**FrontEnd Batch**

**HTML** (Hypertext Mark-up Language)**:**

**Tags:**

Div,span,p,h1 to h6,a,input.

**Types of Tags:**

1.Self closing Tags: Ex:<img/>

2.Double Closing Tags Ex:<div><div/>

**Div:**

Its block element when we use div tag it occupies the entire space on screen by default

**Span:**

Its inline element it occupies only space of the content. In exception case if we use p or h tags inside span it occupies 100% width why because h and p tag normally consists of 100% width.

**H1 to h6:**

These h tags usually represent heading and 1 to 6 indicates the bold and size.Every number in h tags are contains specific size and weight.

**A:**

A tag usually represent link tags used to navigate to some other page or url.A tags has multiple attributes which allows us to open the url in same page or new page.

**Input:**

Input tag nothing but user can enter values in multiple ways like text,number,date and time,checkbox,and radio button.

**Button:**

Button tags name itself represent this is used to submit a form. Ex:<button>Submit<button/>

**Table:**

Syntax of table tags:

<table>

<tr>

<th> Name</th>

<tr/>

<tr>

<td>XYZ<td/>

<tr/>

<table/>

**Comments:**

Comments are used to identify the other user’s to understand what we had done in this code.Ex: <!-- Write your comments here -->

**HTML5 Features and Tags:**

**Header and Footer:**

Usually we can able to design the header and footer of the page using normal div and for understaning logics we can also make use of header and footer tags.Ex:<header><header/>

<footer><footer/>

**Audio and Video:**

Audio and video tags usually use to play audio or video content in screen. Both tag contains multiple inbuilt functions like autoplay which will play when page reloaded and mute will mute the audio in beginning.Ex:<audio><source src=”audio.mp3”><source/><audio/>

**Img tag:**

Img tag used to insert image into the screen we can also modify the height and width of image using inbuilt functions. Ex:<img src=”123.jpeg”/>

**Localstorage:**

Localstorage is mostly used in real time scenarios also to logout the user after certain period such that 24 to 48 hours.These things are mostly used in day to day applications like amazon,facebook and flipkart.Hence it stores some token in our computer for certain period of time.These feature can be achieved with help of javascript.

**Figure and FigCaption:**

Image and Figure tag are mostly common but here figcaption tool used to give caption for the image.

Usage:

<figure><img src=’img.jpeg’/><figcaption>name of the image<figcaption/><figure/>

**Nav:**

Nav tags used to place content in top of the page which we had seen the application has some Home,Service and about option in top right corner.

**Progress:**

Progress tool is most important tool used to know how much is completed here we have to define two attributes one is max and other is value.

**Placeholder Attribute:**

Placeholder attribute used to be in default on input tag.Mostly used to tell the user what they can enter in particular input tag.Ex: <input placeholder=”Enter your name”/>

**List:**

UL represent unorderlist and OL represents Ordered List.

EX:

<ul>

<li>Home<li/>

<li>About</li>

<ul>

**CSS3:**

**Background:**

background-image:url(“img.jpeg”) [Used to insert image in the background]

background-size:contain(Adjust the image size according to the container) or cover(Enlarge the image size) or we can give pixels

background-repeat:repeat(by default) [Will repeat the image to cover the div) no-repeat[Not repeat the image)

background-position:left,right,top,bottom [helps us to place image in required places]

background-attachments:fixed[will not move image when scroll] or scroll[scroll image along with content]

**Border,Margin,Padding:**

Border has multiple property like border top,bottom,left,right color.But we can use short end property like border:1px(border left right top bottom) solid(border- width) black(border-color).

Same as border margin and padding also has the same attribute but we can use shorthand property like margin:10px and padding:20px;

**Height and Width:**

* + Min height represent the height will give take atleast 100px and max height represent maximum height of div ie.atmost 200px.Max height allows us to enable scroll option.
  + Width also had same properties like min and max.
  + Height and width we can give it in pixels,percentage and vh(view height-for height),vw(view width-for width)

**Fonts or Text:**

Font-style:italic,normal

Font-size:10px

Font-family:Times new roman

Color:blue

These above are the attributes of fonts and text used to customize the fonts and change the style.

**Icons:**

If we want use icons first we want to import CDN link from bootstrap,fontawesome.With the help of their customized class we can able to use the icons.

Ex: <i class="fas fa-cloud"></i>

**Display:**

Display:none(hides the element),block(display the content in horizontal manner),inline(display the contents in vertical manner),inline-block(same as inline but also we can able to give height and width),flex(display the content in row and column wise,mostly used style in css).

**Position:**

Position:absolute(overlaps the another div),fixed(stays in same place),relative(places below the div)

These position tags will be used with combination of left right top and bottom property.

**Z-Index:**

Z-Index property allows the picture to overlap or placed behind based on values given.

z-index:-30 place the img behind the content.

**Overflow:**

Overflow:hidden(hides the content if the div doesn’t fit the content),scroll(allows the content to read by scrolling into horizontal or vertical).

**Opacity:**

Opacity:0 hides the element and opacity:50 shows the image partially.

**Pseudo-class:**

This refers to mouse action for example div:hover{ we can add styles here}

**Pseudo-element:**

This refers to particular letter for example p::first-letter{we can make styles here}

**Comments:**

Comments are used to identify the other user’s to understand what we had done in this code.Ex: /\* Write your comments here \*/

**Counters:**

This style used to give automatic numbering system.

Usage:

body {  
  counter-reset: section;  
}  
  
h2::before {  
  counter-increment: section;  
  content: "Section " counter(section) ": ";  
}

**Transistion:**

Transistion usually enables smoothing of action designed by CSS.We need to add transistion to parent of element and transistion will work with combination of pseudo class elements.Transistion has multiple attributes such as  transition-property: width;  
transition-duration: 2s;transition-timing-function: linear; transition-delay: 1s;

Ex: div{transistion:1s ease-in;}

div:hover{width:300px}

**Transform:**

Transform is used to move or rotate or enlarge the element.

Ex: transform : translateY(100px) moves in y direction same for X also. And transform:rotateX(200deg) rotate the object in 200degree and transform:scaleX(10px) enlarges the element

**Shadow:**

Shadow will add shadow around box or text and we have property like [text-shadow: 2px 2px 5px red;] and for box [box-shadow: 2px 2px 5px red;]

**Animation:**

Animation is same as transition property we have to give animation in parent element but here we have to declare new name for animation for example:

Ex: div{background-color:red;animation:example} after this we have to create @keyframes and assign the animation name to it ex: @keyframes example{background-color:blue}

Other inbuilt properties of animations are:

animation-delay, animation-iteration-count, animation-duration…

**BootStrap:**

What is bootstrap?

Bootstrap is a CSS framework where creators modified the designs and we can take their design by simply adding class name or id.

How to use?

In order to use bootstrap we need to link CDN link in our HTML.This link can be found in official bootstrap website.

**Reference to learn HTML and CSS:**

<https://www.w3schools.com/css>

**Projects to be done after this:**

1.Create simple portfolio website

2.Display a any name with animation.

3.Take reference of any single page website and design exactly

**JavaScript**

**Why Javascript?**

Using HTML CSS we can do desing but for functional perspective we need javascript to perform for example if we have designed input box for login and password so we need to check whether username and password is correct so javascript will work on API sent request and get response and navigate our page to dashboard or something based on requirement.

**Variables:**

Variables is nothing but its name for some word, number or collections we have data types in JavaScript known as array, string, number and object. For every values we have to give one name to access anywhere in future.

Example: let name=”Angular” let number=5 let arr=[5,8,6,7] let ob={“id:1,”name”:XYZ}

We have let, const and var as key word.

Let and var known as global variable and these can be **updated** after initialization but const **cant be updated** its one time assignment.

To access array we have to use variable\_name[index]

To access Object we have to use variable\_name.key

**Operators:**

Operators are used to perform any operations with variables we have logical, ternary, arithmetic and assignment operators.

**Logical Operators:**

A logical operator usually compares with two variables and return as true or false.

Example: >,<,<=,>=,==,!=

10>5 returns true

**Ternary Operator:**

This operator behaves like if clause so syntax for this operator is Condition ? (if true then) : (else)

10 > 5 ? “Yes its greater” : “No its not greater”

**Arithmetic Operator:**

This operator used to perform mathematical operations like addition, subtraction, multiplication, divisions

Example: 10+5,10-8, 5\*5, 10/2 (gives quotient), 10%5 (gives remainder)

**Assignment Operator (=):**

This operator used to assign value to the variables.

Example: let num2 = 5;

**Inbuilt Loops:**

**If or If-else loop :**

This loops checks for true or false condition i.e Boolean conditions and execute the expression.

**Syntax:**

If(condition){ //condition must be return type of Boolean ( true or false)

Expression

}else{

Expression

}

**For loop :**

For Loop mostly used to iterate the collections like array.

Syntax:

for(initialization; condition; increment or decrement){

expression

}

Example:

for(let i=0;i<5;i++){

console.log(arr[i])

}

**While loop:**

For Loop mostly used to iterate the collections like array and its same as for loop.

Example:

Initialization

While(condition){

Expression

Increment/decrement

}

Let i=0;

While(i<6){

console.log(arr[i])

i++;

}

**String Functions/Methods:**

String length - returns the length of string

String slice(startIndex,endIndex) -Cut the characters based on index

String substring(startIndex,endIndex) -Cut the characters based on index

String substr(startIndex,length) -Cut the characters based on index and again we can cut characters

String replace(wordtoreplace,word) –replace the word with another word

String replaceAll(wordtoreplace,word) –replace the all the word with another word

String toUpperCase() – transform the all the word with UpperCase

String toLowerCase() – transform the all the word with LowerCase

String concat() – combine the two word

String trim() – remove the whitespace from the word

String trimStart() – remove the whitespace from the starting only

String trimEnd() – remove the whitespace from the ending only

String charAt(index) – remove the whitespace from the ending only

String split(seperator) – Change string to array

**String Search Methods**

String indexOf(word) – return the index of word from start.

String lastIndexOf(word)– return the index of word from end.

String search(word) - return the index of word from start and here we can use regular expression.

String includes(word) - return the true/false if word is found.

String startsWith(word) - return the true/false if word is found in start of sentence.

String endsWith()- return the true/false if word is found in last of sentence.

**Number Methods:**

toString() -Returns a number as a string

toExponential() -Returns a number written in exponential notation

toFixed() -Returns a number written with a number of decimals

toPrecision() -Returns a number written with a specified length

**Array Methods:**

Array length – returns the length of the array

Array toString() – converts array into string

Array pop() – removes the last element of the array

Array push() – add the last element of the array

Array shift(index) – removes the element based on index

Array unshift(word) – add the first element of the array

Array join(seperator) – convert “ , ” separator to anything

Array delete(index) – removes the element but not the memory

Array concat()- merge two arrays

Array splice(param1,param2,param2) :

fruits.splice(2, 0, "Lemon", "Kiwi");

The first parameter (2) defines the position where new elements should be added (spliced in).

The second parameter (0) defines how many elements should be removed.

The rest of the parameters ("Lemon" , "Kiwi") define the new elements to be added.

Array slice(index)

The slice() method slices out a piece of an array into a new array.

Array sort – sort the array based on ascending order

Array reverse – sort the array based on descending order

**Array Iteration:**

Map – returns the array and used for iteration

Filter – returns the array with only condition passed

Find – returns the single value with condition passed at first instance

Some – returns true or false If condition passed for one of element

Every – returns true or false if condition passed for every element

Spread Operator:

(…) - appends the array with many arrays

**MAP:**

Map is used like object.Syntax to create map

Let newMap=new Map([“id”,2], [“name”,”XYZ”]);

**SET:**

Set is used like array only difference is it doesn’t accept duplicate values in array

Example: let newSet=new Set([5,8,9,8)]

Output: Set {3} [5,8,9]

**Date:**

Date objects are created with the new Date() constructor.

Ex: let newDt=new Date() //returns current date and time

**Date Methods:**

getFullYear() Get year as a four digit number (yyyy)

getMonth() Get month as a number (0-11)

getDate() Get day as a number (1-31)

getDay() Get weekday as a number (0-6)

getHours() Get hour (0-23)

getMinutes() Get minute (0-59)

getSeconds() Get second (0-59)

getMilliseconds() Get millisecond (0-999)

getTime() Get time (milliseconds since January 1, 1970)

**Set Methods:**

setDate() Set the day as a number (1-31)

setFullYear() Set the year (optionally month and day)

setHours() Set the hour (0-23)

setMilliseconds() Set the milliseconds (0-999)

setMinutes() Set the minutes (0-59)

setMonth() Set the month (0-11)

setSeconds() Set the seconds (0-59)

setTime() Set the time (milliseconds since January 1, 1970)

Math Important Methods:

Math.min(5,10) // We can give any number of values which returns smallest

Math.max(80,20) // We can give any number of values which returns greatest

**Functions:**

Two types of function:

**1.Default Function :**

function myfunc(parameter1,parameter2,..){ // Declaration of function

}

myfunc(parameter1,parameter2,..) // Calling function

**2.Arrow Function**

let myfunc=( parameter1,parameter2,..)=>{ // Declaration of function

}

myfunc(parameter1,parameter2,..) // Calling function

**Class and Constructor:**

Class is a parent of entire thing and constructor is used to create object and can be only used inside class

Example:

public class Example1{

let name,age;

constructor(name,age)

this.name=name;

this.age=age; //this is a keyword used to tell that we are calling variable inside our class

}

**JSON:**

JSON is a format most widely used in getting response from api and passing request to api because its like object and easy to handle.

**Converting text to json**

let text = '{ "employees" : [' +

'{ "firstName":"John" , "lastName":"Doe" },' +

'{ "firstName":"Anna" , "lastName":"Smith" },' +

'{ "firstName":"Peter" , "lastName":"Jones" } ]}';

const obj = JSON.parse(text);

**Converting JSON to text:**

const obj = {name: "John", age: 30, city: "New York"};

const text = JSON.Stringify(obj);