

JavaScript

JavaScript is a Programming language for the Web.

JavaScript **getElementById("Name of the ID")** :

getElementById("id name") is used to Search for the id name in HTML.

Example: Create a Button Based JS using getElementById():

```
<html>
<body>
<h1>This is the Heading</h1>
<p id = "demo">Js getElementById("demo") usecase.</p>
<button type="button" onclick='document.getElementById("demo").innerHTML = "H3110 J5 !"'>Click Me</button>
</body>
</html>
```

JavaScript **Scripting <script></script>** :

Example: Using Scripting to Print Some thing :

```
<html>
<body>
<h1>This is the Heading</h1>
<p id = "demo">Replaceing text</p>
<script>
document.getElementById("demo").innerHTML="JS Text is Replaced"
</script>
</body>
</html>
```

JavaScript **function function_name(param)** :

Example : Using JS Function for Button :

```
<html>
<head>
<script>
function myFunction(){
  document.getElementById("demo").innerHTML = "Replaced by Function"
}
</script>
</head>
<body>
<h1>This is the Header</h1>
<p id = "demo">Replace this with the Function</p>
<button type="button" onclick="myFunction()">Replace</button>
</body>
</html>
```

Using External Source File :

```
<script src="myScript.js"></script>
```

javascript

Placing Script file outside the main file has some advantages :

1. Seperate Code
2. Easy to Read and Maintain
3. Speed Up the Load time.

External Source: Using Website to import a Script with full file path.

```
<script src="https://mywebsite/js/myScript.js"></script>
```

javascript

Displaying Data in JavaScript

1. Writing into an HTML element using innerHTML
2. Writing into the HTML output using document.write()
3. Writing into an alert Box using window.alert()
4. Writing into the Browser console , using console.log()

Using **document.write()** ("Text to be Printed on the Screen") :

Example : Using the document.write() to directing print in Output:

```
<html>
<body>
<h1>This is the Heading</h1>
<p>This example is not any demo id bytheway</p>
<script>
document.write("Sujal")
</script>
</body>
</html>
```

javascript

Using document.write() after an HTML document is loaded will **delete all the Existing HTML**. Should be Used for only Testing.

```
<html>
<body>
<h1>This is the Heading</h1>
<p id="demo">This is why you should use document.write()</p>
<button type="button" onclick="myfunc()">try</button>
<script>
function myfunc(){
document.write("Sujal")
}
</script>
```

javascript

```
</body>
```

Using **window.alert()** :

```
<html>
<body>
<h1>This is the Heading</h1>
<p>Using window.alert()</p>
<script>
window.alert(1+2)
</script>
</body>
</html>
```

javascript

Using window in window.alert() is not mandatory since windows is a Global object in the JS library

Using the **windows.print()** to print the entire page :

```
<html>
<body>
<h1>This is the heading</h1>
<p>Using the window.print() to print the page.</p>
<button type="button" onclick="Function()">Try</button>
<script>
function Function(){
window.print()
}
</script>
</body>
</html>
```

javascript

In JavaScript **Programming Instructions** are called **Statements**.

Assigning variables :

```
let a,b,c;
a = 1;
b = 2;
c = a+b;
let person = "Sujal";
```

javascript

Keyword	Description
var	Declare a Variable
let	Declare a Block Variable
const	Declares a Block constant
if	Condition Execution
function	Declaring a Function
for	Declaring a Conditional Loop

try	Implementing error handling to a block of statment.
-----	---

Comments :

```
// for Single line Comment
/* Is used to create a Multi-Line Comment */
```

javascript

Always use const for declaring values that doesnt Change, Use let for Changeable Values.

JavaScript **let** :

JS let have a Block Scope , Because of which it must be declared before use and can cannot be Redeclared.

JavaScript **var** :

Javascript var have a global Scope , Because of which you can redeclare the var variable.

JavaScript **typeof** :

To Return the Type of data of the variable supplied.

JavaScript Functions() :

```
<html>
<body>
<h1>Website</h1>
<button type="button" onclick="myfunc(1,2,3)">try</button>
<script>
function myfunc(a,b,c){
    window.alert(a*b+c)
}
</script>
</body>
</html>
```

javascript

Accessing a function with () results in returning the value of the return while accessing a function without () results in returning the entire function LOL ;0

JavaScript Objects :

```
const car = {type:"Fiat", model:"500", color:"white"};
let x = car.model
```

javascript

To Access the Object Properties. [objectName.propertyName](#)

Methods : Methods are actions that can be performed on object.methods are stored in properties as function definitions.

```
const person = {
    firstname = "John",
    lastname = "Doe",
    id = 5566,
    fullname = function(){
        return this.firstname + " " + this.lastname;
    }
}
```

javascript

```
}  
}
```

this is used to refer the firstname and lastname of the Object .

Common HTML Events :

Events	Descriptions
onchange	HTML element has changed
onclick	The Users Click an HTML elements
onmouseover	The user moves the mouse over the HTML Element.
onmouseout	The user moves the mouse away from an HTML element.
onkeydown	The user pushes a keyboard key.
onload	The Browser has finished loading the page.

Example : Using onmouseover and onmouseout Events :

```
<html>  
<body>  
<h1>Trend</h1>  
<button onmouseover="visible()" onmouseout="invisible()">Cat</button>  
  
<script>  
function visible(){  
    document.getElementById("demo").innerHTML = "Chipi Chipi Chapa Chapa Dubi Dubi Daba Daba";  
}  
function invisible(){  
    document.getElementById("demo").innerHTML = "Magicomi Dubi Dubi Boom Boom Boom ;0";  
}  
</script>  
<p id ="demo"></p>  
</body>  
</html>
```

JavaScript String :

To find the length of the string **length** can be use .

```
let text = 'sujal machhale';  
let length= text.length;
```

JavaScript **Escape Characters** :

In Javascript \ is the Escape Character .

```
let car = "car names \"BMW\" is Awesome"
```

\b	backspace
----	-----------

\f	form feed
\n	New Line
\r	Carriage Return
\t	Horizontal tabulator
\v	Vertical tabulator

JavaScript String Methods :

To Return a Character at Specific Index we can use **charAt()** , **at()** , **[]** .

```
let text = "ABCDEFGHJKLMNOPQRSTUVWXYZ";
let len = text.length; //return 26
let index1 = text.charAt(0); //return A
let index2 = text.at(1); //return B
let index3 = text[3]; //return D
```

javascript

To Extract certian Part of the String we can use :

slice(start,end) , **substring(start, end)** , **substr(start, length)**

```
let string = "apple, bannana, kiwi";
let part1 = string.slice(7,13); //return bannana
let part2 = string.substring(7,13); //return bannana
let part3 = string.substr(7,6); //return bannana
```

javascript

JavaScript String to Upper and Lower Case :

To Convert a string to upper case we use **toUpperCase()** and for Lower Case **toLowerCase()** .

```
let x = "Hello World!";
let upper = x.toUpperCase(); //HELLO WORLD
let lower = x.toLowerCase(); //hello world
```

javascript

JavaScript join() :

join() method is used to join arrays in js with some seperator.

```
const fruit = ["bannana","orange","apple"]
const joinArray = fruit.join("*")
```

javascript

JavaScript push and pop :

push() is used to add element to array at the end . **pop()** is used to remove array element from the end .

```
fruits = ["bannana","Apple"]
fruit.pop()
fruits.push("Orange")
// [bannana , orange]
```

javascript

JavaScript shift() & unshift():

shift() is used to remove the first element of the array and re-set the index of the array .

unshift() is used to add element to the start of the array and re-set the index of the array .

```
const fruits = ["bannana","orange"]
fruit.shift()
fruit.unshift("apple")
//[apple,orange]
```

javascript

JavaScript Concatinating Arrays :

```
const array1 = ["one" , "two"];
const array2 = ["three","Four"];
const array3 = ["five" , "six"];
const merge_array = array1.concat(array2,array3);
```

javascript

JavaScript splice() :

splice() method is used to add new items to an array at a specific location.

```
const fruits = ["Bannana","Orange","Apple,Mango"];
fruit.splice(2,0,"lemon","kiwi")
```

javascript

(2) parameter is used to start adding the element from the specified index.

(0) parameter is used to tell how many elements to remove from the array.

JavaScript slice() :

slice() method is used to peice out a peice of an array into a new array .

```
const fruit = ["bannana","orange","Lemon"];
const array = fruit.slice(1,3);
//orange , lemon
```

javascript

JavaScript indexOf() :

JavaScript indexOf() is used to search for an element in an array . it finds and return the index of the array.

```
const fruits = ["apple","orange","mango"];
let position = fruits.indexOf("apple");
```

javascript

JavaScript find() in array :

find() in array search for first element of the array .

```
<html>
<body>
<p id = "demo"></p>
<script>
```

javascript

```
const number = [1,3,19,10,24,25];
let greatest = number.find(myfunc);
document.getElementById("demo").innerHTML = greatest
function myfunc(value){
  return value > 10 }
</script>
</body>
</html>
```

JavaScript Sort() and Reverse():

sort() and **reverse()** is used to sort an array alphabetically and reverse of alphabetic order. it creates an new array .

While toSorted() sorts in array without the new to create a new array .

```
const fruits = ["bannana","orange","Apple"]
fruits.sort()
fruits.reverse()
```

javascript