# JavaScript

What is **JS (Java Script):** is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

## JavaScript Where To?

In HTML, JavaScript code is inserted between **<script> and </script>** tags. You can place any number of scripts in an HTML document. Scripts can be placed in the `<body>`, or in the `<head>` section of an HTML page, or in both.

## External JavaScript

Scripts can also be placed in external files:

**<script src="myScript.js"></script>** you can place an external script reference in <head> or <body> as you like.

The script will behave as if it was located exactly where the <script> tag is located.

When you write your JavaScript in a separate file **script tag is not required** in the js file. Save the file with **.js extension**

## JavaScript Output

JavaScript can "display" data in different ways:

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

### 1. Using innerHTML

To access an HTML element, JavaScript can use the document.getElementById(id) method.

The id attribute defines the HTML element. The innerHTML property defines the HTML content:

**Example**

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My First Paragraph</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```

## 2. Using document.write()

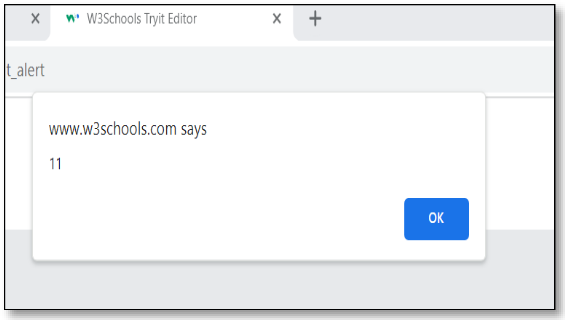
For testing purposes, it is convenient to use document.write() document.write only works while the page is loading; If you call it after the page is done loading, **it will overwrite the** whole page.

**Example**

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<button type="button" onclick="document.getElementById(">Try it</button>

</body>
</html>
```

## 3. Using window.alert()

You can use an alert box to display data:



www.w3schools.com says

11

**Example**

```
<!DOCTYPE html>
<html>
<body>
<h2>My First Web Page</h2>
<p>My first paragraph.</p>
<script>
window.alert(5 + 6); </script> </body> </html>
```

## JavaScript Programs

- A **JavaScript program** is a list of programming **statements**. In HTML, JavaScript programs are executed by the web browser.
- JavaScript statements are composed of: Values, Operators, Expressions, Keywords, and Comments.
- The statements are executed, one by one, in the same order as they are written.
- Semicolons separate JavaScript statements. Add a semicolon at the end of each executable statement.
- JavaScript is a case-sensitive language. This means that language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

## JavaScript Keywords

JavaScript statements often start with a **keyword** to identify the JavaScript action to be performed.

| **Keyword** | **Description** |
|---|---|
| var | Declares a variable |
| const | Declares a block constant |
| if | Marks a block of statements to be executed on a condition |
| switch | Marks a block of statements to be executed in different cases |
| for | Marks a block of statements to be executed in a loop |
| function | Declares a function |
| return | Exits a function |

# Different Examples Show the Effects of Java Script.

## 1. Change the elements content using JS, when the page loaded.

```
*document.getElementById("id_a").innerHTML=newvalue
```

**Example**:

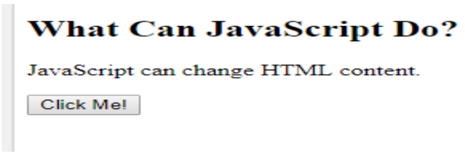
```
<!DOCTYPE html>
<html>
<body>
<h1>My Web Page</h1>

<p id="myP1">This is a p element.</p>

<p id="myP2">This is a p2 element.</p>
<script>
document.getElementById("myP1").innerHTML = "Hello Student.";
document.getElementById("myP2").innerHTML = "How are you?";
</script>

</body></html>
```

## 2. Change the elements content using JS, when the button clicked.

This example uses the **getElementById()** method to "find" an HTML element (with id="demo") and changes the element content (**innerHTML**) to "Hello JavaScript"**(on button click)**

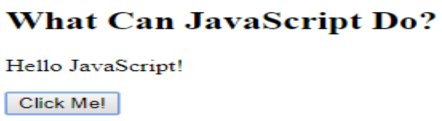
**Example:**

```
<html>
<body>
<h1>What can JavaScript do?</h1>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!"' >Click Me!</button>

</body></html>
```



What Can JavaScript Do?

JavaScript can change HTML content.



What Can JavaScript Do?

Hello JavaScript!

## 3. Change the style of element using JS.

```
*document.getElementById("id_a").style.propertyname=newvalue;
```

1. **Change single style property**: by using style attribute to change an element style.

**document.getElementById("id_a").style.backgroundColor = "yellow";**

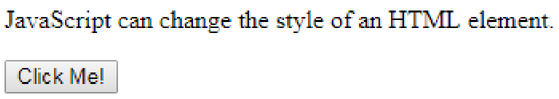
### Example

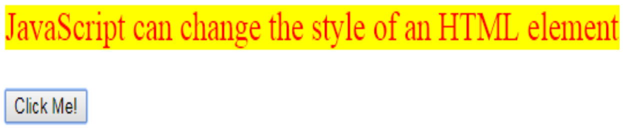
```
<button onclick=" document.getElementById('demo').style.backgroundColor ='yellow';
">
 Click Me!
</button>
```

**2. Change more than style at once:** by using functions, if there is more than one change you should write them in a function in a script which can be written **in body or head**.

### Example

```
<p id="p1">Javascript can change the style of HTML element</p>

<button onclick=" myFunction(); "> Click Me! </button>

<script>
function myFunction() {
document.getElementById("p1").style.fontSize = "25px";
document.getElementById("p1").style.color = "red";
document.getElementById("p1").style.backgroundColor = "yellow";
}</script>
```


JavaScript can change the style of an HTML element.


JavaScript can change the style of an HTML element

## 4. Change HTML elements attributes using JS.

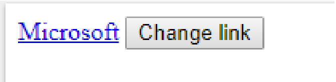
***document.getElementById("id_a").attribute-name=new value**

### Example1. Change img src attribute

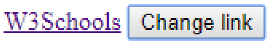
```
<html>
<body>

<img id="myImage" src="off.gif" style="width:100px">

<button onclick="document.getElementById('myImage').src='on.png' " >
Turn on the light </button>

<button onclick="document.getElementById('myImage').src='off.png' " >
Turn off the light</button>

</body> </html>
```

### Example2. (Change the HTML content, URL, and target of a link)

```
<!DOCTYPE html>
<html> <body>

<a id="myAnchor" href="http://www.microsoft.com">Microsoft</a>

<button onclick="myFunction()">Change link</button>

<script>
function myFunction() {
document.getElementById("myAnchor").innerHTML = "W3Schools";
document.getElementById("myAnchor").href =
"https://www.w3schools.com";
document.getElementById("myAnchor").target = "_blank";
}
</script>
</body>
</html>
```



→



# JS Functions

- A JavaScript function is a block of code designed to perform a particular task.
- A JavaScript function is executed when "something" invokes it (calls it).
- A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ().

## Advantage of JavaScript function

There are mainly two advantages of JavaScript functions.

1. **Code reusability**: We can call a function several times so it save coding.
2. **Less coding**: It makes our program compact. We don’t need to write many lines of code each time to perform a common task.

## JavaScript Function Syntax

The syntax of declaring function is given below.

```
function functionName([arg1, arg2, ...argN]){
 //code to be executed
}
```

JavaScript Functions can have 0 or more arguments.

**Example1**:

```
<script>
function msg(){
alert("hello! this is message");
}
</script>
<input type="button" onclick="msg()" value="call function"/>
```

### Example2: Call external functions

```
<html>
<head>
<script type = "text/javascript" src="function.js"></script>
</head>
<body>
<p>Click the following button to see the function in action</p>
<input type = "button" onclick = "myfunction()" value = "Display">
</body>
</html>
```

### In the external js

```
functionmyfunction()
{   document.write("welcome to Javatpoint");
    }
```

### Example3: Function with Return Value

```
<script>
function getInfo(){
return "hello javatpoint! How r u?";
}  </script>
<script>  document.write(getInfo());  </script>
```

### Example4: To set the value of an element such as input field.

```
<html>
<body>
Name: <input type="text" id="myText" value="Mickey">
<p>Click the button to change the value of the text field.</p>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {  document.getElementById("myText").value = "Johnny Bravo";}
</script>
</body></html>
```

### Example5: To get the value of an element such as input field.

```
<script>
function getcube( ){
var number= parseInt(document.getElementById("num").value);
window.alert(number*number*number);
}
</script>
<form>
Insert a number: <input type="number" id="num"/>  <br>
  <input type="button" value="click" onclick="getcube()"/>
</form>
```

### Example6: Declare variables and using for loop in js

```
<script>
  function sumMe(number){
var sum=0;
for(i=1;i<=number;i++)
{sum+=i;
}
window.alert("Sum of all numbers less than"+number+"="+sum);
}
</script>
<form name="myform">
Insert a number: <input type="number" id="num"/>  <br>
<input type="button" value="click"
onclick="sumMe(document.getElementById('num').value)"/>
</form>
```

## Example7: Function Convert to Celsius Example

```
<html>
<body>
<h2>Temperature Converter</h2>
<p>Type a value in the Fahrenheit field to convert the value to Celsius:</p>
<p>
Fahrenheit:<input id="inputFahrenheit" type="number"
oninput="temperatureConverter(this.value)"
onchange="temperatureConverter(this.value)">
</p>
<p>Celcius: <span id="outputCelcius"></span></p>

<script>
function temperatureConverter(valNum) {
  valNum = parseFloat(valNum);
  document.getElementById("outputCelcius").innerHTML=(valNum-32)/1.8;
}
</script>
</body>
</html>
```

## Comment in JS

- Comment multiple lines by using the /* and */ symbols.
- A single line comment can be added by using two forward slashes //.

## Form validation in js

**Example: check the insertion of numbers only.**

```
<script>
function validate(){
var num=document.getElementById("num").value;
if (isNaN(num)){
  document.getElementById("numloc").innerHTML="Enter Numeric value only";
  return false;
}else{
  return true;
  }
}  </script>
<form name="myform" onsubmit="return validate()" >
Number: <input type="text" id="num"><span id="numloc"></span><br/>
<input type="submit" value="submit">
</form>
```

**Note**: onsubmit="return validate()" → you need the return so the true/false gets passed up to the form's submit event (which looks for this and prevents submission if it gets a false).

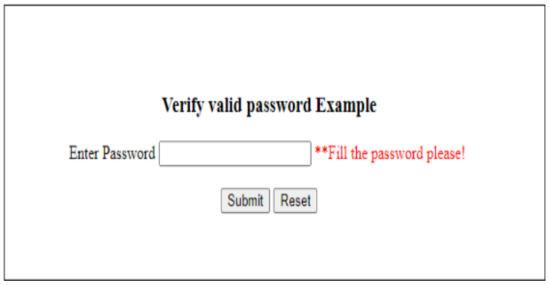
# Homework:

Write a JavaScript code to validate a from password with the next conditions:

- Password length >6 and <12
- Password shouldn't be empty

**Output 1**

Output on leaving the password field blank.



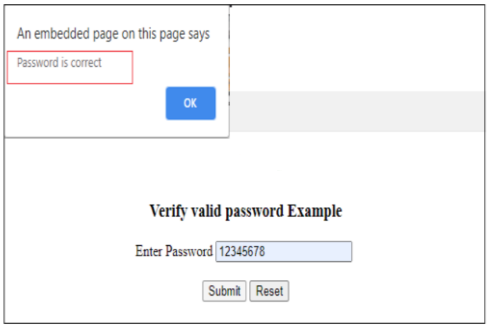
Verify valid password Example

Enter Password

**Fill the password please!

**Output 2**

Output on entering a valid password.



An embedded page on this page says

Password is correct

Verify valid password Example

Enter Password 12345678