

# UI FULL STACK WEBDEVELOPMENT NOTES

## INTRODUCTION:

A full stack web developer is a person who can develop both **client** and **server** software.

In addition to mastering HTML and CSS, he/she also knows how to:

- Program a **browser** (like using JavaScript, jQuery, Angular, or Vue)
- Program a **server** (like using PHP, ASP, Python, or Node)
- Program a **database** (like using SQL, SQLite, or MongoDB)

## Front End:

HTML4&5  
CSS2&3  
JAVA SCRIPTS  
ADVANCED JAVA SCRIPTS  
JQUERY  
ECMA 6  
BOOTSTRAP 5

## BACK END:

NODE JS  
EXPRESS JS  
SOCKET IO

## DATA BASE:

MANGO DB

## Frontend:

It only defines UI which a user or a client will unable to access the data in the database.

### **HTML (Hyper Text Markup Language):**

HTML is the language used to create the websites you visit every day, and it provides a logical way to structure content for websites. We can also define HTML as the language that helps in creating the backbone of any website.

Below mentioned are the basic HTML tags which divides the whole document into various parts like head, body etc.

- Every HTML document begins with a HTML document tag. Although this is not mandatory but it is a good convention to start the document with this below mentioned tag:  
`<!DOCTYPE html>`

- **<html>**: Every HTML code must be enclosed between basic HTML tags. It begins with **<html>** and ends with **</html>** tag.
- **<head>**: The head tag comes next which contains all the header information of the web page or document like the title of the page and other miscellaneous information. These information's are enclosed within head tag which opens with **<head>** and ends with **</head>**. The contents will of this tag will be explained in the later sections of course.
- **<title>**: We can mention the title of a web page using the **<title>** tag. This is a header information and hence mentioned within the header tags. The tag begins with **<title>** and ends with **</title>**
- **<body>**: Next step is the most important of all the tags we have learned so far. The body tag contains the actual body of the page which will be visible to all the users. This opens with **<body>** and ends with **</body>**. Every content enclosed within this tag will be shown on the web page be it writings or images or audios or videos or even links. We will see later in the section how using various tags we may insert mentioned contents into our web pages.

## CSS (Cascading Style Sheets):

CSS is used to stylize the HTML contents present on a website. This includes modifying the page colour, font-family, font-size, element positioning and more.

There are three types of CSS:

1. In a separate file (external)
2. At the top of a web page document (internal)
3. Right next to the text it decorates (inline)

## External Style Sheets:

Separate files having CSS instructions with the file extension (**.css**). The main advantage of using external style sheet is that you can change the whole website's style at once, without rewriting or modifying the style tag every page. Thus, saving a lot of time and energy. However, the external style sheet must be linked into the HTML file by using the tag between for making it work.

## Internal Styles:

Placed at the top of each web page document before any of the content is listed. The internal style CSS codes are written between the head tags in the of the HTML file itself. Internal styles are very easy to find and they are given the second highest priority, next to the external style sheets.

## Inline Styles:

Placed right where you need them, next to the text or graphics you wish to decorate. The inline styles can be inserted in the middle of the HTML code. This gives the real freedom to specify each web page element, however, can make the maintenance work of the website difficult.

## **JavaScript:**

JavaScript and Java are completely different languages, both in concept and design, and both have no correlation with each other. Java is an Object-Oriented Programming (OOP) language created by James Gosling of Sun Microsystems. JavaScript is a scripting language and was originally known as Live Script. JavaScript is used in front-end development whereas Java is used for back-end development in web development.

## **ECMA 6:**

ES6 stands for ECMAScript 6. ECMAScript was created to standardize JavaScript, and ES6 is the 6th version of ECMAScript, it was published in 2015, and is also known as ECMAScript 2015.

## **Bootstrap:**

Bootstrap is the most popular CSS Framework for developing responsive and mobile-first Websites.

## **jQuery:**

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API (Application Programming Interface) that works across a multitude of browsers.

## **Back End:**

Back-end Development refers to the server side of development where you are primarily focused on how the site works. ... This type of web development usually consists of three parts: a server, an application, and a database. Code written by back-end developers is what communicates the database information to the browser.

### **Node.JS:**

Node. Js is a JavaScript run-time environment built on Chrome's V8 JavaScript engine. It comes with an http module that provides a set of functions and classes for building a HTTP server. For this basic HTTP server, we will also be using file system, path and URL, all of which are native Node. □

## **Express.JS:**

Express. Js is a free and open-source web application framework for Node. Js. It is used for designing and building web applications quickly and easily.

4 | P a g e

Data Base (DB): A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). ... The data can then be easily accessed, managed, modified, updated, controlled, and organized.

## **Mongo DB:**

Mongo DB is a document database with the scalability and flexibility that you want with the querying and indexing that you need

**DATE:14/6/2021**

## **HIPERTEXT MARKUP LANGUAGE:**

A Predefined language using which we could be able to add content and sources to the field. It comes with the predefined set of html tags used to add different types of content accordingly.

Some of html tags:

- html -> Tag used to hold the complete content of page
- body -> To hold the actual content of page
- title -> Holds the page title
- p -> Paragraph tag used to hold multi line text content
- br -> Break tag to add a single line break
- div -> Div tag to hold block content
- span -> Inline element to hold content in same line
- table -> To render content in row and column way
- ul/ol -> To render content in ordered or unordered way
- li -> List
- h1 to h6 -> Heading tags to render content in the form of heading
- Etc.

### **HTML ATTRIBUTES:**

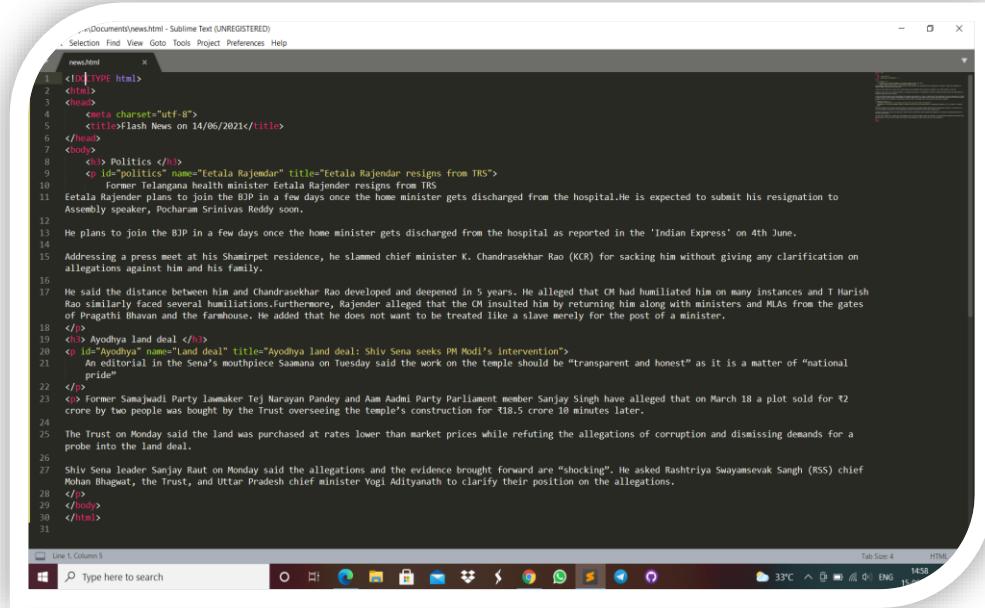
We can add extra information to the html elements through attributes. The attributes can be both predefined or user defined.

Following are the some of predefined attributes that can be added to the elements:

- id -> Using which we can add unique reference to elements
- name -> Name value can be added to elements
- class -> Using which we can add single/multiple CSS classes to elements
- style -> To add single or multiline CSS properties to elements
- alt -> Using which we can alternate text content to img tag

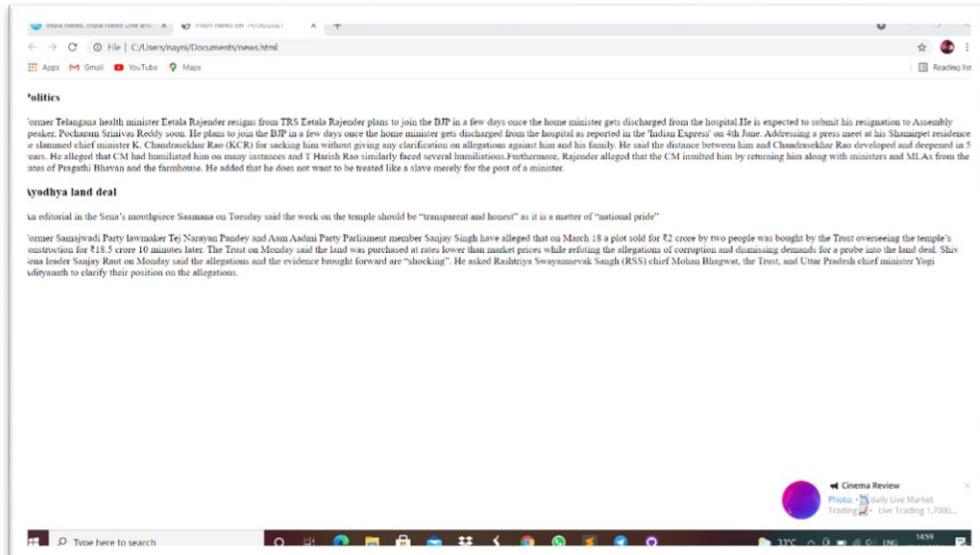
title -> using which we can add title to any html elements.  
Etc.

## Sample Webpage Program Using HTML attributes:



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Flash News on 14/06/2021</title>
</head>
<body>
<h2> Politics </h2>
<p id="politics" name="Eetal Rajendar" title="Eetal Rajendar resigns from TRS">
Former Telangana health minister Eetal Rajendar resigns from TRS
Eetal Rajendar plans to join the BJP in a few days once the home minister gets discharged from the hospital. He is expected to submit his resignation to Assembly speaker, Pocharam Srinivas Reddy soon.
He plans to join the BJP in a few days once the home minister gets discharged from the hospital as reported in the 'Indian Express' on 4th June.
Addressing a press meet at his Shamipet residence, he slammed chief minister K. Chandrasekhar Rao (KCR) for sacking him without giving any clarification on allegations against him and his family.
He said the distance between him and Chandrasekhar Rao developed and deepened in 5 years. He alleged that CM had humiliated him on many instances and T Harish Rao similarly faced several humiliations. Furthermore, Rajendar alleged that the CM insulted him by returning him along with ministers and MLAs from the gates of Pragathi Bhavan and the Farmhouse. He added that he does not want to be treated like a slave merely for the post of a minister.
</p>
<h2> Ayodhya land deal </h2>
<p id="Ayodhya" name="Land deal" title="Ayodhya land deal: Shiv Sena seeks PM Modi's intervention">
An editorial in the Sena's mouthpiece Saamana on Tuesday said the work on the temple should be "transparent and honest" as it is a matter of "national pride"
</p>
<p> Former Samajwadi Party lawmaker Tej Narayan Pandey and Aam Aadmi Party Parliament member Sanjay Singh have alleged that on March 18 a plot sold for ₹2 crore by two people was bought by the Trust overseeing the temple's construction for ₹18.5 crore 10 minutes later.
The Trust on Monday said the land was purchased at rates lower than market prices while refuting the allegations of corruption and dismissing demands for a probe into the land deal.
Shiv Sena leader Sanjay Raut on Monday said the allegations and the evidence brought forward are "shocking". He asked Rashtriya Swayamsevak Sangh (RSS) chief Mohan Bhagwat, the Trust, and Uttar Pradesh chief minister Yogi Adityanath to clarify their position on the allegations.
</p>
</body>
</html>
```

## OUTPUT:



**DATE:16/6/2021**

Today class is taken about GITHUB account:  
Steps to be followed to install this GitHub account  
/\*

Install Git on Windows

Navigate to the latest [Git for Windows installer](#) and download the latest version.

Once the installer has started, follow the instructions as provided in the **Git Setup** wizard screen until the installation is complete. Open the windows command prompt (or **Git Bash** if you selected not to use the standard Git Windows Command Prompt during the Git installation).

Type git version to verify Git was installed.

\*/

- 1.Browse to the official Git website: <https://git-scm.com/downloads>
2. Click the download link for Windows and allow the download to complete.



3. Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.

4. If you want you can allow the app to make changes to your device by clicking **Yes** on the User Account Control dialog that opens.

5. The installer will ask you for an installation location. Leave the default, unless you have reason to change it, and click **Next**.

6. A component selection screen will appear. Leave the defaults unless you have a specific need to change them and click **Next**.

7. The installer will offer to create a start menu folder. Simply click **Next**.

8. Select a text editor you'd like to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click **Next**.

After all the necessary steps taken (After installation process) Sign up to GitHub account using your Email-id and password (necessary details asked at the time of signup).

After that process has been completed open the GitHub Desktop on your pc such that you can clone the codes to your pc by using the other person repositories.

**Steps followed to clone the codes from other repositories (Durgaprasad sir repository):**

1.Open the GitHub Desktop on your pc.

2.Sign in to GitHub account (use your registered email-id and password)

3.Click on file on top left corner and click on the clone repository option available there.

4.After opening that option u can find the name of repository from which you want to clone the data available. Click on that.

5.Choose the path where you want to clone the data (Better to choose windows D or E).

6.Click on the clone option.

### **PUSH AND PULL THE CODES:**

#### **❖ What Does git push Do?**

git push updates the remote branch with local commits. It is one of the four commands in Git that prompts interaction with the remote repository. You can also think of git push as *update* or *publish*.

By default, git push only updates the corresponding branch on the remote. So, if you are checked out to the master branch when you execute git push, then only the master branch will be updated. It's always a good idea to use git status to see what branch you are on before pushing to the remote.

#### **❖ Common usages and options for git push**

- git push -f: Force a push that would otherwise be blocked, usually because it will delete or overwrite existing commits (*Use with caution!*)
- git push -u origin [branch]: Useful when pushing a new branch, this creates an upstream tracking branch with a lasting relationship to your local branch
- git push --all: Push all branches
- git push --tags: Publish tags that aren't yet in the remote repository

#### **❖ What Does git pull Do?**

##### **git pull**

git pull is one of the 4 remote operations within Git. Without running git pull, your local repository will never be updated with changes from the remote. git pull should be used every day you interact with a repository with a remote, at the minimum. That's why git pull is one of the most used Git commands.

#### **▪ git pull and git fetch**

git pull, a combination of git fetch + git merge, updates some parts of your local repository with changes from the remote repository. To understand what is and isn't affected by git pull, you need to first understand the concept of remote tracking branches. When you clone a repository, you clone one working branch, master, and all of the remote tracking branches. git fetch updates the remote tracking branches. git merge will update your current branch with any new commits on the remote tracking branch.

git pull is the most common way to update your repository.

#### **➤ Common usages and options for git pull**

- git pull: Update your local working branch with commits from the remote, *and* update all remote tracking branches.
- git pull --rebase: Update your local working branch with commits from the remote, but rewrite history so any local commits occur after all new commits coming from the remote, avoiding a merge commit.
- git pull --force: This option allows you to force a fetch of a specific remote tracking branch when using the <refspec> option that would otherwise not be fetched due to conflicts. To force Git to overwrite your current branch to match the remote tracking branch, read below about using git reset.
- git pull --all: Fetch *all* remotes - this is handy if you are working on a fork or in another use case with multiple remotes.

**REFERENCE:** <https://github.com/git-guides/git-clone>

#### **Some commands which relate to repository structure:**

- ❖ git add // transfers your project from working directory// to staging area.
- ❖ git commit // transfers your project from staging area to // Local Repository.
- ❖ git push// transfers project from local to central repository.// (requires internet)

/\*For Easy Reference:

1. Find a project you want to contribute to
2. Fork it
3. Clone it to your local system
4. Make a new branch
5. Make your changes
6. Push it back to your repo
7. Click the **Compare & pull request** button
8. Click **Create pull request** to open a new pull request\*/

**Date:18/06/2021**

#### CSS:

Cascading Style Sheets (CSS) is used to format the layout of a webpage. With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colours are to be used, different displays for different devices and screen sizes

#### **CSS**

#### **Syntax:**

A CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your

document.

A style rule set consists of a selector and declaration block.

```
Selector -- h1Declaration -- {color:pink;font size:12px;}  
/*Not Spoken Topic
```

## CSS SELECTORS:

CSS selectors are used to “find” (or select) HTML elements based on their element name, id, class, attribute.

There are six types of selectors in CSS:

**1.THE UNIVERSAL SELECTORS:** Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type.

**Example:**

```
* {  
color: #000000;  
}*  
/*
```

## OUTPUT:

Hello World!

These paragraphs are styled with CSS.

**2.THE ELEMENT SELECTOR:** The element selector selects elements based on the element name. You can select all p elements on a page like this.

**Example:**

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

## OUTPUT:

Every paragraph will be affected by the style.

Me too!

And me!

**3. THE DESCENDANT SELECTOR:** Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, the style rule will apply to the em element only when it lies inside the ul tag.

**Example:**

```
* ul em {  
color: #000000;  
}*  
}
```

**OUTPUT:**

Hello World!

These paragraphs are styled with CSS.

**4. THE ID SELECTOR:** The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within a page, so the id selector is used to select one unique element!To select an element with a specific id, write a hash (#) character, followed by the id of the element.The style rule below will be applied to the HTML element with id="para1":

**5. THE CLASS SELECTORS:** The class selector selects elements with a specific class attribute.To select elements with a specific class, write a period (.) character, followed by the name of the class.

**Example:**

```
* <p class="center large">This paragraph refers to two classes.</p>*
```

**OUTPUT:**

This heading will not be affected

This paragraph will be red and center-aligned.

This paragraph will be red, center-aligned, and in a large font-size.

**6. GROUPING SELECTORS:** If you have elements with the same style definitions,

```
*h1 {  
text-align: center;  
color: blue;  
}  
h2 {  
text-align: center;  
color: blue;  
}  
p {  
text-align: center;
```

```
color: blue;  
} *
```

it will be better to group the selectors, to minimize the code. To group selectors, separate each selector with a comma.

```
*h1, h2, p {  
text-align: center;  
color: red;  
}*
```

#### OUTPUT:

Hello World!

Smaller heading!

This is a paragraph.

Not	Spoken	Topic*/
-----	--------	---------

#### Applications of CSS:

- **CSS saves time** - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** - Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

Styles can be added in three following ways:

#### Inline, Internal, and External.

#### 1. Inline CSS:

For inline styles, we use the style attribute of the HTML tags. CSS is passed as a string to the style attribute which adds the styles to the tags.

For example: If we want to make our web page background pink with inline CSS, we can write something like this:

```
<body style="background-color: pink"></body>
```

## 2.Internal CSS

For Internal CSS, a style tag is used in which all the styles related to the webpage are added. This style tag is added in the head tag of the page so that the styles are added even before the HTML document is rendered.

Example:

```
<head>
  <style>
    body {
      background-color: pink;
    }
  </style>
</head>
```

## 3.External CSS

In external CSS we use a separate file with a .css extension where we write all our styles. This CSS file can be used by multiple webpages by using a link tag which is added under the head tag.

Example:

1.First, we will add all our styles to our CSS file, for our example, it will be styles.css

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

2.Then we will link this CSS file to our html file as follows:

```
<head>
<link rel="stylesheet" href="styles.css">
</head>
```

### ★ CSS Colors, Fonts and Sizes:

The CSS **color** property defines the text color to be used.

The CSS **font-family** property defines the font to be used.

The CSS **font-size** property defines the text size to be used.

## EXAMPLE PROGRAM FOR CSS:

```

C:\Users\nayni\Documents\news.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
news.html login.html Audio.html ssids.py meg.java layout.html news.html
1 <html>
2   <head>
3     <title>
4       News Page
5     </title>
6     <style type="text/css">
7       .newsTitle {
8         background: red;
9         color: green;
10        width: 300px;
11      }
12      .newsContent {
13        background: pink;
14        font-size: 20px;
15      }
16    </style>
17  </head>
18  <BODY>
19  <!--
20    This page is to demonstrate css classes and tags...
21
22
23  -->
24
25  <h3 class="newsTitle">New Item 2:</h3>
26  <p id="specialItem2" class="newsContent" storytype="education" dataContnet="newsscontentlist">
27    Shops, malls and restaurants in
28    Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken if the Covid numbers rise. Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.
29  </p>
30
31
32  <h3 class="newsTitle">New Item 3:</h3>
33  <p style="background-color: yellow; border: 2px solid black; padding: 5px; font-size: 18px;">
34    Shops, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken if the Covid numbers rise. Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.
35
36
37
38
39
40
41
42
43

```

Line 9, Column 30

31°C 10:11 ENG 18-06-2021

File Edit Selection Find View Goto Tools Project Preferences Help

news.html login.html Audio.html ssids.py meg.java layout.html news.html

RESTAURANTS -- WHICH WERE OPEN ONLY FOR TAKEAWAYS AND HOME DELIVERIES -- CAN NOW HAVE DINERS BUT WITH ONLY 50 PER CENT OF SEATING CAPACITY.

35 </p>
36 <h3 class="newsTitle">New Item 4:</h3>
37 <p style="background-color: pink;font-size:20px;">
38 Shops, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken if the Covid numbers rise. Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.

39
40
41
42
43

Line 9, Column 30

31°C 10:14 ENG 18-06-2021

## OUTPUT:

```

news today - Google Search | News Page
← → ⌂ File | C:/Users/nayni/Documents/news.html
_apps Gmail YouTube Maps
New Item 2:
Shops, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken if the Covid numbers rise. Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.
New Item 3:
Shops, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken if the Covid numbers rise. Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.
New Item 4:
Shops, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken if the Covid numbers rise. Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.

```



**Date: 21/06/2021**

## CSS CLASSES:

A concept of grouping all the required CSS properties as an individual block (under the {}) assigning a user defined name to it is called a CSS classes.

- 1.Every CSS class name should always start with. Operator
- 2.in a single page we can define any number of CSS classes
3. style is a predefined tag capable of holding any number of CSS classes.
4. class is a predefined attribute using which we could be able to inject one or multiple CSS classes to a single html element.
5. once a CSS class been defined it can be used any number of times to any number of elements.

syntax:

```
<style>
  .classname {
    ...
    ...
    .... /Set of CSS property classes
}
</style>
```

Topic: COLORS IN HTML: -

**HTML color codes** are hexadeciml triplets representing the colours red, green, and blue (#RRGGBB).

For example, in the color red, the color code is #FF0000, which is '255' red, '0' green, and '0' blue. These color codes can change the color of the background, text, and tables on a web page.

Here are some of the color codes mentioned for reference:

Color Name	Color Code	Color Name	Color Code
Red	#FF0000	White	#FFFFFF
Cyan	#00FFFF	Silver	#C0C0C0
Blue	#0000FF	Gray or Grey	#808080
DarkBlue	#00008B	Black	#000000
LightBlue	#ADD8E6	Orange	#FFA500
Purple	#800080	Brown	#A52A2A
Yellow	#FFFF00	Maroon	#800000
Green	#00FF00	Green	#008000
Magenta	#FF00FF	Olive	#808000
Pink	#FFC0CB	Aquamarine	#7FFFDD

The <body> tag has following attributes which can be used to set different colours –

- **bgcolor** – sets a color for the background of the page.

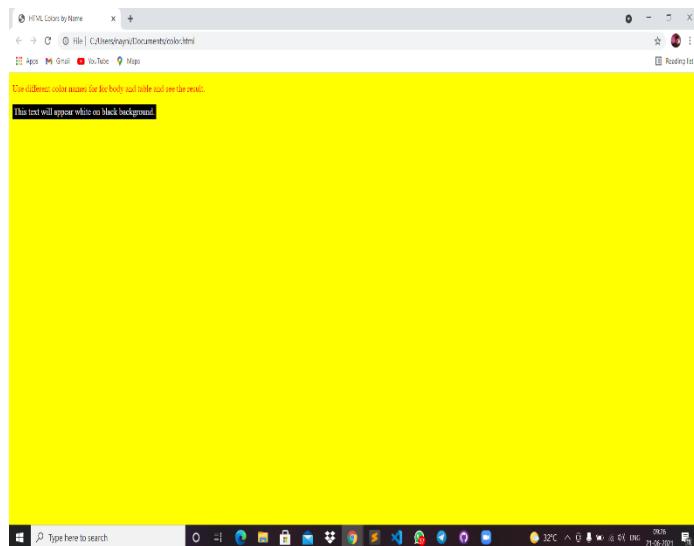
- **text** – sets a color for the body text.
  - **alink** – sets a color for active links or selected links.
  - **link** – sets a color for linked text.
  - **vlink** – sets a color for *visited links* – that is, for linked text that you have already clicked on

## Example program:

```
[1] C:\Users\9192\OneDrive\Desktop\1.html for 05W88ST98D
File Edit View Insert Tools Help Preferences Help

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML colors by names</title>
5   </head>
6
7   <body style="text-align: center; background-color: yellow;">
8     <p>Use different color names for body and table and see the result.</p>
9
10    <table border="1" style="width: 100%; border-collapse: collapse; background-color: black;">
11      <tr>
12        <td>
13          <span style="color: white;">This text will appear white on black background.</span>
14        </td>
15      </tr>
16    </table>
17
18  </body>
19
20 </html>
```

## OUTPUT:

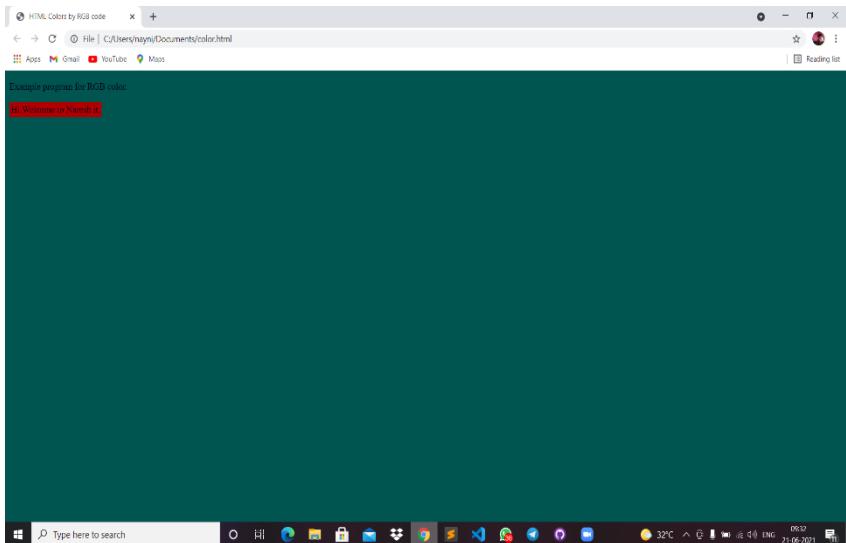


# HTML Colors - RGB Values

This color value is specified using the `rgb()` property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

---

## OUTPUT:



## COLOR: HASHCODES:-

Hex color code contains the symbol “#” followed by 6 alphabets or numbers. Numbers are in the hexadecimal numeric system. **00** value range of the code represents the lowest intensity of color on the other hand **FF** value range of code represents the highest intensity.

- ❖ The **white color** is a mixture of the three primary colours at full intensity representing the Hex color code **#FFFFFF**.
  - ❖ The **black color** is a mixture of the three primary colours at the lowest intensity representing the color code **#000000**.

EXAMPLE: - color: #FFFFFF

Note: if anytime all 6 characters are same, we can give 3 characters.

/\*Reference – geeksforgeeks, Tutorial point, W3schools\*/

ASCII code:

ASCII stands for the "American Standard Code for Information Interchange". ASCII is a 7-bit character set containing 128 characters. It contains the numbers from 0-9, the upper- and lower-case English letters from A to Z, and some special characters. The character sets used in modern computers, in HTML, and on the Internet, are all based on ASCII.

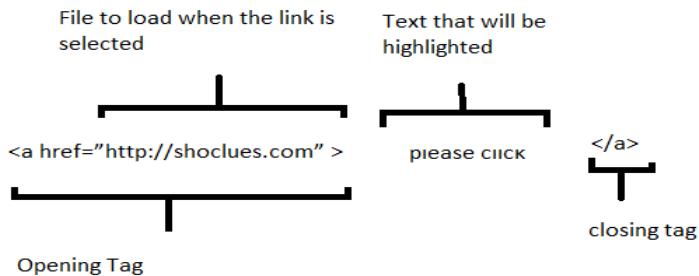
Also let us discuss about some of the anchor tags:

#### Anchor tag:

The **HTML anchor tag** defines a *hyperlink that links one page to another page*. It can create hyperlink to other web page as well as files, location, or any URL. The "href" attribute is the most important attribute of the HTML a tag. and which links to destination page or URL. use to make links like third party page like amazon, flipkart etc...

**NOTE:** - Href is used for third party page opening external and internal web page.

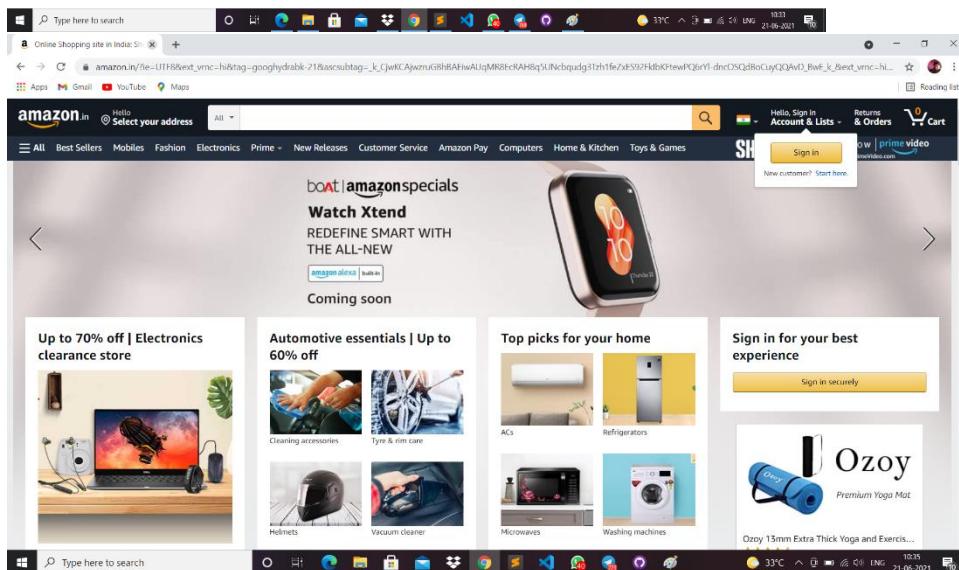
<a href="http://shoclues.com" >please click </a>



A screenshot of the Sublime Text editor showing the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
<title>Image Tag</title>
</head>
<body>
<p><a href="https://www.amazon.in/?ie=UTF8&__mk_Uncode=&tag=goohydrat-21&ascsubtag=_K_CjwKAjzruGhBAF1wUqPRBFcRA8qSUNchqudp5Tzh1fexZFS92fk1hFfew
pjqE1_dnc7Q9R0cuyQQUVO_Buf_k_Kext_vmc-h1&gclid=CjwKCAjzruGhBAF1wUqPRBFcRA8qSUNchqudp5Tzh1fexZFS92fk1hFfew
This link </a>to open a website.</p>
</body>
</html>
```

OUTPUT:



## Pseudo classes:

A pseudo-class is used to define a special state of an element. For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

## Target attribute:

The **target** attribute specifies where to open the linked document.  
**target=" \_blank"** open in very blank page.

## Underlining text:

The **<u>** tag in HTML stands for underline and it's used to underline the text enclosed within the **<u>** tag. This tag is generally used to underline misspelled words. This tag requires a starting as well as ending tag.

## EXAMPLE:

```
<html>
```

```

<head>
    <title>u Tag</title>
    <style>
        body {
            text-align:center;
        }
        .meg {
            font-size:40px;
            font-weight:bold;
            color:green;
        }
        .project {
            font-size:25px;
            font-weight:bold;
        }
        p {
            font-size:20px;
        }
    </style>
</head>
<body>
    <div class = "meg">Meghana </div>
    <div class = "project"><u> Tag</div>
    <p>Meghana <u>computer science</u>
        project</p>
</body>
</html>

```

OUTPUT:



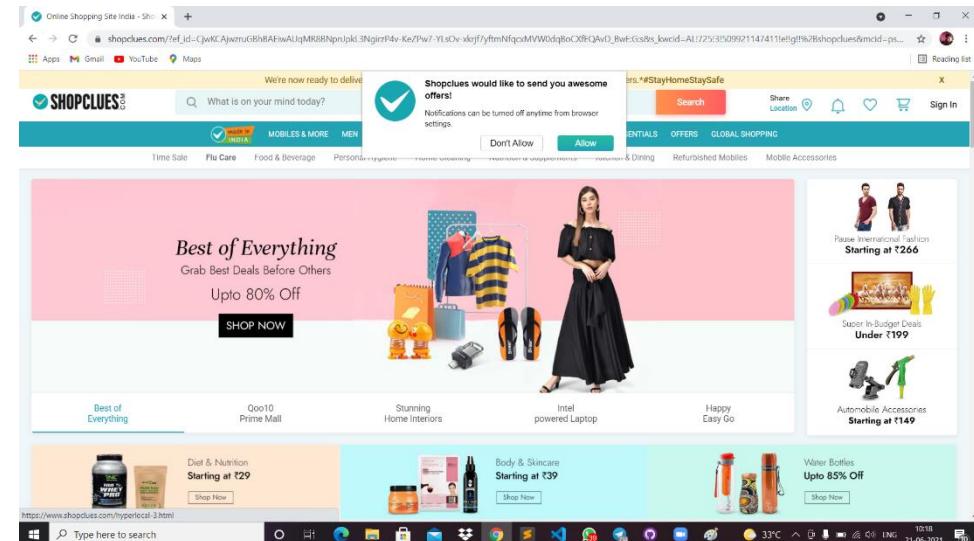
## Adding images to web page:

**img tag: src attribute:**

Syntax: 
<html>
<body>
<p>Click on the image to open web page.</p>
<!-- anchor tag starts here -->
<a href="https://www.shopclues.com/?ef_id=CjwKCAjwruGBhBAEiwAqM8B8NpnJpk3NgirP4v-KzPw7-YLsOv-xkrjI7yfmlNfqxMNdqBoCXFEQAvD_BwE:Gisks_kwcid=AL!725!315
09921147411elIg!!1%2Bshopclues&mcid=p&utm_source=Google&utm_medium=cpc&utm_campaign=Search_Text_RLSA_BMW_Br
_and_Shopclues&gclid=CjwKCAjwruGBhBAEiwAqM8B8NpnJpk3NgirP4v-KzPw7-YLsOv-xkrjI7yfmlNfqxMNdqBoCXFEQAvD_BwE">
<img alt="https://www.shopclues.com/g/p polycarbonate gym bottle grey set of 2.html"
      width="300"
      height="250"
/>
</a>
<!-- anchor tag ends here -->
</body>
</html>

```

**OUTPUT: -**



**DATE: 22/06/2021**

**GitHub:**

GitHub is an independent platform for full stack users basically helpful for cloning or pushing or pulling the data origin from the repositories URL address.

What is cloning? How it's used?

Cloning a GitHub repository creates a local copy of the remote repository. This allows you to make all of your edits locally rather than directly in the source files of the origin repository.

*/\*Note: U have to have access to get the code from the repository member if its private repository. That is, they have to add us a collaborator. \*/*

Steps for cloning:

After getting the access as a collaborator:

1. Open the file explorer and create a separate folder in D or E Drive (UIClasses file as we are cloning the data for UI classes from Durgaprasad sir).

2. Then Open GitHub desktop (which is installed) and check whether the collaborator repository (Durgaprasad sir repository) is available or not.

3. If available click on that choose the path (the folder you have created(UIClasses folder in D or E Drive)). Then simply click on the clone option shown on the GitHub Desktop.

What is push and pull origins?

Push origin:

Used to push the assignment after fetching the data

Pull origin:

Used to pull the origins or files or codes of others (If any changes were made and resubmitted again) by other collaborators.

## Submitting an assignment in GitHub: -

open GitHub account and first fetch the origin. After fetching you will get a notification to pull the origin then click on that so that what you have submitted gets saved in the GitHub. Then see on down left corner ull see a comment window keep the title as Assignment1\_studentdetails and click on commit to master.

Again, ull see option on top like pull the origin do that. After all these processes completed open GitHub in chrome browser ull see Durgaprasad sir folder click on that open 7:30Am batch see whether Ur name is registered there or simplest method is open file explorer and go to D:drive and click on uiclasses(or the name ull saved) then open 7:30 Am and click on assignment folder created check Ur name is shown or not.

/\*Avoid giving spaces between the file you are saving\*/  
/\*This will be useful for all windows 64-bit compilers\*/

DATE: 23/06/2021

Gmail: [webdev.prasad@gmail.com](mailto:webdev.prasad@gmail.com)

Assignment 1 – Batch 7Am (DOM)

Body: file location

Notes:

Different ways of injecting CSS to an element:

1. Adding Inline CSS through style Attribute. Use. (dot) operator
2. Creating a class, injecting through class attribute.
3. Creating a class through id of an element. Use # (hash) operator

4. Creating a CSS class using the tag name itself. No need to make use of any operators.

- Creating a class through id of an element:

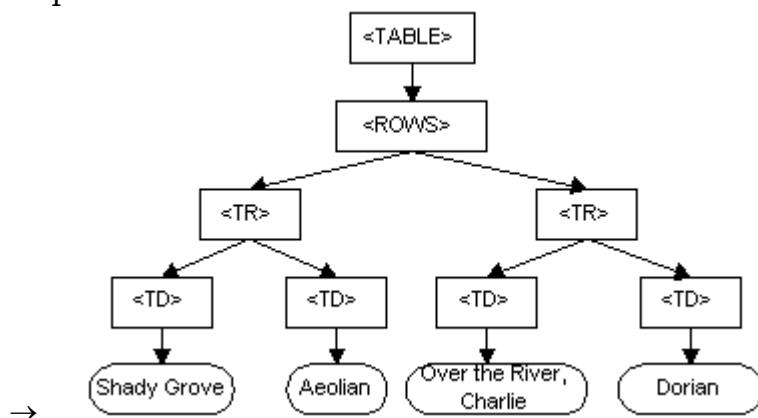
If there is an element for which we are adding CSS and there is id attribute already injected then we can define a class through id of the element using # operator.

DOM Structure (DOCUMENT OBJECT MODEL): -

The Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. In the DOM specification, the term "document" is used in the broad sense - increasingly, XML is being used as a way of representing many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM may be used to manage this data.

- For Every web page browser dynamically creates a DOM structure which indicates the tree structure of current webpage with all the elements.
- DOM structure specifies the relationship between the elements, list of elements participated within the page, attributes and its corresponding values etc.

- Example:



## DEBUGGER TOOL:

Every browser by default comes with a default debugger tool using which we could be able to debug any webpage running under the browser.

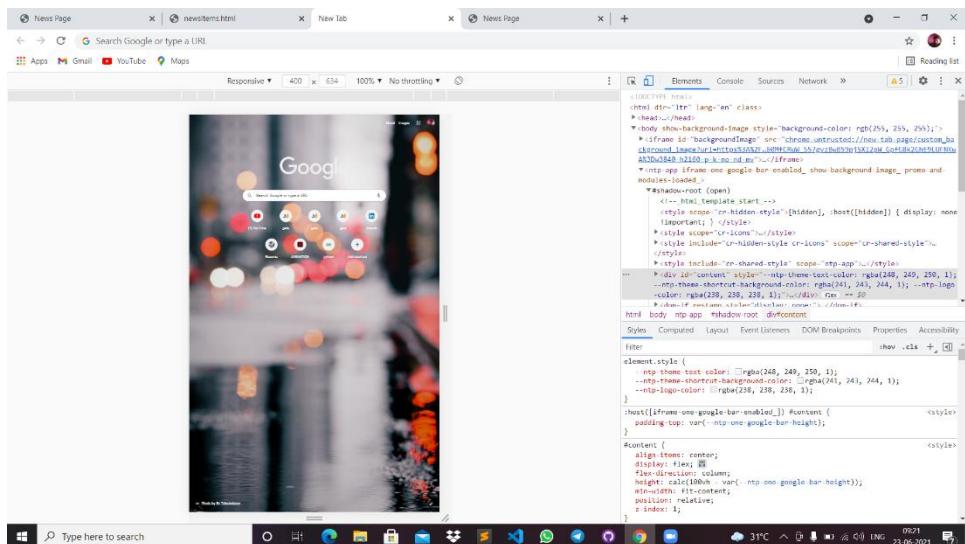
1. It shows the DOM structure of the current webpage.
2. It provides an option to create a dynamic HTML element while the page is running.

3. It provides an option to edit, delete, update any HTML element and its corresponding attributes while the page is running.
4. Any CSS property of any HTML element can be edited, deleted, add a new property can be done through debugging tool.
5. It also provides an option to explore list of resources been loaded in the current webpage.
6. It provides an option to explore list of resources getting loaded through network or server (images, js files, CSS files, XHR requests etc.)

*/\* It is a temporary changeable tool\*/*

*/\* It is one of very important tool for UI development\*/*

Example of Debugging window:



## TODAY'S EXAMPLE:

```

<!DOCTYPE html>
<html>
  <head>
    <title>News Page</title>
    <style type="text/css">
      h3 {
        background: red;
        color: green;
        width: 300px;
      }
      .news_content {
        background: pink;
        font-size: 20px;
      }
      #special_news {
        background: blue;
        font-weight: bold;
      }
      body {
        background: #ccc;
      }
    </style>
  </head>
  <body>
    <h2>New Item 1</h2>
    <div class="news_content">
      Shops, malls and restaurants in
      Delhi
      will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the
      current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken
      if the Covid numbers rise. Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for
      takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.
    </div>
    <h2>New Item 2</h2>
    <div class="news_content">
      Shops, malls and restaurants in
      Delhi
      will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the
      current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken
    </div>
  </body>
</html>

```

```

39 if the Covid numbers rise.Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for
40 takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.
41 </div>
42 <div id="special_News">
43 Shops, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be
44 open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a
45 week and strict action will be taken if the Covid numbers rise.Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said.
46 Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.
47 </div>
48 <div id="New_Item_4">
49 <div class="content">
50 Shop, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be
51 open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a
52 week and strict action will be taken if the Covid numbers rise.Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said.
53 Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.
54 </div>
55 <div id="Student_detail">
56 Name: raj
57 </div>
58 </body>
59 </html>

```

## OUTPUT:

New Item 1

Shops, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken if the Covid numbers rise.Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.

New Item 2

Shops, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken if the Covid numbers rise.Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.

New Item 3

Shops, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken if the Covid numbers rise.Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.

New Item 4

Shops, malls and restaurants in Delhi will open from today as Covid numbers in the national capital drop to a three-month low. Shops will be open seven days a week instead of the current odd-even system. Chief Minister Arvind Kejriwal, however, said this will be on trial basis for a week and strict action will be taken if the Covid numbers rise.Shop timings will remain the same, from 10 am to 8 pm, Mr Kejriwal said. Restaurants -- which were open only for takeaways and home deliveries -- can now have diners but with only 50 per cent of seating capacity.

Student details

Name: raj

## Assignment: Practice with debugging tool.

DATE: 24/06/2021

Debugger tool:

Debugging is the process of finding and fixing errors within a script. All modern browsers and most other environments support debugging tools – a special UI in developer tools that makes debugging much easier. It also allows to trace the code step by step to see what exactly is going on.

Mainly debugging tool consists of 8 parts:

## 1.Elements

Add attribute — add a new attribute to the selected element  
Edit attribute — edit an attribute, available only when you click on the attribute

Edit as HTML — by picking this one, you can edit the whole element; also, useful to copy part of an element that you want to use somewhere else

Copy:

Copy outer HTML — copy tag, including a tag itself and a child element

Copy selector — copy of a CSS selector (div > span > #id)

Copy XPath — copy of an XPath //\*[@id="answer11208745-20"]/div/div

[3]/time, more in further reading

Cut element — cuts element

Copy element — copy element and children's elements

Hide element — hide element temporarily by adding display: none; (cmd + H / ctrl+H)

Delete element — delete element and children's elements, can be reversed by cmd + z

Expand all — expand all nodes

Collapse all — collapse all nodes

: active — set element in active state\*

: hover — set element in hover state\*

: focus — set element in focus state\*

: visited — set element in visited state\*

Scroll into view — gets you immediately to the selected element on web page

2.Resources

3.Network

4.Sources

The Sources panel has 3 parts:

The **File Navigator** pane lists HTML, JavaScript, CSS and other files, including images that are attached to the page. Chrome extensions may appear here too.

The **Code Editor** pane shows the source code.

The **JavaScript Debugging** pane is for debugging,

5.Timeline

6.Profile

7.Audits

8.Console

If we press Esc, then a console opens below. We can type commands there and press Enter to execute.

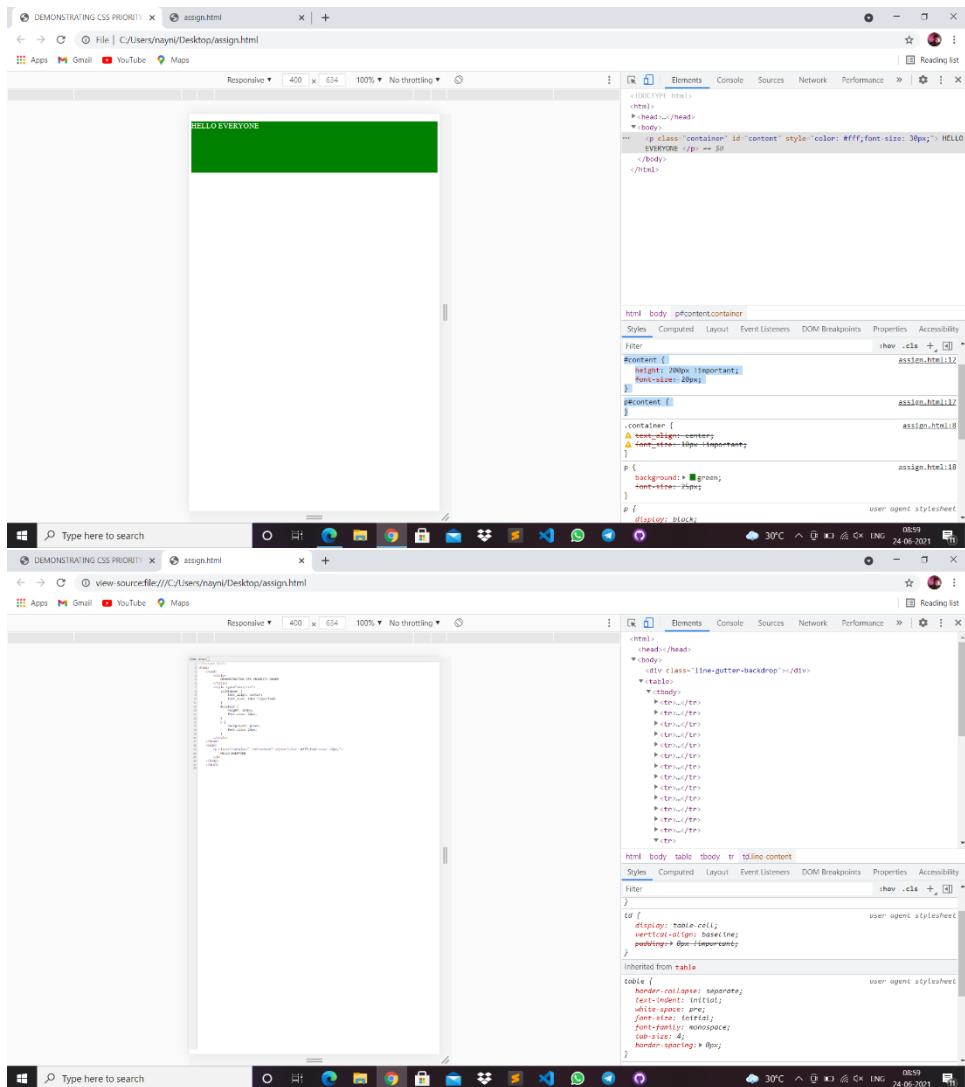
After a statement is executed, its result is shown below.

For example, here `1+2` results in `3`, and `hello("debugger")` returns nothing, so the result is undefined:

## EXAMPLE:

```
<!Doctype html>
<html>
<head>
    <title>
        DEMONSTRATING CSS PRIORITY ORDER
    </title>
    <style type="text/css">
        .container {
            text_align: center;
            font_size: 10px !important;
        }
        #content {
            height: 200px !important;
            font-size: 20px;
        }
        p {
            background: green;
            font-size: 25px;
        }
    </style>
</head>
<body>
    <p class="container" id="content" style="color: #fff; font-size: 30px;">
        HELLO EVERYONE
    </p>
</body>
</html>
```

## OUTPUT:



## CSS priorities:

As we have the 4 different ways to apply CSS properties for any HTML element, if the same CSS property being applied to a single element using all the 4 different ways with different values, browser by default follows the below priority order.

1. Among all the 4 different ways CSS been applied inline (through style attribute) will always takes higher priority in order.
2. CSS been applied through id based takes second priority in order.
3. CSS been applied through class takes third priority in order.
4. CSS applied through tag-based class takes least priority in order.

Note: - Irrespective of above default priority order any CSS property being added with! important will always takes highest priority in order.

**DATE:25/06/2021**

## Interview Questions

### ❖ Inline and block level elements:

All the html elements are been categorised into two types:

- 1.Block level element
- 2.Inline element

1.Block level element: Any HTML element which comes under the block level category holds the following properties:

- It occupies 100% width of its container by default
- Even though it occupies 100% width we can still control the dimensions of block level elements through CSS width and height properties.
- While getting rendered with in the page by default it comes to new line and gets rendered.
- The element which is following a block level element also automatically comes to a new line and gets rendered.
- Block level elements are mainly used to hold group of relative items as lightly individual block.
- `<div>` tag is best example for block level element.

The screenshot shows a Sublime Text 3 interface with the following code in the 'block.html' file:

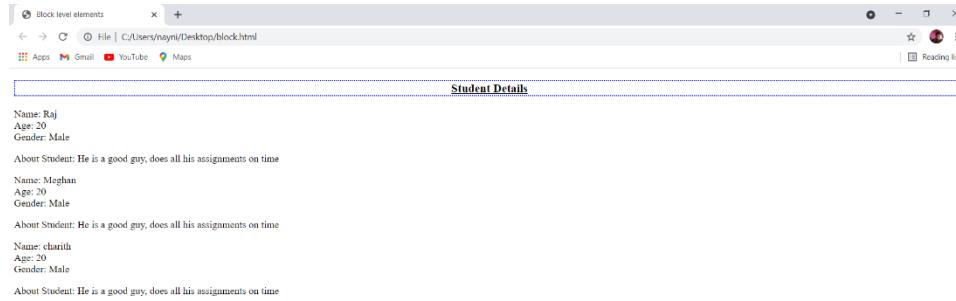
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Block level elements </title>
5     <style type="text/css">
6       h3 {
7         text-align: center;
8         text-decoration: underline;
9         border: 2px dotted blue;
10      }
11    </style>
12  </head>
13  <body>
14    <h3> Student Details </h3>
15    <div>
16      Name: Raj <br/>
17      Age: 20 <br/>
18      Gender: Male <br/>
19      <p> About Student: He is a good guy, does all his assignments on time </p>
20    </div>
21    <div>
22      Name: Meghan <br/>
23      Age: 20 <br/>
24      Gender: Male <br/>
25      <p> About Student: He is a good guy, does all his assignments on time </p>
26    </div>
27    <div>
28      Name: charith <br/>
29      Age: 20 <br/>
30      Gender: Male <br/>
31      <p> </p>
32    </div>
33  </body>
34 </html>
```

The screenshot shows a Sublime Text window with the file 'block.html' open. The code defines a CSS style for 'blocklevel' and a block-level element 'block2html'. It contains three 

blocks, each with a heading and some descriptive text. The Sublime Text interface includes tabs for 'block.html' and 'inline.html', a status bar at the bottom, and a taskbar at the bottom.

```
File Edit Selection Find View Goto Tools Project Preferences Help
C:\Users\nayni\Desktop\block.html - Sublime Text (UNREGISTERED)
block.html
block2html
inline.html
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
<style>
  blocklevel {
    text-align: center;
    text-decoration: underline;
    border: 2px dotted blue;
  }
</style>
</head>
<body>
  <div>
    Name: Raj <br/>
    Age: 20 <br/>
    Gender: Male <br/>
    <p>
      About Student: He is a good guy, does all his assignments on time
    </p>
  </div>
  <div>
    Name: Meghan <br/>
    Age: 20 <br/>
    Gender: Male <br/>
    <p>
      About Student: He is a good guy, does all his assignments on time
    </p>
  </div>
  <div>
    Name: charith <br/>
    Age: 20 <br/>
    Gender: Male <br/>
    <p>
      About Student: He is a good guy, does all his assignments on time
    </p>
  </div>
</body>
</html>
```

## OUTPUT:



## Example 2:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Demonstrating block level elements
5     <style type="text/css">
6       .container {
7         background: blue;
8         width: 300px;
9         height: 300px;
10      }
11      p {
12        background: yellow;
13        width: 300px;
14        height: 30px;
15      }
16    </style>
17  </head>
18  <body>
19    Item 1
20    Item 2
21    <div class="container"> Item 3</div>
22    Item 4
23    Item 5
24    <p>Item 6</p>
25    Item 7
26  </body>
27 </html>

```

## OUTPUT:



2. Inline element: Any HTML element comes under inline category holds following properties:

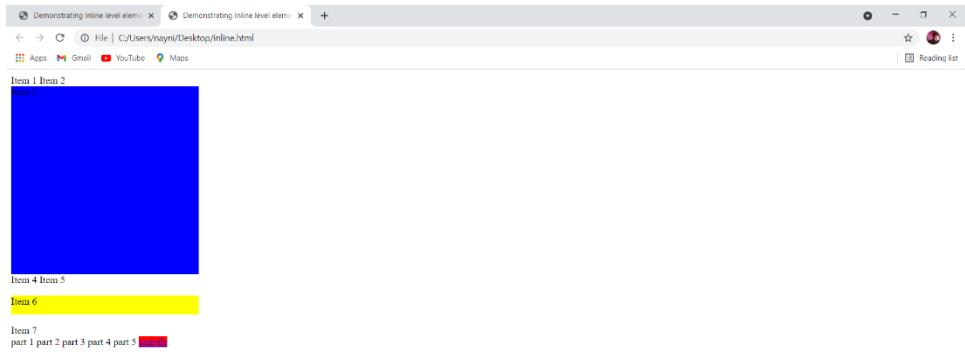
- While rendering on the page all the inline elements try to render within the same line.
  - Inline element always occupies the width within the container based on the content it is holding.
  - We cannot control the dimensions of inline element.
  - ‘span’ tag is the best example for inline element.
- /\* Any element in inline block doesn't occupy new line\*/

The screenshot shows the Sublime Text editor with an open file named 'inline.html'. The code is as follows:

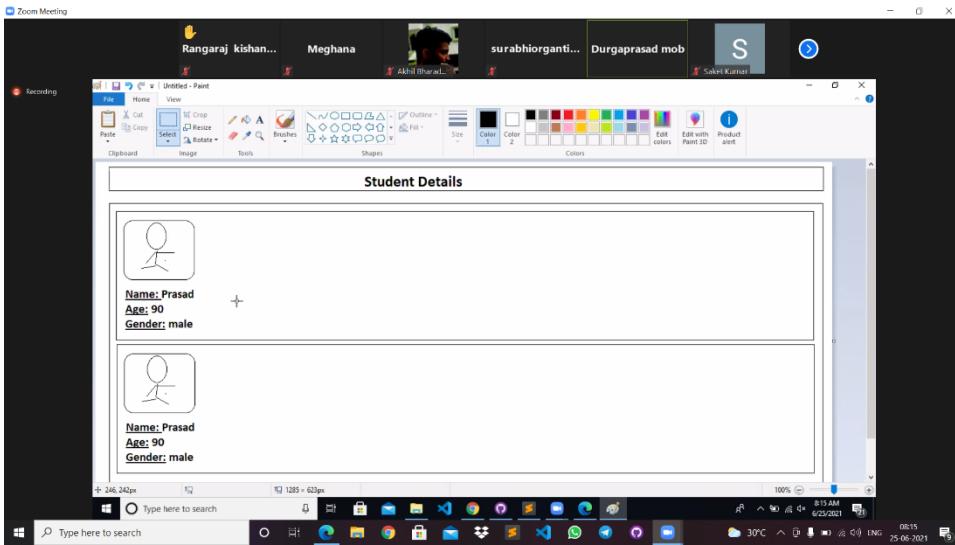
```
<!DOCTYPE html>
<html>
<head>
<title> Demonstrating inline level elements </title>
<style type="text/css">
    .container {
        background: blue;
        width: 300px;
        height: 300px;
    }
    p {
        background: yellow;
        width: 300px;
        height: 30px;
    }
    a {
        background: red;
        width: 60px;
    }
</style>
</head>
<body>
    Item 1
    Item 2
    <div class="container"> Item 3</div>
    Item 4
    Item 5
    <p>Item 6</p>
    Item 7
    <br/>
    part 1
    part 2
    <span class="part 3">part 3</span>
    part 4
    part 5
    <a href="http://abc.com">sample</a>
</body>
</html>
```

Line 1, Column 16

## OUTPUT:



## ASSIGNMENT-2:



\*Minimum of 10 students\*

Overflow:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>overflow property</title>
5     <style type="text/css">
6       .container {
7         border: 2px solid;
8         width: 500px;
9         height: 450px;
10        background-color: #ccc;
11        overflow: auto;
12      }
13      .block {
14        width: 150px;
15        height: 150px;
16        border: 1px;
17        text-align: center;
18      }
19    </style>
20  </head>
21  <body>
22    <div class="container">
23      <div class="block">Block1</div>
24      <div class="block">Block2</div>
25      <div class="block">Block3</div>
26      <div class="block">Block4</div>
27      <div class="block">Block5</div>
28    </div>
29  </body>
30 </html>
31
32
33
34

```

OUTPUT:



### **Span Tag:**

- The span tag is an inline container used to mark up a part of text, or a part of document.
- It is easily styled by the CSS or manipulated with JavaScript using the class or id attribute.
- The <span> tag is much like the <div> element, but <div> is a block-level element and <span> is an inline element.

NOTE: The <span> tag also supports the Global attribute in HTML.

And also supports the Event attribute.

### **Div Tag:**

- ★ The div tag is known as Division tag. The div tag is used in HTML to make divisions of content in the web page like (text, images, header, footer, navigation bar, etc).
- ★ Div tag has both open(<div>) and closing (</div>) tag and it is mandatory to close the tag.
- ★ The Div is the most usable tag in web development because it helps us to separate out data in the web page.
- ★ we can create a particular section for particular data or function in the web pages.

Properties	Div Tag	Span Tag
Elements Types	Block-Level	Inline
Space/Width Available	Contain Whole Width	Takes only required Width
Examples	Headings, Paragraph, form	Attribute, image
Uses	Web-layout	container for some text
Attributes	Not required, with common css, class	Not required, with common css, class

### CSS Overflow: -

The CSS overflow property controls what happens to content that is too big to fit into an area.

The overflow property has the following values:

- Visible - Default. The overflow is not clipped. The content renders outside the element's box
- div {  
width: 200px;

- ```

height:          50px;
background-color: #eee;
overflow:        visible;
}

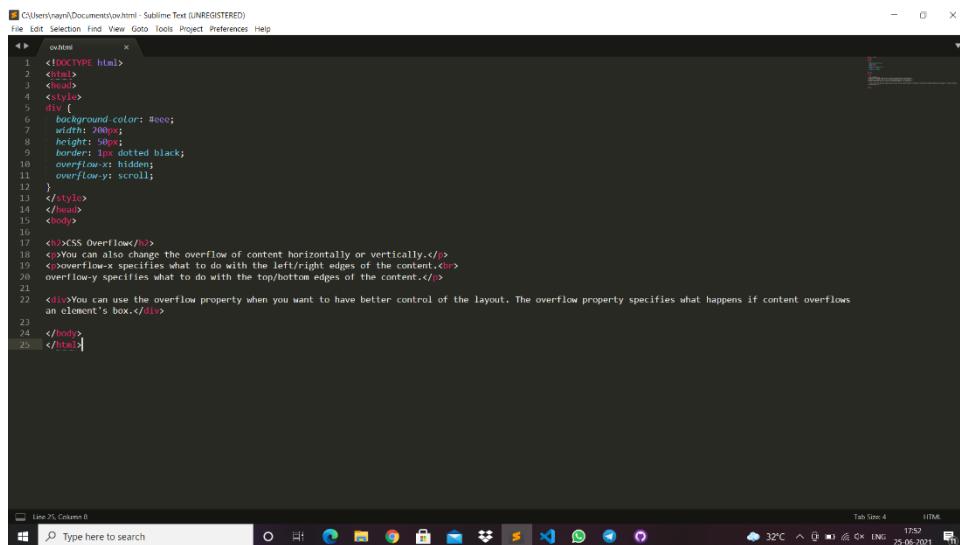
• hidden - The overflow is clipped, and the rest of the content will be invisible
• 
• scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content
• 
• auto - Similar to scroll, but it adds scrollbars only when necessary
• 
```

### **Overflow-x and Overflow-y:**

The Overflow-x and Overflow-y properties specifies whether to change the overflow of content just horizontally or vertically (or both):

1. Overflow-x specifies what to do with the left/right edges of the content.
2. Overflow-y specifies what to do with the top/bottom edges of the content.

### **EXAMPLE:**

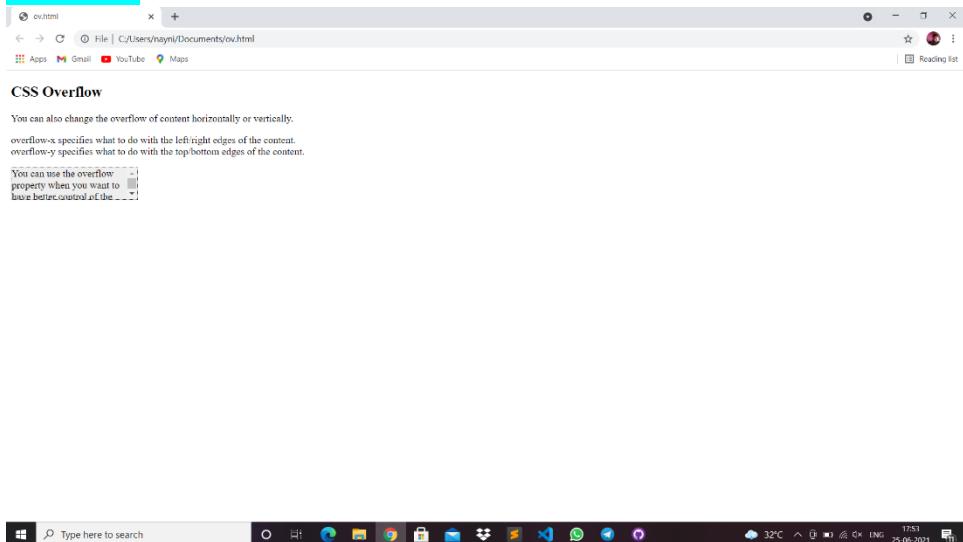


```

File Edit Selection Find View Goto Tools Project Preferences Help
ov.html <FILETYPE html>
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 div {
6   background-color: #eee;
7   width: 100px;
8   height: 50px;
9   border: 1px dotted black;
10  overflow-x: hidden;
11  overflow-y: scroll;
12 }
13 </style>
14 </head>
15 <body>
16
17 <!--CSS Overflow-->
18 <p>You can also change the overflow of content horizontally or vertically.</p>
19 <p><b>overflow-x</b> specifies what to do with the left/right edges of the content.</p>
20 <p><b>overflow-y</b> specifies what to do with the top/bottom edges of the content.</p>
21
22 <p>You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.</p>
23
24 </body>
25 </html>

```

## OUTPUT:



DATE:26/06/2021

### CSS PADDING AND MARGIN properties:

By default, while the elements getting rendered on the page, they try to occupy very next line of the previous element without maintaining any space. In order to add space or gap between or within the elements we make use of the CSS property called as 'Padding' or 'Margin'.

#### CSS 'Padding' Property: -

Through padding property, we could able to add space or gap between the border and content of the container.

- Syntax: padding: 10px; Adds padding of 10px to all directions (top, left, right and bottom)
- Padding: 5px 10px; Adds padding of 5px to top and bottom, 10px to left and right.
- Padding: 5px 6px 10px 15px;  
Top right Bottom left
  - Padding-top: 10px;
  - Padding-left: 35px;
  - Padding-right: 10px;
  - Padding-bottom: 10px;

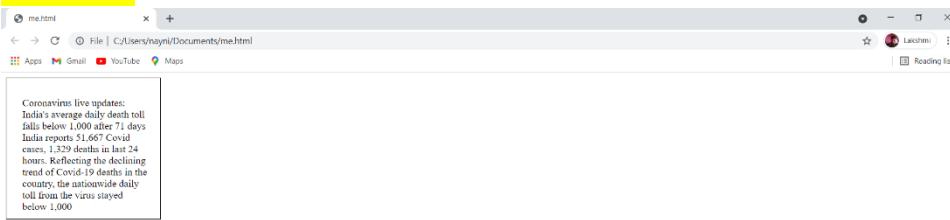
#### EXAMPLE: -

```

C:\Users\nayni\Documents\me.html - Sublime Text (UNREGISTERED)
File Edit Selection Find Goto Tools Project Preferences Help
<html>
<head>
<style type="text/css">
div.special2
{
    width:200px;
    border-style: solid;
    border-width: thin;
    border-color: #000000;
    padding: 10px 20px 10px 25px;
}
</style>
</head>
<body>
<div class="special2">
<p>Coronavirus live updates: India's average daily death toll falls below 1,000 after 71 days<br/>India reports 51,667 Covid cases, 1,329 deaths in last 24 hours. Reflecting the declining trend of Covid-19 deaths in the country, the nationwide daily toll from the virus stayed below 1,000</p>
</div>
</body>
</html>

```

## OUTPUT:



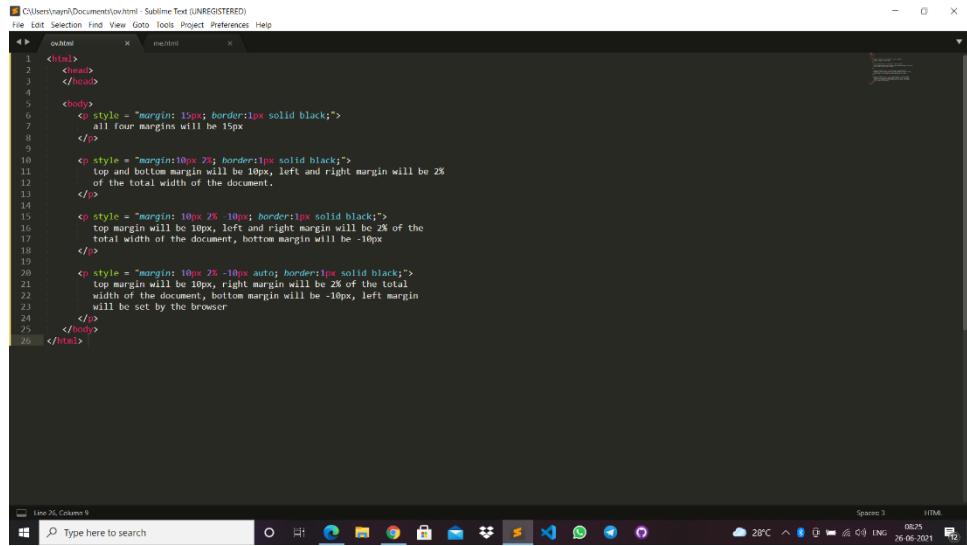
## CSS 'Margin' property:

Through which we could be able to add space or gap above the border of container.

- Syntax: margin: 10px; Adds padding of 10px to all directions (top, left, right and bottom)
- margin: 5px 10px; Adds padding of 5px to top and bottom, 10px to left and right.
- margin: 5px 6px 10px 15px;  
Top right Bottom left
  - margin-top: 10px;
  - margin-left: 35px;
  - margin-right: 10px;
  - margin-bottom: 10px;

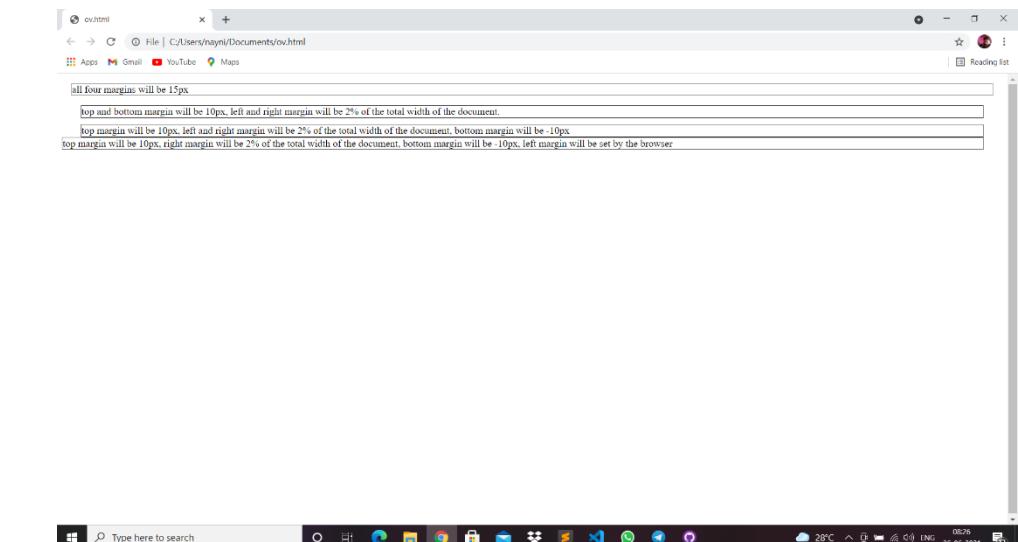
→ Margin:10px auto; Adds 10px of margin for top and bottom, moves the element to the center of container horizontal.

### EXAMPLE:

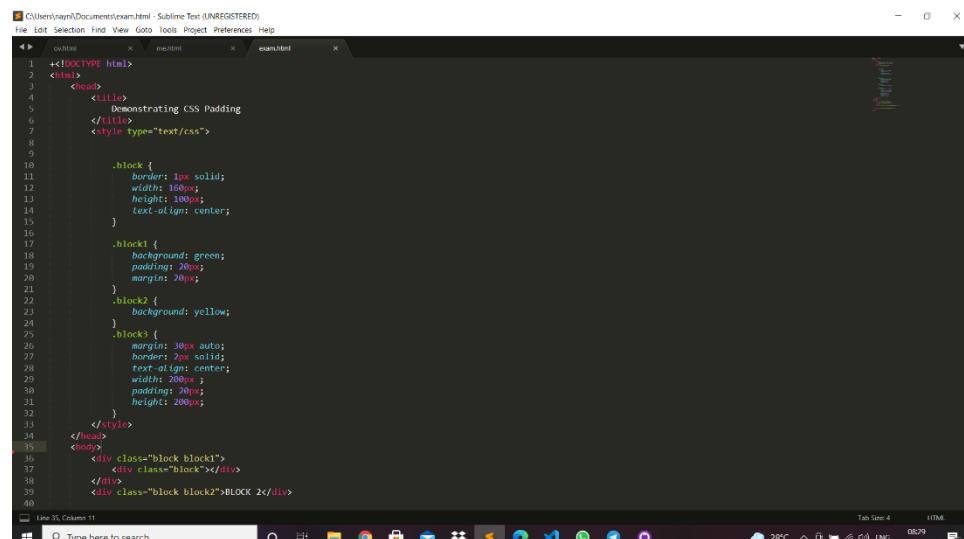


```
C:\Users\nayni\Documents\ov.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
ov.html X mentml X
1 <html>
2   <head>
3     </head>
4
5   <body>
6     <p style="margin: 15px; border:1px solid black;">
7       all four margins will be 15px
8     </p>
9
10    <p style="margin:10px 20px; border:1px solid black;">
11      top and bottom margin will be 10px, left and right margin will be 20px
12      of the total width of the document.
13    </p>
14
15    <p style="margin: 10px 25px; border:1px solid black;">
16      top margin will be 10px, left and right margin will be 25px of the
17      total width of the document, bottom margin will be -10px
18    </p>
19
20    <p style="margin: 10px 25px -10px auto; border:1px solid black;">
21      top margin will be 10px, right margin will be 25% of the total
22      width of the document, bottom margin will be -10px, left margin
23      will be set by the browser
24    </p>
25  </body>
26</html>
```

### OUTPUT:



### TODAY'S EXAMPLE:

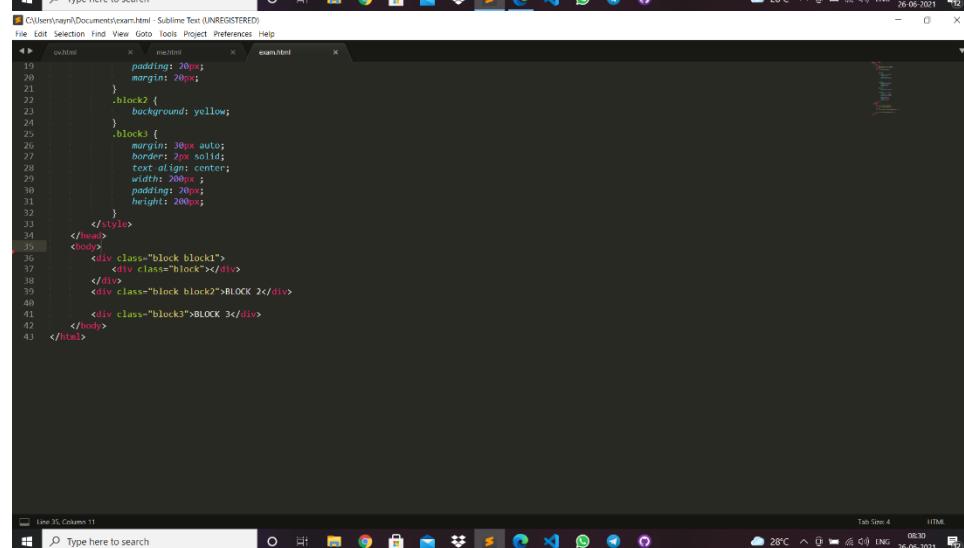


```

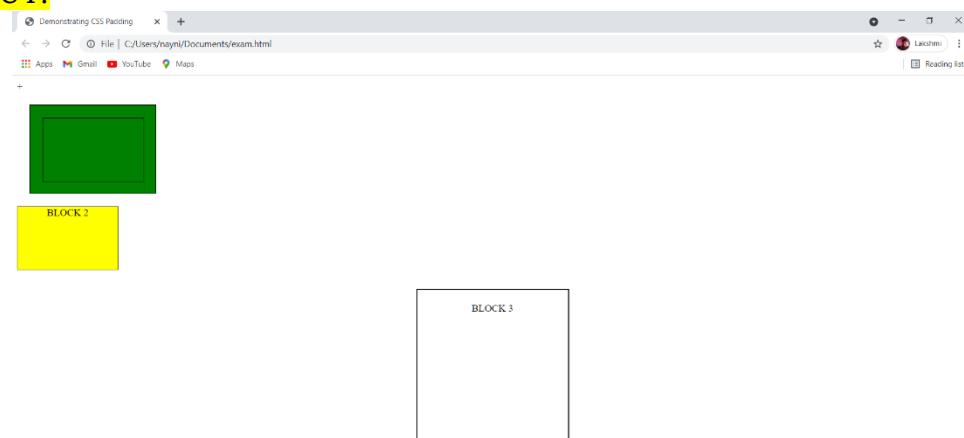
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Demonstrating CSS Padding </title>
5     <style type="text/css">
6
7       block {
8         border: 1px solid;
9         width: 100px;
10        height: 100px;
11        text-align: center;
12      }
13
14      block1 {
15        background: green;
16        padding: 20px;
17        margin: 20px;
18      }
19      block2 {
20        background: yellow;
21      }
22      block3 {
23        margin: 30px auto;
24        border: 2px solid;
25        text-align: center;
26        width: 200px;
27        padding: 20px;
28        height: 200px;
29      }
30
31    </style>
32  </head>
33  <body>
34    <div class="block block1">
35      <div class="block"></div>
36    </div>
37    <div class="block block2">BLOCK 2</div>
38
39    <div class="block block3">BLOCK 3</div>
40
41  </body>
42 </html>

```

Line 35, Column 11



## OUTPUT:



DATE:28/06/2021

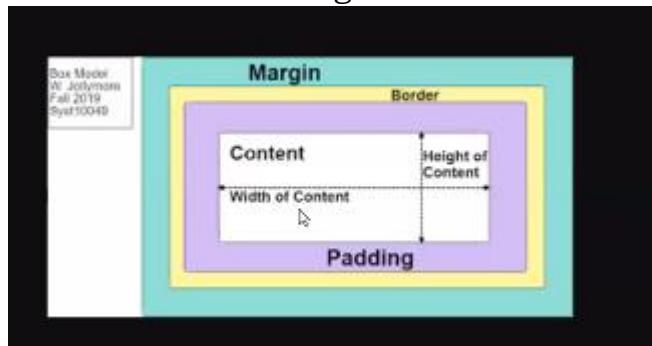
## CSS Box Model:

When an element gets rendered on a page, browser calculates the actual dimensions of the existing element on the page, accordingly it renders the new DOM element.

While calculating the dimensions of an element, it follows Box model feature.

In CSS box model property while calculating the actual dimensions of an element, it considers the following CSS properties:

- ◆ The total border space been occupied.
- ◆ The total margin space is occupied
- ◆ The total padding space is occupied.
- ◆ Actual width and height of Dom Element.



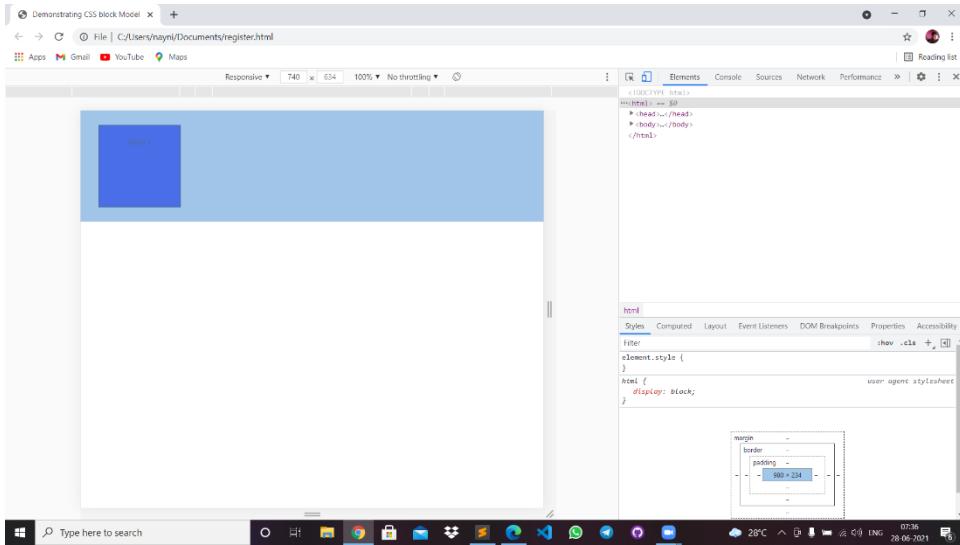
### EXAMPLE:

A screenshot of a code editor window titled 'codeModel.html'. The code is as follows:

```
<!DOCTYPE html>
<html>
    <head>
        <title>
            Demonstrating CSS Box Model
        </title>
        <style type="text/css">
            .block1 {
                border: 2px solid;
                margin: 30px;
                padding: 25px;
                height: 120px;
                width: 120px;
                text-align: center;
                background: blue;
            }
        </style>
    </head>
    <body>
        <div class="block1">
            Block 1
        </div>
        <div class="block2">TEST</div>
    </body>
</html>
```

The code defines two div elements. The first div, with class 'block1', has a blue background, a 2px solid border, 30px of margin, and 25px of padding. The second div, with class 'block2', contains the text 'TEST'.

OUTPUT: -



## Sample Example to find width: -

```

<div class="container"> </div>
File Edit Selection Find Goto Tools Project Preferences Help
content.html <div class="container"> </div>
1 <div class="container">
2   CONTENT
3 </div>
4
5
6 .container {
7   border: 1px solid;
8   margin: 5px 0px 11px 7px;
9   padding: 5px 8px;
10  height: 134px;
11  width: 198px;
12  text-align: center
13 }
14
15 Actual space/width of element with class 'container' ->

```

Activate Windows  
Go to Settings to activate Windows.

Width of element: -  $11+11+7+10+8+8+198=253$

## HTML UL tags and OL tags:

The two predefined HTML tags using which we could able to group relative items as an individual block.

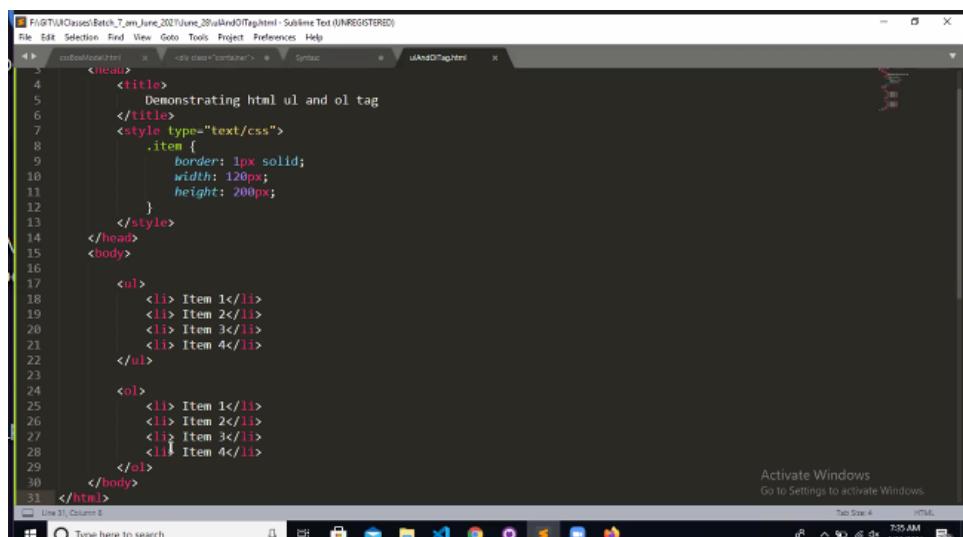
'ul' and 'ol' tag renders the element almost same where the only difference is ul tag (unordered list) renders the element with default bullet symbols, Ol tag (ordered list) render the element with default serial number.

'List-style' is the CSS property through which we can control the type of list style getting rendered for both ul and ol tag.

Syntax:

```
<ul>
<li>item1</li>
<li>item2</li>
<li>item3</li>
.....
</ul>
<ol>
<li>item1</li>
<li>item2</li>
<li>item3</li>
.....
</ol>
```

Example: -

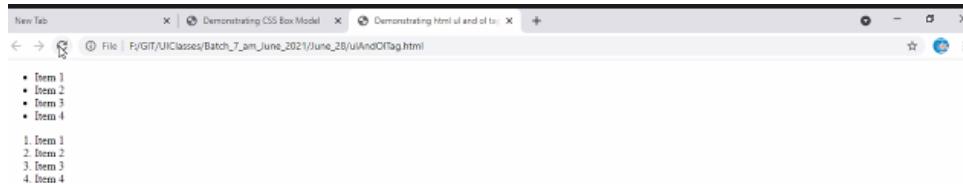


The screenshot shows a Sublime Text editor window with two tabs open. The left tab contains the following HTML code:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Demonstrating html ul and ol tag</title>
        <style type="text/css">
            .item {
                border: 1px solid;
                width: 120px;
                height: 200px;
            }
        </style>
    </head>
    <body>
        <ul>
            <li> Item 1</li>
            <li> Item 2</li>
            <li> Item 3</li>
            <li> Item 4</li>
        </ul>
        <ol>
            <li> Item 1</li>
            <li> Item 2</li>
            <li> Item 3</li>
            <li> Item 4</li>
        </ol>
    </body>
</html>
```

The right tab is titled "uAndOlTag.html". The status bar at the bottom indicates "Tab Size: 4" and "HTML".

OUTPUT: -



Sample Example: -

The image shows two side-by-side code editors. Both editors have the same title bar: "F:\GIT\UI\Classes\Batch\_7\_am\_June\_2021\June\_20\AndOrTag.html - Sublime Text (UNREGISTERED)". The top editor contains the following CSS:

```

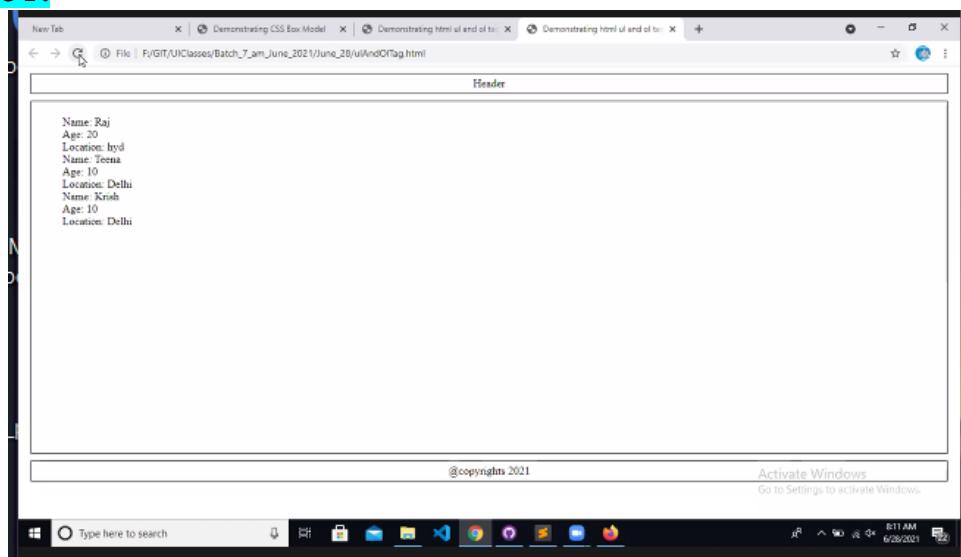
24     .container {
25         height: 500px;
26     }
27     .detailsContent {
28         text-align: left;
29     }
30 
```

The bottom editor contains the following HTML:

```

31 </head>
32 <body>
33     <div class="header">Header</div>
34     <div class="container">
35         <div class="detailsContent">
36             <ul>
37                 <li><div>Name: Raj</div><div>Age: 20</div><div>Location: hyd</div></li>
38                 <li><div>Name: Teena</div><div>Age: 10</div><div>Location: Delhi</div></li>
39                 <li><div>Name: Krish</div><div>Age: 10</div><div>Location: Delhi</div></li>
40             </ul>
41         </div>
42     </div>
43 
```

## OUTPUT:



DATE: 29/06/2021

### **Background-Property:**

- Using the background attribute in the HTML code of your pages, you can reach really attractive color effects for your web presence. You can use that attribute in two basic ways - through the Cascading Style Sheets (CSS).
- The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.
- You can set the following positions:
  - center
  - top
  - bottom
  - left
  - Right

Properties of Background image:

Property	Description
<a href="#"><u>background</u></a>	Sets all the background properties in one declaration
<a href="#"><u>background-attachment</u></a>	Sets whether a background image is fixed or scrolls with the rest of the page
<a href="#"><u>background-clip</u></a>	Specifies the painting area of the background
<a href="#"><u>background-color</u></a>	Sets the background color of an element
<a href="#"><u>background-image</u></a>	Sets the background image for an element
<a href="#"><u>background-origin</u></a>	Specifies where the background image(s) is/are positioned
<a href="#"><u>background-position</u></a>	Sets the starting position of a background image
<a href="#"><u>background-repeat</u></a>	Sets how a background image will be repeated
<a href="#"><u>background-size</u></a>	Specifies the size of the background image(s)

**Reference:-**[\*\*https://www.ntchosting.com/encyclopedia/scripting-and-programming/background/\*\*](https://www.ntchosting.com/encyclopedia/scripting-and-programming/background/)

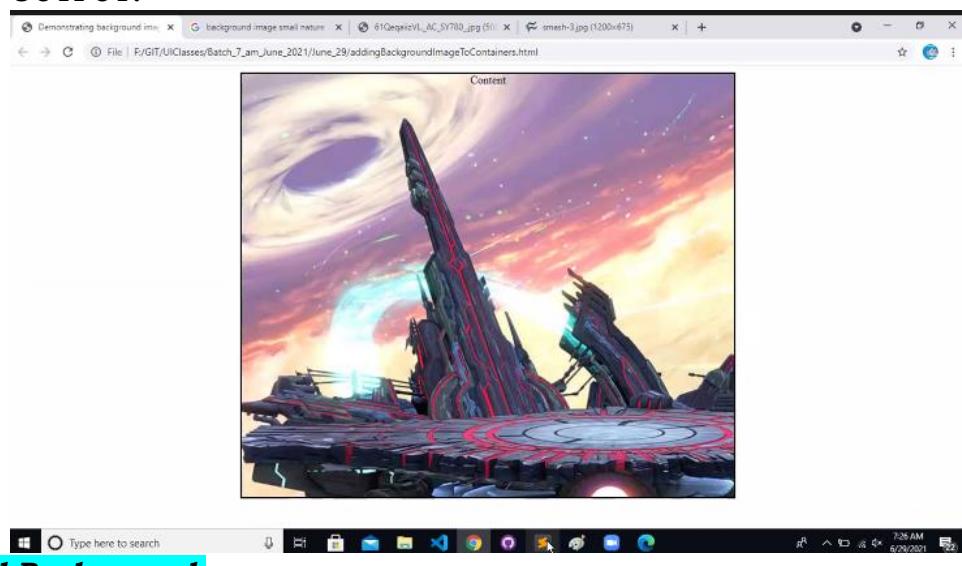
### **Background-image property:**

Using this property, you can set an image as a background for either a separate element on your site or for the whole site.

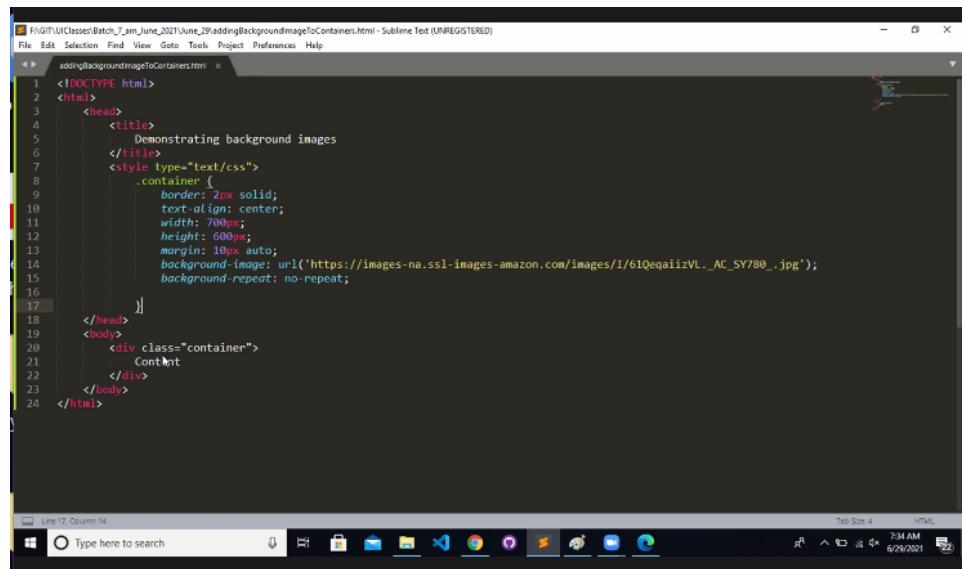
**Example:**

```
F:\GIT\UIClasses\Batch_7_am_June_2021\June_29\addingBackgroundImageToContainers.html - Sublime Text (UNREGISTERED)
addingBackgroundImageToContainers.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>
5       Demonstrating background images
6     </title>
7     <style type="text/css">
8       .container {
9         border: 2px solid;
10        text-align: center;
11        width: 700px;
12        height: 600px;
13        margin: 10px auto;
14        background-image: url('https://images-na.ssl-images-amazon.com/images/I/61QeqaiizVL._AC_SX700_.jpg');
15        background-repeat: no-repeat;
16      }
17    </style>
18  </head>
19  <body>
20    <div class="container">
21      Content
22    </div>
23  </body>
24</html>
```

## OUTPUT:

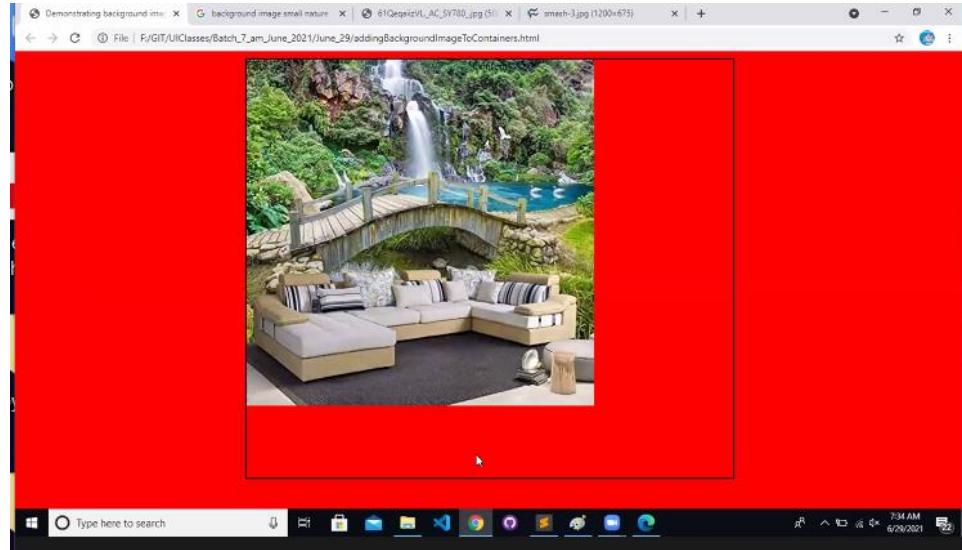


To Cover full Background:



```
File Edit Selection Find View Goto Tools Project Preferences Help
addingBackgroundImageToContainers.html - Sublime Text (UNREGISTERED)
<!DOCTYPE html>
<html>
    <head>
        <title>
            Demonstrating background images
        </title>
        <style type="text/css">
            .container {
                border: 2px solid;
                text-align: center;
                width: 700px;
                height: 600px;
                margin: 10px auto;
                background-image: url('https://images-na.ssl-images-amazon.com/images/I/61QeqaiizVL.AC_SX780_.jpg');
                background-repeat: no-repeat;
            }
        </style>
    </head>
    <body>
        <div class="container">
            Content
        </div>
    </body>
</html>
```

## OUTPUT:

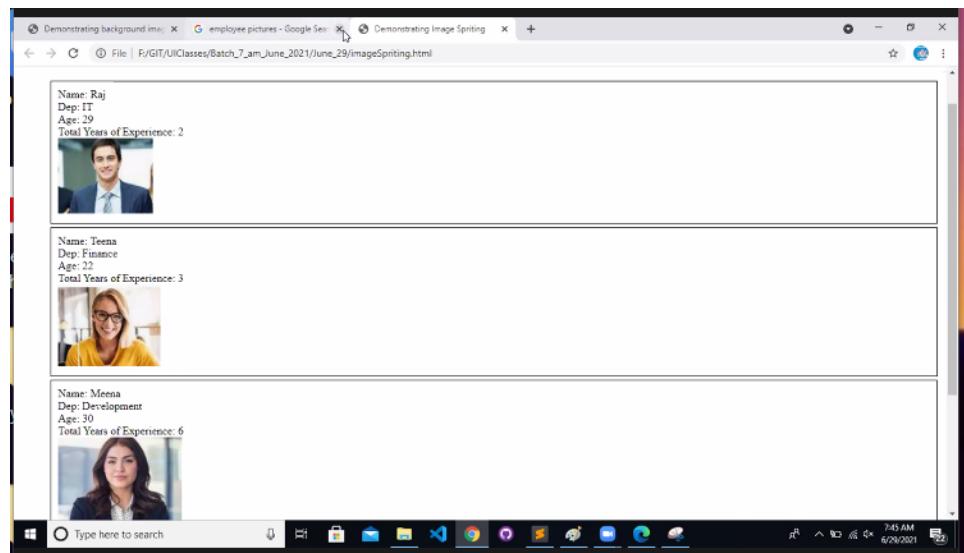


Employee details using image property:

```
File Edit Selection Find View Goto Tools Project Preferences Help
addingBackgroundImageForCorollaries.html imageSpring.html
<!DOCTYPE html>
<html>
    <head>
        <title>Demonstrating Image Spriting</title>
        <style type="text/css">
            h3 {
                text-align: center;
                padding: 10px;
            }
            li {
                list-style: none;
                border: 1px solid;
                margin: 5px;
                padding: 10px;
            }
        </style>
    </head>
    <body>
        <h3>Employee Details</h3>
        <ol>
            <li>
                <div>Name: Raj</div>
                <div>Dep: IT</div>
                <div>Age: 29</div>
                <div>Total Years of Experience: 2</div>
                
            </li>
            <li>
                <div>Name: Teena</div>
            </li>
        </ol>
    </body>
</html>
```

```
File Edit Selection Find View Goto Tools Project Preferences Help
addingBackgroundImageForCorollaries.html imageSpring.html
<!DOCTYPE html>
<html>
    <head>
        <title>Demonstrating Image Spriting</title>
        <style type="text/css">
            h3 {
                text-align: center;
                padding: 10px;
            }
            li {
                <div>Name: Raj</div>
                <div>Dep: IT</div>
                <div>Age: 29</div>
                <div>Total Years of Experience: 2</div>
                
            </li>
            <li>
                <div>Name: Teena</div>
                <div>Dep: Finance</div>
                <div>Age: 22</div>
                <div>Total Years of Experience: 3</div>
                
            </li>
            <li>
                <div>Name: Meena</div>
                <div>Dep: Development</div>
                <div>Age: 30</div>
                <div>Total Years of Experience: 6</div>
                
            </li>
            <li>
                <div>Name: Krish</div>
                <div>Dep: Testing/QA</div>
                <div>Age: 32</div>
                <div>Total Years of Experience: 7</div>
                
            </li>
        </ol>
    </body>
</html>
```

## OUTPUT:



To Upload the Image in Database it takes only three steps: -

- Make a HTML form to upload the image
- Connect to the database and store image
- Displaying the Image

<https://nodejs.org/en/>

Install Nodejs for windows (14.17.1 LTS Mostly recommended).

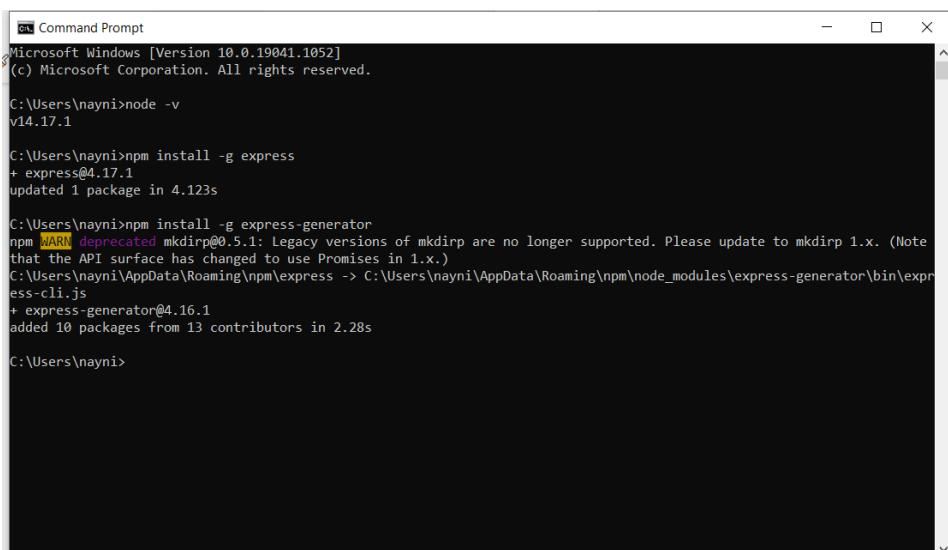
Open new command prompt and node -v

Create a server:

To do this we need to install a new server npm(node package manager)

cmd: npm install -g express

Then give command: npm install -g express-generator



```
Microsoft Windows [Version 10.0.19041.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nayni>node -v
v14.17.1

C:\Users\nayni>npm install -g express
+ express@4.17.1
updated 1 package in 4.123s

C:\Users\nayni>npm install -g express-generator
npm WARN deprecated mkdirp@0.5.1: Legacy versions of mkdirp are no longer supported. Please update to mkdirp 1.x. (Note
the API surface has changed to use Promises in 1.x.)
C:\Users\nayni\AppData\Roaming\npm\express -> C:\Users\nayni\AppData\Roaming\npm\node_modules\express-generator\bin\express-cli.js
+ express-generator@4.16.1
added 10 packages from 13 contributors in 2.28s

C:\Users\nayni>
```

DATE:30/06/2021

**create a server:**

Switch to drive where you want to create the server folder.

Step:1 Change folder into Batch code and do the following steps.

- The commands are:

npm install express

    npm install express-generator

    go the drive/folder where u want to create server - with in command  
    prompt

        express server name

e.g.:

        express batch\_7am....

        go inside folder

        cd batch\_7am....

        npm install

        open app.js -> add code for making server to listing at given port no.

        npm start -> start Ur server

        the npm install express

```
npm install express-generator
```

---  
go the drive/folder where u want to create server - with in commandprmppt

express server name

e.g.:

express batch\_7am....

go inside folder

cd batch\_7am....

npm install

open app.js -> add code for making server to listing at given port no.

npm start -> start Ur server

```
npm install express
npm install express-generator
-----
go the drive/folder where u want to create server - with in commandprmppt
eg:
express batch_7am....
go inside folder
cd batch_7am....
npm install
open app.js -> add code for making server to listing at given port no.
npm start -> start ur server
```

```
Select Command Prompt
Microsoft Windows [Version 10.0.17134.1384]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>f:
F:\>cd GIT
F:\GIT>cd UIClasses
F:\GIT\UIClasses>dir
Volume in drive F has no label.
Volume Serial Number is 8A59-7CE3

Directory of F:\GIT\UIClasses

06/26/2021  09:22 AM    <DIR>        .
06/26/2021  09:22 AM    <DIR>        ..
06/11/2021  07:06 AM           597 .gitignore
06/30/2021  06:52 AM    <DIR>        Batch_7_am_June_2021
06/11/2021  07:03 AM    <DIR>        Batch_8_B_Mar_2021
03/13/2021  03:05 PM    <DIR>        ProjectMockup
                           1 File(s)      597 bytes
                           5 Dir(s)  45,007,183,872 bytes free

F:\GIT\UIClasses>
```

```

generator
Command Prompt

create : Batch_7am_June_2021\
create : Batch_7am_June_2021\public\
create : Batch_7am_June_2021\public\javascripts\
create : Batch_7am_June_2021\public\images\
create : Batch_7am_June_2021\public\stylesheets\
create : Batch_7am_June_2021\public\stylesheets\style.css
create : Batch_7am_June_2021\routes\
create : Batch_7am_June_2021\routes\index.js
create : Batch_7am_June_2021\routes\users.js
create : Batch_7am_June_2021\views\
create : Batch_7am_June_2021\views\error.jade
create : Batch_7am_June_2021\views\index.jade
create : Batch_7am_June_2021\app.js
create : Batch_7am_June_2021\package.json
create : Batch_7am_June_2021\bin\
create : Batch_7am_June_2021\bin\www

change directory:
> cd Batch_7am_June_2021

install dependencies:
> npm install

run the app:
> SET DEBUG=batch-7am-june-2021:* & npm start

```

Step:6 Now open **app.js** file in Batch\_7am\_June\_2021 folder which was created by us and at line no 24 give the following command.

```

app.listen(8081, function () {
  console.log ("Server is listing at 8081");
});

```

```

F:\GIT\URClasses\Batch_7am_June_2021\app.js - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
express app.js
6
7 var indexRouter = require('./routes/index');
8 var usersRouter = require('./routes/users');
9
10 var app = express();
11
12 // view engine setup
13 app.set('views', path.join(__dirname, 'views'));
14 app.set('view engine', 'jade');
15
16 app.use(logger('dev'));
17 app.use(express.json());
18 app.use(express.urlencoded({ extended: false }));
19 app.use(cookieParser());
20 app.use(express.static(path.join(__dirname, 'public')));
21
22 app.use('/', indexRouter);
23 app.use('/users', usersRouter);
24
25
26 app.listen(8081, function() {
27   console.log("Server is listing at 8081")
28 }
29
30 // catch 404 and forward to error handler
31 app.use(function(req, res, next) {
32   next(createError(404));
33 });
34
35 // error handler
36 app.use(function(err, req, res, next) {

```

Step:7 So now we need to check the server by giving following command in command prompt or simply open in browser and check using this link <http://localhost:8081/>

```

[npm] Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>f:
F:\>cd GIT
F:\GIT>cd UIClasses
F:\GIT\UIClasses>cd Batch_7am_June_2021
F:\GIT\UIClasses\Batch_7am_June_2021>npm start
> batch-7am-june-2021@0.0.0 start
> node ./bin/www

Server is listing at 8081

```

Step:8 Now open public folder and clear the folders available there and simply copy them under public folder.

---



---

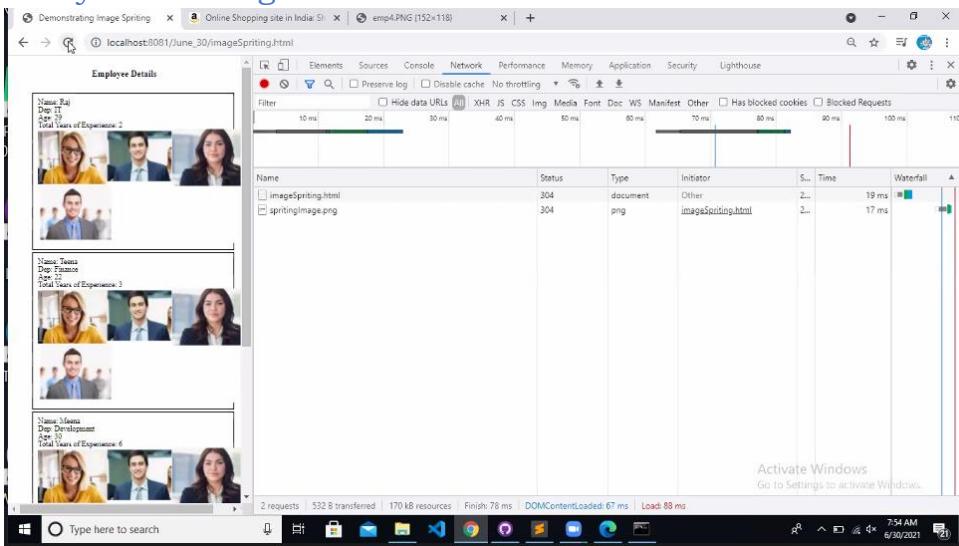


---

## Image Spriting

- The concept of merging all the static images to form a single image, through using background position property we show only the required image is called image spiriting.
- Through image spiriting we increase the performance of the page by loading all the static images in a single call

**Note:** Image spriting is only recommended for static images but not for dynamic images

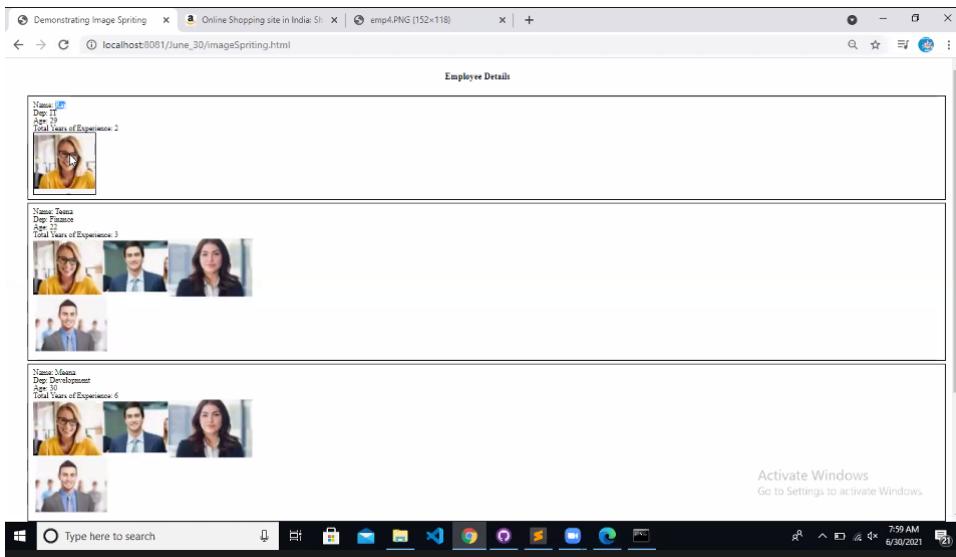


**TO LOCK AND MOVE THE IMAGES IN BACKGROUND:**

```
F:\GIT\UI\Classes\Batch_7am_June_2021\public\June_30\imageSpriting.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
imgSpriting.html
12
13     list-style: none;
14     border: 1px solid;
15     margin: 5px;
16     padding: 10px;
17   }
18   .imgBlock {
19     width: 130px;
20     height: 130px;
21     border: 1px solid;
22     background-image: url('images/spritingImage.png');
23   }
24   </style>
25 </head>
26 <body>
27   <h1>Employee Details</h1>
28   <ul>
29     <li>
30       <div>Name: Raj</div>
31       <div>Dep: IT</div>
32       <div>Age: 29</div>
33       <div>Total Years of Experience: 2</div>
34       <div class="imgBlock"></div>
35     </li>
36     <li>
37       <div>Name: Teena</div>
38       <div>Dep: Finance</div>
39       <div>Age: 22</div>
40       <div>Total Years of Experience: 3</div>
41       
42     </li>
43   </ul>
```

---

**OUTPUT:**

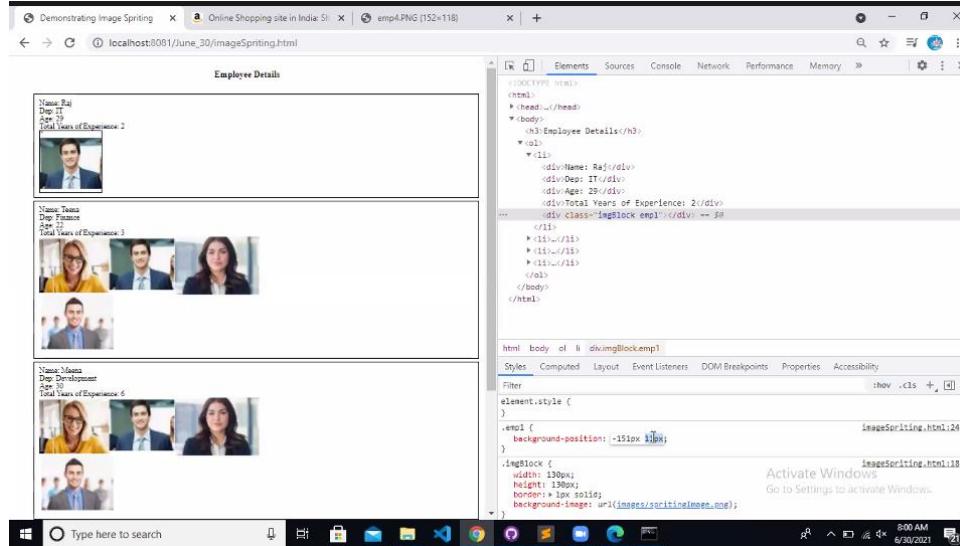


## MOVING THE IMAGE:

File Edit Selection Find View Goto Tools Project Preferences Help

```
11 {
12     list-style: none;
13     border: 1px solid;
14     margin: 5px;
15     padding: 10px;
16 }
17 .imgBlock {
18     width: 130px;
19     height: 130px;
20     border: 1px solid;
21     background-image: url('images/spritingImage.png');
22 }
23 .empl {
24     background-position: 10px 10px;
25 }
26 }
27 </style>
28 </head>
29 <body>
30 <h2>Employee Details</h2>
31 <ol>
32     <li>
33         <div>Name: Raj</div>
34         <div>Dep: IT</div>
35         <div>Age: 29</div>
36         <div>Total Years of Experience: 2</div>
37         <div class="imgBlock empl1"></div>
38     </li>
39     <li>
40         <div>Name: Teena</div>
41         <div>Dep: Finance</div>
42         <div>Age: 28</div>
43         <div>Total Years of Experience: 2.2</div>
44     </li>
45 </ol>
```

## OUTPUT:



## FINAL CODE:

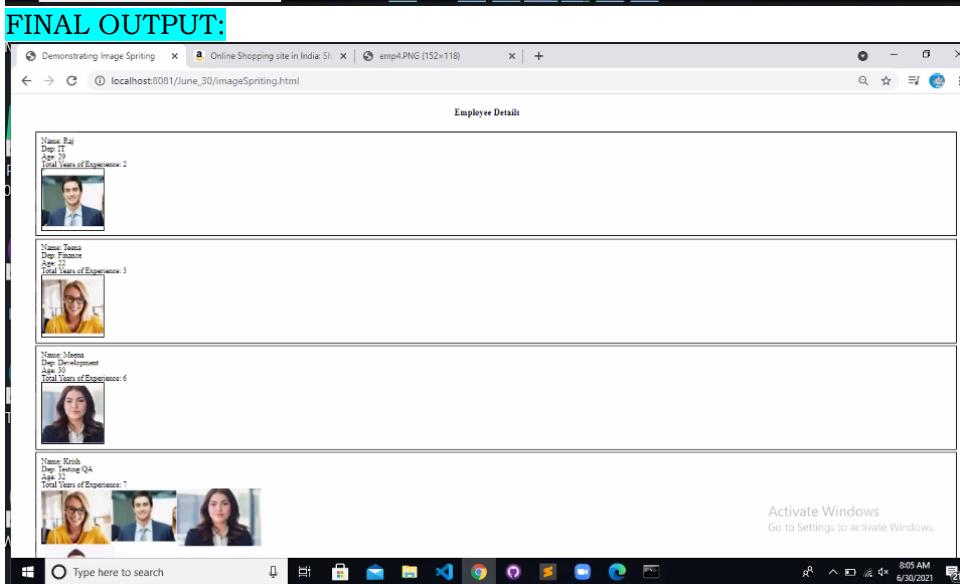
```
F:\GIT\UI\Classes\Batch_7am_June_2021\public\June_30\imageSpriting.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
imgSpriting.html x
1 <head>
2   <style>
3     .imgBlock {
4       width: 150px;
5       height: 130px;
6       border: 1px solid;
7       background-image: url('images/spritingImage.png');
8     }
9     .emp1 {
10       background-position: -151px 10px;
11     }
12     .emp2 {
13       background-position: -4px 5px;
14     }
15     .emp3 {
16       background-position: 10px 10px;
17     }
18   </style>
19 </head>
20 <body>
21   <h3>Employee Details</h3>
22   <ol>
23     <li>
24       <div>Name: Raj</div>
25       <div>Dep: IT</div>
26       <div>Age: 29</div>
27       <div>Total Years of Experience: 2</div>
28       <div class="imgBlock emp1"></div>
29     </li>
30     <li>
31       <div>Name: Teena</div>
32       <div>Dep: Finance</div>
33       <div>Age: 22</div>
34       <div>Total Years of Experience: 3</div>
35       <div class="imgBlock emp2"></div>
36     </li>
37     <li>
38       <div>Name: Meena</div>
39       <div>Dep: Development</div>
40       <div>Age: 30</div>
41       <div>Total Years of Experience: 6</div>
42       <div class="imgBlock emp3"></div>
43     </li>
44   </ol>
45   <div style="text-align: right; margin-top: 10px;">
46     Activate Windows
47     Go to Settings to activate Windows.
48   </div>
49
50
51 Line 31, Column 40, Saved F:\GIT\UI\Classes\Batch_7am_June_2021\public\June_30\imageSpriting.html (UTF-8)
```

F:\GIT\UIClasses\Batch\_7em\_June\_2021\public\June\_30\imageSpriting.html - Sublime Text (UNREGISTERED)

```

File Edit Selection Find View Goto Tools Project Preferences Help
imageSpriting.html x
46   <div>Name: Teena</div>
47   <div>Dep: Finance</div>
48   <div>Age: 22</div>
49   <div>Total Years of Experience: 3</div>
50   <div class="imgblock emp2"></div>
51   </li>
52   <li>
53     <div>Name: Meena</div>
54     <div>Dep: Development</div>
55     <div>Age: 30</div>
56     <div>Total Years of Experience: 6</div>
57     <div class="imgblock emp3"></div>
58   </li>
59   <li>
60     <div>Name: Krish</div>
61     <div>Dep: Testing/QA</div>
62     <div>Age: 32</div>
63     <div>Total Years of Experience: 7</div>
64     
65   </li>
66 
67 
68 </ol>
69 </body>
69 </html>

```



DATE: 1/07/2021

## CSS Positions:

Till now in order to move the DOM elements to a specified position we were using padding and margin properties, while using padding or margin properties to move the DOM elements, it is not actually moving the DOM element but it is increasing its dimensions.

In order to actually move the DOM element to a required position without increasing its dimensions we use the following CSS properties:

- Top
- Left
- Right

→ Bottom

- Not every DOM element is capable of considering the above 4 CSS properties but the DOM elements which are positioned can only consider the above properties.
- CSS ‘position’ is the property through which we could be able to control the position of any DOM element. Following are possible values the position attribute takes:
  - Static
  - Relative
  - Absolute
  - Fixed
  - Sticky

❖ Element with position static:

- Every DOM element by default holds the static position which indicates the DOM element cannot be moved to any position from its default (it will not consider the properties top left right and bottom).

★ Syntax:

Position: static;

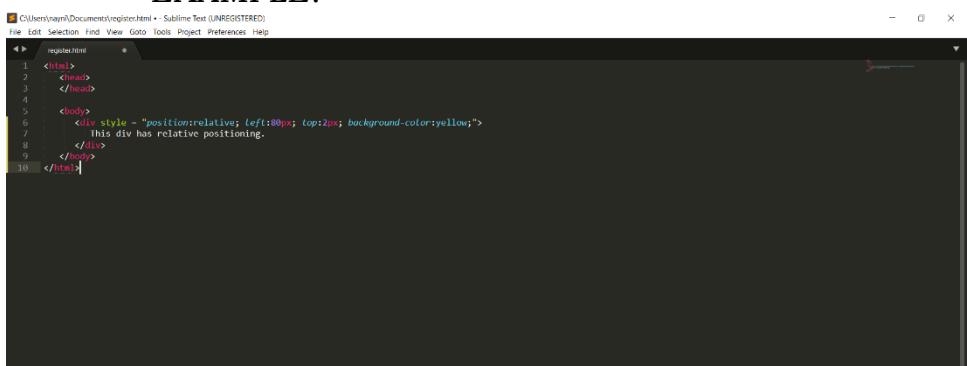
❖ Element with position Relative:

- Any DOM element with position relative holds the following properties:
  - It is capable of moving to any required position within the page (it considers top left right and bottom properties).
  - While moving to a new position it never loses space been occupied on load of the page.
  - While moving to a new position it always moves relevant to its default position.

★ Syntax:

position: relative;

★ EXAMPLE:



```
<html>
<head>
</head>
<body>
<div style = "position:relative; left:80px; top:2px; background-color:yellow;">
    This div has relative positioning.
</div>
</body>
</html>
```

❖ Element with position Absolute:

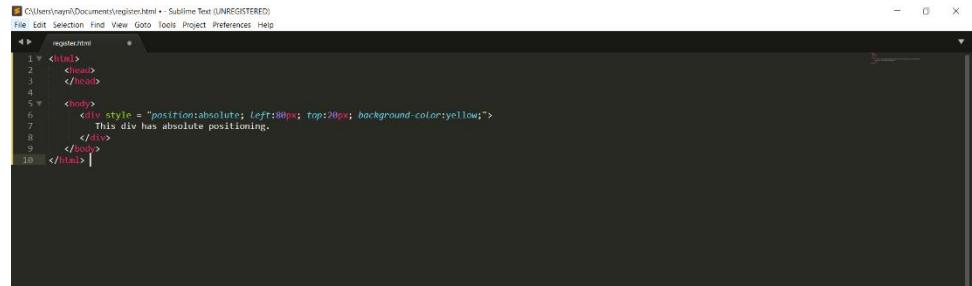
- Any DOM element with position absolute holds the following properties:
  - It is capable of moving to any required position (considers top left right and bottom properties)
  - Element with position absolute will automatically lose the default space it occupies on the page (loses space within x-y axis).
  - While moving to a new position it always moves relevant to its parent's position.
  - While depending on the parent position it only depends on parent whose position value is non static.
  - If its immediate parent doesn't hold the position non-static, it traverses to its grandparents until it finds a parent with position on static.

**NOTE: ELEMENT WITH POSITION ABSOLUTE AUTOMATICALLY JUMPS FROM DEFAULT X-Y AXIS TO Z-AXIS.**

★ Syntax:

`position: absolute;`

★ EXAMPLE:



```

1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     <div style="position: absolute; left: 80px; top: 20px; background-color: yellow;">
7       This div has absolute positioning.
8     </div>
9   </body>
10 </html>

```

TODAYS'EXAMPLE FOR RELATIVE:

```

F:\GIT\UI\Classes\Batch_7am_June_2021\publicJuly_1\cssPositions.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
cssPositions.html • top top exportAndImportDOM X app.js X mathUtilities X Session X
1.1      .block1 {
1.2        background: green;
1.3      }
1.4      .block2 {
1.5        background: blue;
1.6        left: 500px;
1.7        top: 300px;
1.8        position: relative;
1.9      }
1.10     .block3 {
1.11       background: yellow;
1.12     }
1.13     .block4 {
1.14       background: pink;
1.15     }
1.16   
```

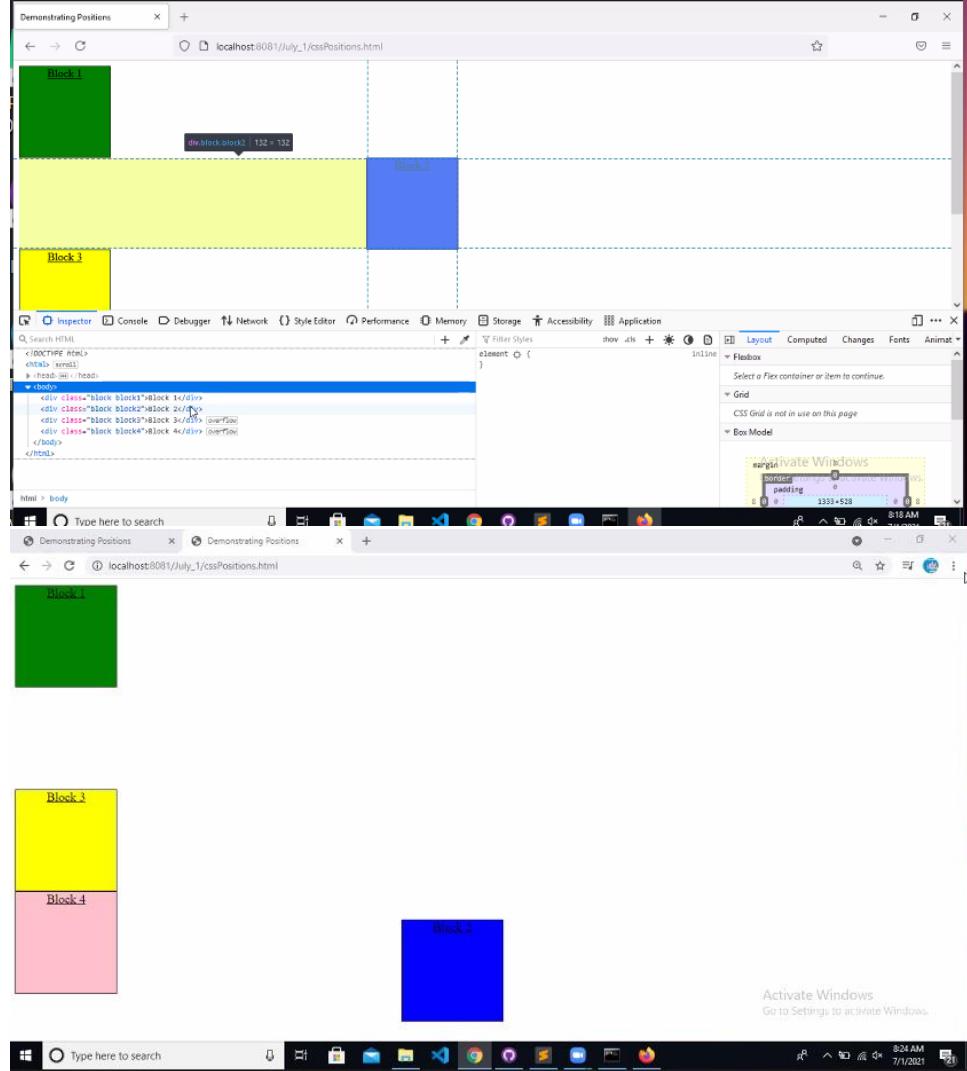
Activate Windows  
Go to Settings to activate Windows.

Tab Size 4 HTML

8:23 AM

7/1/2021

## OUTPUT:



**DATE: 2/07/2021**

EXAMPLE FOR ABSOLUTE:

cssPosition.html

```

13     height: 130px;
14   }
15   .block1 {
16     background: green;
17   }
18   .block2 {
19     background: blue;
20     top: 100px;
21     left: 200px;
22     position: absolute;
23   }
24   .block3 {
25     background: yellow;
26   }
27   .block4 {
28     background: pink;
29   }
30   
```

</style>

</head>

<body>

<div class="block block1">  
 Block 1  
</div>

<div class="block block2">  
 Block 2  
</div>

<div class="block block3">  
 Block 3  
</div>

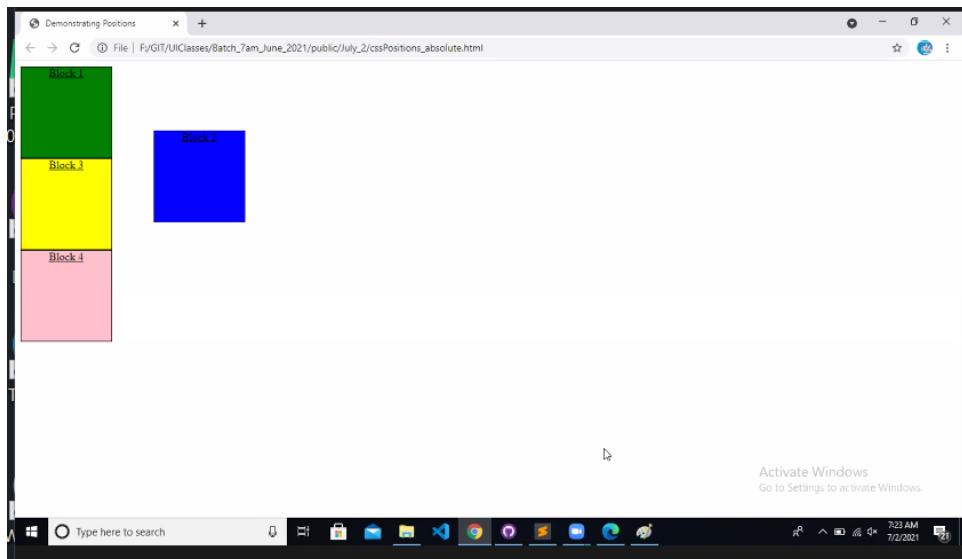
<div class="block block4">  
 Block 4  
</div>

</body>

</html>

Activate Windows  
Go to Settings to activate Windows.

## OUTPUT:



## EXAMPLE-2:

cssPosition.html

```

23   }
24   .block3 {
25     background: yellow;
26   }
27   
```

.iparent {  
border: 1px solid;  
padding: 10px;  
height: 150px;  
}  
.gparent {  
border: 2px solid;  
padding: 30px;  
height: 250px;  
}

</style>

</head>

<body>

<div class="block block1">  
 Block 1  
</div>

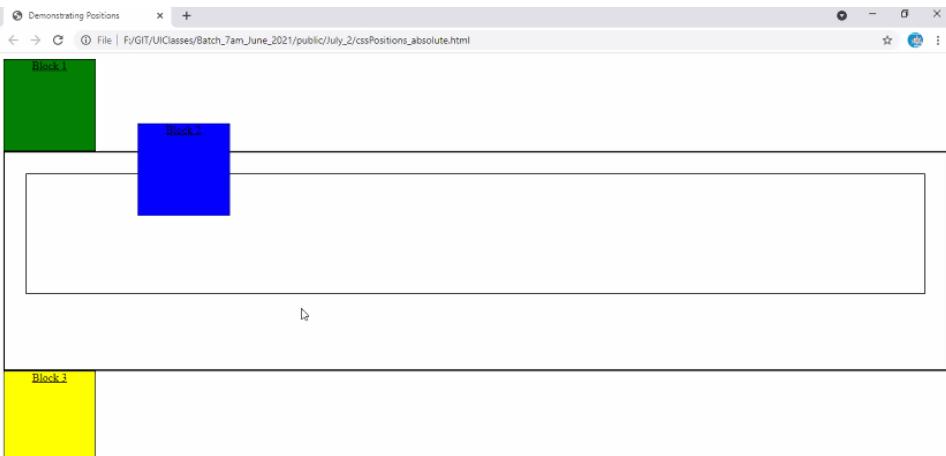
<div class="gparent">  
 <div class="iparent">  
 <div class="block block2">  
 Block 2  
</div>  
 </div>  
<div class="block block3">  
 Block 3  
</div>

</body>

</html>

Activate Windows  
Go to Settings to activate Windows.

## OUTPUT:

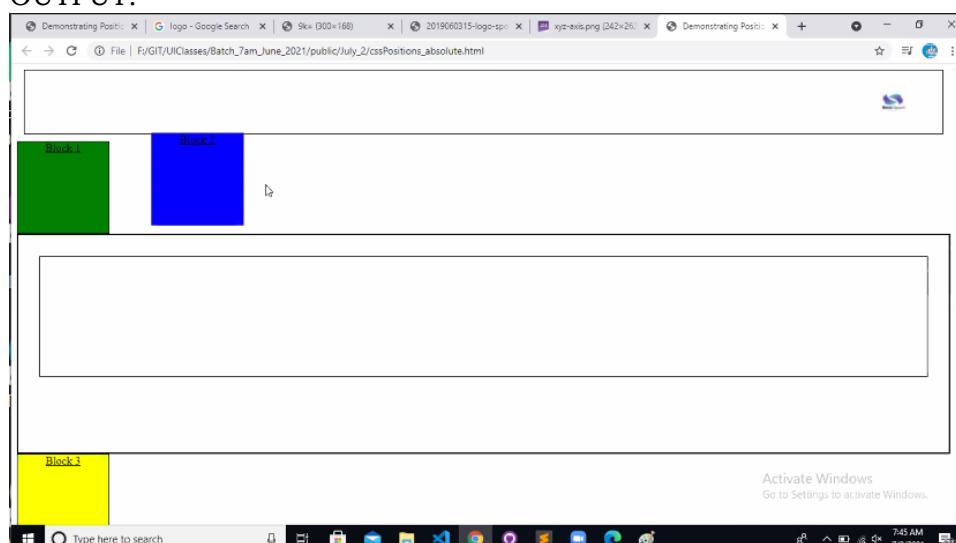


## PLACING LOGO IN OUR WEB PAGE:

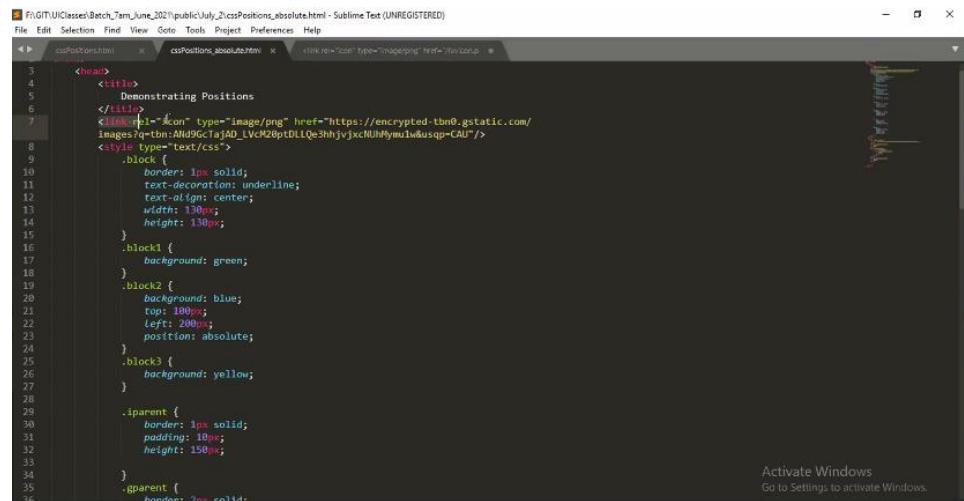
```
F:\GIT\UIClasses\Batch_7am_June_2021\public\July_2\cssPositions_absolute.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
cssPositionAbsolute.css
Activate Windows
Go to Settings to activate Windows.

25     background: yellow;
26 }
27 .iparent {
28     border: 1px solid;
29     padding: 10px;
30     height: 150px;
31 }
32 .gparent {
33     border: 2px solid;
34     padding: 30px;
35     height: 250px;
36 }
37 img {
38     width: 60px;
39     height: 50px;
40     position: absolute;
41     right: 0;
42 }
43 .header {
44     border: 1px solid;
45     padding: 20px;
46     height: 50px;
47     margin: 10px;
48 }
49 
```

## OUTPUT:



## PLACING AN ICON IN THE BROWSER:

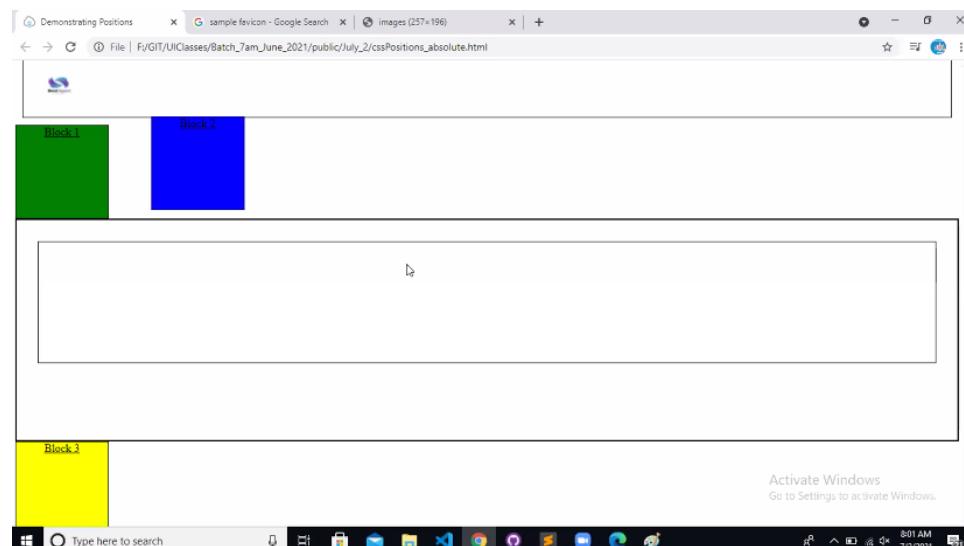


```

3 <head>
4   <title>
5     Demonstrating Positions
6   </title>
7   <link rel="icon" type="image/png" href="https://encrypted-tbn0.gstatic.com/
8     images?q=tbn:ANd9GcJaAD_LVcM28ptDLLe3hhjvjxcNUHMymuIw&usqp=CAU"/>
9   <style type="text/css">
10    .block {
11      border: 1px solid;
12      text-decoration: underline;
13      text-align: center;
14      width: 170px;
15      height: 130px;
16    }
17    .block1 {
18      background: green;
19    }
20    .block2 {
21      background: blue;
22      top: 100px;
23      left: 200px;
24      position: absolute;
25    }
26    .block3 {
27      background: yellow;
28    }
29    .parent {
30      border: 1px solid;
31      padding: 10px;
32      height: 150px;
33    }
34    .parent {
35      border: 2px solid;

```

OUTPUT:



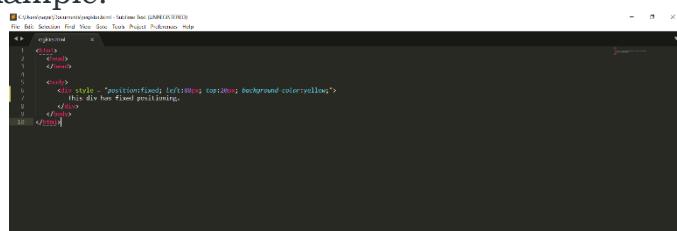
#### ❖ Element with position Fixed:

- Any DOM element with position fixed is almost like an element with position absolute where the only difference is once the element gets its fixed position it doesn't even move from its original position even while scroll.

#### ★ Syntax:

Position: fixed;

#### ★ Example:



```

1 <html>
2   <head>
3     <title>
4       Position Fixed Example
5     </title>
6   </head>
7   <body>
8     <div style="position:fixed; top:0px; left:0px; background-color:yellow">
9       This div has fixed positioning.
10    </div>
11  </body>
12</html>

```

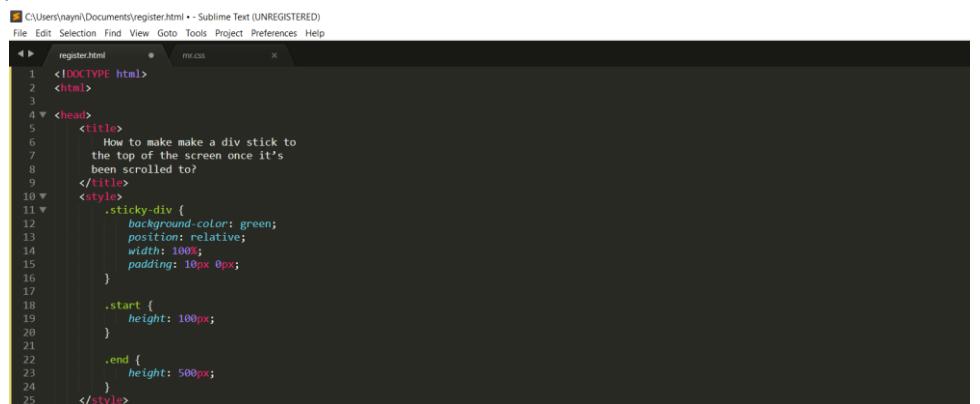
❖ Element with position sticky:

- Any DOM element with position Sticky is almost like element with position relative where the only difference is once positions values been given (top, left, right, bottom) if we tried to scroll the elements out of its view port, it automatically turns to fixed position and doesn't get scrolled.

★ Syntax:

Position: sticky;

★ Example:

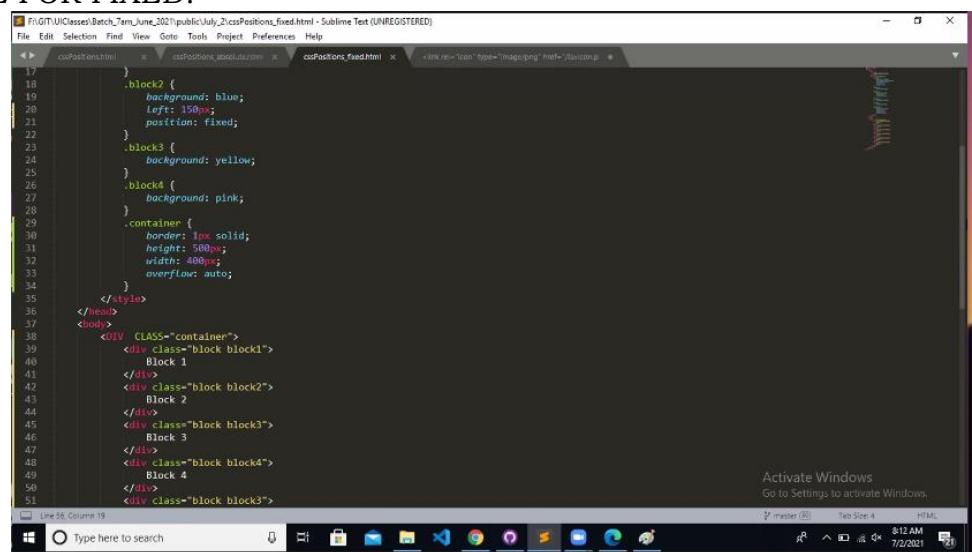


A screenshot of Sublime Text showing a file named register.html. The code defines a CSS class .sticky-div with properties: background-color: green; position: relative; width: 100%; padding: 10px;. It also defines two inner classes: .start { height: 100px;} and .end { height: 500px;}. The file is located at C:\Users\nayn\Documents\register.html.

```
<!DOCTYPE html>
<html>
<head>
    <title>
        How to make a div stick to
        the top of the screen once it's
        been scrolled to?
    </title>
    <style>
        .sticky-div {
            background-color: green;
            position: relative;
            width: 100%;
            padding: 10px;
        }
        .start {
            height: 100px;
        }
        .end {
            height: 500px;
        }
    </style>

```

TODAY'S EXAMPLE FOR FIXED:



A screenshot of Sublime Text showing a file named cssPositions.html. The code defines a CSS class .container with properties: border: 1px solid black; height: 500px; width: 400px; overflow: auto;. It also defines four block classes: .block1, .block2, .block3, and .block4 with various background colors. The file is located at F:\GIT\UI\Classes\Batch\_7\Jun,2021\public\July\_2\cssPositions\_fixed.html.

```

    }
    .block2 {
        background: blue;
        left: 150px;
        position: fixed;
    }
    .block3 {
        background: yellow;
    }
    .block4 {
        background: pink;
    }
    .container {
        border: 1px solid black;
        height: 500px;
        width: 400px;
        overflow: auto;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="block block1">
            Block 1
        </div>
        <div class="block block2">
            Block 2
        </div>
        <div class="block block3">
            Block 3
        </div>
        <div class="block block4">
            Block 4
        </div>
    </div>
</body>

```

The screenshot shows a Sublime Text window with three tabs: 'cssPositions.html', 'cssPosition\_absolute.css', and 'cssPositions\_fixed.html'. The 'cssPositions.html' tab contains the following HTML code:

```
<html>
  <head>
    <style>
      .block {
        width: 100px;
        height: 100px;
        background-color: black;
        color: white;
        text-align: center;
        margin-bottom: 10px;
      }
      .block1 {background-color: red;}
      .block2 {background-color: green;}
      .block3 {background-color: yellow;}
      .block4 {background-color: blue;}
    </style>
  </head>
  <body>
    <div CLASS="container">
      <div class="block block1">
        Block 1
      </div>
      <div class="block block2">
        Block 2
      </div>
      <div class="block block3">
        Block 3
      </div>
      <div class="block block4">
        Block 4
      </div>
      <div class="block block3">
        Block 3
      </div>
      <div class="block block1">
        Block 1
      </div>
    </div>
  </body>
</html>
```

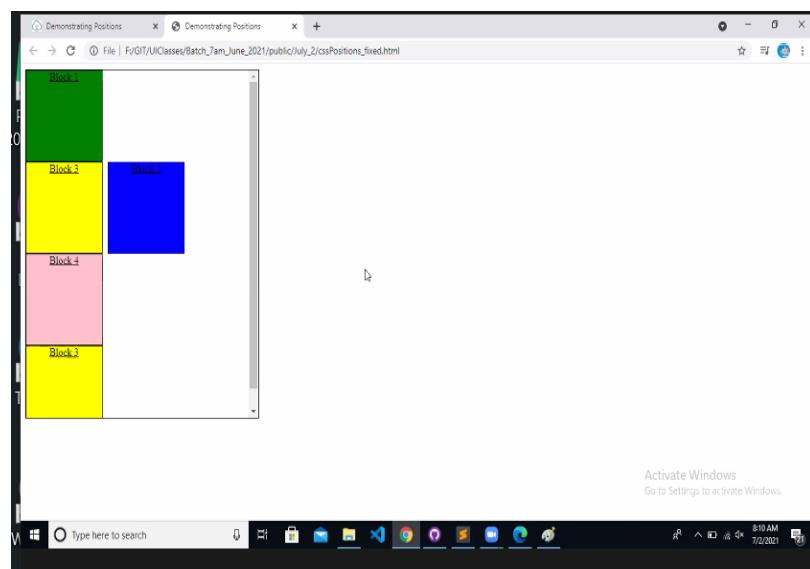
The 'cssPosition\_absolute.css' tab contains the following CSS code:

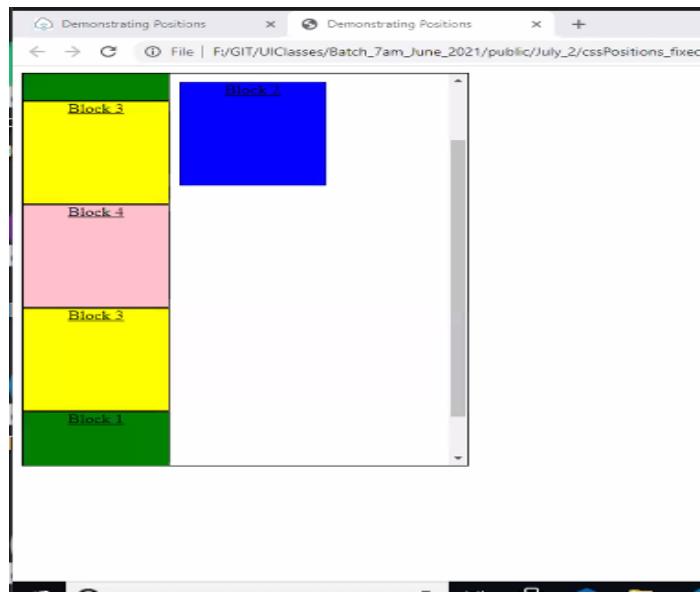
```
.block {
  position: absolute;
  left: 0;
  top: 0;
}
```

The 'cssPositions\_fixed.html' tab contains the following HTML code:

```
<link rel="icon" type="image/png" href="favicon.png" />
```

OUTPUT:





### TODAY'S EXAMPLE FOR STICKY:

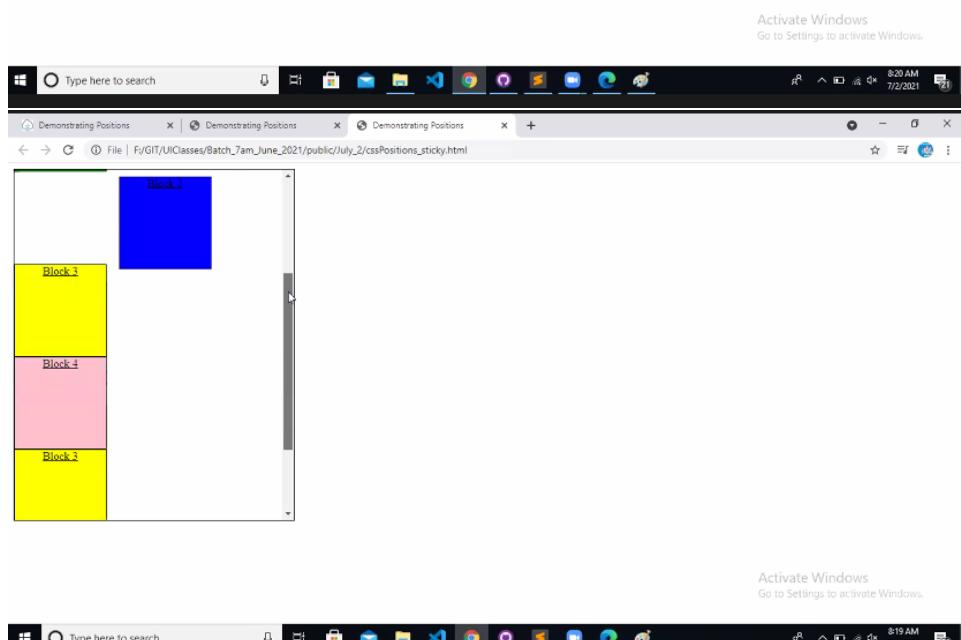
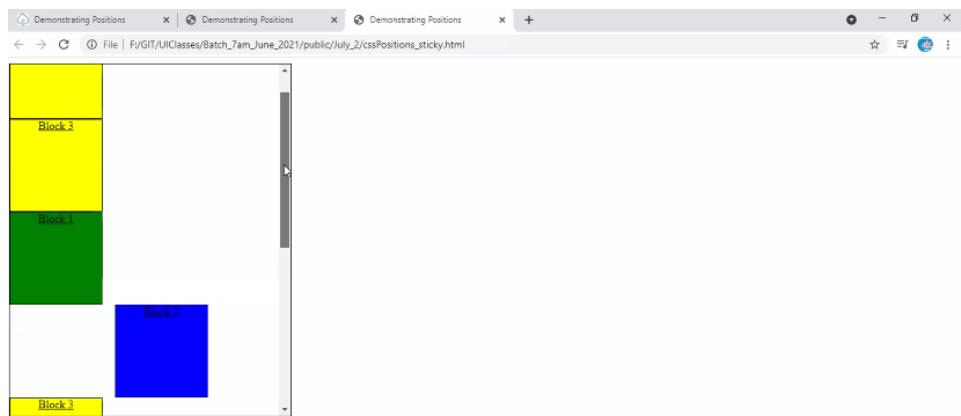
A screenshot of the Sublime Text editor showing the code for a sticky position example. The code consists of a CSS file (csspositions.css) and an HTML file (csspositions\_sticky.html).

```
csspositions.css
11     text-align: center;
12     width: 130px;
13     height: 130px;
14 }
15 .block1 {
16     background: green;
17 }
18 .block2 {
19     background: blue;
20     position: sticky;
21     left: 150px;
22 }
23 .block3 {
24     background: yellow;
25 }
26 .block4 {
27     background: pink;
28 }
29 .container {
30     border: 1px solid;
31     height: 500px;
32     width: 400px;
33     overflow: auto;
34 }
35 
```

```
</style>
</head>
<body>
<div CLASS="container">
    <div class="block block3">
        Block 3
    </div>
    <div class="block block1">
        Block 1
    </div>

```

OUTPUT:



## Placing elements to center:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Demonstrating Positions</title>
    <style type="text/css">
      .block {
        border: 1px solid;
        text-decoration: underline;
        text-align: center;
        width: 130px;
        height: 130px;
      }
      .block2 {
        background: blue;
        position: absolute;
        left: 45%;
        top: 45%;
        bottom: 20px;
      }
    </style>
  </head>
  <body>
    <DIV CLASS="container">
      <div class="block block2">Block 2</div>
    </DIV>
  </body>
</html>

```

A screenshot of Sublime Text showing the source code for a CSS demonstration. The code defines a container DIV containing a blue block element. The blue block is positioned absolutely at 45% from the top and left, and has a height of 20px. The browser interface at the bottom shows tabs for other files and a status bar indicating the file is 8:19 AM on 7/2/2021.

DATE: 3/7/2021

### CSS Z-Index Property:

The elements which are falling under z-axis, there is a chance for multiple elements override each other while rendering on the page.

While the elements are overriding, we can control the rendering order through CSS Z-Index property.

Z-Index a CSS property takes a positive number as a value, through which we can specify the priority order, the element with higher priority order always renders on the top.

NOTE: Z-Index property can only be applied to elements which fall under Z-axis (Elements with position non-static).

EXAMPLE:

```
<head>
<style>
img {
    position: absolute;
    left: 0px;
    top: 0px;
    z-index: -1;
}
</style>
</head>
<body>
<h1>The z-index Property</h1>
<imgsrc="http://www.vanseodesign.com/blog/wp-
content/uploads/2009/11/z-index.png"
width="100" height="140">
<p>Because the image has a z-index of -1, it will be
placed behind the heading. </p>
</body>
</html>
```

OUTPUT:



## CSS ‘FLOAT’ PROPERTY:

By default, while the elements getting rendered on the page, they try to render from the left top corner/position of the container, the block level elements fall under the new line, inline elements get rendered within the same line. Using CSS ‘float’ property we can make the elements to get rendered either to the right side or left side of the container.

Following are the possible values a float property takes:

float: left/right;

1. Element with float left property will render to complete left side of the container, element with float right property will render to complete right side of the container.
  2. Multiple continuous DOM elements applied with CSS float property will always tries to render within the same line irrelevant of block or inline type of elements.
  3. If an element changes its default direction through float property, the element which is following it will also try to render in the same line.
  4. To make the element not to follow previous element direction we make use of CSS property: “clear”.

### i. Syntax:

Clear: left/right/both;

F:\GIT\UI\Classes\Batch\_7\mune\_2021\public\July\_3\cssFloatProperty.html - Sublime Text (UNREGISTERED)

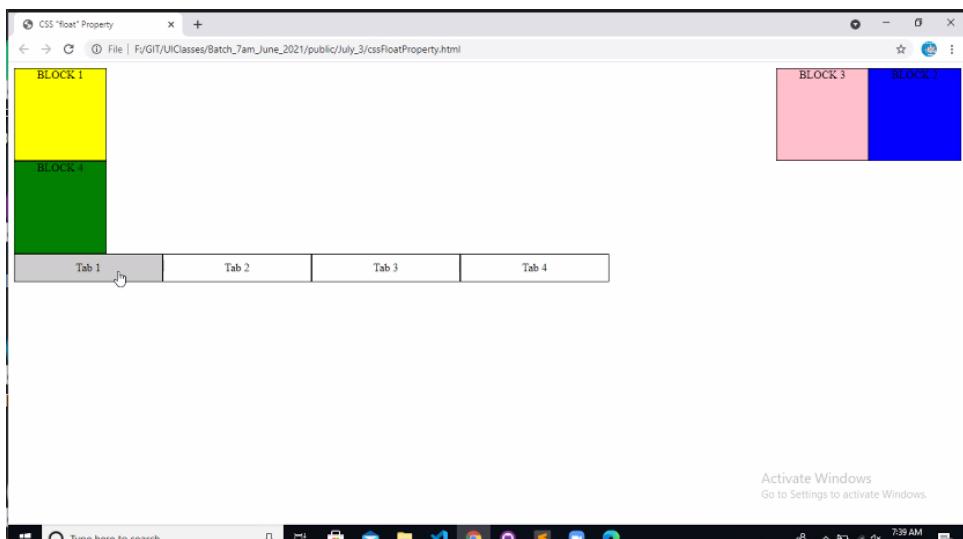
File Edit Selection Find View Goto Tools Project Preferences Help

```
<html>
<head>
    <style>
        .block1 {
            background: pink;
            float: right;
        }
        .block4 {
            background: green;
            clear: both;
        }
        .tabContainer div {
            border: 1px solid;
            padding: 10px 30px;
            width: 150px;
            text-align: center;
            float: left;
        }
        .tabContainer div:hover {
            background: #ccc;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <div class="block block1">
        BLOCK 1
    </div>
    <div class="block block2">
        BLOCK 2
    </div>
    <div class="block block3">
        BLOCK 3
    </div>
    <div class="block block4">
        BLOCK 4
    </div>
<div class="tabContainer">
```

Activate Windows  
Go to Settings to activate Windows.

Line 49, Column 35. Severe F:\GIT\UI\Classes\Batch\_7\mune\_2021\public\July\_3\cssFloatProperty.html (UTF-8)

Type here to search



## HTML TABLES:

following are predefined tables supported in html using which we could be able to render any data in the form of row and column wise.

- table -> Holds the complete table content
- thead -> To hold table header
- tbody -> holds the body of table
- th -> table header cell
- tr -> hold row of table
- td-> holds table data cell
- tfoot -> Holds footer of table

```
<table>
  <thead>
    <tr>
      <th>heading1</th>
      <th>heading 2</th>
      <th>...</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>..</td>
      <td>...</td>
    </tr>
    <tr>
      ...
    </tr>
    ...
  </tbody>
  <tfoot>
    <tr>
      <td>..</td>
    </tr>
    <tr>...</tr>
  </tfoot>
</table>
```

### EXAMPLE-1:

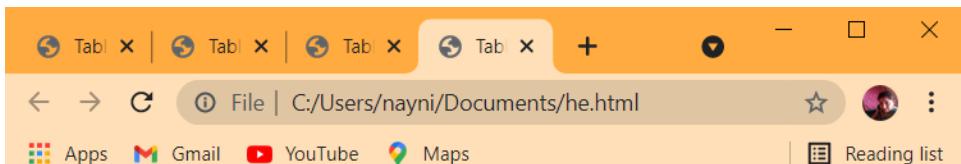
```
<!DOCTYPE html>
<html>
  <head>
    <title>Tables</title>
    <table border="1">
```

```

<tr>
    <th>Name</th>
    <td>Meghana</td>
    <td>Ravi</td>
    <td>Sreenu</td>
</tr>
<tr>
    <th>Height</th>
    <td>5'4"</td>
    <td>6'5"</td>
    <td>5'1"</td>
</tr>
<tr>
    <th>Age</th>
    <td>22</td>
    <td>25</td>
    <td>19</td>
</tr>
<tr>
    <th>Place</th>
    <td>Singapore</td>
    <td>Vizag</td>
    <td>rajasthan</td>
</tr>
</table>
</head>
</html>

```

#### OUTPUT:



The screenshot shows a web browser window with the address bar set to 'File | C:/Users/nayni/Documents/he.html'. Below the address bar, there are standard browser controls like back, forward, and search. The main content area displays a table with four rows and four columns. The first row has th cells labeled 'Name', 'Meghana', 'Ravi', and 'Sreenu'. The second row has th cells labeled 'Height', '5'4"', '6'5"', and '5'1''. The third row has th cells labeled 'Age', '22', '25', and '19'. The fourth row has th cells labeled 'Place', 'Singapore', 'Vizag', and 'rajasthan'. All cells appear to have a border.

Name	Meghana	Ravi	Sreenu
Height	5'4"	6'5"	5'1"
Age	22	25	19
Place	Singapore	Vizag	rajasthan

#### EXAMPLE-2:

Sublime Text (UNREGISTERED)

```

<!DOCTYPE html>
<html>
    <head>
        <title> Demonstrating Tables </title>
        <style type="text/css">
            table, tr, td, th {
                background: green;
                border: 2px dotted black;
            }
        </style>
    </head>
    <body>
        <table border="1px solid black" style="">
            <thead>
                <tr>
                    <th>Sno</th>
                    <th>Name</th>
                    <th>Age</th>
                    <th>Current Location</th>
                    <th>Prmt location</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>1</td>
                    <td>Raj</td>
                    <td>10</td>
                    <td>Hyderabad</td>
                    <td>New Delhi</td>
                </tr>
                <tr>
                    <td>2</td>
                    <td>Teena</td>
                    <td>11</td>
                    <td>Mumbai</td>
                    <td>Chennai</td>
                </tr>
                <tr>
                    <td>3</td>
                    <td>Krish</td>
                    <td>9</td>
                    <td>Chennai</td>
                    <td>Vizag</td>
                </tr>
                <tr>
                    <td>4</td>
                    <td>Meena</td>
                    <td>5</td>
                    <td colspan="2">>Chennai</td>
                </tr>
            </tbody>
            <tfoot></tfoot>
        </table>
    </body>
</html>

```

Activate Windows  
Go to Settings to activate Windows.

Line 10, Column 36: Seized F:\GIT\UI\classes\Batch\_7am\_June\_2021\public\July\_3\tablesData.html (UTF-8)

F:\GIT\UI\classes\Batch\_7am\_June\_2021\public\July\_3\tablesData.html - Sublime Text (UNREGISTERED)

Sublime Text (UNREGISTERED)

```

<table border="1px solid black" style="">
    <thead>
        <tr>
            <th>Current location</th>
            <th>Prmt location</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>Raj</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Teena</td>
        </tr>
        <tr>
            <td>3</td>
            <td>Krish</td>
        </tr>
        <tr>
            <td>4</td>
            <td>Meena</td>
        </tr>
        <tr>
            <td>5</td>
            <td colspan="2">>Chennai</td>
        </tr>
    </tbody>
    <tfoot></tfoot>
</table>

```

Activate Windows  
Go to Settings to activate Windows.

Line 27, Column 35: Seized F:\GIT\UI\classes\Batch\_7am\_June\_2021\public\July\_3\tablesData.html (UTF-8)

F:\ Type here to search 8:03 AM 7/3/2021

## OUTPUT:

File | F:\GIT\UI\classes\Batch\_7am\_June\_2021\public\July\_3\tablesData.html

Sno	Name	Age	Current Location	Prmt Location
1	Raj	10	Hyderabad	New Delhi
2	Teena	11	Mumbai	Chennai
3	Krish	9	Chennai	Vizag
4	Meena	5	Chennai	

DATE: 6/7/2021

## CSS 'display' property:

'display' is a CSS property through which we could be able to change the default rendering type of any DOM element.

Following are the possible values:

- ✗ display: block; -> makes the Dom element to render like a block level element.
- ✗ display: inline; -> makes any Dom element to render like an inline element.
- ✗ display: inline-block -> Makes any Dom element to render like a block level element, but occupies in same line as like inline element.
- ✗ display: none; -> makes element to not be shown on page. (It still exists in Dom structure)
- ✗ display: flex -> A css3 property to render flexible items on page.
- ✗ Etc...

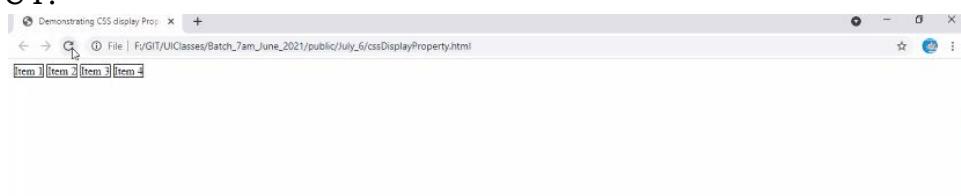
### Example:

The screenshot shows a Sublime Text editor window with the title bar "F:\GIT\UI\Classes\Batch\_7am\_June\_2021\public\July\_6\cssDisplayProperty.html - Sublime Text (UNREGISTERED)". The code in the editor is as follows:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>
5       Demonstrating CSS display Property
6     </title>
7     <style type="text/css">
8       .item {
9         width: 120px;
10        height: 100px;
11        text-align: center;
12        border: 1px solid;
13        display: inline; }
14     </style>
15   </head>
16   <body>
17     <div class="item"> Item 1</div>
18     <div class="item"> Item 2</div>
19     <div class="item"> Item 3</div>
20     <div class="item"> Item 4</div>
21   </body>
22 </html>
```

The status bar at the bottom shows "Line 13, Column 33: Saved F:\GIT\UI\Classes\Batch\_7am\_June\_2021\public\July\_6\cssDisplayProperty.html (UTF-8)".

### OUTPUT:



- Display Block:

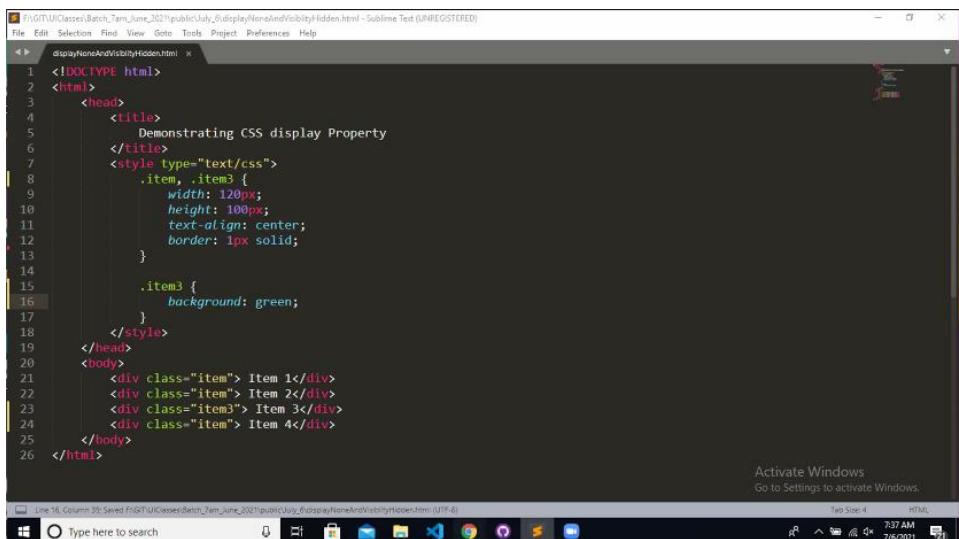
```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>
5       Demonstrating CSS display Property
6     </title>
7     <style type="text/css">
8       .item {
9         width: 120px;
10        height: 100px;
11        text-align: center;
12        border: 1px solid;
13        display: inline;
14      }
15
16      .block {
17        width: 120px;
18        height: 100px;
19        text-align: center;
20        border: 1px solid;
21        display: block;
22      }
23    </style>
24  </head>
25  <body>
26    <div class="item"> Item 1</div>
27    <div class="item"> Item 2</div>
28    <div class="item"> Item 3</div>
29    <div class="block"> Item 4</div>
30  </body>

```

- Difference between display ‘none’ and visibility ‘hidden’ properties:  
Both CSS properties are used to make the DOM element to be not visible on the page where the only difference is display ‘none’ makes the element to be not visible on the page, it doesn’t even occupy its own space.  
Visibility ‘hidden’ also makes the element to be not shown on the page but it still occupies its own space with in the page.  
Note: Both visibility hidden and display ‘none’ properties makes element to be not visible on the page but the element still exists within the DOM structure.

Example:

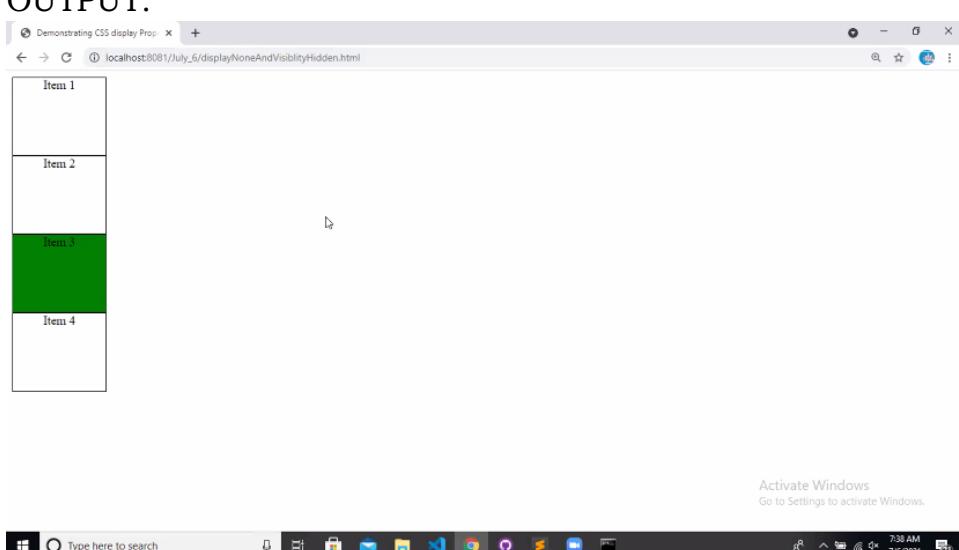


```

<!DOCTYPE html>
<html>
  <head>
    <title> Demonstrating CSS display Property </title>
    <style type="text/css">
      .item, .item3 {
        width: 120px;
        height: 100px;
        text-align: center;
        border: 1px solid;
      }
      .item3 {
        background: green;
      }
    </style>
  </head>
  <body>
    <div class="item"> Item 1</div>
    <div class="item"> Item 2</div>
    <div class="item3"> Item 3</div>
    <div class="item"> Item 4</div>
  </body>
</html>

```

Activate Windows  
Go to Settings to activate Windows.



Now if we want to hide item-3 we use following properties:  
 Display none;  
 Visibility hidden;

**Example for display: none;**

```

<!DOCTYPE html>
<html>
<head>
<title> Demonstrating CSS display Property </title>
<style type="text/css">
.item, .item3 {
width: 120px;
height: 100px;
}
```

```

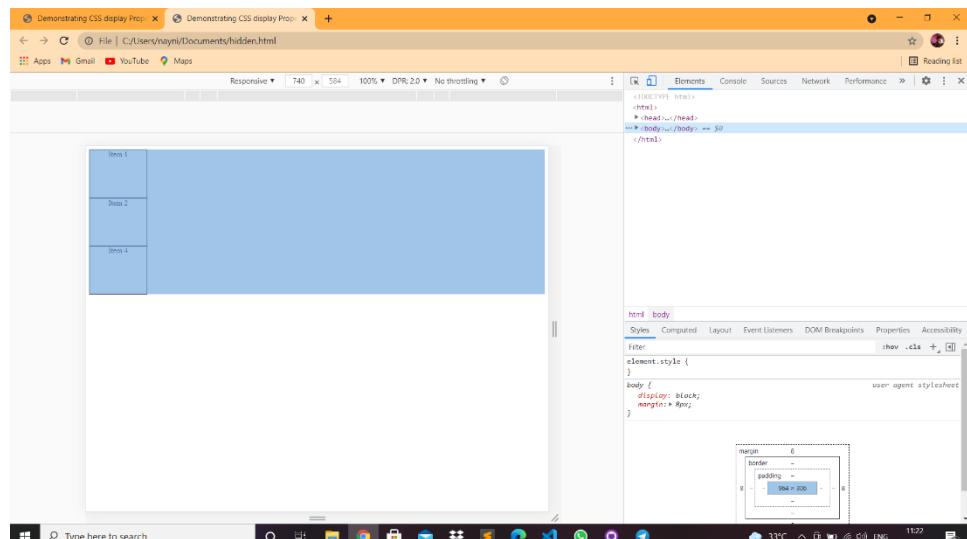
text-align: center;
border: 1px solid;
}

.item3 {
background: green;
display: none;
}

</style>
</head>
<body>
<div class="item"> Item 1</div>
<div class="item"> Item 2</div>
    <div class="item3">Item 3</div>
<div class="item"> Item 4</div>
</body>
</html>

```

## OUTPUT:



Example for visibility: hidden;

```

<!DOCTYPE html>
<html>
<head>
<title>
Demonstrating CSS display Property
</title>
<style type="text/css">
.item,.item3 {
width: 120px;

```

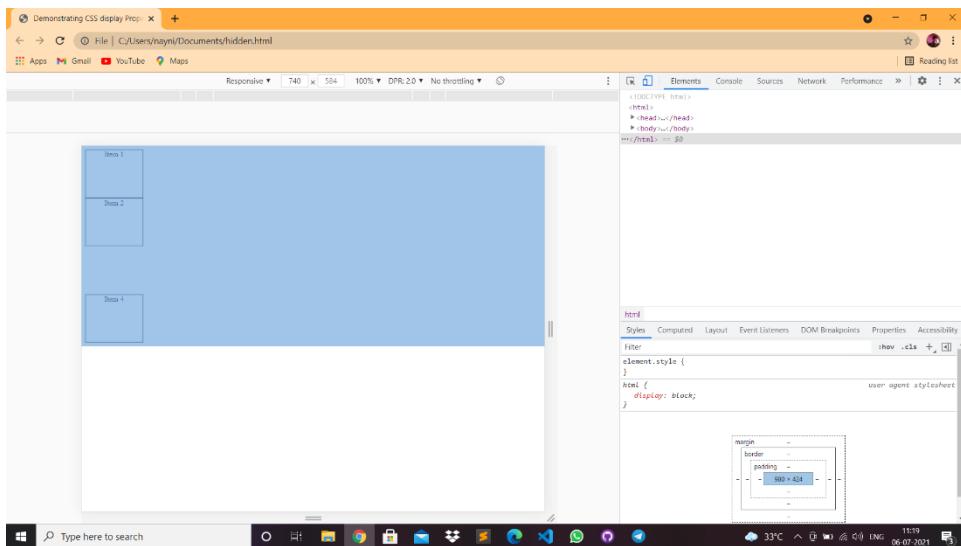
```

height: 100px;
text-align: center;
border: 1px solid;
}

.item3 {
background: green;
/*display: none; */
visibility: hidden;
}
</style>
</head>
<body>
<div class="item"> Item 1</div>
<div class="item"> Item 2</div>
<div class="item3"> Item 3</div>
<div class="item"> Item 4</div>
</body>
</html>

```

#### OUTPUT:



**CSS pseudo classes:** Following are the predefined pseudo classes been supported using which we could able to apply the CSS on elements not on load of the page but based on current state of the element.

- **: hover -> applies set of css when there is hover**
- **: empty -> Applied to any element which doesn't have child elements**
- **: disabled -> To element with disable state**
- **: enabled -> To element with enable state**

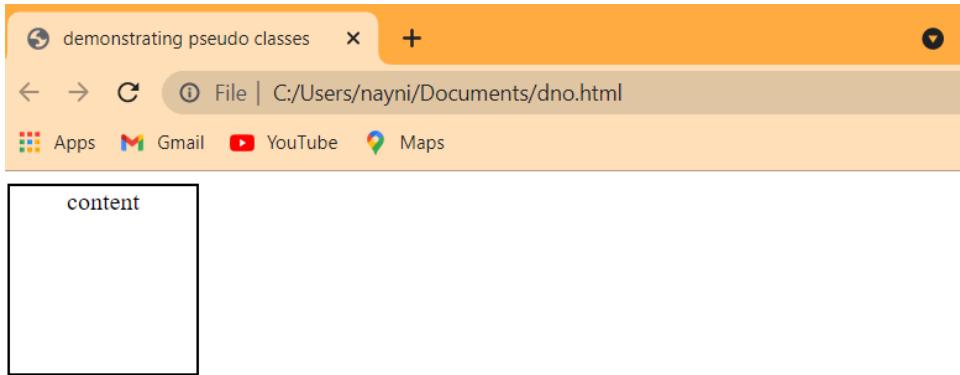
- **: active** -> Selects any active link
- **: checked** -> selects checkbox element with checked state
- **: focus** -> Selects element when it's in focus state
- **: first-child** -> element which is in first child state.
- **: last-child** -> element which is in last child state.
- **div: first-of-type** -> to select every div element which is the first div of its parent.
- **a: link** -> selects unvisited links
- **a: visited** -> Selects visited links
- **nth-child (2)** -> selects element in 2nd position
- **p:only-child** - selects p tag which is only child of its element
- etc.

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>
demonstrating pseudo classes
</title>
<style type="text/css">
.container {
border: 2px solid;
width: 120px;
height: 120px;
text-align: center;
}
.container:hover {
background: green;
font-size: 20px
}

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

Output:

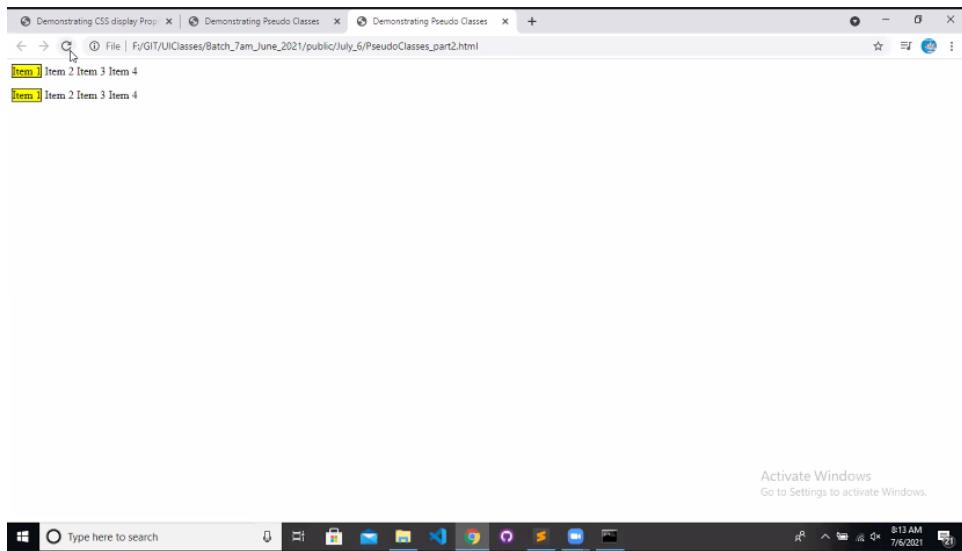


## Pseudo Classes example 2:

```
<!DOCTYPE html>
<html>
<head>
    <title> Demonstrating Pseudo Classes </title>
    <style type="text/css">
        span:first-child {
            border: 1px solid;
            background: yellow;
        }
    </style>
</head>
<body>
    <div>
        <span>Item 1</span>
        <span>Item 2</span>
        <span>Item 3</span>
        <span>Item 4</span>
    </div>
    <p>
        <span>Item 1</span>
        <span>Item 2</span>
        <span>Item 3</span>
        <span>Item 4</span>
    </p>
</body>
</html>
```

The screenshot shows a Sublime Text 3 editor window with the file "PseudoClasses\_part2.html" open. The code uses the `:first-child` pseudo-class to style the first child `span` element within both the `div` and `p` elements. The first `span` in each container has a 1px solid border and a yellow background. The rest of the `span` elements are plain black text. A status bar at the bottom indicates the file is at line 27, column 12, and shows the Windows activation message: "Activate Windows Go to Settings to activate Windows."

Output:



### Example 3 (for middle i.e., nth value)

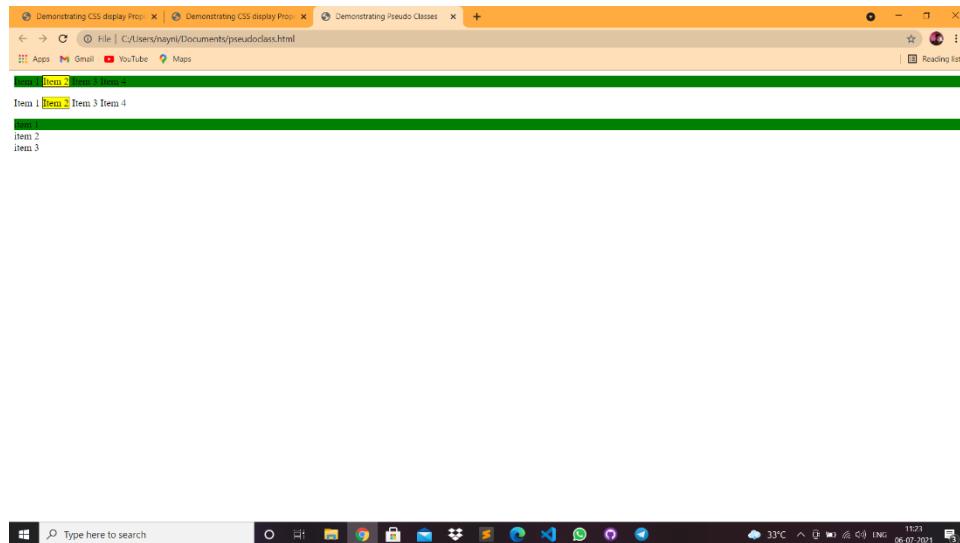
```
<!DOCTYPE html>
<html>
<head>
<title>
Demonstrating Pseudo Classes
</title>
<style type="text/css">
span: nth-child (2) {
border: 1px solid;
background: yellow;
}

div: first-of-type {
background: green;
}
</style>
</head>
<body>
<div>
<span>Item 1</span>
<span>Item 2</span>
<span>Item 3</span>
<span>Item 4</span>
</div>
<p>
<span>Item 1</span>
<span>Item 2</span>
<span>Item 3</span>
```

```
<span>Item 4</span>
</p>
```

```
<div>
<div>item 1</div>
<div>item 2</div>
<div>item 3</div>
</div>
</body>
</html>
```

Output:



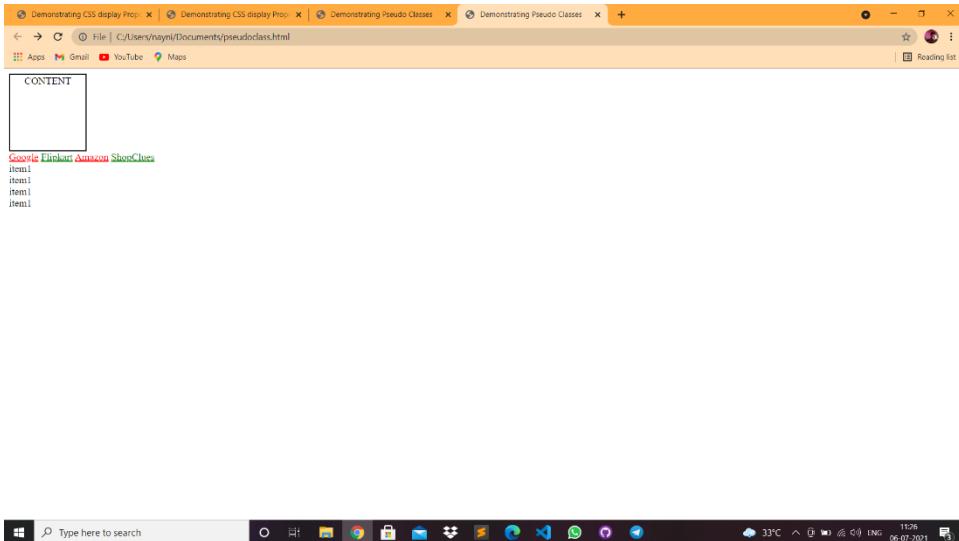
### Highlighting the image or links in pseudo-Classes:

```
<!DOCTYPE html>
<html>
<head>
<title>
Demonstrating Pseudo Classes
</title>
<style type="text/css">
.container {
border: 2px solid;
width: 120px;
height: 120px;
text-align: center;

}
```

```
.container: hover {  
background: green;  
font-size: 20px;  
}  
a: link {  
color: green;  
}  
a: visited {  
color: red;  
}  
a: active {  
background: red;  
}  
a: focus {  
background: yellow;  
}  
div: empty {  
border: 2px solid;  
}  
</style>  
</head>  
<body>  
<div class="container">  
CONTENT  
</div>  
<a href="http:www.google.com">Google</a>  
<a href="http:www.flipkart.com">Flipkart</a>  
<a href="http:www.amazon.com">Amazon</a>  
<a href="http:www.shopclues.com">Shop Clues</a>  
<div>  
<div>item1</div>  
<div>item1</div>  
<div>  
<span>item1</span>  
</div>  
<div>item1</div>  
</div>  
</body>  
</html>
```

Output:



DATE: 7/07/2021

#### Pseudo Elements:

Following are the predefined pseudo elements been supported using which we could able to apply CSS for not to the complete element but partially to the content of the element.

- ❖ ::after -> Add any content after selected element.
  - It inserts the content after the content of the element. It is used to add something after the specific part of an element. Generally, it is used with the **content** property.
  - It also allows us to add regular strings or images after the content.
- ❖ ::before -> Add any content before the selected element.
  - It allows us to add something before the element's content. It is used to add something before the specific part of an element. Generally, it is used with the **content** property.
  - We can also add the regular strings or images before the content using this pseudo-element.
- ❖ ::first-letter -> Only gets applied to first letter of the container.
  - It affects the first letter of the text. It can be applied only to block-level elements. Instead of supporting all CSS properties
- ❖ ::first-line -> Only gets applied to first line in content.
  - It is similar to the ::first-letter pseudo-element, but it affects the entire line. It adds the special effects to the first line of the text.
- ❖ ::marker -> to set CSS for makers of list items.

❖ ::selection -> Selects the part/portion of element content which is selected.

- It is used to style the part of an element that is selected by the user. We can use the following CSS properties with it:
- **color.**
- **background-color.**
- Other properties include **cursor**, **outline**, etc.

**Sample Definition:** CSS pseudo-elements are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-element is as follows  
= selector: pseudo-element {property: value};

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>
Demonstrating Pseudo Elements
</title>
<style type="text/css">
p::first-letter {
font-size: 25px;
color: green;
font-weight: bold;
}
p::first-line {
background: yellow;
}
p::selection {
background: red;
color: green;
}
p::after {
content: 'Continue reading from main page';
color: blue;
text-decoration: underline;
}
.test::marker {
color: blue;
```

```
font-size: 20px;  
}  
</style>  
</head>  
<body>  
<ul>  
<li class="test">  
item 1  
</li>  
<li>  
item 2  
</li>  
<li>  
item 3  
</li>  
</ul>  
<h3>  
News Item 1:  
</h3>  
<p>
```

In a historic move, a new 'Ministry of Cooperation' has been created by the Narendra Modi Government to realise the vision of 'Shakar se Samriddhi' (prosperity from cooperation).

Government sources told CNN-News18 that the ministry would provide a separate administrative, legal and policy framework to strengthen the cooperative movement in the country. It will also help deepen cooperatives as a true people-based movement reaching upto the grassroots.

The creation of a new ministry comes a day ahead of the much-awaited cabinet reshuffle.

"In our country, a co-operative based economic development model is very relevant where each member works with a spirit of responsibility," a source said.

```
</p>  
<h3>  
News Item 1:  
</h3>  
<p>
```

In a historic move, a new ‘Ministry of Cooperation’ has been created by the Narendra Modi Government to realise the vision of ‘Shakar se Samriddhi’ (prosperity from cooperation).

Government sources told CNN-News18 that the ministry would provide a separate administrative, legal and policy framework to strengthen the cooperative movement in the country. It will also help deepen cooperatives as a true people-based movement reaching up to the grassroots.

The creation of a new ministry comes a day ahead of the much-awaited cabinet reshuffle.

“In our country, a co-operative based economic development model is very relevant where each member works with a spirit of responsibility,” a source said.

</p>

<h3>

News Item 1:

</h3>

<p>

In a historic move, a new ‘Ministry of Cooperation’ has been created by the Narendra Modi Government to realise the vision of ‘Shakar se Samriddhi’ (prosperity from cooperation).

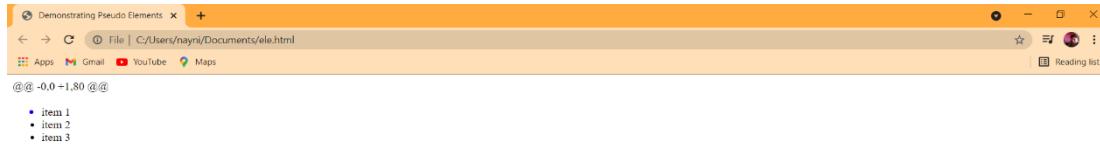
Government sources told CNN-News18 that the ministry would provide a separate administrative, legal and policy framework to strengthen the cooperative movement in the country. It will also help deepen cooperatives as a true people-based movement reaching up to the grassroots.

The creation of a new ministry comes a day ahead of the much-awaited cabinet reshuffle.

“In our country, a co-operative based economic development model is very relevant where each member works with a spirit of responsibility,” a source said.

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

Output:



News Item 1:

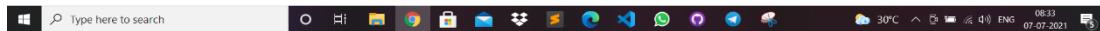
In a historic move, a new 'Ministry of Cooperation' has been created by the Narendra Modi Government to realise the vision of 'Sahkar se Samridhhi' (prosperity from cooperation). Government sources told CNN-News18 that the ministry would provide a separate administrative, legal and policy framework to strengthen the cooperative movement in the country. It will also help deepen cooperatives as a true people-based movement reaching upto the grassroots. The creation of a new ministry comes a day ahead of the much-awaited cabinet reshuffle. "In our country, a co-operative based economic development model is very relevant where each member works with a spirit of responsibility," a source said. [Continue reading from main page](#)

News Item 1:

In a historic move, a new 'Ministry of Cooperation' has been created by the Narendra Modi Government to realise the vision of 'Sahkar se Samridhhi' (prosperity from cooperation). Government sources told CNN-News18 that the ministry would provide a separate administrative, legal and policy framework to strengthen the cooperative movement in the country. It will also help deepen cooperatives as a true people-based movement reaching upto the grassroots. The creation of a new ministry comes a day ahead of the much-awaited cabinet reshuffle. "In our country, a co-operative based economic development model is very relevant where each member works with a spirit of responsibility," a source said. [Continue reading from main page](#)

News Item 1:

In a historic move, a new 'Ministry of Cooperation' has been created by the Narendra Modi Government to realise the vision of 'Sahkar se Samridhhi' (prosperity from cooperation). Government sources told CNN-News18 that the ministry would provide a separate administrative, legal and policy framework to strengthen the cooperative movement in the country. It will also help deepen cooperatives as a true people-based movement reaching upto the grassroots. The creation of a new ministry comes a day ahead of the much-awaited cabinet reshuffle. "In our country, a co-operative based economic development model is very relevant where each member works with a spirit of responsibility," a source said. [Continue reading from main page](#)



DATE:8/7/2021

- CSS 'Opacity' property: While elements get override each other we could able to control the transparency level of elements through CSS 'opacity' property.
- It takes a value between 0-1.

Example:

opacity: 0.3;

**Example program:**

```
<!DOCTYPE html>
    <html>
        <head>
            <style>
                div {
                    background-color: #4CAF50;
                    padding: 10px;
                }

                div.First {
                    opacity: 0.1;
                }
            </style>
        </head>
        <body>
            <div>
                <h1>Hello World</h1>
            </div>
        </body>
    </html>
```

```
}
```

```
div. Second {  
    opacity: 0.3;  
}
```

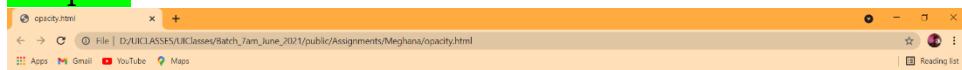
```
div. Third {  
    opacity: 0.6;  
}  
</style>  
</head>  
<body>
```

## <h1>The opacity Property</h1>

<p>The opacity property adds transparency to the background of an element, and to all of its child elements as well. This makes the text inside a transparent element hard to read:</p>

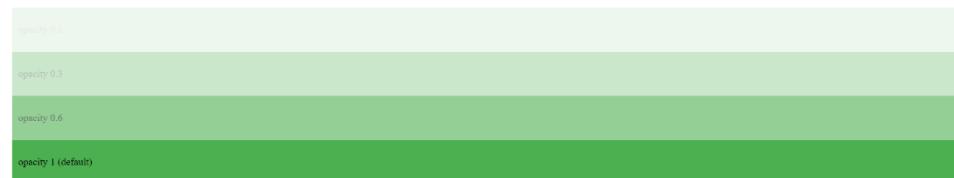
```
<div class="first"><p>opacity 0.1</p></div>  
<div class="second"><p>opacity 0.3</p></div>  
<div class="third"><p>opacity 0.6</p></div>  
<div><p>opacity 1 (default)</p></div>  
  
</body>  
</html>
```

### Output:



#### The opacity Property

The opacity property adds transparency to the background of an element, and to all of its child elements as well. This makes the text inside a transparent element hard to read:



### Example-2:

The screenshot shows a Sublime Text window with the following CSS code:

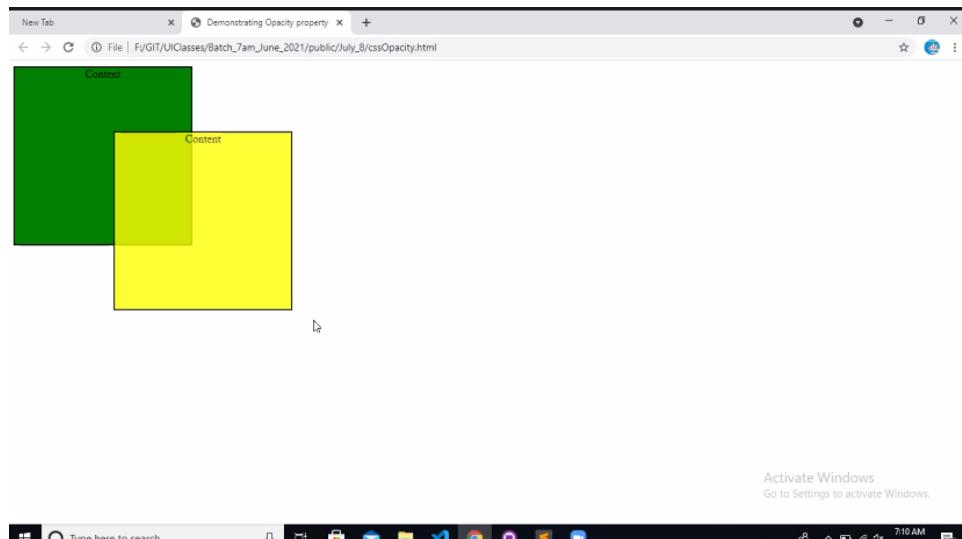
```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>
5       Demonstrating Opacity property
6     </title>
7     <style type="text/css">
8       .container {
9         width: 250px;
10        height: 250px;
11        border: 2px solid;
12        text-align: center;
13      }
14      .container1 {
15        background: green;
16      }
17      .container2 {
18        background: yellow;
19        position: absolute;
20        left: 150px;
21        top: 100px;
22        opacity: 0.8;
23      }
24    </style>
25  </head>
26  <body>
27    <div class="container container1">Content</div>
28    <div class="container container2">Content</div>
29  </body>

```

The code defines a container with a width and height of 250px, containing two nested divs: container1 (green) and container2 (yellow). Container2 is positioned absolutely at left: 150px and top: 100px, with an opacity of 0.8.

**Output:**



**HTML 'Input' Elements:**

Following are the predefined HTML elements supported using which we could able to read different types of data from the user.

- <input type="text" > --> To read the type of content.
- <input type="password" > --> To read sensitive data.
- <input type="radio" > --> creates a radio button.
- <input type="checkbox" > --> creates a checkbox.
- <input type="button" > --> creates a button.

textarea --> creates a multiline text container.

```
<select>
<option>one</option>
<option>two</option>
</select> --> creates a dropdown with multiple options.  
Etc.
```

**Example-1:**

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Student Registration Form</title>
    <style>
        * {
            font-family: Arial, Helvetica, sans-serif;
        }

        input {
            padding: 10px;
            margin: 10px;
        }

        button {
            margin: 10px;
        }
        form {
            margin: 10px;
            padding: 10px 20px;
            border: rgb(93, 1, 180) solid 2px;
        }
        h1 {
            text-decoration: underline;
            color: blueviolet;
        }
        a {
            text-decoration: none;
        }
    </style>
</head>

<body>
```

```

<center>
  <div>
    
  </div>
  <h1>STUDENT REGISTRATION FORM</h1>
</center>
<form action="./thankYou.html" style="background-color: blueviolet;
color: white;">
  <table>
    <tr>
      <td>First Name:</td>
      <td><input type="text" maxlength="30" placeholder="Enter
first name" /> (max 30 characters a-z A-z)
        </td>
    </tr>
    <tr>
      <td>Last Name:</td>
      <td><input type="text" maxlength="30" placeholder="Enter
last name" /> (max 30 characters a-z A-z)
        </td>
    </tr>
    <tr>
      <td>Date of Birth</td>
      <td>
        <select name="" id="">
          <option value="null">Day:</option>
          <option value="1">1</option>
          <option value="2">2</option>
          <option value="3">3</option>
          <option value="4">4</option>
          <option value="5">5</option>
          <option value="6">6</option>
          <option value="7">7</option>
          <option value="8">8</option>
          <option value="9">9</option>
          <option value="10">10</option>
          <option value="11">11</option>
          <option value="12">12</option>
          <option value="13">13</option>
          <option value="14">14</option>
          <option value="15">15</option>
          <option value="16">16</option>
          <option value="17">17</option>
        </select>
      </td>
    </tr>
  </table>
</form>

```

```

<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
<select name="Month" id="">
    <option value="null">Month:</option>
    <option value="Jan">Jan</option>
    <option value="Feb">Feb</option>
    <option value="Mar">Mar</option>
    <option value="Apr">Apr</option>
    <option value="May">May</option>
    <option value="Jun">Jun</option>
    <option value="Jul">Jul</option>
    <option value="Aug">Aug</option>
    <option value="Sept">Sept</option>
    <option value="Oct">Oct</option>
    <option value="Nov">Nov</option>
    <option value="Dec">Dec</option>
</select>
<select name="" id="">
    <option value="null">Year:</option>
    <option value="2020">2020</option>
    <option value="2019">2019</option>
    <option value="2018">2018</option>
    <option value="2017">2017</option>
    <option value="2016">2016</option>
    <option value="2015">2015</option>
    <option value="2014">2014</option>
    <option value="2013">2013</option>
    <option value="2012">2012</option>
</select>
</td>
</tr>
<tr>

```

```

<td>Email Id</td>
<td><input type="email" name="" id="" placeholder="Enter
email address"></td>
</tr>
<tr>
    <td>Gender</td>
    <td>
        <label for="male">Male</label><input type="radio"
name="gender" id="male">
        <label for="female">Female</label><input type="radio"
name="gender" id="female">
    </td>
</tr>
<tr>
    <td>Address</td>
    <td>
        <textarea name="" id="" cols="30" rows="4"></textarea>
    </td>
</tr>
<tr>
    <td>City</td>
    <td><input type="text" name="City" id="" maxlength="30"
placeholder="Enter City Name"> (max 30
characters a-z A-Z) </td>
</tr>
<tr>
    <td>PIN CODE</td>
    <td><input type="number" name="" id="" maxlength="6"
placeholder="Enter pincode"> (6-digit number)
    </td>
</tr>
<tr>
    <td>STATE</td>
    <td><input type="text" name="" id="" placeholder="Enter
state" maxlength="30"> (max 30 characters
a-z
A-Z) </td>
</tr>
<tr>
    <td>COUNTRY</td>
    <td><input type="text" name="" id="" placeholder="Enter
Country"></td>
</tr>
<tr>
    <td>HOBBIES</td>

```

```

<td>
    <label for="drawing">Drawing</label><input type="checkbox" name="hobbies" id="drawing">
    <label for="singing">Singing</label><input type="checkbox" name="hobbies" id="singing">
    <label for="dancing">Dancing</label><input type="checkbox" name="hobbies" id="dancing">
    <label for="sketching">Sketching</label><input type="checkbox" name="hobbies" id="sketching">
</td>
</tr>
<tr>
    <td></td>
    <td><label for="other">Others</label><input type="checkbox" name="hobbies" id="other"><input type="text" name="" id=""></td>
</tr>
</table>
<table>
    <tr>
        <td>QUALIFICATION</td>
        <td>Sr.No.</td>
        <td>Examination</td>
        <td><center>Board</center></td>
        <td><center>Percentage</center></td>
        <td><center>Year of Passing</center></td>
    </tr>
    <tr>
        <td rowspan="4"></td>
        <td>1</td>
        <td>Class X</td>
        <td><input type="text" name="" id="" maxlength="10"></td>
        <td><input type="text" name="" id=""></td>
        <td><input type="text" name="" id=""></td>
    </tr>
    <tr>
        <td>2</td>
        <td>Class XII</td>
        <td><input type="text" name="" id="" maxlength="10"></td>
        <td><input type="text" name="" id=""></td>
        <td><input type="text" name="" id=""></td>
    </tr>
    <tr>
        <td>3</td>
        <td>Graduation</td>
    </tr>

```

```

<td><input type="text" name="" id="" maxlength="10"></td>
<td><input type="text" name="" id=""></td>
<td><input type="text" name="" id=""></td>
</tr>
<tr>
    <td>4</td>
    <td>Masters</td>
    <td><input type="text" name="" id="" maxlength="10"></td>
    <td><input type="text" name="" id=""></td>
    <td><input type="text" name="" id=""></td>
</tr>
<tr>
    <td colspan="3"></td>
    <td><center>(10 char max) </center></td>
    <td><center>(up to 2 decimal) </center></td>
</tr>
<tr>
    <td>COURSES <br />APPLIED FOR</td>
    <td>
        <label for="bca">BCA</label><input type="radio" name="COURSES" id="bca">
        <label for="BCom">BCom</label><input type="radio" name="COURSES" id="BCom">
        <label for="bsc">B.Sc.</label><input type="radio" name="COURSES" id="bsc">
        <label for="ba">B. A</label><input type="radio" name="COURSES" id="ba">
    </td>
</tr>
</table>
<center>
    <div>
        <button type="submit">Submit</button>
        <button type="button"><a href=". ./index.html">Reset</a></button>
    </div>
</center>
</form>
</body>
</html>

```

**Output:**

**Student Registration Form**

File | D:\UICLASSES\UIClasses\Batch\_7am\_June\_2021\public\Assignments\Meghana\loc.html

Apps Gmail YouTube Maps

### STUDENT REGISTRATION FORM

First Name:  (max 30 characters a-zA-Z)

Last Name:  (max 30 characters a-zA-Z)

Date of Birth  Day  Month  Year

Email Id:  Enter email address

Gender: Male  Female

Address:

City:  Enter City Name (max 30 characters a-zA-Z)

**Student Registration Form**

File | D:\UICLASSES\UIClasses\Batch\_7am\_June\_2021\public\Assignments\Meghana\loc.html

Apps Gmail YouTube Maps

PIN CODE:  (6 digit number)

STATE:  Enter state (max 30 characters a-zA-Z)

COUNTRY:  Enter Country

HOBBIES: Drawing  Singing  Dancing  Sketching   
Others:

QUALIFICATION Sr.No.	Examination	Board	Percentage	Year of Passing
1	Class X	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	Class XII	<input type="text"/>	<input type="text"/>	<input type="text"/>
3	Graduation	<input type="text"/>	<input type="text"/>	<input type="text"/>
4	Masters	<input type="text"/>	<input type="text"/>	<input type="text"/>

(10 char max) (upto 2 decimal)

COURSES APPLIED FOR: BCA  B.Com  B.Sc  B.A

## TODAY'S EXAMPLE:

D:\UICLASSES\UIClasses\Batch\_7am\_June\_2021\public\Assignments\Meghana\loc.html - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Demonstrating HTML Input elements </title>
5     <style type="text/css">
6       ul {
7         border: 2px solid;
8         width: 500px;
9         margin: 10px auto;
10      }
11      li {
12        list-style: none;
13        border: 1px solid;
14        margin: 5px;
15        padding: 5px;
16      }
17      h3 {
18        text-decoration: underline;
19        text-align: center;
20        padding: 10px;
21        margin: 5px;
22      }
23      ul {
24        padding: 0;
25      }
26    </style>
27  </head>
28  <body>
29    <h3> Student Details Registration Page </h3>
30    <form action="http://www.sample2342.com/data/uinfo" method="POST">
31      <ul>
32        <li>
33
34
35
36

```

Line 14, Column 34

File | D:\UICLASSES\UIClasses\Batch\_7am\_June\_2021\public\Assignments\Meghana\loc.html

Apps Gmail YouTube Maps

32°C 08:20 08-07-2021

```

1 <label for="uname">Enter user Name:</label>
2 <input type="text" name="uname" required>
3 </li>
4 <li>
5     <input name="upwd" type="password" maxlength="5" placeholder="Acnt password">
6 </li>
7 <li>
8     Gender:
9     <input type="radio" name="ugender" value="male">Male
10    <input type="radio" name="ugender" value="female">Female
11 </li>
12 <li>
13     Known Languages:
14     <input type="checkbox" name="lng_eng">English
15     <input type="checkbox" name="lng_tel">Telugu
16     <input type="checkbox" name="lng_hin">Hindi
17     <input type="checkbox" name="lng_urdu">Urdu
18 </li>
19 <li>
20     Country:
21     <select name="ucountry">
22         <option></option>
23         <option>India</option>
24         <option>USA</option>
25         <option selected>Japan</option>
26         <option>Singapore</option>
27     </select>
28 </li>
29 <li>
30     Tell about Urself:
31     <textarea name="aboutUser"></textarea>
32 </li>
33 <li>
34 </li>
35 </ul>
36 </form>
37 </body>
38 </html>

```

## OUTPUT:

Student Details Registration Page

Enter user Name:	<input type="text" value="Meghana"/>
Acnt password:	<input type="password"/>
Gender:	<input type="radio"/> Male <input checked="" type="radio"/> Female
Known Languages:	<input checked="" type="checkbox"/> English <input checked="" type="checkbox"/> Telugu <input checked="" type="checkbox"/> Hindi <input type="checkbox"/> Urdu
Country:	<input type="button" value="India"/>
Tell about Urself:	<input type="text" value="sample"/>
	<input type="button" value="Register user"/>



DATE: 9/7/2021

### **HTML form tag:**

A predefined tag using which we could able to send user input data to server. It takes following mandatory attributes.

- . Method
- . Action

\*Method attribute through which we could specify the type of communication while sending or receiving the data.

\*Action attribute through which we specify the path of the server to which communication should happen.

```
<form action=" path/url of server "method=" GET/POST">  
...  
... // set of input elements.  
</form>
```

#### **❖ Types of communications:**

While communicating with a server, it could be either secure or non-secure type of communication.

1.non-secured type of communication (GET): In this type of communication, data will be sent to the server by appending to the URL as query parameters

E.g:

<http://www.sample.com/data/user/info?uname=test&age=20&gender=male>

2.Secured communication (POST): In this type of communication the data will be sent to the server by adding it to the request header, which is not exposed to the end user.

\*Any time we send sensitive data to server we use post type of communication.

---

---

---

---

#### ❖ JAVASCRIPT:

- ★ JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document.
  - ★ JavaScript was first known as Live Script, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java.
- ◆ Advantages of JavaScript
- ✖ The merits of using JavaScript are:
    - Less server interaction: You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
    - Immediate feedback to the visitors: They don't have to wait for a page reload to see if they have forgotten to enter something.
    - Increased interactivity: You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
    - Richer interfaces: You can use JavaScript to include such items as drag and drop.

#### ❖ Applications of JavaScript:

- ✖ Client-side validation,
- ✖ Dynamic drop-down menus,
- ✖ Displaying date and time,

- \* Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
- \* Displaying clocks etc.

Syntax of JavaScript:

```
<script ...>
    JavaScript code
</script>
```

EXAMPLE

```
<html>
<body>
<script language = "javascript" type = "text/javascript">
    <!--document.write("Hello World!")// -->
</script>
</body>
</html>
```

PROGRAM:

The Structure of JavaScript:

When you include any JavaScript code in an HTML document (apart from using the `<script>` tag), you must follow the few conventions:

1. HTML standards prior to HTML5 required that `<script>` tag to be placed between `<head>` and `</head>` tags at the starting of the document.
2. Unlike HTML which uses the `<! --- comment tag --->` commits inside Javascript code use the `//` symbol at the start of the line.

So, as I was always saying we use `<script>` tag let us know what is this `<script>` tag:

The `<script>` tag:

The `<script>` tag is used to include a JavaScript script in a webpage.

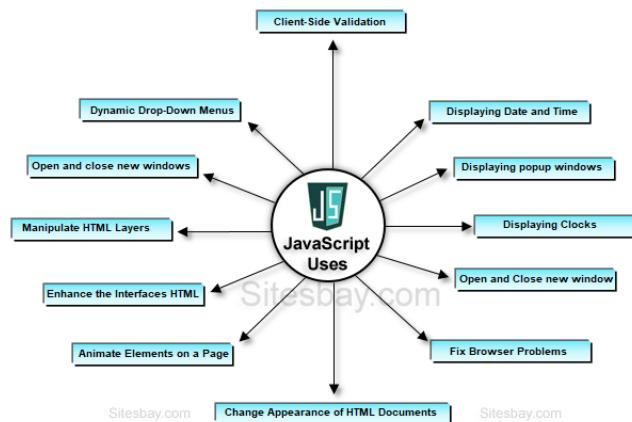
- @ The HTML `<script>` tag is used to define a client-side script (JavaScript).
- @ The `<script>` element either contains script statements, or it points to an external script file through the `src` attribute.
- @ Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.
- @ To select an HTML element, JavaScript most often uses the `document.getElementById()` method.
- @ This JavaScript example writes "Hello JavaScript!" into an HTML element with `id="demo"`:
- @

```
<script>
document.getElementById("demo"). inner HTML = "Hello
JavaScript!";
</script>
```

Why we use JavaScript:

JavaScript is mainly used for web-based applications and web browsers. But JavaScript is also used beyond the Web in software, servers and embedded hardware controls. Here are some basic things JavaScript is used for:

1. Adding interactive behaviour to web pages
2. Creating web and mobile apps
3. Building web servers and developing server applications
4. Game Development.



Date: 12/07/2021

Datatypes in JavaScript:

Datatypes is an important concept in JavaScript that will be able to operate on variables. The datatype specifies the type of data that a variable can hold. In JavaScript while declaring a variable it is not required to specify a datatype which declares the variables.

“VAR” is a pre-defined keyword through which we could declare the variables in JavaScript.

Basically, there are seven datatypes in JavaScript:

- \$ Numbers
  - o A=90
  - o B=9.4
- \$ String
  - o Code = 'a'
  - o Name = 'India'
- \$ Boolean

- IsFeePaid = true;
  - Isfemale = false;
- \$ Undefined  
\$ Null  
\$ function  
\$ Object

\*In side JavaScript we need not decide what type of data to store and how much memory it has to be occupied.

Ex: var a --> declaring a variable a capable of data

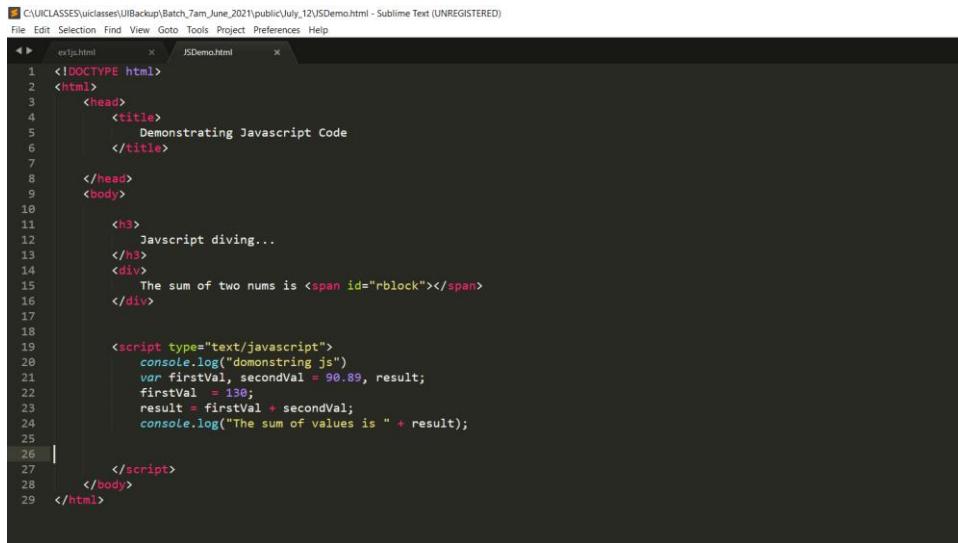
var b --> variable b capable of data

var c

a=4.5 -->number or a = "Hello" --> string

\*In JavaScript any line we are declaring should end with the semicolon. It indicated that is the end of the code.

Example:



```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>
5              Demonstrating Javascript Code
6          </title>
7
8      </head>
9      <body>
10         <h3>
11             Javascript diving...
12         </h3>
13         <div>
14             The sum of two nums is <span id="rblock"></span>
15         </div>
16
17
18     <script type="text/javascript">
19         console.log("domonstring js")
20         var firstVal, secondVal = 90.89, result;
21         firstVal = 138;
22         result = firstVal + secondVal;
23         console.log("The sum of values is " + result);
24
25     |
26
27     </script>
28 </body>
29 </html>

```

Example JavaScript program for adding two numbers:

```

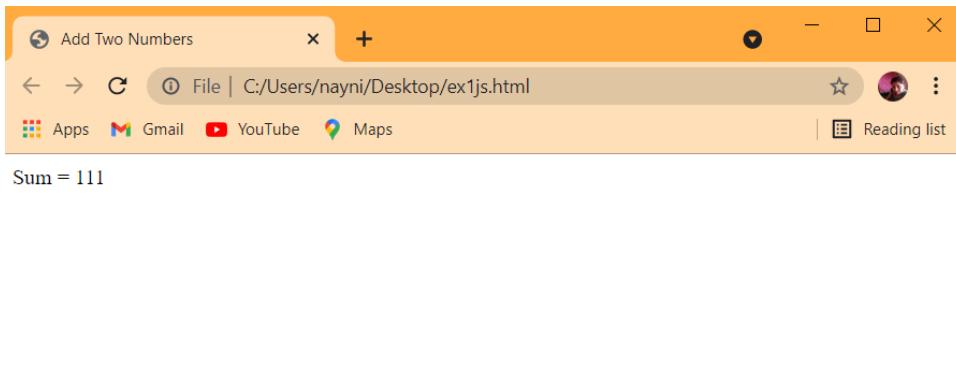
<!DOCTYPE html>
<html>
    <head>
        <title>Add Two Numbers</title>
    <script>
        var m = 91;
        var a = 20;
        var sum = m + a;
        document. Write ("Sum = " + sum);
    </script>

```

```
</head>
<body>

</body>
</html>
```

Output:



Date: 13/07/2021

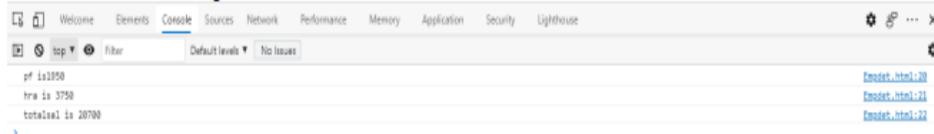
Write a program for employee details like sname,sage,basic sal,totsal,hra,pf and department and calculate the total salary?

```
<script type="text/javascript">
var sname="Meghana";
var sage=22;
var gender =" Female";
var dept=" CSE";
var basicsal=15000;

var totalsal,hra,pf;
pf=13/100*basic;
hra=25/100*basic;
totalsal = hra + basic + pf;

console.log ("pf is "+ pf);
console.log ("hra is "+ hra);
console.log ("totalsal is "+ totalsal);
```

```
</script>
```



## Type of Method:

A predefined method being supported in JavaScript using which we could able to get type of data a variable is holding. It takes a variable as a parameter and returns, the type of data the variable is holding.

Syntax: `typeof(Variable_Name);`

```
Ex: var num = 10;  
Var name= 'Meghana';  
console.log(typeof(age));  
console.log(typeof(name));
```

If any variable is not initialized it will be defined as undefined. It takes the variable parameter and returns the type of data the variable is holding.

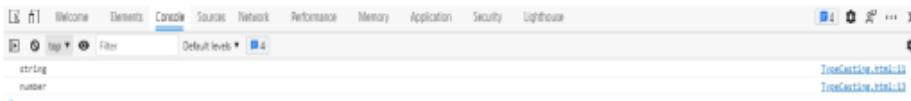
## Type Casting in JavaScript:

Typecasting, or type conversion, is a method of changing an entity from one datatype to another. It is used in computer programming to ensure variables are correctly processed by a function.

An example of typecasting is converting an integer to a string.

Example:

```
<script type="text/javascript">  
var num = 10;  
Var name= 'Meghana';  
console.log(typeof(num));  
console.log(typeof(name));  
</script>
```



## TODAYS EXAMPLE STUDENT DETAILS:

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

```

<title>
Demonstrating Student Details
</title>
</head>
<body>
<h3>
Student Details.
</h3>
<script type="text/javascript">

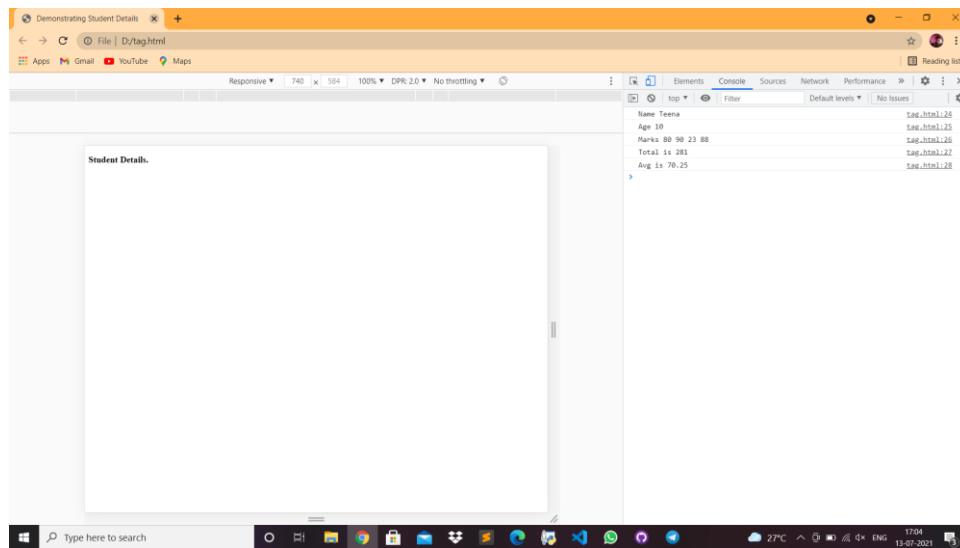
var sname = "Teena";
var sage = 10;
var gender = 'Female';
var m1 = 80;
var m2 = 90, m3 = 23, m4 = 88;
var total, avg;

total = m1 + m2 + m3 + m4;

avg = total / 4;
console.log ("Name " + sname);
console.log ("Age " + sage);
console.log ("Marks " + m1 + ' ' + m2 + " " + m3 + ' ' + m4);
console.log ("Total is " + total);
console.log ("Avg is " + avg);

</script>
</body>
</html>

```



Date: 14/07/2021

### Type Casting:

The process of converting datatype of a variable into another is called type casting.

This is classified into two types:

ParseInt () - Takes any datatype value and returns the number.

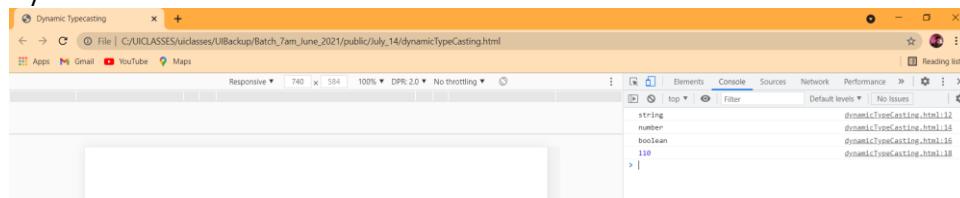
Parse float () - Takes any datatype value and returns the decimal number.

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>
Dynamic Typecasting
</title>
</head>
<body>
<script type="text/javascript">
var z = 80
var a = "30";
console.log(typeof(a));
a = 90;
console.log(typeof(a));
a = true;
console.log(typeof(a));
console.log (30 + z);

var c = 10 + z
```

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



Today's topic: Dynamic Type casting:

JavaScript internally implements and supports a feature called dynamic type casting in which it automatically converts the datatype of a variable based on type of data it is actually holding.

### STUDENT DETAILS EXAMPLE:

```

<!DOCTYPE html>
<html>
<head>
<title>
Demonstrating Student Details
</title>
</head>
<body>
<h3>
Student Details.
</h3>
<script type="text/javascript">
console.log ("About to start student details")
var sname = "Teena";
var sage = 10;
var gender = 'Female';
var m1 = 45;
var m2 = 50, m3 = 60, m4 = 40;
var total, avg;

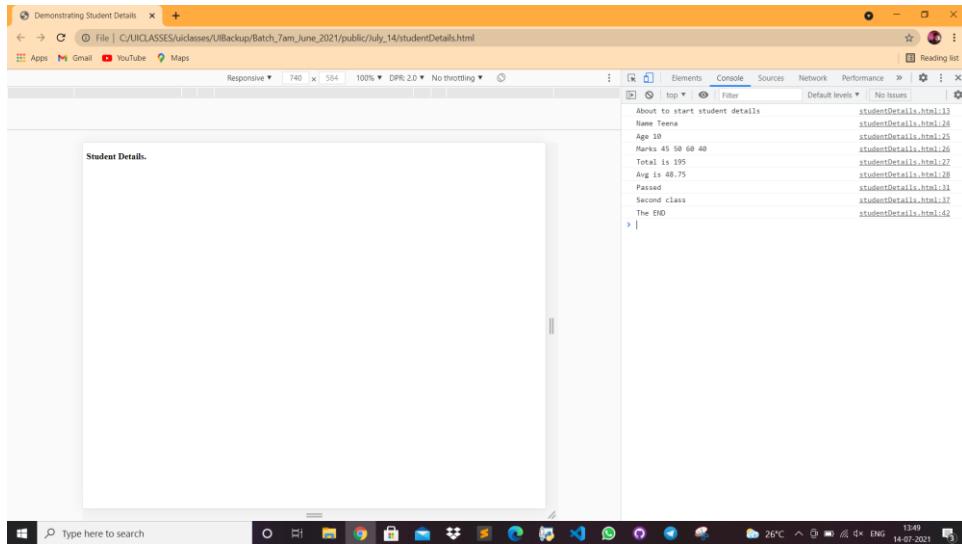
total = m1 + m2 + m3 + m4;

avg = total / 4;
console.log ("Name " + sname);
console.log ("Age " + sage);
console.log ("Marks " + m1 + ' ' + m2 + " " + m3 + ' ' + m4);
console.log ("Total is " + total);
console.log ("Avg is " + avg);

if (avg >= 40) {
  console.log("Passed");
  if (avg >= 90) {
    console.log ("you got Distinction")
  } else if(avg >= 60) {
    console.log ("you got first class");
  } else {
    console.log ("Second class");
  }
} else {
  console.log("Failed");
}
console.log ("The END");
</script>

```

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



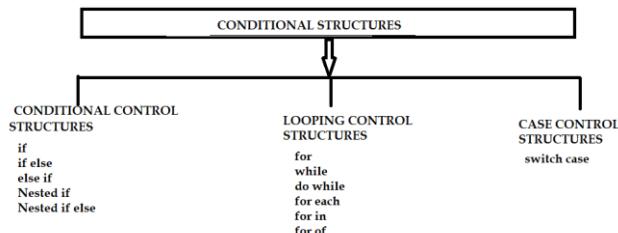
## Control structures:

Following are the set of control structures being supported through which we could able to control the sequence of execution flow in an application.

### 1. Conditional Control Structures.

### 2. Looping Control Structures.

### 3. Case Control Structures.



### 1. Conditional Control Structures:

- æ **IF Condition:** The **if** statement is the fundamental control statement that allows JavaScript to make decisions and execute statements conditionally.

#### æ Syntax:

```
if (condition) {
    // block of code to be executed if the condition is true}
```

}

æ Else: We use the else statement to specify a block of code to be executed if the condition is false.

æ Syntax:

```
if (condition) {  
    // block of code to be executed if the condition is  
    true  
}  
else {  
    // block of code to be executed if the condition is  
    false  
}
```

æ Else if Statement: Use the else if statement to specify a new condition if the first condition is false.

æ Syntax:

```
if (condition1) {  
    // block of code to be executed if the condition1 is true.  
}  
else if (condition2) {  
    // block of code to be executed if the condition1 is false  
    and condition 2 is true.  
}  
else {  
    // block of code to be executed if the condition1 is false  
    and condition 2 is False.  
}
```

æ Nested if: A nested if is an if statement that is the target of another if or else. Nested if statements mean an if statement inside an if statement. JavaScript allows us to nest if statements within if statements. i.e., we can place an if statement inside another if statement.

æ Syntax:

```
if (condition1) {  
    // block of code to be executed if the condition1 is true  
} if (condition2) {  
    // block of code to be executed if the condition 2 is true.  
}
```

Date: 15/07/2021

2. Looping control structures:

Looping control structures has been supported in Javascript takes the set of instructions and repeat them for multiple times.

JavaScript supports different kinds of loops:

- ❖ For - loops through a block of code a number of times
- ❖ for/in - loops through the properties of an object
- ❖ for/off- loops through the values of an iterable object
- ❖ while - loops through a block of code while a specified condition is true
- ❖ do/while - also loops through a block of code while a specified condition is true

➤ For Loop: Loops through a block of code a number of times.

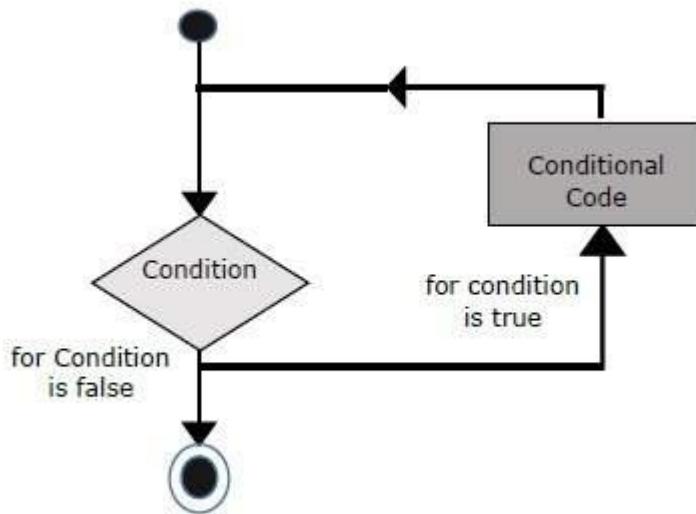
¶ The '**for**' loop is the most compact form of looping. It includes the following three important parts –

- The **loop initialization** where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.
- The **test statement** which will test if a given condition is true or not. If the condition is true, then the code given inside the loop will be executed, otherwise the control will come out of the loop.
- The **iteration statement** where you can increase or decrease your counter.

Syntax:

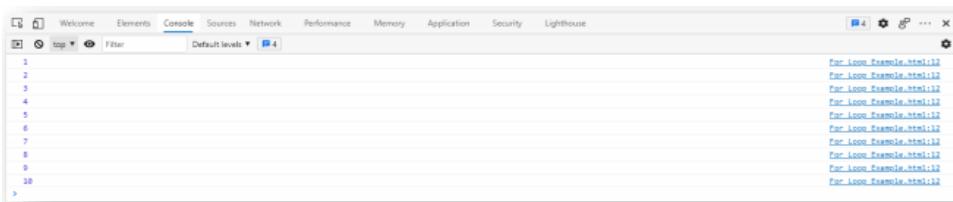
```
for (initialization, <condition>, increment/decrement)
    // set of lines to be Executed
}
```

¶ The flow chart of a for loop in JavaScript would be as follows –



Ex:

```
<script type="text/javascript">
    Var i
    Var j=10;
    For (i=1; i<=j; i++) {
        Console.log(i);
    }
</script>
```



### TODAY'S EXAMPLE:

```
<!DOCTYPE html>
<html>
<head>
<title>
for Looping CS
</title>
</head>
<body>
<script type="text/javascript">
console.log("Start")
for (var i = 1; i <= 4; i++) {
    console.log ("----- " + i)
    console.log ("Hello ");

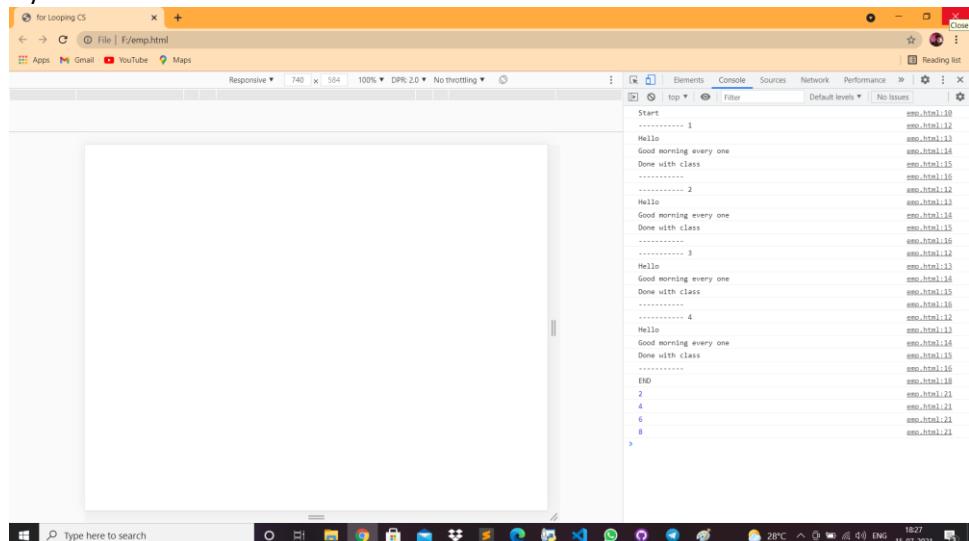
```

```

        console.log ("Good morning every one");
        console.log ("Done with class");
        console.log ("-----");
    }
    console.log("END");
    for (var i = 1; i <= 9; i++) {
        if ( i % 2 == 0) {
            console.log(i);
        }
    }

```

</script>  
</body>  
</html>



## EXAMPLE-2:

```

<!DOCTYPE html>
<html>
<head>
<title>
for Looping CS
</title>
</head>
<body>
<script type="text/javascript">
/*console.log("Start")
for (var i = 1; i <= 4; i++) {
console.log ("----- " + i)
console.log ("Hello ");
console.log ("Good morning every one");
console.log ("Done with class");

```

```

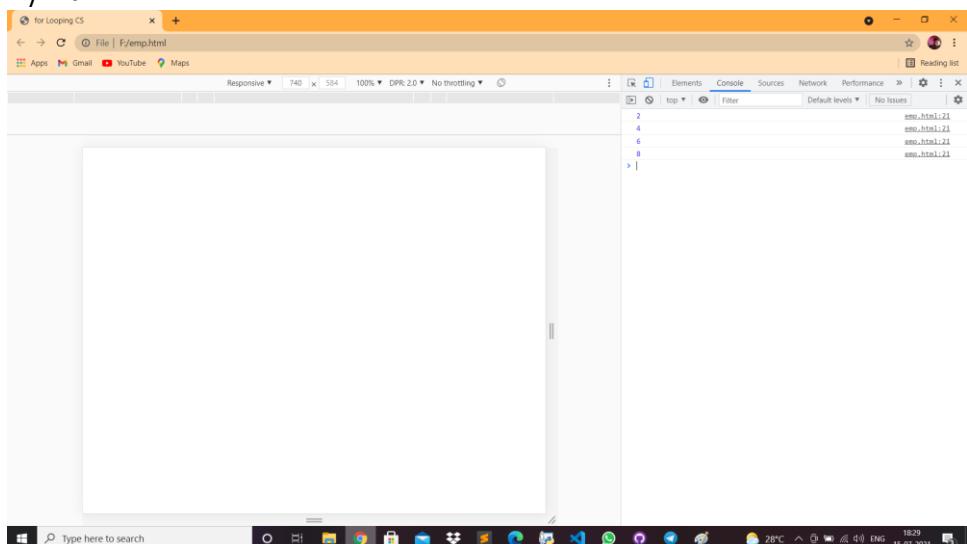
        console.log ("-----");
    }
    console.log("END"); */
    for (var i = 1; i <= 9; i++) {
        if ( i % 2 == 0) {
            console.log(i);
        }
    }
}

```

```

</script>
</body>
</html>

```



### Assignment:

Emp details – age, name, dept, basic Sal, gender  
Cal pf, total Sal,hra

---

Also calculate tax value

male emp – total Sal -> 2lakh -> tax percent is 15%  
male emp – total Sal -> 1lakh -> tax percent is 10%  
male emp – total Sal -> 50k -> tax percent is 5%  
Female emp – total Sal -> 2lakh -> tax percent is 10%  
Female emp – total Sal -> 1lakh -> tax percent is 5%  
Female emp – total Sal -> 50k -> tax percent is 0%

Difference between ‘=’, ‘==’ and ‘====’ operators:

- ❖ Equal to (=) is an assignment operator, which sets the variable on the left of the = to the value of the expression that is on its right. This operator assigns lvalue to rvalue.
- ❖ For example, writing a=10 is fine. If we write 10=10, 'a' = 10 or 'a' = 'a', it will result in a reference error.

- ❖ Double equals (==) are a comparison operator, which transforms the operands having the same type before comparison. So, when you compare string with a number, JavaScript converts any string to a number. An empty string is always converts to zero. A string with no numeric value is converts to NaN (Not a Number), which returns false.
- ❖ Note: == operator while comparing values only checks for value equality but not type equality
  
- ❖ === (Triple equals) is a strict equality comparison operator in JavaScript, which returns false for the values which are not of a similar type. This operator performs type casting for equality. If we compare 2 with "2" using ===, then it will return a false value.
- ❖ Note: === it is almost like '==' operator used to compare value equality where the only difference is while comparing values it not just checks for value equality but it also checks for type equality of data

Date: 17/07/2021

**Write a program to show the list of odd numbers between 59-11.**

```
<script type="text/javascript">
  for (var i =59; i>=11; i--)
  {
    if (i%2!=0)
    {
      console.log ("odd number +i");
    }
  }
</script>
```

**Write a program to show the list of even numbers between 1-100.**

```
<script type="text/javascript">
  for (var i =1; i>=100; i--)
  {
    if (i%2 ==0)
    {
      sum = sum + i;
    }
  }
  console.log (sum is "+sum);
</script>
```

**Write a program to show the given the number is prime or not.**

```

<script type="text/javascript">

    function checkPrime() {

        var n, i, flag = true;

        n = document.myform.n.value;
        n = parseInt(n)
        for (i = 2; i <= n - 1; i++)
            if (n % i == 0) {
                flag = false;
                break;
            }

        // Check and display output
        if (flag == true)
            console.log (n + " is prime");
        else
            console.log (n + " is not prime");
    }
</script>

```

## **ASSIGNMENT:**

Write a program to find sum of all the prime numbers between 1 –100  
 Write a program to show the table of a given number.

Date: 19/07/2021

- While loop control structure: It is almost like for loop control structure used to iterate instructions more than once where the only difference is while loop only takes the condition initialization, increment or decrement can be placed anywhere with the code.

Syntax:

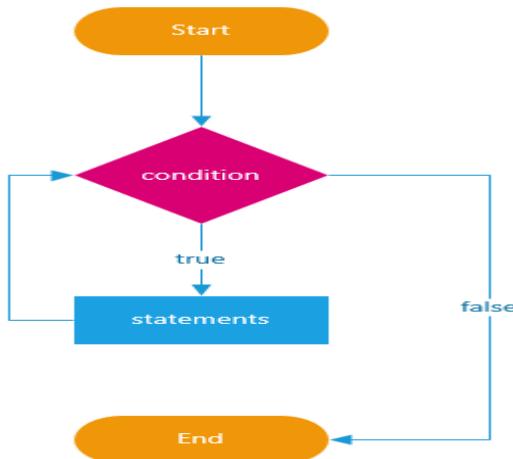
```

While (<condition>) {
.....
.....
..... // set of instructions gets executed for multiple
times, until the provided condition is unsatisfied.

```

- ¶ The while statement evaluates the expression before each iteration of the loop.

- If the expression evaluates to true, the while statement executes the statement. If the expression evaluates to false, execution continues with the statement after the while loop.
- The while loop evaluates the expression before each iteration; therefore, the while loop is known as a pre-test loop. For this reason, it is possible that the statement inside the while loop is never executed.
- The following flowchart illustrates the while loop statement:



**Example:**

```

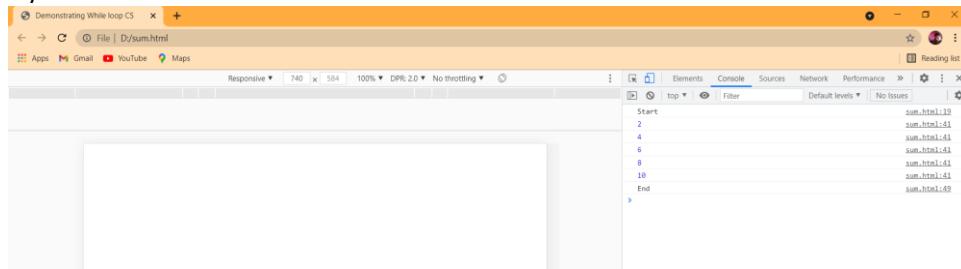
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
Demonstrating While loop CS
</title>
</head>
<body>
<script type="text/javascript">
console.log("Start");
/*for (var i = 1; i <= 10; i++) {

if (i % 2 == 0) {
console.log(i);
}
}*/
var i = 1;
while ( i <= 10) {
  
```

```

if (i % 2 == 0) {
    console.log(i);
}
i++;
}
console.log("End")
</script>
</body>
</html>

```



### SUM OF GIVEN NUMBERS:

```

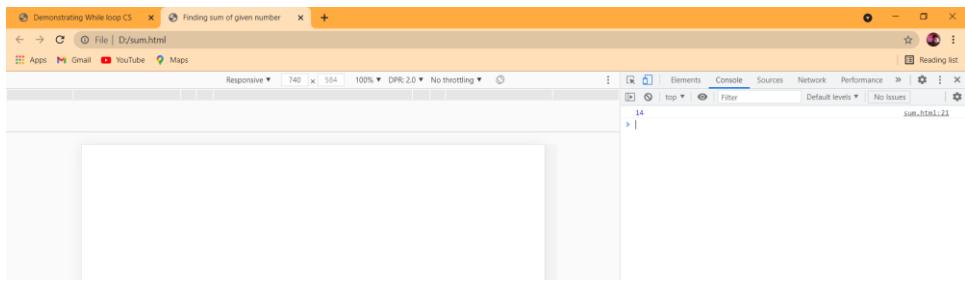
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
Finding sum of given number
</title>
</head>
<body>
<script type="text/javascript">

var n = 2345;
var rem, sum = 0;
while (n > 0) {
rem = n % 10;
sum = sum + rem;
n = n / 10;
n = parseInt(n);

};

console.log(sum)
</script>
</body>
</html>

```



## ASSIGNMENTS:

1. Write a program to find out the lucky number of a given number.
2. Write a program to find out the reverse of a given number.
3. Write a program to check whether given number is palindrome number or not.
4. Write a program to list out all the palindrome numbers between 300-21.
5. Write a program to list out reverse of numbers between 32-950.
6. Write a program to check whether given number is Armstrong number or not.

Date: 20/07/2021

- Prompt Method: This method is used to display a dialog box, which would require a person to type in input.
- The JavaScript prompt () method is commonly used for prompting the user for input before entering a web page.
- [REDACTED]

## PROMPT WINDOW

**Note:** Once a prompt box appears, the user has to click either ok or cancel after entering the input. However, do not use this method excessively, as it prevents access to parts of the webpage until submitted, making the user experience less smooth.

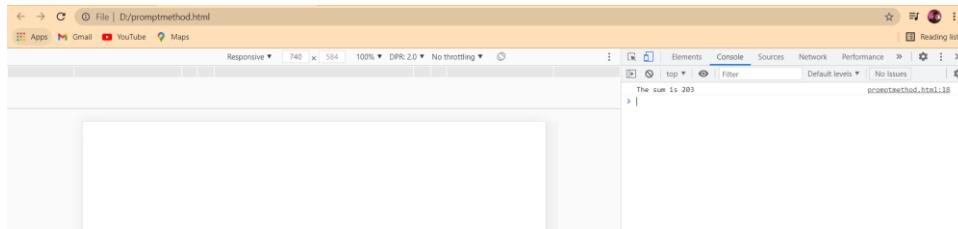
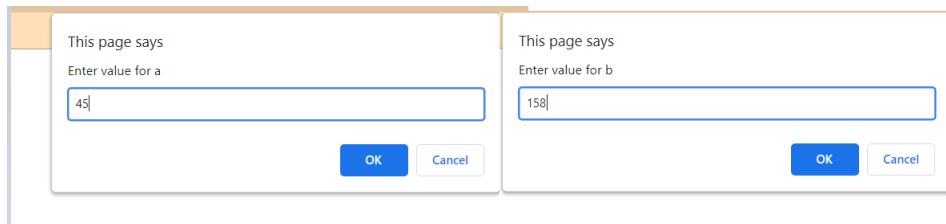
Example:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
Demonstrating Prompt Method
</title>
```

```

</head>
<body>
<script type="text/javascript">
var a = prompt ("Enter value for a");
var b = prompt ("Enter value for b");
a = parseInt(a);
b = parseInt(b);
var result = a + b;
console.log ("The sum is " + result);
</script>
</body>
</html>

```

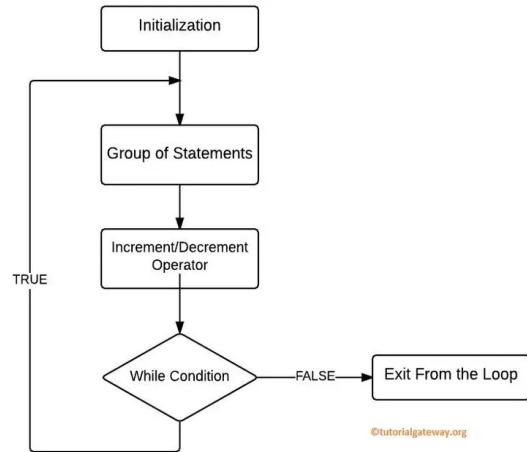


## DO-WHILE LOOP:

- The do-while creates a loop that executes a specified statement until the test condition evaluates to false. The condition is evaluated after executing the statement, resulting in the specified statement executing at least once.
  - It is almost like a while loop control structure used to repeat a set of instructions for more than once where the only difference is while loop until executes the instructions only when the provided condition is true, whereas do-while executes set of instructions at least for once irrelevant of where the provided where the condition given is true or false.
  - Syntax:
- ```

do
  code      block      to      be      {
    }
  while (condition);

```



Example:

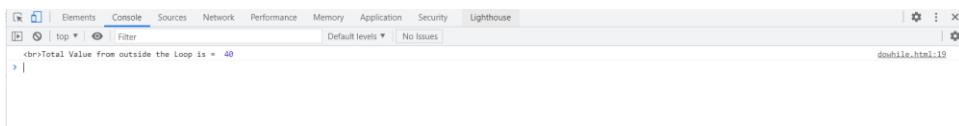
```

<!DOCTYPE html>
<html>
<head>
    <title> JavaScript Do While Loop </title>
</head>

<body>
    <h1> JavaScript Do While </h1>
<script>
    var number = 6, total=0;
    do
    {
        total = total + number;
        document.write("<br>Number = " + number);
        document.write("<br>Total Value is = " + total);
        number++;
    }
    while (number <= 10);

    console.log("<br>Total Value from outside the Loop is = ", total);
</script>
</body>
</html>

```



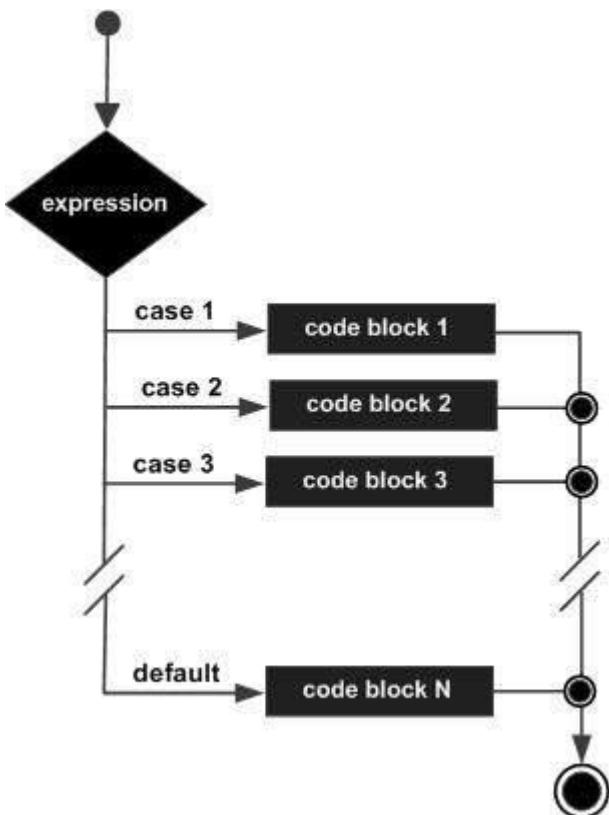
Date: 21/07/2021

#### Variable Hosting:

- ❖ The var statement declares a function-scoped or globally-scoped variable, optionally initializing it to a value. The var declarations, wherever they occur, are processed before any code is executed. This is called hosting.
- ❖ In JavaScript, Hoisting is the default behaviour of moving all the declarations at the top of the scope before code execution. Basically, it gives us an advantage that no matter where functions and variables are declared, they are moved to the top of their scope regardless of whether their scope is global or local. It allows us to call functions before even writing them in our code.

#### Case Control Statements--->Switch Case:

- ❖ It takes multiple cases within it and invokes corresponding particular case based on the value that is been passed.
- ❖ The switch statement evaluates an expression. The value of the expression is then compared with the values of each case in the structure. If there is a match, the associated block of code is executed.
- ❖ The switch statement is often used together with a break or a default keyword (or both). These are both optional:
- ❖ The break keyword breaks out of the switch block. This will stop the execution of more execution of code and/or case testing inside the block. If break is omitted, the next code block in the switch statement is executed.
- ❖ The default keyword specifies some code to run if there is no case match. There can only be one default keyword in a switch. Although this is optional, it is recommended that you use it, as it takes care of unexpected cases



Syntax:

```

Switch (<value>) {
  case <value 1>:
  .....
  break;
  case <value 2>:
  .....
  break;
  .....
  default :
  .....// gets executed automatically if none of cases matches
}
```

Example:

```

<script type = "text/javascript">

<! -- var grade = 'A';

document.write("Entering switch block<br />");

switch (grade)
```

```

{
  case 'A': document.write("Good job<br />");
  case 'B': document.write("Pretty good<br />");
  case 'C': document.write("Passed<br />");
  case 'D': document.write("Not so good<br />");
  case 'F': document.write("Failed<br />");
  default: document.write("Unknown grade<br />")

}
document.write("Exiting switch block");
//--> </script>

```

---



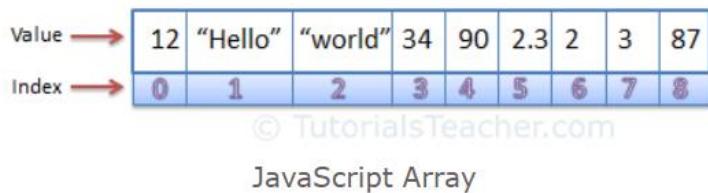
---

## **ARRAYS:**

- A predefined data structure through which we could be able to hold group of multiple values within it.
- In JavaScript arrays are heterogeneous data structure capable of holding different types of data.
- Array is a single variable that is used to store different elements.
- It is often used when we want to store list of elements and access them by a single variable.
- '[]' braces are always used to represent arrays in any language.
- All the values within the array will automatically assigned a numeric index value "ZERO.

Here are the two different ways we can create an array in JavaScript:

- Var a= new array () // Dynamic allocation.  
(or)
- Var a = []; // Empty Array  
Var a = [25,25,25,5]; // Declaring an array with value



Example: A sample example for 2-D element of Array:

```
var a = [[1, 2, 1, 24], [8, 11, 9, 4], [7, 0, 7, 27], [7, 4, 28, 14], [3, 10, 26, 7]];
```

```
for (var i in a)
{
    console.log ("row " + i);
    for (var j in a[i])
    {
        console.log (" " + a[i][j]);
    }
}
```

Date: 24/07/2021

### **Assignment:**

- ❖ Write a program to find out all the even numbers with in n-array?
- ❖ Write a program to find Sum of odd numbers within array?
- ❖ Write a program to list out the prime numbers within given array?
- ❖ Write a program to find out the palindrome numbers within the given array?
- ❖ Write a program to find largest and smallest value from an array?
- ❖ Write a program to find the second largest value from an array?
- ❖ Write a program to take an array and sort array in both ascending and descending order?
- ❖ Write a program to find out total number of times a given value is getting repeated in an array?

### → **Pre-defined Methods in arrays:**

- ❖ The strength of JavaScript arrays lies in the array methods. Array methods are functions built-in to JavaScript that we can apply to our arrays
- ❖ Each method has a unique function that performs a change or calculation to our array

❖ The pre-defined methods in javascript are given below that basically supports the javascript which can be applied on arrays:

- length – It is a property which returns the total number of values in an array.
- push(<values>) - This method is used to insert single or multiple values to an array from its right direction
- pop () - This method is used to delete the single value from an array from its right direction.
- shift () - This method is used to delete a single value from an array from its left direction.
- unshift (<values>) - This method is used to insert single or multiple values to an array from its left direction
- splice (start position, <no of values to delete>, <optional values to be inserted>) - This splice method is used to insert or delete the values in an array.
- join () - This method is used to merge all the values of an array to a single value with or without Seperator.
- includes () - This method checks if an array contains the specified element

Example:

```
<script type="text/javascript">
var details = [34, 34, 78, 90, 56, 89, 100, 102];
/*console.log(details);
console.log(details.length);
details.push(100);
console.log(details);
details.unshift(101);
console.log(details);
//-----
details.pop();
console.log(details);
details.shift();
console.log(details);
details.splice(4, 0, 201, 202, 304);
console.log(details);

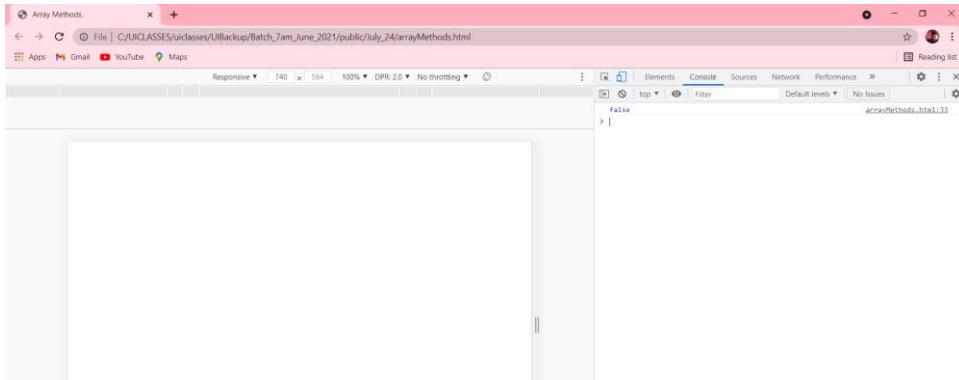
console.log(details.join(""))
```

```

var a = [45, 67, 90];
var b = [34, 11, 22, 33];
var c = a.concat(b);
console.log(c) */

var details = [34, 56, 89, 90, 100];
console.log(details.includes(99))
</script>

```



### Example-2:

```

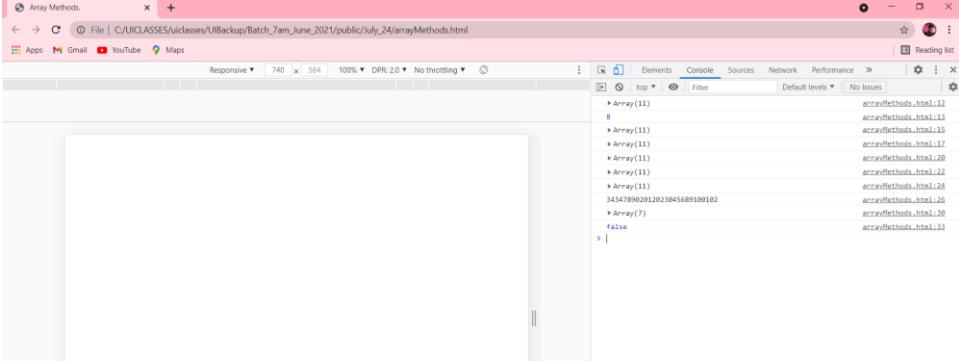
<script type="text/javascript">
var details = [34, 34, 78, 90, 56, 89, 100, 102];
console.log(details);
console.log(details.length);
details.push(100);
console.log(details);
details.unshift(101);
console.log(details);
//-----
details.pop();
console.log(details);
details.shift();
console.log(details);
details.splice(4, 0, 201, 202, 304);
console.log(details);

console.log(details.join())
var a = [45, 67, 90];
var b = [34, 11, 22, 33];

```

```
var c = a.concat(b);
console.log(c)
```

```
var details = [34, 56, 89, 90, 100];
console.log(details.includes(99))
</script>
```



A screenshot of a browser's developer tools console window. The title bar says "Array Methods." The tabs at the top are Elements, Console, Sources, Network, and Performance. The Console tab is selected. The status bar shows "Responsive" mode, "740" width, "504" height, "100%" zoom, and "DPR 2.0". The console output is as follows:

```
Array(11)
B
Array(11)
Array(11)
Array(11)
Array(11)
Array(11)
341478902012023045689100102
Array(7)
false
```

The code being run is:

```
var details = [34, 56, 89, 90, 100];
console.log(details.includes(99))
</script>
```