JavaScript is a very powerful **client-side scripting language** by Brendan Eich.

data types: numbers, strings, objects and more

lively and interactive

Variable:

var two = 3;

var add = one + two;

- x ** y produces the same result as Math.pow(x,y)

Display:-

- Writing into an HTML element, using **innerHTML**.
- Writing into the HTML output using **document.write()**.
- Writing into an alert box, using **window.alert()**.
- Writing into the browser console, using **console.log()**.

Javascript in css

```
document.getElementById("demo").style.fontSize = "35px";
```

What is an Array?

An array is an object that can store a **collection of items**

```
var students = ["John", "Ann", "Kevin"];
```

You can also create an array using Array constructor like this:

```
var students = new Array("John", "Ann", "Kevin");
```

<script>

var vr1 = ["hello","how","are"];

document.write(vr1);

</script>

Reverse:

<script>

var vr1 = ["hello","how","are"];

document.write(vr1);

document.write("
");

var new_arr = vr1.reverse();

document.write(new_arr);

</script>

Or for reverse sort

var vr1 = ["hello","how","are"];

vr1.sort();

```
vr1.reverse();
document.write(vr1);
```

Push:add last element

```
var vr1 = ["hello","how","are"];
vr1.push("you");
document.write(vr1);
```

Pop:-Remove last element

```
var vr1 = ["hello","how","are"];
vr1.pop(vr1);
document.write(vr1);
```

unShift:-Add first element

```
var vr1 = ["hello","how","are"];
vr1.unshift("first");
document.write(vr1);
```

shift:-remove first elementr

```
var vr1 = ["hello","how","are"];
vr1.shift();
document.write(vr1)
```

Length:

```
var vr1 = ["hello","how","are"];
document.write(vr1.length);
```

Sort:-

```
var vr1 = ["hello","how","are"];
vr1.sort();
document.write(vr1);
```

Slicing an Array

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

This example slices out a part of an array starting from array element 1 ("Orange"):

Example

```
var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];
var citrus = fruits.slice(1);
```

Concat two array:

```
var vr1 = ["hello","how","are"];
```

```
var vr2 = ["hello","how","are"];
var arra = vr1.concat(vr2);
document.write(arra);
```

Max or min number:

```
<div id="demo">
        <p>here is code</p>
    </div>
<script>
var arr1=[5,7,100,167];
document.getElementById("demo").innerHTML=arrf(arr1)
function arrf(arr)
{
return Math.max.apply(null,arr);  //here min for minimum
</script>
```

For loop:-

```
var arr=new Array("1","2","3");
for(i=0;i<arr.length;i++)
{
document.write(arr[i]+"<br>");
}
```

While:

```
var arr=["1","2","3"];
var i=3;
while(i<1);
{
document.write(arr);
}
</script>
```

Do_while:

The do…while loop is very similar to while loop. The only difference is that in do…while loop, the block of code gets executed once even before checking the condition.

```
var i=2;
```

```
        do
        {
            document.write(i);
            i=i+2;
        }while(i<20)
    </script>
```

Conditional statements:-

1. If statement

```
var age=17;
if(age<18);
console.log('age is under 18');
```

2. If…Else statement

```
var age = 17;

if(age < 18)

{

document.write('age is under 18');

}

else
{

document.write('age is above');

}
```

3. If…Else If…Else statement

```
var age = 3;

if(age < 18)

{

document.write('age is under 18');

}

else if(age<7)
{

document.write('age is above 50');
```

```
}
else
{
document.write('old');
```

Function:

```
function myfun()
{
document.write("she is here");
}
myfun()
</script>
```

JavaScript Return Value

You can also create JS functions that return values. Inside the function, you need to use the keyword **return** followed by the value to be returned.

```
<script>
    var S=sum(8,22);
    document.write(S);
    function sum(first,second)
     {
       return first + second;
     }
  </script>
```

After return you write anything it will not afffect....

OBJECT:-

```
var person = {
first:"Nilofar",
last:"shaikh",
}
document.write(person.first);
//function as an object
var person = {
first:"Nilofar",
last:"shaikh",
fullname:function(){
return person.first+' '+person.last;
```

```
}
};
document.write(person.fullname());
```

//number string boolean as an object

```
var person ="hello";
alert(person.toUpperCase());
```

****Events:-interactive page:-**

Onclick:

```
<body>
<input type="button" onclick="fun()" value="click">
</body>


<script>
function fun(){
alert("go there");
}
fun();
</script>
Ondblclick:
onmouseover
```

getElementById: To access elements and attributes whose id is set.

innerHTML: To access the content of an element.

```
////javascript by selectores//
Id:-
<body>
<p id="name"></p>
</body>


<script>
document.getElementById('name').innerHTML="here is your code";
```

```
</script>
```

Class:-

```
<body>
<p class="name"></p>
<p class="name"></p>

</body>
<script>
document.getElementsByClassName('name')[0].innerHTML="here is your code";
</script>
```

2]

```
<body>
<p class="name"></p>
<p class="name"></p>
</body>


<script>
document.querySelectorAll("p")[1].innerHTML="here is your code";


</script>
```

3]

```
body>
<div>
<h1></h1>
</div>
</body>


<script>
document.querySelector("div > h1").innerHTML="here is your code";
```

```
</script>

4]<body>
<div>
<h1 class="new"></h1>
<h2 class="new"></h2>
</div>
</body>


<script>
document.querySelector("h2.new").innerHTML="here is your code";


</script>
CHANGING CSS WITH JS:
<p id="p2">Hello World!</p>


<script>
document.getElementById("p2").style.backgroundColor = "blue";
</script>
2]
<p id="p2">Hello World!</p>


<script>
var tet=document.getElementById("p2");
tet.style.backgroundColor="red";
tet.style.borderColor="black";
tet.style.border="solid";
</script>
AddEventListner:-
```

```
<input type="button" id="btn" value="click">
<script>
document.getElementById("btn").addEventListener("click",function(){
alert("this is addeventlistner");
})
</script>
2]<input type="text" id="greens">
<script>
function green(){
document.getElementById("greens").style.backgroundColor="red";
}
document.getElementById("green").addEventListener("focus",green())
</script>
```

FORM VALIDATION:-

-YOU CAN CHECK FORM BROUSER ITSELF

```
function validate(){
var name=document.getElementById("name").value;
var email=document.getElementById("email").value;
var message=document.getElementById("message").value;
if(name=="")
{
alert("name should filled");
return false;


}
else if(email==""){
alert("email should filled");
return false;
}
else if(message=="")
```

```
{
alert("message should filled");
return false;
}
return false;
}
```

Practice.html

```
<!DOCTYPE html>
<html>
<head>
<script src="new.js"></script>
<title>
My website
</title>
</head>


<body>
<form method="post" onsubmit="return validate();">
<input type="text" id="name" name="name"><br><br>
<input type="text" id="email" name="email"><br><br>
<textarea id="message" name="message"></textarea><br>
<input type="submit" id="submit" value="submit" name="submit">
</form>
</body>


</html>
```

Cookies in JavaScript: Set, Get & Delete Example:

A cookie is a piece of data that is stored on your computer to be accessed by your browser.

Browser session expire cookies delete automatically
Session cookies:-no lifetime...for specific session
Persistent cookies:-time set...

JavaScript **escape()** function to encode the value before storing it in the cookie. If you do this, you will also have to use the corresponding **unescape()** function when you read the cookie value.

```
function setcookie(){

var name=document.getElementById("name").value;

var cookievalue= escape(name);

document.cookie="name=" + cookievalue;

document.write("cookies" + cookievalue);

document.cookie = "cookiename=cookievalue; expires= Thu, 3 Jan 2019 20:00:00
UTC; path=/ "

}


<body>

<form action="practice.html" method="post" onsubmit="return setcookie();">

<input type="text" id="name" name="name"><br><br>

<input type="submit" id="submit" value="submit" name="submit">

</form>

</body>
```