>
```

## Null Type

- Value is not defined into a reference dynamically during run time.
- Null is reference type, which indicates that value is not supplied to variable during run time.

**Ex:**

```
<script>

function f1(){
 var uname = prompt("Enter Name");
 if(uname==null) {
 document.write("You canceled");
 } else {
 document.write(`Hello ! ${uname}`);
 }
}

f1();
</script>
```

## Summary

- Number
- String
- Boolean
- Null
- Undefined

## Non-Primitive Types

- The non-primitive types are “Mutable” types.
- Their reference can be changed according to state and situation.
- They don't have fixed range of values.
- The value range varies according to the memory available.
- JavaScript non-primitive types are
  - Array
  - Object
  - Regular Expression

## Array Type

- Arrays in computer programming are used to **reduce overhead and complexity**.
- JavaScript **Array can store different types of values** in sequential order.
- It can reduce overhead by storing values in sequential order.
- It can reduce complexity by storing multiple values under one name.
- **Array size can be changed dynamically.**
- Array in JavaScript have the behaviour of collections like stack, queue, hash table.

### Declaring Array:

- Array can be declared by using
  - Array Meta Character “[]”
  - Array Constructor “Array()”

### Ex:

```
<script>
function f1(){
 var categories = [];
 var products = new Array();
}
f1();
</script>
```

### Initialize values into Array:

```
<script>
function f1(){
 var categories = ["Electronics","Footwear"];
 var products = new Array("Speaker","Nike Causals");
}
```

```
f1();
```

```
</script>
```

### **Assign Values by using Array Property**

- Property is used to map with index number in memory.
- So you can use property to access and or send value into memory.

**Ex:**

```
<script>
```

```
function f1(){
 var categories = [];
 categories["0"] = "Electronics";
 categories["1"] = "Footwear";
 for(var property in categories) {
 document.write(`${property} : [${typeof property}]
`);
 }
}
f1();
```

```
</script>
```

**Ex:**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Slide Show</title>
```

```
<link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
<link rel="stylesheet" href="../Fonts/css/all.css">
```

```
<script>
```



```

 var products = ["../Images/jblspeaker.jpg", "../Images/earpods.jpg",
"../Images/shoe.jpg", "../Images/shirt.jpg"];

 var productNames = ["JBL Speaker", "Earpods", "Nike Casuals", "Shirt"];

 function loadImage(index){

 var pic = document.getElementById("pic");

 pic.src= products[index];

 var lblName = document.getElementById("lblName");

 lblName.innerHTML = productNames[index];

 }

 var index = 0;

 function NextClick(){

 index++;

 loadImage(index);

 }

 function PreviousClick(){

 index--;

 loadImage(index);

 }

 function SlideShow(){

 var txtRange = document.getElementById("txtRange").value;

 loadImage(txtRange);

 }

</script>

</head>

<body class="container-fluid" onload="loadImage(0)">

 <div class="card">

 <div class="card-header text-center">

```

```

 <h3 id="lblName"></h3>
 </div>
 <div class="card-body text-center">
 <button onclick="PreviousClick()" class="btn btn-outline-danger">
 <
 </button>

 <button onclick="NextClick()" class="btn btn-outline-danger">
 >
 </button>
 </div>
 <div class="card-footer text-center">
 <input onchange="SlideShow()" class="form-control-range"
id="txtRange" type="range" min="0" value="0" max="3">
 </div>
</div>
</body>
</html>

```

## Array Manipulation

### Read Array Elements:

Method	Description
toString()	Returns array elements separated with comma. Ex: <pre> &lt;script&gt;     function f1(){         var products = ["TV", "Mobile", "Shoe"];         document.write(products.toString());     }     f1(); &lt;/script&gt; </pre>
join()	Returns array elements separated with custom delimiter.

	<p>Ex:</p> <pre>&lt;script&gt; function f1(){     var products = ["TV", "Mobile", "Shoe"];     document.write(products.join("--&gt;")); } f1(); &lt;/script&gt;</pre>
slice()	<p>Return array element between specified index.</p> <p>Ex:</p> <pre>&lt;script&gt; function f1(){     var products = ["TV", "Mobile", "Shoe"];     document.write(products.slice(1,2)); } f1(); &lt;/script&gt;</pre>
for..of	<p>It reads and return all array elements in sequential order.</p> <p>Ex:</p> <pre>&lt;script&gt; function f1(){     var products = ["TV", "Mobile", "Shoe"];     for(var item of products) {         document.write(item + "&lt;br&gt;");     } } f1(); &lt;/script&gt;</pre>
for..in	<p>It reads and return all array properties.</p> <p>Ex:</p> <pre>&lt;script&gt; function f1(){     var products = ["TV", "Mobile", "Shoe"];     for(var item in products) {         document.write(item + "&lt;br&gt;");     } } f1(); &lt;/script&gt;</pre>
for	<p>It uses a loop to read all elements by using initialization, condition and counter.</p> <p>Syntax:</p> <pre>for(initializer, condition, iterator) {</pre>

	<pre> }  Ex: &lt;script&gt;   function f1(){     var products = ["TV", "Mobile", "Shoe"];     for(var i=0; i&lt;products.length; i++) {       document.write(products[i] + "&lt;br&gt;");     }   }   f1(); &lt;/script&gt; </pre>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Add Array Elements into HTML Page to present as DOM elements:

- To Add any element you have first create element by using the method `document.createElement("elementName")`
- You can add element by using the method `append()`, `appendChild()`

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Array</title>
```

```
<link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
<script>
```

```
var categories = ["All", "Electronics", "Footwear", "Fashion", "Accessories"];
```

```
function bodyload(){
```

```
 var lstCategories = document.getElementById("lstCategories");
```

```
 var optCategories = document.getElementById("optCategories");
```

```
 for(var item of categories) {
```

```
 var li = document.createElement("li");
```

```
 li.innerHTML = item;
```

```
 lstCategories.appendChild(li);
```

```

 var option = document.createElement("option");

 option.text = item;

 option.value = item;

 optCategories.appendChild(option);

 }

}

</script>
</head>
<body onload="bodyload()" class="container-fluid">

 <div class="form-group">

 <h3>Select a Category</h3>

 <ol id="lstCategories">

 </div>

 <div class="form-group">

 <h3>Select Category</h3>

 <select class="form-control" id="optCategories">

 </select>

 </div>

 </body>
</html>

```

### Adding and Removing Elements from Array:

Method	Description
push()	Add new elements as last item.
unshift()	Add new elements as first item.
pop()	Remove and return last item.
shift()	Remove and return first item.

splice()	<p>It is used to add or remove item at any specific index.</p> <p>Syntax: splice(startIndex, removeCount, NewItems...)</p>
----------	--------------------------------------------------------------------------------------------------------------------------------

**Ex:**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Array</title>
```

```
<link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
<script>
```

```
var categories = ["All", "Electronics", "Footwear"];
```

```
function bodyload(){
```

```
var lstCategories = document.getElementById("lstCategories");
```

```
var optCategories = document.getElementById("optCategories");
```

```
lstCategories.innerHTML="";
```

```
optCategories.innerHTML="";
```

```
for(var item of categories) {
```

```
var li = document.createElement("li");
```

```
li.innerHTML = item;
```

```
lstCategories.appendChild(li);
```

```
var option = document.createElement("option");
```

```
option.text = item;
```

```
option.value = item;
```

```
optCategories.appendChild(option);
```

```
}
```

```
}
```

```
function AddClick(){
```

```
var txtName = document.getElementById("txtName");
```

```

 categories.splice(1,0,txtName.value);

 alert("Item Added");

 txtName.value="";

 bodyload();
 }

 function RemoveClick(){

 var item = categories.shift();

 alert(`${item} Removed`);

 bodyload();
 }

 function RemoveSelected(){

 var selectedItem = document.getElementById("optCategories").value;

 var selectedIndex = categories.indexOf(selectedItem);

 var c = confirm("Are you Sure Want to Delete?");

 if(c==true) {

 categories.splice(selectedIndex,1);

 bodyload();

 }

 }

</script>
</head>
<body onload="bodyload()" class="container-fluid">
 <div class="form-group">
 <label>Add Category</label>

 <div>

 <input id="txtName" type="text">

 <button onclick="AddClick()">Add</button>

 </div>
 </div>

```

```
<div class="form-group">
 <h3>Select a Category</h3>
 <ol id="lstCategories">

 <div>
 <button onclick="RemoveClick()">Remove</button>
 </div>
</div>

<div class="form-group">
 <h3>Select Category</h3>
 <select size="3" class="form-control" id="optCategories">

 </select>
 <div>
 <button onclick="RemoveSelected()">Remove Selected</button>
 </div>
</div>
</body>
</html>
```




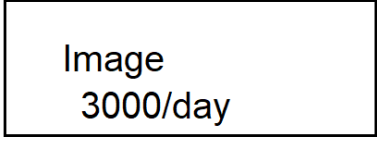
## Hotel Registration

### Customer Info

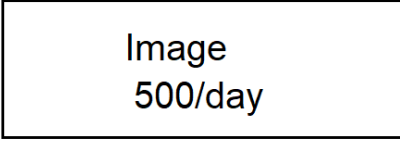
Customer Name	<input type="text"/>
Check in Date	<input type="date"/>
Total No of Days	<input type="number"/>
Total No of People	<input type="number"/>

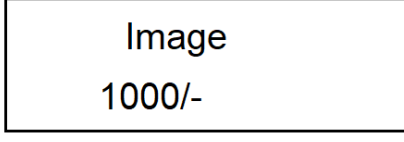
### Room Type

☐  2000/day  
Delux Room

☐  3000/day  
Suite Room

### Aminities

☐  500/day  
Locker

☐  1000/-  
A/C

### Advance

Mandatory

Register

### Bill Summary

Customer Name :  
Check in Date :  
Total No of Days:  
Total No People :  
Room Type:  
Aminities:  
Advance :  
Balance :

Tdays = 2

Delux: 2000/day 2x2000

Aminities 4000  
A/C = 1000x2

2000  
Advance = 2000 - 2000

=====

Balance

Searching for Elements in Array

indexOf()	It can search for element in array based on given string and returns the “index” number.
lastIndexOf()	It returns the last occurrence index number.
find()	<p>It finds and returns the first occurrence element that matches the given condition.</p> <p>Ex:</p> <pre> &lt;script&gt;   function f1(){     var sales = [34500, 20000, 45000, 12000, 30000];     var result = sales.find(function(val){       return val&gt;30000;     });     document.write(result);   }   f1(); &lt;/script&gt; </pre>
filter()	<p>It finds and returns all elements that matches the given condition.</p> <p>Ex:</p> <pre> &lt;script&gt;   function f1(){     var sales = [34500, 20000, 45000, 12000, 30000];     var result = sales.filter(function(val){       return val&lt;=30000;     });     document.write(result.toString());   }   f1(); &lt;/script&gt; </pre>

Ex:

```
<script>
```

```

function f1(){
 var sales = [34500, 20000, 45000, 12000, 30000];
 function search(val){
 return val>=30000;
 }
 var result = sales.find(search);
 document.write(result);
}

```

```
f1();
</script>
```

### Sort Array Elements

- sort() arranges elements in ascending order.
- reverse() arranges elements in reverse order [bottom to top]

**Ex:**

```
<script>
function f1(){
 var sales = [34500, 20000, 45000, 12000, 30000];
 function search(val){
 return val>=30000;
 }
 var result = sales.filter(search);
 result.sort();
 result.reverse();
 document.write(result.toString());
}
f1();
</script>
```

### FAQ:

#### 1. What type of values we can store in array?

You can store any type of value.

#### 2. We store function in Array?

Yes.

Ex:

```
<script>
function f1(){
```

```

 var methods = [function(){return "Hello !"}, function(a, b){return a +
b}];
 document.write(methods[0]() + "
");
 document.write(methods[1](10,20));
 }
 f1();
</script>

```

### 3. What is Array Destruction?

It is a technique used to access array elements and store in individual memory references.

**Ex:**

```

<script>
 function f1(){
 var methods = [function(){return "Hello !"}, function(a, b){return a +
b}];
 //Without Destruction
 var m1 = methods[0];
 var m2 = methods[1];
 document.write(m1() + "
");
 document.write(m2(10,30) + "
");
 // With Destruction
 var [x1, x2] = methods;
 document.write(x1() + "
");
 document.write(x2(10,20));
 }
 f1();
</script>

```

### 4. Can we define Array inside Array [Multi Dimension]?

Yes.

**Ex:**

```

<script>
 function f1(){
 var values = [[10,20],["A","B"]];
 document.write(values[0][1]);
 }
 f1();
</script>

```

## Object Type

- Object in computer programming was introduced in early 1960's by "Alan Kay".
- Object can keep all related data and functionality at one memory reference.
- Object comprises of data and functionality.
- Object is a set of properties and methods.

```
object1
{
 name: tv,
 price: 57000.66,
 qty : 2,
 total: function(){}
}
```

Properties —

Methods —

```
object2
{
 name:"mobile"
 price: 12000,
}
```

- Data is stored in properties.
- Functionality is defined in methods.
- In JavaScript early version object is also known as "pseudo class".

### Syntax:

```
var object = {
 property: value,
 method: function(){}
}
```

```
}
```

- You can access object property within the object by using “this” keyword.
- You can access object property outside the object by using object name.

Syntax:

```
object
```

```
{
```

```
 this.property
```

```
 this.method()
```

```
}
```

```
object.property
```

```
object.method()
```

- Later in early 1967 “**Johan Olay, Kristian Nygaard**” introduced the concept of reusing object with class. [OOP]
- The first OOP language was **SIMULA 67**, Small Talk, C++, Java, .NET Languages

**Ex:**

```
<script>
```

```
function f1(){
```

```
 var product = {
```

```
 Name: "",
```

```
 Price: 0,
```

```
 Qty: 0,
```

```
 Total: function(){
```

```
 return this.Qty * this.Price;
```

```
 },
```

```
 Print: function(){
```

```
 document.write(`
```

```
 Name : ${this.Name}

```

```
 Price: ${this.Price}

```

```

 Qty: ${this.Qty}

 Total: ${this.Total()}

`);
 }
}

product.Name = "Samsung TV";
product.Price = 4000.44;
product.Qty = 2;
product.Print();
document.write("<hr>");
product.Name = "Nike Casuals";
product.Price = 2000.44;
product.Qty = 3;
product.Print();

}

f1();
</script>

```

## JSON Type Data

### [JavaScript Object Notation]

- It is a format for data.
- It is a collection objects.

Ex:

```

<script>
function f1(){
 var products = [
 {Name: "TV", Price: 45000.44, Cities:['Delhi', 'Hyd']},
 {Name: "Mobile", Price: 12000.33, Cities: ['Hyd','Chennai']}
];
}

```

```

 for(var product of products) {
 document.write(product.Name + "-" + product.Price + "-" +
product.Cities.toString() + "
");
 }
 }
 f1();
</script>

```

### Ex: Filtering of data

```

<script>

function f1(){

 var products = [

 {Name: "TV", Price: 45000.44, Cities:['Delhi', 'Hyd'], Category:
"Electronics"},

 {Name: "Mobile", Price: 12000.33, Cities: ['Hyd','Chennai'],
Category:"Electronics"},

 {Name: "Nike Casuals", Price: 4000.44, Cities: ['Chennai', 'Hyd'],
Category: "Footwear"}

];

 var result = products.filter(function(product){

 return product.Category=="Electronics";

 })

 for(var item of result) {

 document.write(item.Name + "-" + item.Price + "
");

 }

}

f1();
</script>

```

### Ex: JSON Type Data for Products



```

<!DOCTYPE html>

<html>

 <head>

 <title>Slide Show</title>

 <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <link rel="stylesheet" href="../Fonts/css/all.css">

 <script>

 var products = [

 {Name: "JBL Speaker", Price: 4500.55, Photo:
"../Images/jblspeaker.jpg"},

 {Name: "EarPods", Price: 2500.55, Photo: "../Images/earpods.jpg"},

 {Name: "Nike Casuals", Price: 6500.55, Photo: "../Images/shoe.jpg"},

 {Name: "Lee Boot", Price: 1500.55, Photo: "../Images/shoe1.jpg"},

];

 function loadImage(index){

 var pic = document.getElementById("pic");

 pic.src= products[index].Photo;

 var lblName = document.getElementById("lblName");

 var lblPrice = document.getElementById("lblPrice");

 lblName.innerHTML = products[index].Name;

 lblPrice.innerHTML = "₹" + products[index].Price;

 }

 var index = 0;

 function NextClick(){

 index++;

 loadImage(index);

```

```
}

function PreviousClick(){
 index--;
 loadImage(index);
}

function SlideShow(){
 var txtRange = document.getElementById("txtRange").value;
 loadImage(txtRange);
}

</script>
</head>
<body class="container-fluid" onload="loadImage(0)">
 <div class="card">
 <div class="card-header text-center">
 <h3 id="lblName"></h3>
 </div>
 <div class="card-body text-center">
 <button onclick="PreviousClick()" class="btn btn-outline-danger">
 <
 </button>

 <button onclick="NextClick()" class="btn btn-outline-danger">
 >
 </button>
 </div>
 <div class="card-footer text-center">
```

```

 <h3 id="lblPrice"></h3>

 <input onchange="SlideShow()" class="form-control-range"
id="txtRange" type="range" min="0" value="0" max="3">

 </div>

</div>

</body>

</html>

```

### Ex: Adding Rows into Table Dynamically

```

<!DOCTYPE html>

<html>

 <head>

 <title>Dynamic Table</title>

 <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <script>

 var products = [

 {Name: "JBL Speaker", Price: 4500.55, Photo:
"../Images/jblspeaker.jpg"},

 {Name: "EarPods", Price: 2500.55, Photo: "../Images/earpods.jpg"},

 {Name: "Nike Casuals", Price: 6500.55, Photo: "../Images/shoe.jpg"},

 {Name: "Lee Boot", Price: 1500.55, Photo: "../Images/shoe1.jpg"},

];

 function bodyload(){

 var tbody= document.getElementById("tbody");

 for(var item of products)

 {

```

```
var tr = document.createElement("tr");
var tdName = document.createElement("td");
var tdPrice = document.createElement("td");
var tdPhoto = document.createElement("td");

tdName.innerHTML = item.Name;
tdPrice.innerHTML = item.Price;

var pic = new Image();
pic.src= item.Photo;
pic.height="50";
pic.width="50";

tdPhoto.appendChild(pic);
tr.appendChild(tdName);
tr.appendChild(tdPrice);
tr.appendChild(tdPhoto);

tbody.appendChild(tr);
}
}
</script>
</head>
<body onload="bodyload()" class="container-fluid">
 <h2>Product Details</h2>
 <table class="table table-hover">
```

```

<thead>
 <tr>
 <th>Name</th>
 <th>Price</th>
 <th>Preview</th>
 </tr>
</thead>
<tbody id="tbody">

</tbody>
</table>
</body>
</html>

```

### Ex: **Dynamic List with Nested Data**

```

<!DOCTYPE html>
<html>
 <head>
 <title>Dynamic List</title>
 <script>
 var data = [
 {Category: "Electronics", Products:["JBL Speaker", "EarPods"]},
 {Category: "Footwear", Products: ["Nike Casuals", "Lee Cooper
Boot"]},
 {Category: "Fashion", Products: ["Watch", "Shirt", "Jeans"]}
];
 function bodyload(){

```

```

var lstCategories = document.getElementById("lstCategories");
for(var item of data)
{
 var parentLi = document.createElement("li");
 parentLi.innerHTML = item.Category;
 lstCategories.appendChild(parentLi);
 for(var product of item.Products)
 {
 var ul = document.createElement("ul");
 var childLi = document.createElement("li");
 childLi.innerHTML = product;
 ul.appendChild(childLi);
 parentLi.appendChild(ul);
 }
}
}
</script>
</head>
<body onload="bodyload()">
 <ol id="lstCategories">

</body>
</html>

```

Ex: Dynamically Adding into Table

```
<!DOCTYPE html>

<html>

 <head>

 <title>Dynamic Insert</title>

 <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <script>

 var data = [

 {Name: "Samsung TV", Price: 45000.55},

 {Name: "Nike Casuals", Price: 4200.33}

];

 function LoadTable(){

 var tbody = document.getElementById("tbody");

 tbody.innerHTML="";

 for(var item of data){

 var tr = document.createElement("tr");

 var tdName = document.createElement("td");

 var tdPrice = document.createElement("td");

 tdName.innerHTML = item.Name;

 tdPrice.innerHTML = item.Price;

 tr.appendChild(tdName);

 tr.appendChild(tdPrice);

 tbody.appendChild(tr);

 }

 }

 </script>

 </head>

</html>
```

```
 }
 }
 function bodyload(){
 LoadTable();
 }
 var newObject = {
 Name: "",
 Price: 0
 };
 function AddClick(){
 var txtName = document.getElementById("txtName");
 var txtPrice = document.getElementById("txtPrice");

 newObject = {
 Name: txtName.value,
 Price: txtPrice.value
 }
 data.push(newObject);
 alert("Record Added");
 txtName.value="";
 txtPrice.value="";
 LoadTable();
 }
</script>
</head>
<body onload="bodyload()" class="container-fluid">
```



```
<div class="row">
 <div class="col-3">
 <h3>Register Product</h3>
 <div class="form-group">
 <label>Name</label>
 <div>
 <input type="text" id="txtName" class="form-control">
 </div>
 </div>
 <div class="form-group">
 <label>Price</label>
 <div>
 <input type="text" id="txtPrice" class="form-control">
 </div>
 </div>
 <div class="form-group">
 <button onclick="AddClick()" class="btn btn-primary btn-block">Add Product</button>
 </div>
 </div>
 <div class="col-9">
 <table class="table table-hover">
 <thead>
 <tr>
 <th>Name</th>
 <th>Price</th>
 </tr>
```

```

 </thead>

 <tbody id="tbody">

 </tbody>

 </table>

</div>

</div>

</body>

</html>

```

## Regular Expression

- Regular expression is used to verify the format of input value.
- Expression is defined by using meta character and quantifiers enclosed in “/ /”
- Regular expression is verified by using “match()”.
- match() is a boolean method that return true if expression is matching with value.

Ex:

```

<!DOCTYPE html>

<html>

 <head>

 <title>Regular Expression</title>

 <script>

 function VerifyPassword(){

 var txtPwd = document.getElementById("txtPwd").value;

 var regExp = /(?!.*[A-Z])\w{4,10}/;

 var msg = document.getElementById("msg");

 var grade= document.getElementById("grade");

```

```
function ShowGrade(mn, mx, val){
 grade.min = mn;
 grade.max= mx;
 grade.value = val;
}

if(txtPwd.match(regExp)){
 msg.innerHTML="Strong Password".fontcolor('green');
 ShowGrade(1,100,100);
} else {
 if(txtPwd.length<4){
 msg.innerHTML="Poor Password".fontcolor('red');
 ShowGrade(1,100,20);
 } else {
 msg.innerHTML="Weak Password".fontcolor('orange');
 ShowGrade(1,100,70);
 }
}
}

</script>

<style>
progress{
 height: 20px;
 width: 150px;
}
```

```

 </style>
</head>
<body>
 <fieldset>
 <legend>Password</legend>
 <div>
 <input onkeyup="VerifyPassword()" id="txtPwd"
type="password">
 <div>
 <progress id="grade" min="1" value="0"
max="100"></progress>
 </div>
 <div id="msg"></div>
 </div>
 </fieldset>
</body>
</html>

```

### JavaScript Date Type

- JavaScript date values are handle by using "Date()" constructor.
- It allocates memory to store date and time type value.
- In the memory date is defined as string with "yy-mm-dd" format.

Syntax:

```
var mfd = new Date("YY-MM-DD");
```

```
var mfd = new Date(); // it loads current date into memory.
```

- To Access date values JavaScript provides several methods
  - getHours() – 0 to 23
  - getMinutes() – 0 to 59
  - getSeconds() – 0 to 59
  - getMilliseconds() – 0 to 999

- getDate() – returns the date number
- getDay() – returns the weekday number [0=Sunday]
- getMonth() – returns the month number [0=January]
- getFullYear()
- toString()
- toLocaleDateString()
- toLocaleTimeString()
- toString()
- toTimeString()

Ex:

```
<script>
```

```
function f1(){
 var product = {
 Name: "Samsung TV",
 Price: 45000.55,
 InStock: true,
 Mfd: new Date("2020-03-18")
 };
 var months = ["January", "Feb", "March", "Apr"];
 var weekdays = ["Sunday", "Monday", "Tue", "Wednesday", "Thu"];
```

```
document.write(`Name=${product.Name}
Price=${product.Price}
InStock=${product.InStock}

```

```
 Manufactured Month: ${months[product.Mfd.getMonth()]}

```

```
 Manufactured Weekday: ${weekdays[product.Mfd.getDay()]}

```

```
 Manufactured Date: ${product.Mfd.getDate()}

```

```
 Manufactured Year: ${product.Mfd.getFullYear()}

```

```
 Date: ${product.Mfd.toLocaleDateString()}
```

```
`);
}
f1();
</script>
```

- You can set date dynamically by using “set” methods.
  - o setMonth()
  - o setDate()
  - o setFullYear()
  - o setHours()
  - o setMinutes()
  - o setSeconds()
  - o setMilliseconds()

Ex:

```
<script>
function f1(){
 var product = {
 Name: "Samsung TV",
 Price: 45000.55,
 InStock: true,
 Mfd: new Date("2020-03-18")
 };

 var months = ["January", "Feb", "March", "Apr"];
 var weekdays = ["Sunday", "Monday", "Tue", "Wednesday", "Thu"];
 product.Mfd.setFullYear(2021);
 product.Mfd.setMonth(1);

 document.write(`Name=${product.Name}
Price=${product.Price}
InStock=${product.InStock}

 Manufactured Month: ${months[product.Mfd.getMonth()]}

 Manufactured Weekday: ${weekdays[product.Mfd.getDay()]}

```

```
 Manufactured Date: ${product.Mfd.getDate()}

 Manufactured Year: ${product.Mfd.getFullYear()}

 Date: ${product.Mfd.toLocaleDateString()}
 `);
}
f1();
</script>
```

Ex:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Date</title>
 <style>
 .container{
 width: 600px;
 height: 400px;
 justify-content: center;
 align-items: center;
 margin:auto;
 }
 </style>
 <script>
 function bodyload(){
 var now = new Date();
 var hrs = now.getHours();
 var pic = document.getElementById("pic");
```

```
 if(hrs>=00 && hrs<=12) {
 pic.src="../Images/morning.gif";
 } else if(hrs>12 && hrs<=17) {
 pic.src="../Images/afternoon.gif";
 } else {
 pic.src="../Images/evening.gif";
 }
 }
</script>
</head>
<body onload="bodyload()">
 <div class="container">

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

## Summary

- Primitive Types
  - Number
  - String
  - Boolean
  - Null
  - Undefined
- Non-Primitive Types
  - Array
  - Object
  - Regular Expression
  - Date



## JavaScript Operators and Expressions

- Operator is an object that evaluates a value.
- Based on what type of value an operator evaluates the operators are classified into following types:
  - Arithmetic Operators
  - Logical Operators
  - Comparison Operators
  - Assignment Operators
  - Special Operators
- Operators are also classed into 3 type based on the number of operands they can handle.
  - Unary : One Operand [x++]
  - Binary : Two operands [x + y]
  - Ternary : Three operands [ (condition)?true:false]

### Arithmetic Operators:

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus division
**	Exponent [new in ES5] [old version – Math.pow()]
++	Increment
--	Decrement

Ex:

```
<script>
function f1(){
 var x = 2;
```

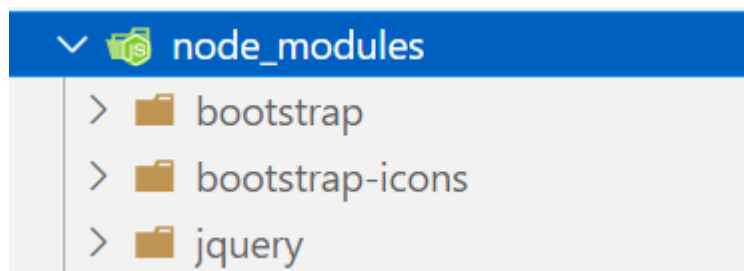
```
var y = 3;
document.write(`x**y=${x**y}
`);
document.write(`Power=${Math.pow(x,y)}`);
}
f1();
</script>
```

# Bootstrap 5

Setup Environment of Bootstrap in your project

- Open Terminal
- Install following libraries
  - > npm install bootstrap --save
  - > npm install bootstrap-icons --save
  - > npm install jquery --save

Note: All libraries are copied into "node\_modules" folder



- Link Bootstrap files to your page
  - o bootstrap.css
  - o bootstrap-icons.css
  - o jquery.js
  - o bootstrap-bundle.js

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Bootstrap</title>
```

```
<link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
<link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">
```

```
</head>
```

```
<body>
```

```
<h2> Amazon Shopping</h2>
```

```
<p>Online Shopping - 50% off on all products.</p>
```

```
<script src="../../node_modules/jquery/dist/jquery.js"></script>
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
</body>
</html>
```

## Containers

.container  
.container-fluid  
.container-sm  
.container-lg  
.container-xl  
.container-xxl

## Margins and Padding

.m margin  
.p padding

### Directions

t top  
b bottom  
s left  
e right  
x both left and right  
y both top and bottom

### Sizes

1 to 5

Ex:

m-5 all directions margin set to 5 [large]  
p-5 all directions padding set to 5  
mt-3 margin top 3  
mx-3 margin left and right 3

pt-3                  padding top 3

Ex:

```
<!DOCTYPE html>

<html>

 <head>

 <title>Bootstrap</title>

 <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">

 </head>

 <body>

 <div class="container-fluid mt-4 mb-4 ms-2 me-2 pt-4 ps-4">

 <h2> Amazon Shopping</h2>

 <p>Online Shopping - 50% off on all products.</p>

 </div>

 <script src="../node_modules/jquery/dist/jquery.js"></script>

 <script src="../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

 </body>

</html>
```

### Background and Foreground Colors

- Bootstrap provides contextual  
bg-contextualName  
text-contextualName
- “bg” background
- “text” for text color
- The contextual names
  - Primary
  - Secondary
  - Dark
  - Light
  - Danger
  - Success

- Warning
- Info etc..

Syntax:

.bg-danger

.bg-warning

.bg-info

.text-primary

.text-secondary

.text-danger

Ex:

```
<div class="container-fluid">
```

```
 <h2 class="bg-primary text-white"> Amazon
 Shopping</h2>
```

```
 <p class="text-danger">Online Shopping - 50% off on all products.</p>
```

```
</div>
```

## Border

.border

.border-top

.border-end

.border-start

.border-bottom

Remove Border

.border-0                      removes border

.border-top-0                removes top border

Border Color

.border-primary, success, danger, warning etc.

Border Width

.border-1 to 5                width

Border Radius

.rounded

.rounded-top  
.rounded-bottom  
.rounded-start  
.rounded-end  
.rounded-circle  
.rounded-pill

Ex:

```
<div class="mt-3">

</div>
```

## Width and Height

.w for width

.h for height

Size

0	0%
25	25%
50	50%
75	75%
100	100%

Syntax:

.w-100	width 100%
.w-25	width 25%
.h-25	height 25%

Ex:

```
<div class="mt-3 w-25 h-100 border border-4 border-danger p-3">
 <h2 class="text-primary"> Amazon Shopping</h2>
```

```
<p class="text-danger">Online Shopping - 50% off on all products.</p>
</div>
```

## Display

.d for display

.d-{value}

### Display Values

- None
- Inline
- Block
- Grid
- Flex

### Syntax:

.d-flex display flex

.d-inline display inline

.d-block

.d-grid

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Bootstrap</title>
```

```
<link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
<link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">
```

```
</head>
```

```
<body class="container-fluid">
```

```
<ul class="d-flex list-unstyled">
```

```
<li class="me-3">Home
```

```
<li class="me-3">About
```

```
<li class="me-3">Contact
```



```


<div class="bg-danger text-white d-inline">

 Block-1

</div>

<div class="bg-warning text-white d-inline">

 Block-2

</div>

<script src="../../node_modules/jquery/dist/jquery.js"></script>

<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</body>
</html>

```

## Position

```

.position-static
.position-relative
.position-absolute
.position-fixed
.position-sticky

```

### Position With Location

```

.top
.start
.end
.bottom

```

### Position Location values

```

1 at 0
50 at 50
100at 100

```

### Position with Translate

- translate is the change the location along x and y axis
  - translate-middle
  - translate-start
  - translate-end

Ex:

```
<h2 class="position-sticky top-0 bg-dark text-white">Home | About | Contact</h2>
```

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
 <head>
```

```
 <title>Bootstrap</title>
```

```
 <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
 <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">
```

```
 </head>
```

```
 <body class="container-fluid">
```

```
 <div class="mt-4">
```

```
 <button class="position-relative">
```

```
 Nofications
```

```
 +30
```

```
 </button>
```

```
 </div>
```

```
 <script src="../node_modules/jquery/dist/jquery.js"></script>
```

```
 <script src="../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
```

```
 </body>
```

```
</html>
```

## Row and Columns

.row

.cols

.cols-1 to 12

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>

 <title>Bootstrap</title>

 <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body class="container-fluid">

 <div class="row bg-danger text-white text-center">

 <div class="col">

 Amazon Shopping - Online Store

 </div>

 </div>

 <div class="row mt-3" style="height: 400px;">

 <div class="col-2">

 Home

 Electronics

 Footwear

 </div>

 <div class="col-8">

 <h2>Home Page</h2>

 </div>

 <div class="col-2">

 <h3>Ads..</h3>

 </div>

 </div>

 <div class="row">

 <div class="col bg-danger text-white text-center">

 © copyright 2021

 </div>

 </div>

</body>

</html>
```

```

</div>

<script src="../../node_modules/jquery/dist/jquery.js"></script>

<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</body>

</html>

```

Ex:

```

<dl class="row">

 <dt class="col-2">Name</dt>

 <dd class="col-10">Samsung TV</dd>

 <dt class="col-2">Price</dt>

 <dd class="col-10">56000.55</dd>

 <dt class="col-2">Stock</dt>

 <dd class="col-10">Available</dd>

</dl>

```

## Bootstrap Components

- Component comprises of pre-defined styles and functionality
- Functionality is configured by using jQuery attributes
- Bootstrap 5 attributes are
  - data-bs-target specifies the target element id  
[data-target] bootstrap 4
  - data-bs-toggle specifies the component name  
[data-toggle] bootstrap 4
  - data-bs-dismiss etc.. configures close function for current component  
[data-dismiss] bootstrap 4

## Alerts

.alert

.alert-primary, secondary, success, danger

.alert-link

.alert.dismissible

.show [display by default on page load]

.fade [to configure fade animation]

Ex:

```
<!DOCTYPE html>

<html>

 <head>

 <title>Alerts</title>

 <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">

 </head>

 <body class="container-fluid">

 <h2>Alerts</h2>

 <div class="alert alert-danger alert-dismissible show fade" id="delete">

 <button class="btn-close" data-bs-dismiss="alert"></button>

 <h2>Delete Record</h2>

 <p>Record will be deleted permanently</p>

 Help?

 </div>

 <script src="../node_modules/jquery/dist/jquery.js"></script>

 <script src="../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

 </body>

</html>
```

## Buttons

.btn

.btn-primary, secondary, danger, etc..

.btn-outline-primary, danger, success etc..

.btn-close

.btn-sm, lg,

.btn-toolbar

.btn-group

Ex:

```
<!DOCTYPE html>

<html>

 <head>

 <title>Alerts</title>

 <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">

 </head>

 <body class="container-fluid">

 <h2>Buttons</h2>

 <button class="btn btn-danger">Delete</button>

 <button class="btn btn-success">Insert</button>

 <h2>Outline Button</h2>

 <button class="btn btn-outline-danger">Delete</button>

 <button class="btn btn-outline-success">Insert</button>

 <h2>Button Group</h2>

 <div class="btn-group">

 <button class="btn btn-success">Insert</button>

 <button class="btn btn-danger">Delete</button>

 </div>

 <h2>Button Toolbar</h2>

 <div class="btn-toolbar bg-danger justify-content-between">

 <div class="btn-group">

 <button class="btn btn-danger"> Home</button>

 <button class="btn btn-danger"> Contact</button>

 <button class="btn btn-danger"> Notifications</button>

 </div>

 <div class="btn-group">

 <button class="btn btn-danger"> Search</button>

 </div>

 </div>

 </body>

</html>
```

```

<h2>Block Button</h2>

<div class="d-grid">

 <button class="btn btn-dark">Back to Top</button>

</div>

<script src="../../node_modules/jquery/dist/jquery.js"></script>

<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</body>

</html>

```

## Badge

```

.badge
.rounded-pill
.bg-contextual

```

Ex:

```

<!DOCTYPE html>

<html>

 <head>

 <title>Alerts</title>

 <link rel="stylesheet" href="../../node_modules/bootstrap/dist/css/bootstrap.css">

 <link rel="stylesheet" href="../../node_modules/bootstrap-icons/font/bootstrap-icons.css">

 </head>

 <body class="container-fluid">

 <h2>Badge</h2>

 <button class="btn btn-primary">

 Notifications 4

 </button>

 <h2>Badge Position</h2>

 <button class="btn btn-primary position-relative">


```

Mails <span class="badge bg-danger start-100 translate-middle position-absolute rounded-pill">10+</span>

</button>

<script src="../../node\_modules/jquery/dist/jquery.js"></script>

<script src="../../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</body>

</html>

## Cards

.card-group

.card

.card-header

.card-footer

.card-body

.card-img-top

.card-img-overlay

.card-title

.card-text

.card-link

Ex:

<!DOCTYPE html>

<html>

<head>

<title>Alerts</title>

<link rel="stylesheet" href="../../node\_modules/bootstrap/dist/css/bootstrap.css">

<link rel="stylesheet" href="../../node\_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body class="container-fluid">

<h2>Products Catalog</h2>

<div class="card-group d-flex flex-wrap flex-row">



```
<div class="card m-3 border border-danger">
 <div class="card-header">
 <h3 class="card-title">JBL Speaker</h3>
 </div>
 <div class="card-body">

 <p class="card-text">₹ 5600.53/-</p>
 more.
 </div>
 <div class="card-footer d-grid">
 <button class="btn btn-danger">

 Add to Cart
 </button>
 </div>
</div>
```

```
<div class="card m-3 border border-success">
 <div class="card-header">
 <h3 class="card-title">Nike Casuals</h3>
 </div>
 <div class="card-body">

 <p class="card-text">₹ 7600.53/-</p>
 </div>
 <div class="card-footer d-grid">
 <button class="btn btn-danger">

 Add to Cart
 </button>
 </div>
</div>
```

</div>

<div class="card m-3 border border-primary">

<div class="card-header">

<h3 class="card-title">Shirt</h3>

</div>

<div class="card-body">



<p class="card-text">&#8377; 5600.53/-</p>

</div>

<div class="card-footer d-grid">

<button class="btn btn-danger">

<span class="bi bi-cart4"></span>

Add to Cart

</button>

</div>

</div>

<div class="card m-3 border border-primary">



<div class="card-header text-center">

<h3 class="card-title">Shirt</h3>

<p class="card-text">&#8377; 5600.53/-</p>

</div>

<div class="card-footer d-grid">

<button class="btn btn-danger">

<span class="bi bi-cart4"></span>

Add to Cart

</button>

</div>

</div>

```
</div>
<script src="../../node_modules/jquery/dist/jquery.js"></script>
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
</body>
</html>
```

## Carousel

```
.carousel
.carousel-dark
.carousel-inner
.carousel-item
.slide
.carousel-fade
.carousel-control-prev
.carousel-control-prev-icon
.carousel-control-next
.carousel-control-next-icon
.carousel-indicators
```

## Attributes

data-bs-ride	starts carousel
data-bs-slide	Previous and Next
data-bs-slide-to	Specific slide number
data-bs-touch="false"	Disable touch sliding
data-bs-interval="false"	Disable auto slide

Ex:

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

```
<link rel="stylesheet" href="../../node_modules/bootstrap/dist/css/bootstrap.css">

<link rel="stylesheet" href="../../node_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body class="container-fluid">

<div class="carousel slide carousel-dark" id="topBanner" data-bs-ride="carousel" >

 <div class="carousel-inner">

 <div class="carousel-item active" data-bs-interval="3000">

 </div>

 <div class="carousel-item">

 </div>

 <div class="carousel-item">

 </div>

 </div>

 <button class="carousel-control-prev" data-bs-target="#topBanner" data-bs-slide="prev" >

 </button>

 <button class="carousel-control-next" data-bs-target="#topBanner" data-bs-slide="next" >

 </button>

 <div class="carousel-indicators">

 <button data-bs-target="#topBanner" data-bs-slide-to="0" class="active"></button>

 <button data-bs-target="#topBanner" data-bs-slide-to="1"></button>

 <button data-bs-target="#topBanner" data-bs-slide-to="2"></button>

 </div>

</div>

<script src="../../node_modules/jquery/dist/jquery.js"></script>

<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</body>
```

```
</html>
```

Collapse

.collapse

.show

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
 <head>
```

```
 <title>Alerts</title>
```

```
 <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
 <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">
```

```
 </head>
```

```
 <body class="container-fluid">
```

```
 <div class="mt-4">
```

```
 <button data-bs-target="#preview" data-bs-toggle="collapse" class="btn btn-primary">Show /
Hide</button>
```

```
 Toggle
```

```
 <div class="collapse show" id="preview">
```

```
 <dl>
```

```
 <dt>Name</dt>
```

```
 <dd>Nike Causals</dd>
```

```
 <dt>Photo</dt>
```

```
 <dd>
```

```

```

```
 </dd>
```

```
 </dl>
```

```
 </div>
```

```
 </div>
```

```
 <script src="../node_modules/jquery/dist/jquery.js"></script>
```

```
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
</body>
</html>
```

## Accordion

```
.accordion
.accordion-item
.accordion-header
.accordion-button
.accordion-body
.accordion-collapse
```

Ex:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Alerts</title>
 <link rel="stylesheet" href="../../node_modules/bootstrap/dist/css/bootstrap.css">
 <link rel="stylesheet" href="../../node_modules/bootstrap-icons/font/bootstrap-icons.css">
 </head>
 <body class="container-fluid">
 <h2>Table of Contents</h2>
 <div class="accordion" id="topics">
 <div class="accordion-item">
 <h2 class="accordion-header d-grid">
 <button class="btn btn-danger" data-bs-target="#html" data-bs-toggle="collapse">
 HTML Tutorial
 </button>
 </h2>
 <div class="collapse accordion-collapse show" id="html" data-bs-parent="#topics">
 <div class="accordion-body">
```

```

 Normal Elements
 Void Elements
 RC Data Elements
 Raw Text Elements

</div>
</div>
</div>
```

```
<div class="accordion-item">
 <h2 class="accordion-header">
 <button class="accordion-button bg-success text-white" data-bs-target="#css" data-bs-
toggle="collapse">
 CSS Examples
 </button>
 </h2>
 <div class="collapse accordion-collapse" id="css" data-bs-parent="#topics">
 <div class="accordion-body">

 Transions
 Transforms
 Responsive Design
 Animations

 </div>
 </div>
</div>
</div>
```

```

<div class="accordion-item">
 <h2 class="accordion-header">
 <button class="accordion-button" data-bs-target="#js" data-bs-toggle="collapse">
 JavaScript Tutorial
 </button>
 </h2>
 <div class="collapse accordion-collapse" id="js" data-bs-parent="#topics">
 <div class="accordion-body">

 Variables
 Data Types
 Operators
 Statements

 </div>
 </div>
</div>
</div>
<script src="../../node_modules/jquery/dist/jquery.js"></script>
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
</body>
</html>

```

## Forms

.form-control	: textbox, email, number, textarea, data elements related to string and number input.
.form-select	: dropdown
.form-check	: checkbox and radio group
.form-check-input	: checkbox and radio
.form-switch	: checkbox style
.form-file	: input type file
.form-range	: input type range



.form-label : labels  
.form-check-label : label of radio and checkbox

Ex:

```
<!DOCTYPE html>

<html>

 <head>

 <title>Form</title>

 <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">

 </head>

 <body class="container-fluid">

 <div class="d-flex justify-content-center align-items-center" style="height: 400px;">

 <form class="w-25">

 <h3>Register Product</h3>

 <div>

 <label class="form-label">Product Name</label>

 <div>

 <input type="text" class="form-control">

 </div>

 </div>

 <div class="mt-2">

 <label class="form-label">Price</label>

 <div>

 <input type="text" class="form-control">

 </div>

 </div>

 <div class="mt-2">

 <label class="form-label">Qty</label>

 <div>

 <input type="range" min="1" max="4" value="1" class="form-range" >

 </div>

 </div>

 </form>

 </div>

 </body>

</html>
```

```
 </div>
</div>
<div class="mt-2">
 <label class="form-label">Shipped To</label>
 <div>
 <select class="form-select">
 <option>Delhi</option>
 <option>Hyd</option>
 </select>
 </div>
</div>
<div class="mt-2">
 <label class="form-label">In Stock</label>
 <div class="form-check form-switch">
 <input class="form-check-input" type="checkbox"> <label class="form-check-label">Yes</label>
 </div>
</div>
<div class="mt-2">
 <label class="form-label">Choose Color</label>
 <div>
 <input type="color" class="form-control-color">
 </div>
</div>
<div class="mt-2">
 <label class="form-label">Upload Photo</label>
 <div>
 <input type="file" class="form-control">
 </div>
</div>
<div class="d-grid mt-2">
```

```

 <button class="btn btn-primary">Register Product</button>
 </div>
</form>
</div>
<script src="../../node_modules/jquery/dist/jquery.js"></script>
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
</body>
</html>

```

### Input Group

```

.input-group
.input-group-text

```

Ex:

```

<!DOCTYPE html>
<html>
 <head>
 <title>Form</title>
 <link rel="stylesheet" href="../../node_modules/bootstrap/dist/css/bootstrap.css">
 <link rel="stylesheet" href="../../node_modules/bootstrap-icons/font/bootstrap-icons.css">
 </head>
 <body class="container-fluid">
 <div class="d-flex justify-content-center align-items-center" style="height: 400px;">
 <div class="input-group w-50">
 <select class="input-group-text">
 <option>All</option>
 <option>Electronics</option>
 <option>Footwear</option>
 </select>
 <input type="text" class="form-control">
 <button class="btn btn-warning">


```

```

 </button>
 </div>
</div>
<script src="../../node_modules/jquery/dist/jquery.js"></script>
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
</body>
</html>

```

Ex: Append [right] Prepend [left]

```

<div class="w-25 input-group">
 ₹
 <input type="text" class="form-control">
 .00
</div>

```

Ex:

```

<div class="d-flex justify-content-center align-items-center" style="height: 400px;">
 <div class="w-25 input-group">

 <input type="email" placeholder="Your Email" class="form-control">

 </div>
</div>

```

## Modal

```

.modal
.modal-dialog
.modal-content
.modal-header
.modal-footer

```

.modal-dialog-center

Ex:

```
<!DOCTYPE html>

<html>

 <head>

 <title>Form</title>

 <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">

 </head>

 <body class="container-fluid">

 <div class="mt-3">

 <button class="btn btn-primary" data-bs-target="#register" data-bs-
toggle="modal">Register</button>

 <button class="btn btn-warning" data-bs-target="#login" data-bs-
toggle="modal">Login</button>

 <div class="modal fade show" id="login">

 <div class="modal-dialog">

 <div class="modal-content">

 <div class="modal-header">

 <h2>Sign-In</h2>

 <button class="btn-close"></button>

 </div>

 <div class="modal-body">

 <dl>

 <dt>User Name</dt>

 <dd><input type="text" class="form-control"></dd>

 </dl>

 </div>

 </div>

 </div>

 </div>

 </div>

 </div>
```

```

<div class="modal show fade" id="register">

 <div class="modal-dialog modal-dialog-centered modal-dialog-scrollable">

 <div class="modal-content">

 <div class="modal-header bg-danger text-white">

 <h2>Register User</h2>

 <button class="btn-close btn-close-white" data-bs-dismiss="modal"></button>

 </div>

 <div class="modal-body">

 <dl>

 <dt>User Name</dt>

 <dd><input type="text" class="form-control"></dd>

 <dt>Password</dt>

 <dd><input type="password" class="form-control"></dd>

 <dt>User Name</dt>

 <dd><input type="text" class="form-control"></dd>

 <dt>Password</dt>

 <dd><input type="password" class="form-control"></dd>

 <dt>User Name</dt>

 <dd><input type="text" class="form-control"></dd>

 <dt>Password</dt>

 <dd><input type="password" class="form-control"></dd>

 <dt>User Name</dt>

 <dd><input type="text" class="form-control"></dd>

 <dt>Password</dt>

 <dd><input type="password" class="form-control"></dd>

 </dl>

 </div>

 <div class="modal-footer">

 <button class="btn btn-primary">Register</button>

 <button class="btn btn-danger" data-bs-dismiss="modal">Cancel</button>

 </div>

 </div>

 </div>

</div>

```

```
 </div>
 </div>
</div>
</div>
</div>
<script src="../../node_modules/jquery/dist/jquery.js"></script>
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
</body>
</html>
```

## Images

.img-fluid  
.img-thumbnail

Ex:

```
<div class="mt-2">

</div>
```

## Tables

.table  
.table-hover  
.table-bordered  
.table-borderless  
.table-striped  
.table-responsive  
.table-primary, secondary, success, danger, warning, info, light, dark  
.caption-top

Ex:

&lt;!DOCTYPE html&gt;

&lt;html&gt;

&lt;head&gt;

&lt;title&gt;Form&lt;/title&gt;

```
<link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
<link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">
```

&lt;/head&gt;

```
<body class="container-fluid">
```

<div class="mt-2">

## Products List

<div class="table-responsive">

--

<caption>Products - Updated on 05-07-2021</caption>

|
 Product Id | Name | Price | Name | Price | Name | Price | Name | Price | Name | Price | Name | Price | Name | Price | Name |



Price | Name | Price | Name | Price | Name | Price | Price | Name | Price | Name | Price | Price | Name | Price | Name | Price | Price | Name | Price | Name | Price | Price | Name | Price | Name | Price | Price | Name | Price | Name |

46000.55 |

1 | Samsung TV | 46000.55 | 1 | Samsung TV | 46000.55 | 1 | Samsung TV | 46000.55 | 1 | Samsung TV | 46000.55 ||
 2 | Mobile | 16000.55 | 2 | Mobile | 16000.55 | 2 | Mobile | 16000.55 | 2 | Mobile | 16000.55 | 2 | Mobile | 16000.55 | 2 | Mobile |

<td>16000.55</td>

<td>2</td>

<td>Mobile</td>

<td>16000.55</td>

<td>2</td>

<td>Mobile</td>

<td>16000.55</td>

<td>2</td>

<td>Mobile</td>

<td>16000.55</td>

<td>2</td>

<td>Mobile</td>

<td>16000.55</td>

<td>2</td>

<td>Mobile</td>

<td>16000.55</td>

</tr>

<tr>

<td>3</td>

<td>Nike Casual</td>

<td>6000.55</td>

<td>3</td>

<td>Nike Casual</td>

<td>6000.55</td>

<td>3</td>

<td>Nike Casual</td>

<td>6000.55</td>

<td>3</td>

<td>Nike Casual</td>

<td>6000.55</td>

<td>3</td>

```
<td>Nike Casual</td>
<td>6000.55</td>
<td>3</td>
<td>Nike Casual</td>
<td>6000.55</td>
<td>3</td>
<td>Nike Casual</td>
<td>6000.55</td>
<td>3</td>
<td>Nike Casual</td>
<td>6000.55</td>
<td>3</td>
<td>Nike Casual</td>
<td>6000.55</td>
<td>3</td>
<td>Nike Casual</td>
<td>6000.55</td>
<td>3</td>
<td>Nike Casual</td>
<td>6000.55</td>
<td>3</td>
<td>Nike Casual</td>
<td>6000.55</td>
<td>3</td>
</tr>
</tbody>
</table>
</div>
</div>
```

```

<script src="../../node_modules/jquery/dist/jquery.js"></script>
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</body>
</html>

```

### Dropdown List

```

.dropdown
.dropdown-toggle button in dropdown menu
.dropdown-menu collections of items
.dropdown-item
.dropup
.dropend [right]
.dropstart [left]
.active
.disabled

```

Ex:

```

<!DOCTYPE html>

<html>

 <head>

 <title>Form</title>

 <link rel="stylesheet" href="../../node_modules/bootstrap/dist/css/bootstrap.css">

 <link rel="stylesheet" href="../../node_modules/bootstrap-icons/font/bootstrap-icons.css">

 </head>

 <body class="container-fluid">

```

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

</p>

<div class="mt-2 w-25 d-grid">

<div class="dropdown">

<button class="btn btn-primary btn-lg dropdown-toggle" data-bs-toggle="dropdown" >

More

</button>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="#"> <span class="bi bi-house-fill"></span>

Home</a></li>

<li><a class="dropdown-item active" href="#"> <span class="bi bi-alarm"></span>

Notifications</a></li>

<li><a class="dropdown-item disabled" href="#"> <span class="bi bi-envelope-fill"></span>

Contact</a></li>

<li class="dropdown-divider"></li>

<li class="dropdown-item">

<div class="dropdown dropend">

<a class="btn dropdown-toggle">

Categories

</a>

</div>

</li>

</ul>

</div>

</div>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

```
</p>
<script src="../../node_modules/jquery/dist/jquery.js"></script>
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
</body>
</html>
```

## Navs and Tabs

```
.nav
.nav-item
.nav-link
.nav-tabs
.nav-pills
.tab-content
.tab-pane
.fade
.show
.active
```

Ex:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Form</title>
 <link rel="stylesheet" href="../../node_modules/bootstrap/dist/css/bootstrap.css">
 <link rel="stylesheet" href="../../node_modules/bootstrap-icons/font/bootstrap-icons.css">
 </head>
 <body class="container-fluid">
 <div class="mt-3">
 <nav>
 <div class="nav nav-tabs">
```



```
 <button class="nav-link active" data-bs-toggle="tab" data-bs-target="#basic">Basic
Details</button>
```

```
 <button class="nav-link" data-bs-toggle="tab" data-bs-
target="#preview">Preview</button>
```

```
 <button class="nav-link" data-bs-toggle="tab" data-bs-
target="#description">Description</button>
```

```
 </div>
```

```
</nav>
```

```
<div class="tab-content mt-3">
```

```
 <div class="tab-pane fade show active" id="basic">
```

```
 <dl>
```

```
 <dt>Name</dt>
```

```
 <dd>JBL Speaker</dd>
```

```
 <dt>Price</dt>
```

```
 <dd>5600.55</dd>
```

```
 <dt>Stock</dt>
```

```
 <dd>Avilable
```

```
 </dl>
```

```
 </div>
```

```
 <div class="tab-pane fade" id="preview">
```

```

```

```
 </div>
```

```
 <div class="tab-pane fade" id="description">
```

```
 <p>something about JBL Speaker..</p>
```

```
 </div>
```

```
</div>
```

```
</div>
```

```
<script src="../../node_modules/jquery/dist/jquery.js"></script>
```

```
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
```

```
</body>
```

```
</html>
```

## Navbar

.navbar

.navbar-expand {sm-md-lg-xl-xxl}

.navbar-brand

.navbar-nav

.navbar-toggler

.navbar-toggler-icon

.navbar-collapse

.navbar-light

.navbar-dark

Ex:

```
<!DOCTYPE html>
```

```
<html>
```

```
 <head>
```

```
 <title>Form</title>
```

```
 <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
```

```
 <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">
```

```
 </head>
```

```
 <body class="container-fluid">
```

```
 <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
```

```
 <div class="container-fluid">
```

```
 <button class="navbar-toggler" data-bs-target="#menuList" data-bs-toggle="collapse">
```

```

```

```
 </button>
```

```
 Amazon Store
```

```
 <div class="collapse navbar-collapse" id="menuList">
```

```
 <ul class="navbar-nav me-auto mb-2">
```

```
 <li class="nav-item">
```

```
 Home
```

```


<li class="nav-item">
 Electronics

<li class="nav-item">
 Footwear

<li class="nav-item">
 Fashion

<form class="d-flex">
 <input type="text" class="form-control me-2">
 <button class="btn btn-warning">Search</button>
</form>
</div>
</div>
</nav>
<script src="../../node_modules/jquery/dist/jquery.js"></script>
<script src="../../node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
</body>
</html>

```

### List Group

```

.list-group
.list-group-item
.list-unstyled
.list-group-flush
.list-group-numbered
.list-group-horizontal
.list-group-item-primary, secondary, dark etc..

```

Ex:

```
<ul class="list-group list-group-numbered ms-3">
 <li class="list-group-item">Item-1
 <li class="list-group-item">Item-2
 <li class="list-group-item">Item-3
 <li class="list-group-item">Item-4

```

### Pagination

.pagination {sm | lg }

.page-item

.page-link

Ex:

```
<ul class="pagination ms-2 mt-3">
 <li class="page-item">«
 <li class="page-item">1
 <li class="page-item active">2
 <li class="page-item">3
 <li class="page-item">»

```

### Progress

.progress

.progress-bar

.progress-bar-striped

.progress-bar-animated

Ex:

```
<div class="mt-3">
```

```

<div class="progress">
 <div class="progress-bar progress-bar-striped progress-bar-animated bg-success"
style="width: 40%;">
 40% Completed
 </div>
 <div class="progress-bar progress-bar-striped bg-danger" style="width: 60%;">
 60% Remaining
 </div>
</div>

```

## Spinners

.spinner-border

.spinner-grow

Ex:

```

<div class="mt-3">
 <button class="btn btn-primary">
 Loading
 </button>
 <button class="btn btn-primary">
 Loading
 </button>
</div>

```

## Scrollspy

Data-bs-spy

Data-bs-offset

Tabindex

Ex:

```
<!DOCTYPE html>
```

```

<html>

<head>

 <title>Form</title>

 <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">

 <link rel="stylesheet" href="../node_modules/bootstrap-icons/font/bootstrap-icons.css">

</head>

<body class="container-fluid">

 <div class="row">

 <div class="col-3">

 <div id="topics" class="list-group">

 HTML

 CSS

 JavaScript

 jQuery

 </div>

 </div>

 <div class="col-9">

 <div data-bs-spy="scroll" data-bs-target="#topics" data-bs-offset="0" tabindex="0"
class="overflow-scroll" style="height: 400px;">

 <h3 id="html">HTML</h3>

 <p>Depending on how you obtained the Windows software, this is a license agreement
between (i) you and the device manufacturer or software installer that distributes the software with
your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business,
where your principal place of business is located, one of its affiliates) if you acquired the software
from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its
affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that
if you are a volume license customer, use of this software is subject to your volume license
agreement rather than this agreement.</p>

 <p>Depending on how you obtained the Windows software, this is a license agreement
between (i) you and the device manufacturer or software installer that distributes the software with
your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business,
where your principal place of business is located, one of its affiliates) if you acquired the software
from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its
affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that

```

if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<h3 id="css">CSS Tutorial</h3>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<h3 id="js">JavaScript</h3>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<h3 id="jq">jQuery</h3>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.</p>

</div>

</div>



</div>

<script src="../../node\_modules/jquery/dist/jquery.js"></script>

<script src="../../node\_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>

</body>

</html>











