

## 1. Basic JS

It is a scripting language  
Mostly used client side scripting language

Used for the web development  
It's an event based programming language

Like :  
Click  
Double click  
Mouse hover  
Mouse out  
Scroll  
Key

We can create website, desktop app development and app development using java script

For  
Web Development : jQuery  
Angular JS  
React JS  
Vue JS  
NodeJS --> server side scripting language

### Uses of java scripting in web Development

- ☐ Dropdown Menu
- ☐ Animated Slider
- ☐ Maps
- ☐ Chart Graphs
- ☐ Pop-up window
- ☐ Audio Players
- ☐ Video Players
- ☐ Zoom effects
- ☐ Animated Gallery
- ☐ Form Validation
- ☐ Accordions
- ☐ Calendar

### Software used for web Development --> Vs Code and web Browser

### Java Script Implementation

1. In Page JS
2. External JS

<script></script> is used to write in page.  
Mostly preferred to be written after the body tag.

- If the external JS is used then it is written inside the head tag.

### Document Command

--> document.write("hello world");

### HTML tag in JS

document.write("hello<br>Devi");

### Comments in JS

1. Single line comment : // ha-ha
2. Multiline Comment /\* hello \*/

### Variables in JS

1. Var
2. Let
3. Const

All the variables are case sensitive in JS

#### Var

Var x = "Devi Prasad Mishra";  
Var z = 10.245;

z= "wow"; // override will be done

#### Ways of writing a variables in JS

- a. firstname
- b. first\_name
- c. first-name
- d. firstName --> CamelCase
- e. Firstname99

#### Let

```
let x = "Devi Prasad Mishra";
```

A var can be declared twice and it will not occur any error but in let it will throw an error. i.e. the variable has been declared already. But it can be assigned again

### Const

```
const x = "hello";
```

It cannot be modified further when a variable is declared constant.

## JS Data Types

1. String
2. Number
3. Boolean
4. Array []
5. Object {}
6. X = null --> Null
7. Var x; --> undefined

To know the type of a variable we can use typeof method in JS

## operators in JS

### Arithmetic operators

- a. Addition +
- b. Subtraction -
- c. Multiplication \*
- d. Exponential \*\*
- e. Division /
- f. Modulus %
- g. Increment ++
- h. Decrement --

### Assignment Operators

- a. =
- b. +=
- c. -=
- d. \*=
- e. /=
- f. %=
- g. \*\*=

### Google Chrome Console CTRL + I

Every error in JS is shown in console.

Every modification is tried in console then we can modify it in the document

Testing is done in console like:

log	console.log is the popular tag used to use the console
table	console.table helps to print the data in table format
error	Used to print the error in the console
warn	Used to show a warning
clear()	Clear the data in console
time and timeEnd	To show how much time it has taken to run

### Comparisons Operators

Used to compare elements

==	Equal value
===	Equal value and equal type
!=	Not equal value
!==	Not equal value or not equal type
>	Greater than
<	Less than
>=	Greater than equal to
<=	Less than equal to

## Statements in JS

### If Statement

```
Var x = 15;  
if (x == 2) return true;  
else return false;
```

**+** is a concatenation operator in JS

### Conditional (Ternary) operator in JS

```
(Condition) ? True : false;

var x = 15;
var z;

(x > 10) ? z = "true" : z = "false";
or
b = ( x > 10) ? "true" : "false";

Or
b = "value is" + ( x > 10) ? "true" : "false";
```

### Switch Statement in JS

```
switch (expression){

case condition 1: statements
break;

case condition 2: statements
break;

Default : statements

}
```

## Pop Up Boxes in JS

### Alert Box in JS

alert("hello Dev!"); --> it's like a pop up from the browser

We can show message and any variable value in the alert box  
Also concatenation can be done and all other remaining operations in the alert box

### Confirm Box in JS

It is similar to alert box but have confirm option or question .

e.g.:  
var a = confirm( "Do you like my channel!"); --> true or false

```
If ( a ){
  alert("Good");
}
else{
  alert("Bad");
}
```

### Prompt Box In JS

Get Text from the user having an field to write the answer asked in the prompt

e.g. :

```
var a = Prompt ( " what is your name ? ");
alert(a);
```

## Functions in JS --> 10 Dec

It increases the code reusability of the code to reduce the complexity and make the code easy to write and understand

```
Syntax : Declaration of a function
function functionName(){
  // statements
}
Calling of a function : functionName();
```

### Functions with Parameters

```
function sum( a , b){
  return a + b;
}
```

```
// calling the function
sum(10,20);
```

-- if we will not pass any argument in the function then it will return undefined.

Set a default value in a function e.g. :

```
function name( fname="ram",lname="kumar"){
  document.write(fname + lname);
}
```

After calling the function with different argument then the value will be overridden automatically

### Functions with return value

```
function fullName(fname = "Devi Prasad" , lname = "Mishra"){

    var a = fname + lname;
    return a;
}

var name = fullName(); output : Devi Prasad Mishra --by default as in function.
var name = fullName("Pawan","Mishra"); ----> output : Pawan Mishra
```

## Global and Local Variables in JS

Those variables which are declared inside the function is local variable or inside any parenthesis is local variable whereas the global variable is declared inside a class but outside a function which can be accessed anywhere in the program.

## Events in JS

**Things which are done after click a button or having some effects in the text or the buttons. Some basic events are :**

1	Click ( onclick )
2	Double click ( ondblclick )
3	Right click ( oncontextmenu )
4	Mouse Hover ( onmouseover )
5	Mouse out ( onmouseout )
6	Mouse Down ( onmousedown )
7	Mouse up ( onmouseup )
8	Key Press ( onkeypress )
9	Key Up ( onkeyup )
10	Load ( onload )
11	Unload ( onunload )
12	Resize ( onresize )
13	Scroll ( onscroll )

**e.g. : 1. onclick event**

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <button onclick="hello()">Click me</button>
</body>
<script>
  function hello() {
    document.write("hi i am devi");
  }
</script>
</html>
```

- ondblclick event same as up code
- Keypress is used in the body or the form
- On resize and on scroll event is called for the body tag

## LOOPS in JS --> 11 dec

Loops can be a single or nested it depends on the use case of the code

### 1. While loop in JS

e.g. :  

```
var a = 1;
while( a <= 20){
  document.write("hii"+a);
  a= a+1;
}
```

### 2. Do / while loop

```
do {
  a++;
}
While(a<=10);
```

### 3. For loop

```
For( var a = 0 ; a < 10 ; a++) {
  document.write(a);
}
```

### 4. For / in loop ( objects

It is made for the printing of the objects values.

e.g. :  

```
for(var i in a){
  // Statement
}
```

```
<script>
var obj = {
  firstName : "Yashoo",
  lastName : "Babu",
  Age : 25,
  email : "hell@yashoobabu.net"
};

for(var key in obj){
  document.write(obj[key] + "<br>");
}
</script>
```

## 5. For Each loop (Array)

```
var a = ["Sanjay", "Aman", "Karan"];
a.forEach(function(){
// Statement
});
```

### E.G. : 1. PRINT THE VALUES IN THE ARRAY

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript</title>
<script>
var a = ["Rahul", "Karan", "Aman", "Neha"];

a.forEach(function(value){
document.write(value + "<br>");
});
</script>
</head>
</body>
```

Rahul  
Karan  
Aman  
Neha

### 2. PRINT THE VALUES AS WELL AS THE INDEX OF THE ARRAY ALONG WITH IT.

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript</title>
<script>
var a = ["Rahul", "Karan", "Aman", "Neha"];

a.forEach(function(value, index){
document.write(index + " : " + value +
});
</script>
</head>
</body>
```

0 : Rahul  
1 : Karan  
2 : Aman  
3 : Neha

# Continue and Break Statements

Continue and break are used in a loop and break is mostly used in a switch cases.

# Arrays in Java Script

Arrays are the collection of data's of homogenous data types with continues memory allocated and have fixed sized.

e.g. : var a = [10,20,30];  
Each have their own indexes starting from 0 to .....  
Like :  
a[0] --> 10  
a[1] --> 20

## Note : we can create mixed data type array in java script

Like:  
var arr = [ 10 , "Devi" , "Mishra" , false , null ];

### CREATE ARRAY

```
var a = new Array();
var b = new Array(3) --> of size 3
var c = new Array ("devi" , "Pawan");
```

### MULTI-DIMENSIONAL ARRAY

Name	Age	Gender	Class
Harry	18	Male	B.Com
Sunny	19	Male	BCA
Sarah	18	Female	BCA
Tom	17	Male	B.A.

```
e.g. :
var arr = [
["Harry", 18, "Male", "B.Com"],
["Sunny", 19, "Male", "BCA"],
["Sarah", 18, "Female", "BCA"],
["Tom", 17, "Male", "B.A." ]
];
```

```
document.write(a);
```

```
---- print the array ----
For(var a = 0; a < 4; a++)
{
document.write(arr[a]);
}
```

```
---- print the array inside access ----
For(var a = 0; a < 4; a++)
{
For(var b = 0; b < 4; b++)
{
document.write(arr[a][b]);
}
```

```
}
}
```

Show multidimensional array in table

```
document.write("<table border='1px' cellspacing='0'>");
```

```
For(var a = 0; a < 4 ; a++)
{
document.write("<tr>");
```

```
For(var b = 0; b < 4 ; b++)
{
document.write("<td>" + arr[a][b] + "</td>");
}
document.write("</tr>");
}
```

**NOTE : TO GET LENGTH OF THE ARRAY WE HAVE THE LENGTH FUNCTION LIKE: ARR.LENGTH OR ARR[0].LENGTH**

### MODIFY THE ARRAY

```
var arr = [ "Harry" , 18,"Male","B.Com"];
arr[0] = "Sunny"; --> changing the value at index 0
```

### DELETE THE VALUE IN ARRAY

```
var arr = [ "Harry" , 18,"Male","B.Com"];
Delete a[1] --> delete the value of the index 1
```

## METHODS IN ARRAY

sort()	Sort the elements in the array	slice(start, end)	divide the array from start index to end index like: arr.slice( 1,5) <b>We can print from the end using neg value</b>
reverse()	Reverse the elements in array	splice( start index, How many delete, New values,..)	Make changes in the existing array Splice (start index , how many, "new Value") --> insert  Splice (start index , how many, "new Value") --> Delete  <b>We can use neg value</b>  <b>Only delete values from arr</b> <b>Then</b> <b>arr.splice(2,2);</b>
pop()	Delete from last	isArray(variable)	To check the variable is array or not e.g. : var a = [ 1 ,2, 3]; var c =1; var b = Array.isArray(a); var d = Array.isArray(c);  Output --> true , false
push(value)	Insert element in array at last	indexOf( value , start index)	search the index of the value in arr e.g. : var a = [ 1 ,2, 3]; var b = a.indexOf( 1 ); var b = a.indexOf( 1 , 0);  Output --> 0 Output --> 0  If not found returns -1
shift()	Delete from front	lastIndexOf(value , start index)	Find the index of the value starting from end e.g. : var a = [ 1 ,2, 3]; var b = a.lastIndexOf( 1 ); var b = a.lastIndexOf( 1 , -2);  Output --> 3 Output --> 3  If not found returns -1
unshift(value)	insert at front	valueOf()	Return the array exactly same as printed by document . Write method
concat(value)	merge two or more arrays like: var a = [ 1 ,2,3]; var b = [ 4 ,5];	every(functionname)	Check for the value satisfying the condition if all the values satisfy the condition then only it will return true otherwise false.

	<pre>var c = a.concat(b);</pre> <p>Output --&gt; 1, 2, 3, 4, 5</p>		<pre>&lt;script&gt; var ages = [10,13,15,2]; document.write(ages + "&lt;br&gt;&lt;br&gt;");  var b = ages.every(checkAdult); document.write(b + "&lt;br&gt;&lt;br&gt;");  function checkAdult(age){   return age &gt;= 18; } &lt;/script&gt;</pre> <p>Output --&gt; false</p>
join(value)	<pre>var a = [ 1,2,3]; var b = [ 4,5]; var d = c.join( " "); d is the string now.</pre> <p>Output --&gt; 1 2 3 4 5</p>	filter(functionName)	<p>It will creates an array filled with all array elements that pass a condition required to pass.</p> <pre>&lt;script&gt; var ages = [10,12,19,20]; document.write(ages + "&lt;br&gt;&lt;br&gt;");  var b = ages.filter(checkAdult); document.write(b + "&lt;br&gt;&lt;br&gt;");  function checkAdult(age){   return age &gt;= 18; } &lt;/script&gt;</pre> <p>Output --&gt; 19,20</p>
find(functionname)	<p>It will return the value of the first element in the array that pass a test And then it will not check further more.</p> <pre>&lt;script&gt; var ages = [10,23,19,20]; document.write(ages + "&lt;br&gt;&lt;br&gt;");  var b = ages.find(checkAdult); document.write(b + "&lt;br&gt;&lt;br&gt;");  function checkAdult(age){   return age &gt;= 18; } &lt;/script&gt;</pre> <p>Output --&gt; 23</p>	findIndex( functionName )	<p>It will return the index of the first element in the array that pass a test And then it will not check further more.</p> <pre>&lt;script&gt; var ages = [10,23,19,20]; document.write(ages + "&lt;br&gt;&lt;br&gt;");  var b = ages.findIndex(checkAdult); document.write(b + "&lt;br&gt;&lt;br&gt;");  function checkAdult(age){   return age &gt;= 18; } &lt;/script&gt;</pre> <p>Output --&gt; 1</p>
includes(value)	<p>Find something present or not in arr. e.g. : var arr = [ "Harry" , 18,"Male","B.Com"]; var b = arr.includes(18);</p> <p>output --&gt;true</p> <p><b>Return value in Boolean form and also it is case sensitive. So, value should be exactly same.</b></p>	some(functionname)	<p>Check for the value satisfying the condition if one of the values satisfy the condition then it will return true otherwise false.</p> <p>Like check for adult :</p> <pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt; &lt;title&gt;JavaScript&lt;/title&gt; &lt;script&gt; var ages = [10,13,15,2]; document.write(ages + "&lt;br&gt;&lt;br&gt;");  var b = ages.some(checkAdult); document.write(b + "&lt;br&gt;&lt;br&gt;");  function checkAdult(age){   return age &gt;= 18; } &lt;/script&gt; &lt;/head&gt; &lt;body&gt;</pre> <p>Output --&gt; true</p> <p>If any values is matching the situation as expected then it will throw true as output.</p>
fill(value)	<p>it fills all the elements in an array with a static value</p> <pre>&lt;script&gt; var a = ["Rahul","Karan","Aman","Neha"]; a.fill("Ram"); document.write(a); &lt;/script&gt;</pre> <p>Output --&gt; Ram,Ram,Ram,Ram</p>	toString()	<p>Convert the array into a string and return the result</p> <pre>&lt;script&gt; var a = ["Rahul","Karan","Aman","Neha"]; a.toString();  document.write(a); &lt;/script&gt;</pre> <p>Output : Rahul,Karan,Aman,Neha</p>
	<p><b>READ ALL THIS METHODS CLEARLY AND UNDERSTAND TO WRITE THE CODE PERFECTLY WHILE MANIPULATING THE ARRAY</b></p>		

## Objects in Java Script --> 12 Dec

Objects are the blueprint of the class

- Method to create object  
e.g. :

```
Var a = {
  firstName : 'Devi',
  middleName : 'Prasad',
  lastName : 'Mishra',
  Age: 22,
  Country : 'Nepal',
}
```

→ Properties

```

Country : 'Nepal',
favMovies : ['Dhoom', 'sholay', 'Hum'],
Salary : function() {
  return 2000;
},

Fullname : function() {
  Return this.firstName + this.LastName; -----> this keyword is used to access the variable present in the same class.
};

console.log(a);
console.log(a.firstName);
document.write(a.firstName);
document.write(a.Salary());

```

Output in console:



```

{ffname: "Yahoo", lname: "Baba", age: 25, email: "hello@yahooaba.net", favMovies: Array(3)}
age: 25
email: "hello@yahooaba.net"
favMovies: Array(3)
  0: "Dhoom"
  1: "Sholay"
  2: "Hum"
  length: 3
  __proto__: Array(0)
ffname: "Yahoo"
lname: "Baba"

```

## 2. Method to create objects

```

var a = new Object();

a.firstName : 'devi';
a.lastName : 'Mishra';
a.age : 15;

document.write(a['age']); --> output :15

```

## ARRAYS OF OBJECTS

```

Var a = [
  { name : 'Devi', age : 18 },
  { name : 'Pawan', age : 14 },
  { name : 'Ram', age : 13 },
];

console.log(a);

For ( var i =0; i<a.length;i++){
  document.write(a[i].name + a[i].age + "<br>");
}

```

Output : Devi 18  
Pawan 14  
Ram 13

## const variable with array and objects

### WITH ARRAY

```

const a = [10,20,30];
console.log(a);

```

Change 1 index value  
a[1]=25; --> no error occurs

a = [40,50,60]; --> throws an error that constant values cannot be re assigned.

### WITH OBJECTS

```

const b = {
  Name : 'Ram',
  Age : 25,
};

```

b.name = 'Devi'; --> changes the value of the variable name

## Array Methods Map() --> 13 Dec

```

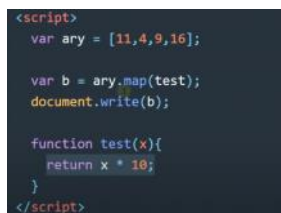
Var a = [ 1 ,2 ,3, 4, 5];

```

```

a.map(function(){
  Statement
});

```



```

<script>
var ary = [11,4,9,16];

var b = ary.map(test);
document.write(b);

function test(x){
  return x * 10;
}
</script>

```



Output : 110 ,40,90,160

### USING ARRAY OF OBJECTS:

Print the names in the arrays of object.

```
<script>
var ary = [
  {fname : "Yahoo" , lname : "Baba"},
  {fname : "Rahul" , lname : "Kumar"},
  {fname : "Karan" , lname : "Sharma"},
];

var b = ary.map(test);
document.write(b);

function test(x){
  return x.fname + " " + x.lname;
}
</script>
```

Output : Yahoo Baba

## Methods in String

THESE ARE THE METHODS HELPS TO MANIPULATE THE STRING AS PER THE REQUIREMENT OF THE PROGRAMMERS.

<b>Length (property)</b>	Find the length of the string	<b>match(value)</b>	Find the word using regex And return it as array.	<b>fromCharCode(ascii value)</b>	It will return the char from The ascii code
<b>toLowerCase()</b>	Converts the string to lowercase	<b>indexOf(value)</b>	Find the index of the Character or word	<b>concat( string)</b>	merge two strings together
<b>toUpperCase()</b>	Converts the string to uppercase	<b>lastIndexOf(value)</b>	Find the index of the char Or word from end of string.	<b>split( value)</b>	Converts an string into char Array.
<b>includes (value)</b>	Finds the matched word or char And returns value in Boolean	<b>replace( value , Replacing Value)</b>	Replace the first word found from the String matching the condition. If regex is used replaces all the Present values.	<b>repeat( times )</b>	Repeats the string at constant Time as given will give output of the String at that time
<b>startsWith (value)</b>	Finds the character starts with is Present or not.	<b>trim()</b>	It removes the extra spaces from The left or right of the string	<b>slice( start , end )</b>	Returns the middle characters present In the start and end points given to them Ans -1 will be last index
<b>endsWith (value)</b>	Finds the character ends with is Present or not.	<b>charAt(index)</b>	Find the character at the given position	<b>substr( start ) / Substr(start , end)</b>	It will work similar to the slice depending On the parameters we uses
<b>search(value)</b>	Return the index if char is present Otherwise -1.	<b>charCodeAt()</b>	Return the ascii code of the character	<b>toString()</b>	Converts the integer value to the string Variable.
<b>Substring(start , end)</b>	It will works similar to slice But won't count the end and Only runs up to end-1;	<b>valueOf()</b>	Return the string as it is		

EXAMPLES IS IN THE FOLDER LEARN JS FILE METHODS OF THE STRING.. FILE NO 1.

## Number Methods

IT CONVERTS THE STRING OR BOOLEAN VALUES TO THE INTEGER VALUE.

<b>Number(value)</b>	It converts the string in integer format ("99") or the Boolean values to number format.
<b>parseInt(value)</b>	It converts decimal values to the int values like 99.9 to 99 and return the integer value present in the starting otherwise NaN
<b>parseFloat(value)</b>	It converts the int value to the decimal value like 22 to 22.00 and Also return the first starting integer value from string and also If 10 20 is in string then it will return the 10 only.
<b>isFinite(value)</b>	Return the Boolean value to check number is countable or not
<b>isInteger(value)</b>	Check the given variable is integer or not
<b>toFixed(value)</b>	Fixed the decimal points of the number like 5.678656 can be up to 2 decimal points like 5.67
<b>toPrecision(value)</b>	return the number of digits we want

EXAMPLES ARE IN THE 2.NUMBER\_METHOD FILE

## Math Methods in JS -->14Dec

These are the methods used to perform the mathematical calculations.

<b>ceil(x)</b>	return the upper value of the nearest float	<b>sqrt(x)</b>	return the square root
<b>floor(x)</b>	return the downward value of the nearest float	<b>cbt(x)</b>	Return the cube root
<b>round(x)</b>	round off to the nearest value in integer form	<b>pow(x,y)</b>	Return the power of number
<b>trunc(x)</b>	it return the integer value only	<b>random()</b>	Generates a random number in given range
<b>max(x,y,z,...,n)</b>	it will return the max value among all	<b>abs(x)</b>	Returns the absolute value -ve to +ve
<b>min(x,y,z,...,n)</b>	It will return min value among all	<b>PI</b>	Pie function having value 3.14

Examples in 3 Maths\_calculation\_Method file.

## Date Methods In JS

It's a date manipulation methods.

Var now = new Date(); --> creating date objects.

We can use now object to perform all the operations. And also use the date methods as per our requirements like:

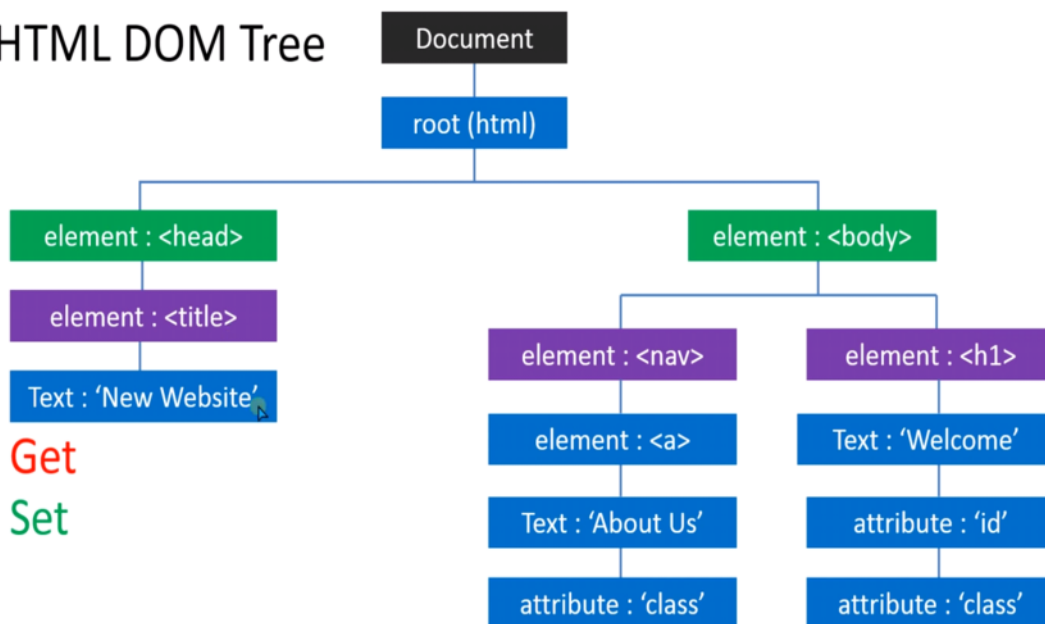
<b>toString()</b>	return the date with day	<b>getMilliseconds()</b>	Return the millisecond
<b>getDate()</b>	Return the date means day	<b>setDate()</b>	Set the date as per our need
<b>getFullYear()</b>	Returns the year	<b>setFullYear()</b>	Set the year
<b>getMonth()</b>	Return the month	<b>setHours()</b>	Set the hour
<b>getDay()</b>	Return the day sun, Mon	<b>setMilliseconds</b>	Set the millisecond
<b>getHours()</b>	Return the present hour	<b>setMinutes()</b>	Set the minutes
<b>getMinutes()</b>	Return the present minutes	<b>setMonth()</b>	Set the month
<b>getSeconds()</b>	Return the present seconds	<b>setSeconds()</b>	Set the seconds

EXAMPLE IN 4. DATE METHODS FILE

# Document Object Model ( DOM )in JS

(DOM) is the data representation of the objects that comprise the structure and content of a document on the web. It is a programming interface that allows us to create, change, or remove elements from a website document. DOM manipulation is when you use JavaScript to add, remove, and modify elements of a website.

## HTML DOM Tree



Get  
Set

## 1. DOM TARGET METHODS

### 1) Id

We can target id by using -----> document.getElementById(id)

### 2) Class Name

We can target class by using ---> document.getElementsByClassName(name)

### 3) Tag Name

We can target tag Name by using ---> document.getElementsByTagName(name)

## Other targeted methods In Dom we uses:

<b>document</b>	It will treat the html file as document And shows the html file in console in document.	<b>document.links</b>	It show the number of anchor tag Present in the html file or the document
<b>document.all</b>	It return all the html tags as an array In the console.	<b>document.froms</b>	It shows the number of forms present In the document
<b>document.documentElement</b>		<b>document.doctype</b>	It returns the document type
<b>document.head</b>	It shows the head tag with all its contains	<b>document.URL</b>	It returns the website URL
<b>document.title</b>	It show all the things in title tag	<b>document.baseURI</b>	It will return all URL of the website
<b>document.body</b>	It shows all the things present in the Body tag	<b>document.domain</b>	It returns the domain name or Ip
<b>document.images</b>	It shows the number of images present in The html body and stores as array.	<b>document.anchors</b>	Return the number of anchor tags

## 2. GET AND SET METHOD IN DOM : ---> 17 DEC

We can get text , html and attribute of the method for that we have some methods like:

1. innerText
2. innerHTML
3. getAttribute
4. getAttributeNode
5. Attributes

We can set the values in the attribute using some methods like:

1. innerText
2. innerHTML
3. setAttribute
4. Attribute
5. removeAttribute

### **Examples in file 5. Document object model**

## 3. QUERY SELECTORS IN DOM :

1. querySelector
2. querySelectorAll

### **Examples in file 5.1 Dom query selector**

## 4. DOM CSS STYLING :

1. Styles
2. className
3. classList

It have its own method to do manipulations with classes:

<b>add(value)</b>	Add a class to a tag
<b>remove(value)</b>	Remove a class to a tag
<b>Toggle(class)</b>	Add the class o 1st click Remove it in another click
<b>Contains(class)</b>	Shows the class is present or not
<b>Item(index)</b>	Check the item present in the index number
<b>Length</b>	Finds the number of classes in a html tag

### **Examples in file 5.2 Dom CSS Styling**

## 5. DOM ADD EVENT LISTENER

As learned before we know the method to add events in html tags now we will be using it.

### 1. addEventListener( event , function );

addEventListener( event , function , useCapture );  
useCapture has 2 values true or false

e.g. : if we have 2 div outer and inner and we have added event listener to both the Div's then  
If we call the inner div listener then automatically outer will be called so to avoid it we can use the  
UseCapture field in the addEventListener like: [in example 5.3.2](#)

### 2. removeEventListener(event , functionname);

if we want to remove some event without touching the count then we Can use this method  
to remove the event in that tag. [In example 5.3.0](#)

## 6. DOM TRAVERSAL METHODS ---> 24 DEC

Methods to access the nodes i.e. it will help to target the node

parentNode	it means the root node	lastChild	returns the last child <b>node</b> (an element node, a text node or a comment node). Whitespace between elements are also text nodes.
ParentElement	it means the root element if Not present it returns null	lastElementChild	returns the last child element of an element and it has a read only property
Children	Return the child of the tag Like in tree.	nextElementSibling	returns the next element in the same tree level and has read only property.
childNodes	Return the name with no. of nodes present In the tag as an array situated With array index.	Nextsibling	returns the next node on the same tree level. The nextSibling returns a node object. The nextSibling property is read-only.
firstChild	Return the HTML content of the first child node and return null if No child is present	previousElementSibling	returns the previous element in the same tree level and has read only property.
firstElementChild	returns the first child <b>element</b> (ignores text and comment nodes).	previousSibling	The previousSibling property returns the previous node on the same tree level. The previousSibling property returns a node object. The previousSibling property is read-only.

**Examples in file 5.4.0 Dom Traversal Method.**

## 7. DOM CREATE METHODS ---> 25 DEC

Basically we have 3 methods to create Dom in JS they are:

createElement	add html element
createTextNode	add a text in html document
createComment	To write comment in html file.

**Examples in file 5.5.1 Dom create Method.**

## 8. DOM APPEND METHODS --> 26 DEC & 27 DEC

There are 2 methods to append elements in the html page after  
Creating it. They are:

appendChild(value)	Attach element to the existing element In the html file. It will always add the element at last.
insertBefore(value, target)	Attach element to the existing element In the html file. It will insert element at any position in html.
insertAdjacentElement (position, element )	method inserts a an element into a specified position. We can insert it into 4 position: 1. beforebegin 2. afterbegin 3. beforeend 4. afterend
insertAdjacentHtml ( position , element )	Inserts an HTML formatted text at a position relative to an element We can insert it into 4 position: 1. beforebegin 2. afterbegin 3. beforeend 4. Afterend
insertAdjacentText (position , element )	Inserts an text at a position relative to an element We can insert it into 4 position: 1. beforebegin 2. afterbegin 3. beforeend 4. Afterend
replaceChild ( new , old )	Replace the element in the html
removeChild ( children )	Remove the child from the html
cloneNode (Boolean)	Copy the element

**Examples in file 5.5.2.0 & 5.5.2.1 Dom Append , Remove , Replace Method.**

