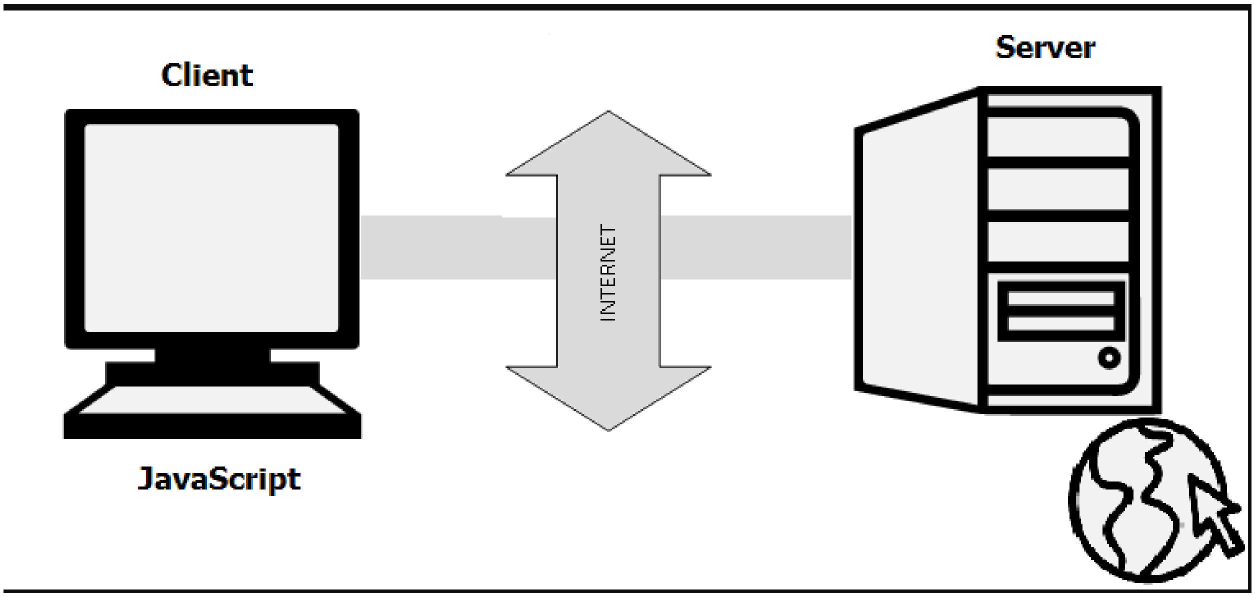
What is JavaScript?

- JavaScript is a very powerful client-side scripting language.
- JavaScript is used mainly for enhancing the interaction of a user with the webpage.
- In other words, you can make your webpage more lively and interactive, with the help of JavaScript.
- JavaScript is also being used widely in game development and Mobile application development.



Javascript History

JavaScript was developed by Brendan Eich in 1995, which appeared in Netscape, a popular browser of that time.

- The language was initially called LiveScript and was later renamed JavaScript.
- There are many programmers who think that JavaScript and Java are the same.
- In fact, **JavaScript and Java are very much unrelated. Java is a very complex programming language whereas JavaScript is only a scripting language**.
- The syntax of JavaScript is mostly influenced by the programming language C

How to Run JavaScript?

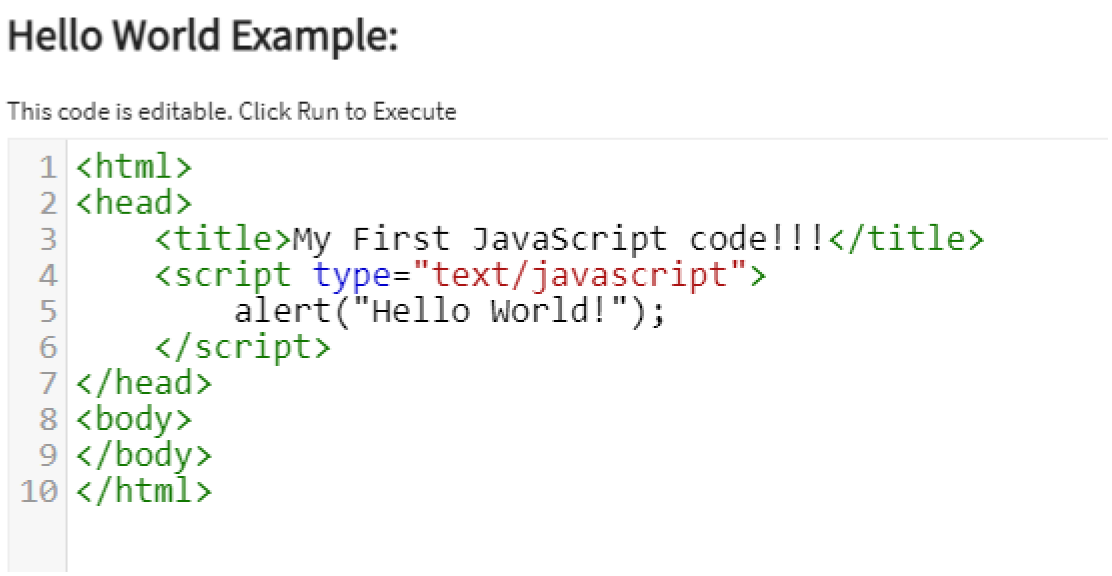
- Being a scripting language, **JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code**.
- When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it.
- The main advantage of JavaScript is that **all modern web browsers support** JavaScript.
- So, you do not have to worry about whether your site visitor uses Internet Explorer, Google Chrome, Firefox or any other browser. JavaScript will be supported.
- Also, JavaScript **runs on any operating system** including Windows, Linux or Mac.

Tools You Need

- To start with, you need a text editor to write your code and a browser to display the web pages you develop.
- You can use a text editor of your choice including Notepad++, Visual Studio Code, Sublime Text, Atom or any other text editor you are comfortable with.
- You can use any web browser including Google Chrome, Firefox, Microsoft Edge, Internet Explorer etc.

A Simple JavaScript Program

- You should place all your JavaScript code within **<script> tags** (<script> and </script>) if you are keeping your JavaScript code within the HTML document itself.
- This helps your browser distinguish your JavaScript code from the rest of the code.
- You have to use the type attribute within the <script> tag and set its value to text/javascript like this:
    - **`<script type="text/javascript">`**

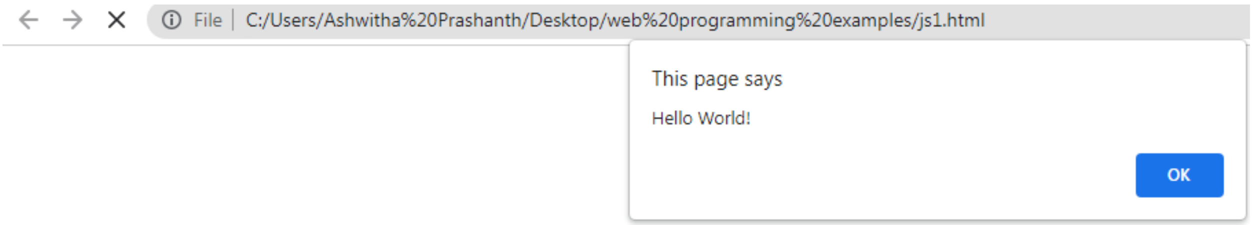


Hello World Example:

This code is editable. Click Run to Execute

```html
<html>
<head>
    <title>My First JavaScript code!!!</title>
    <script type="text/javascript">
        alert("Hello World!");
    </script>
</head>
<body>
</body>
</html>
```

output:

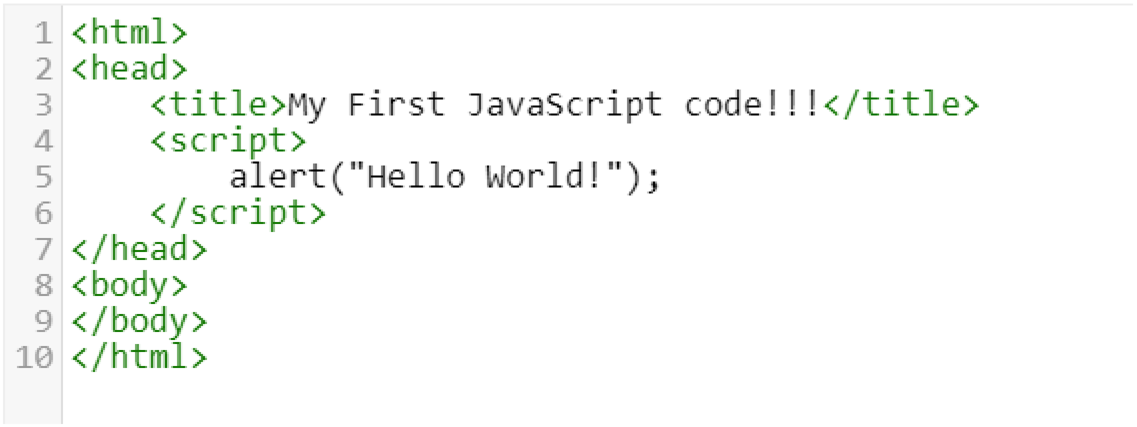


C:/Users/Ashwitha%20Prashanth/Desktop/web%20programming%20examples/js1.html

This page says

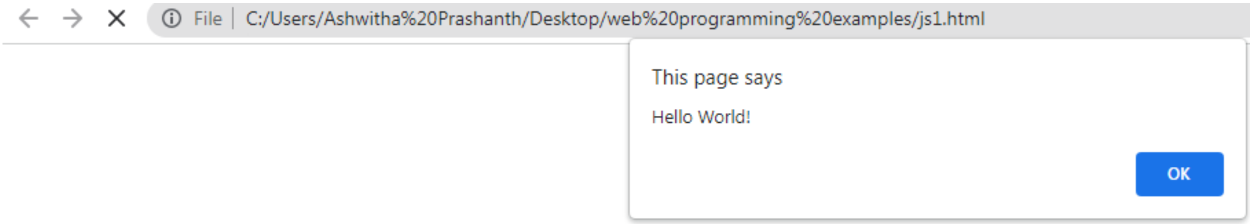
Hello World!

Note: type=”text/javascript” is not necessary in HTML5. Following code will work.



```html
<html>
<head>
    <title>My First JavaScript code!!!</title>
    <script>
        alert("Hello World!");
    </script>
</head>
<body>
</body>
</html>
```

output:



C:/Users/Ashwitha%20Prashanth/Desktop/web%20programming%20examples/js1.html

This page says

Hello World!

JavaScript Variable: Declare, Assign a Value with Example

Variables are used to **store values** (name = “John”) **or expressions** (sum = x + y).

Declare Variables in JavaScript

Before using a variable, you first need to declare it. You have to use the keyword **var** to declare a variable like this:

```
var name;
```

Assign a Value to the Variable

You can assign a value to the variable either while declaring the variable or after declaring the variable.

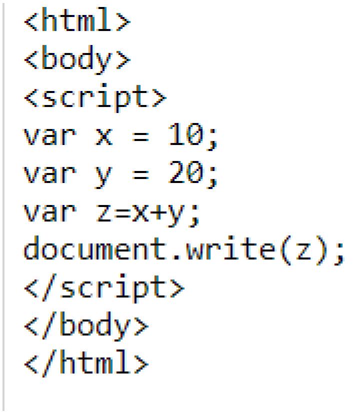
```
var name = "John";
```

OR

```
var name;

name = "John";
```

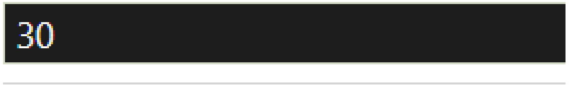
Naming Variable

- variable names should start with a letter and they are case sensitive.
- Hence the variables student name and studentName are different because the letter n in a name is different (n and N).



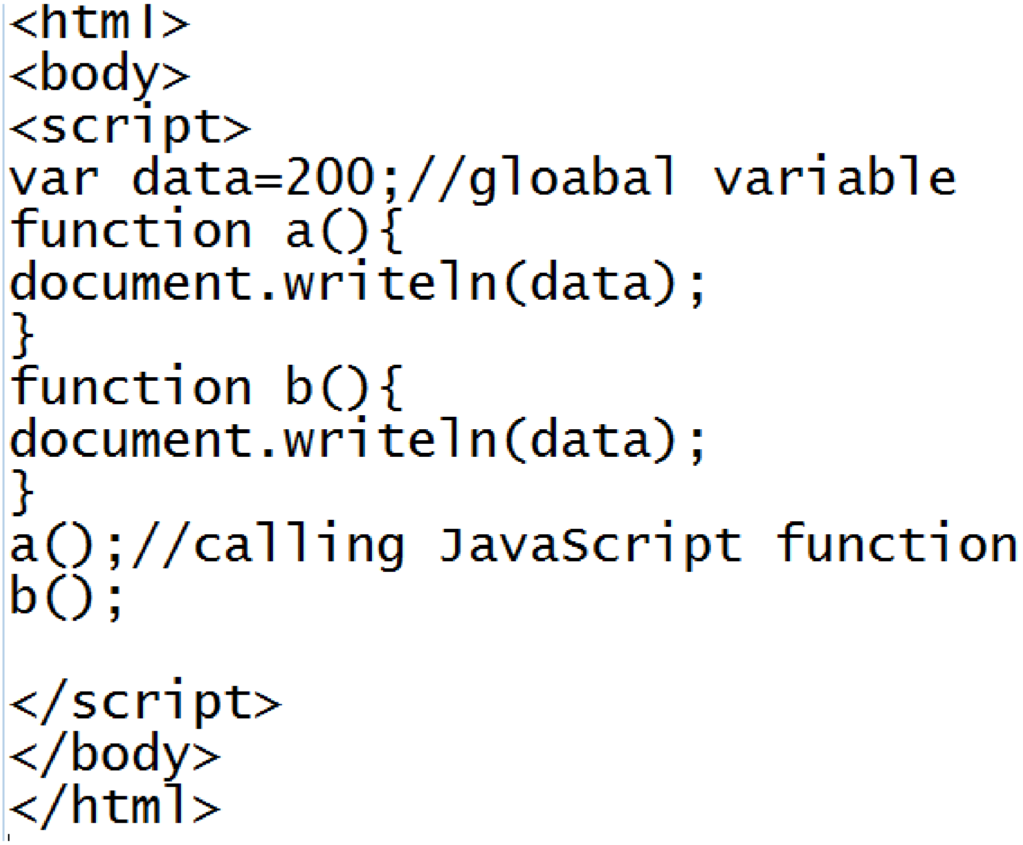
```
<html>
<body>
<script>
var x = 10;
var y = 20;
var z=x+y;
document.write(z);
</script>
</body>
</html>
```

Output of the above example



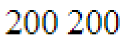
```
30
```

Global variable



```
<html>
<body>
<script>
var data=200;//gloabal variable
function a(){
document.writeln(data);
}
function b(){
document.writeln(data);
}
a();//calling JavaScript function
b();

</script>
</body>
</html>
```

OUTPUT



```
200 200
```

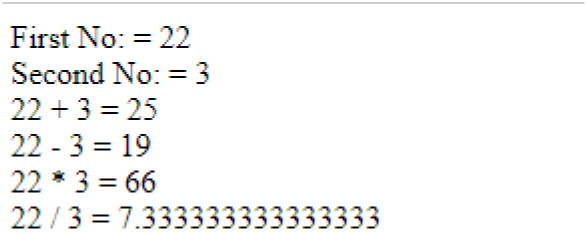
javascript program to declare variable and perform arithmetic operations

```
<html>
<head>
<title>Variables!!!</title>
<script type="text/javascript">
var one = 22;
var two = 3;
var add = one + two;
var minus = one - two;
var multiply = one * two;
var divide = one/two;
document.write("First No: = " + one + "<br />Second No: = " + two + " <br />");
        document.write(one + " + " + two + " = " + add + "<br/>");
        document.write(one + " - " + two + " = " + minus + "<br/>");
        document.write(one + " * " + two + " = " + multiply + "<br/>");
        document.write(one + " / " + two + " = " + divide + "<br/>");
</script>
</head>
<body>
```

</body>

</html>

output



```
First No: = 22
Second No: = 3
22 + 3 = 25
22 - 3 = 19
22 * 3 = 66
22 / 3 = 7.333333333333333
```

4 Ways to Declare a JavaScript Variable:

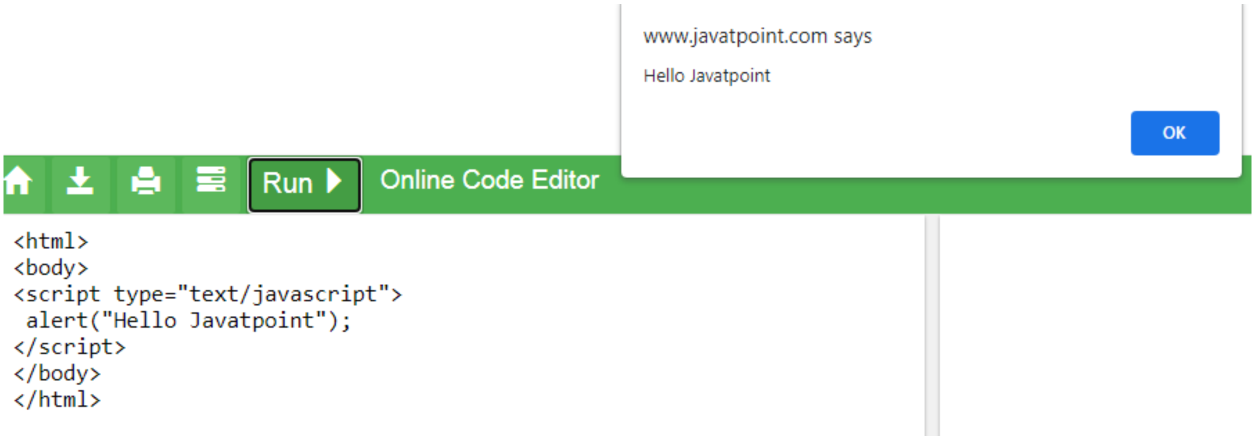
- Using `var`
- Using `let`
- Using `const`
- Using nothing

1) JavaScript Example : code between the body tag

In the above example, we have displayed the dynamic content using JavaScript. Let’s see the simple example of JavaScript that displays alert dialog box.

```
<html>
<body>
<script type="text/javascript">
 alert("Hello Javatpoint");
</script>
</body>
</html>
```

output:



www.javatpoint.com says

Hello Javatpoint

```
<html>
<body>
<script type="text/javascript">
 alert("Hello Javatpoint");
</script>
</body>
</html>
```

2) JavaScript Example : code between the head tag

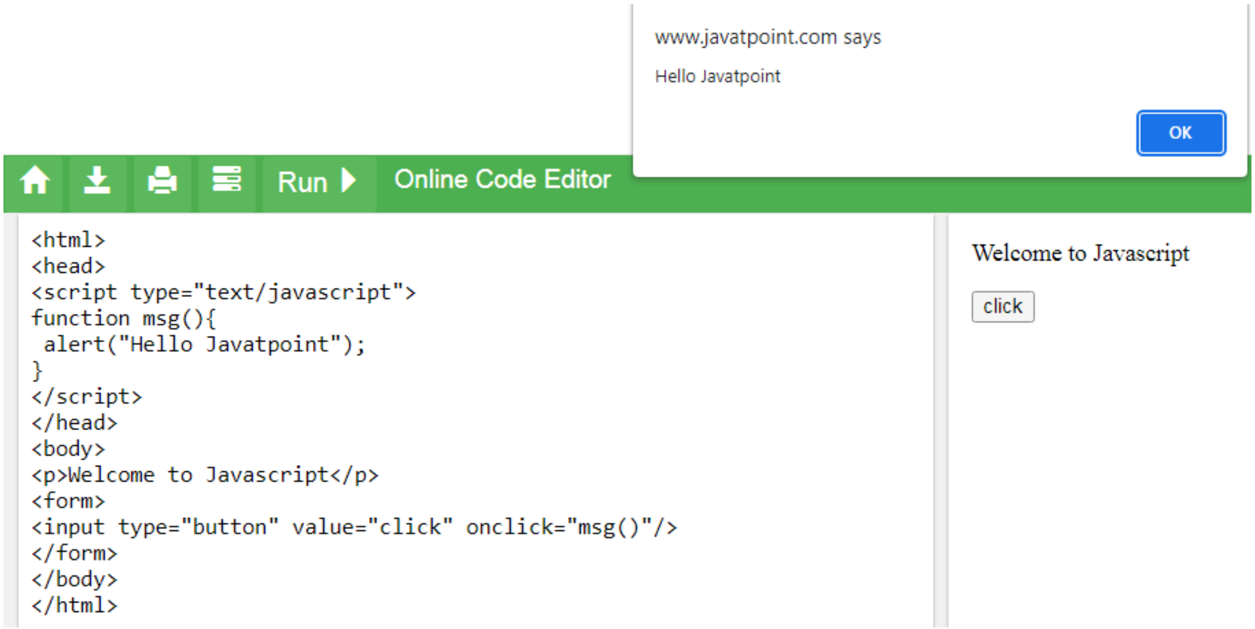
Let’s see the same example of displaying alert dialog box of JavaScript that is contained inside the head tag.

In this example, we are creating a function msg(). To create function in JavaScript, you need to write function with function_name as given below.

To call function, you need to work on event. Here we are using onclick event to call msg() function.

```
<html>
<head>
<script type="text/javascript">
function msg(){
 alert("Hello Javatpoint");
}
</script>
</head>
<body>
<p>Welcome to JavaScript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
</html>
```

output:



www.javatpoint.com says

Hello Javatpoint

```
<html>
<head>
<script type="text/javascript">
function msg(){
 alert("Hello Javatpoint");
}
</script>
</head>
<body>
<p>Welcome to Javascript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
</html>
```

Welcome to Javascript

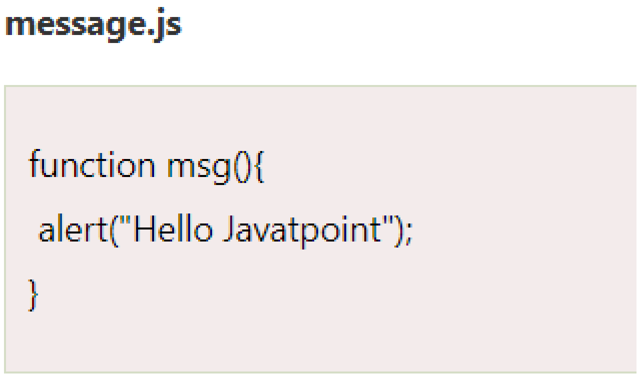
External JavaScript file

We can create external JavaScript file and embed it in many html page.

It provides **code re usability** because single JavaScript file can be used in several html pages.

An external JavaScript file must be saved by .js extension. It is recommended to embed all JavaScript files into a single file. It increases the speed of the webpage.

Let's create an external JavaScript file that prints Hello Javatpoint in a alert dialog box.

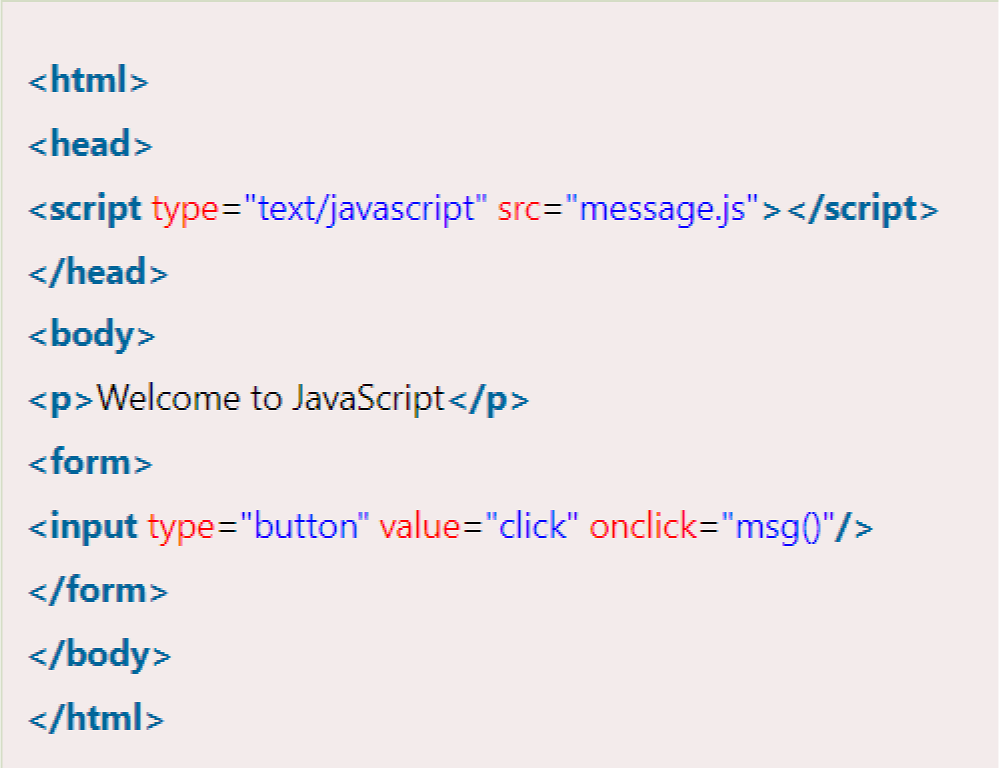


message.js

```
function msg(){
 alert("Hello Javatpoint");
}
```

Let's include the JavaScript file into html page. It calls the JavaScript function on button click.

index.html



```
<html>
<head>
<script type="text/javascript" src="message.js"></script>
</head>
<body>
<p>Welcome to JavaScript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
</html>
```

JavaScript Comment

The **JavaScript comments** are meaningful way to deliver message. It is used to add information about the code, warnings or suggestions so that end user can easily interpret the code.

Types of JavaScript Comments

There are two types of comments in JavaScript.

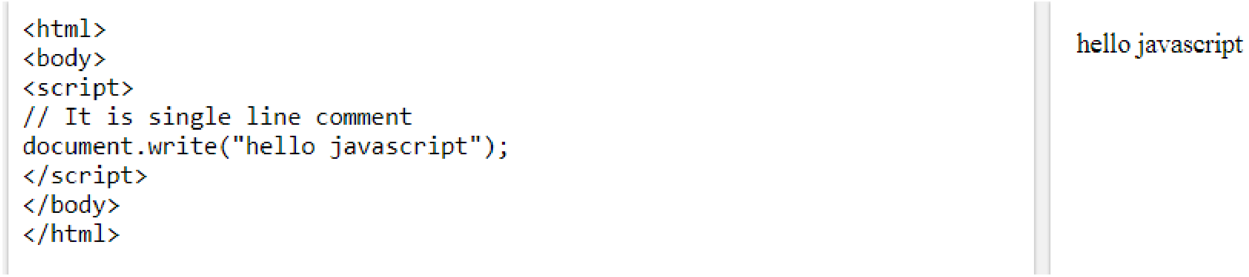
1. Single-line Comment
2. Multi-line Comment

JavaScript Single line Comment

It is represented by double forward slashes (//). It can be used before and after the statement.

Let’s see the example of single-line comment i.e. added before the statement.

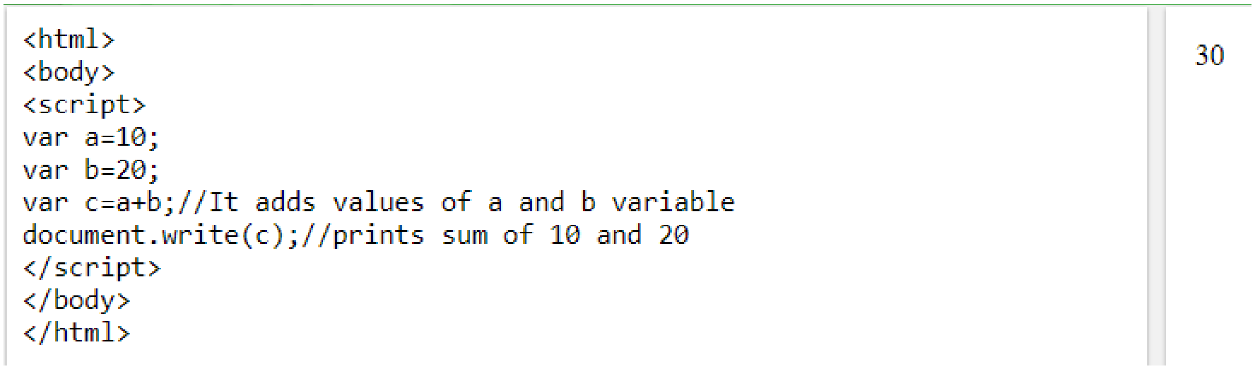
Example 1



```
<html>
<body>
<script>
// It is single line comment
document.write("hello javascript");
</script>
</body>
</html>
```

```
hello javascript
```

Example 2



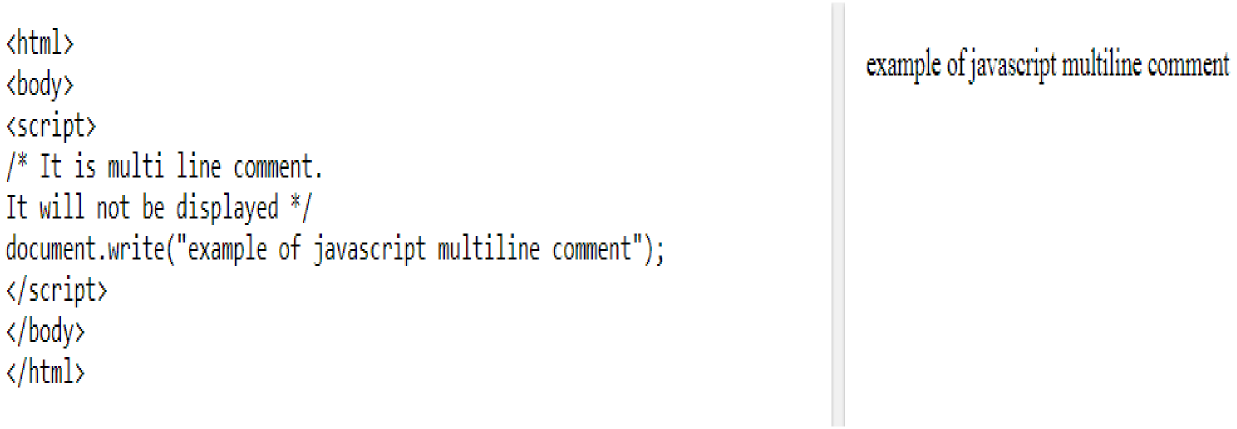
```
<html>
<body>
<script>
var a=10;
var b=20;
var c=a+b;//It adds values of a and b variable
document.write(c);//prints sum of 10 and 20
</script>
</body>
</html>
```

```
30
```

JavaScript Multi line Comment

It can be used to add single as well as multi line comments. So, it is more convenient.

It is represented by forward slash with asterisk then asterisk with forward slash. For example:



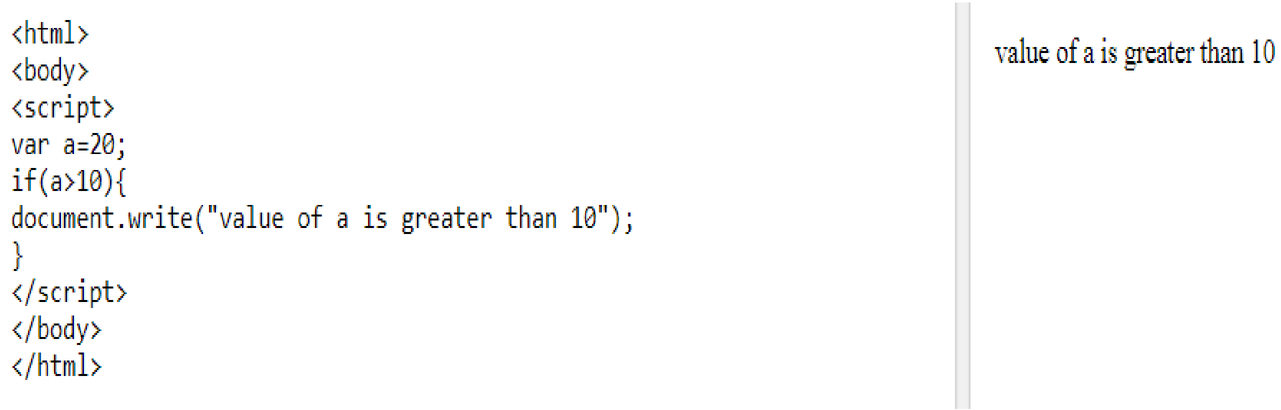
```
<html>
<body>
<script>
/* It is multi line comment.
It will not be displayed */
document.write("example of javascript multiline comment");
</script>
</body>
</html>
```

```
example of javascript multiline comment
```

JavaScript If statement

It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

```
if(expression){
//content to be evaluated
}
```



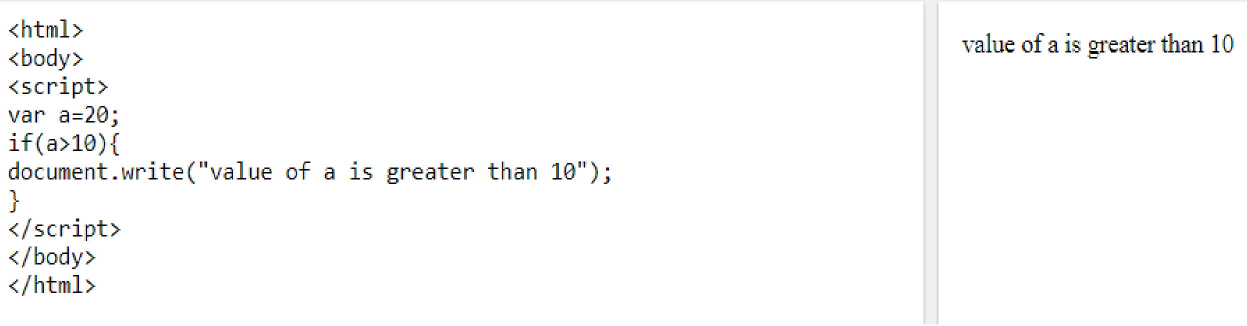
```
<html>
<body>
<script>
var a=20;
if(a>10){
document.write("value of a is greater than 10");
}
</script>
</body>
</html>
```

```
value of a is greater than 10
```

JavaScript If...else Statement

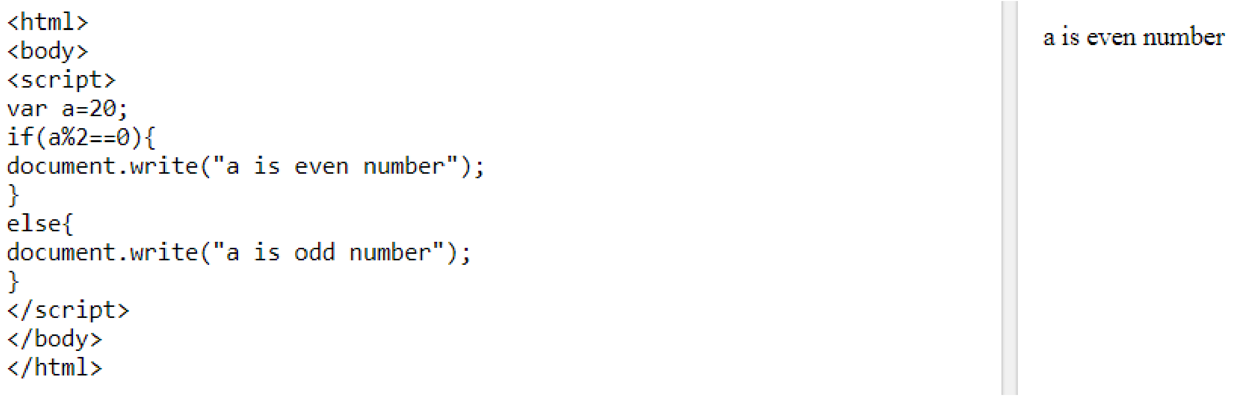
It evaluates the content whether condition is true of false. The syntax of JavaScript if-else statement is given below.

```
if(expression){
//content to be evaluated if condition is true
}
else{
//content to be evaluated if condition is false
}
```



```
<html>
<body>
<script>
var a=20;
if(a>10){
document.write("value of a is greater than 10");
}
</script>
</body>
</html>
```

```
value of a is greater than 10
```



```
<html>
<body>
<script>
var a=20;
if(a%2==0){
document.write("a is even number");
}
else{
document.write("a is odd number");
}
</script>
</body>
</html>
```

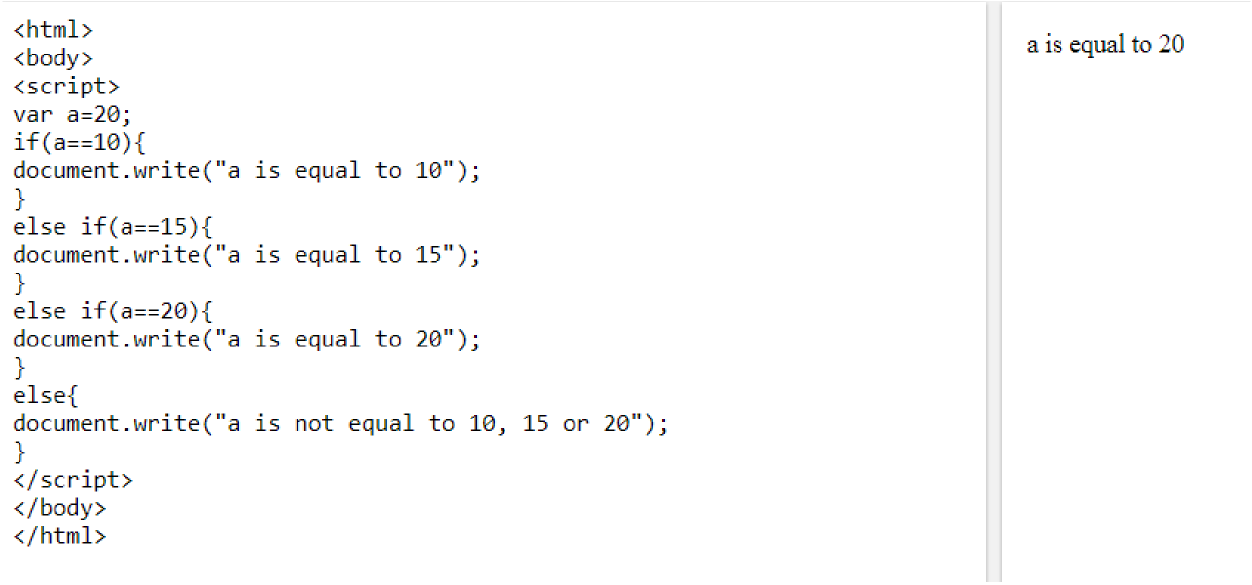
```
a is even number
```

JavaScript If...else if statement

It evaluates the content only if expression is true from several expressions. The signature of JavaScript if else if statement is given below.

```
if(expression1){
//content to be evaluated if expression1 is true
}
else if(expression2){
//content to be evaluated if expression2 is true
}
else if(expression3){
//content to be evaluated if expression3 is true
}
else{
//content to be evaluated if no expression is true
}
```

Example:

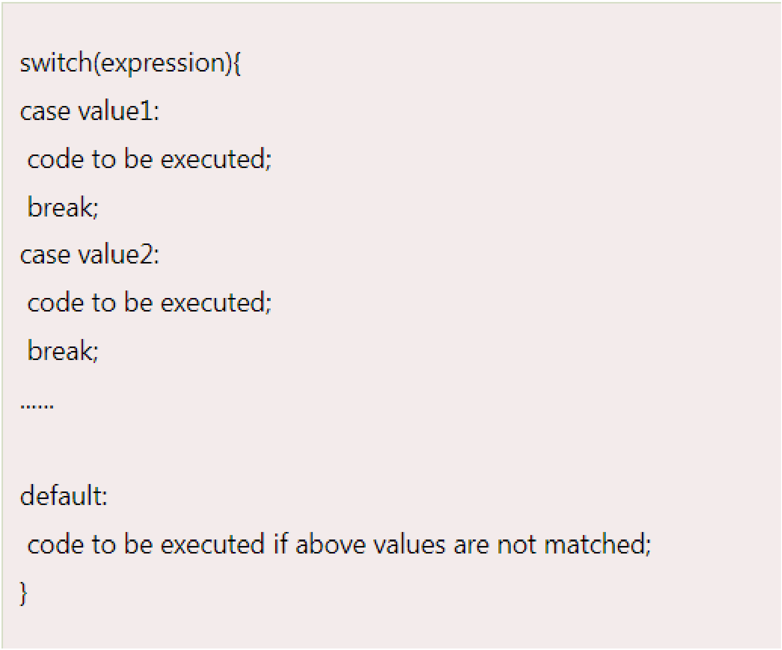


```
<html>
<body>
<script>
var a=20;
if(a==10){
document.write("a is equal to 10");
}
else if(a==15){
document.write("a is equal to 15");
}
else if(a==20){
document.write("a is equal to 20");
}
else{
document.write("a is not equal to 10, 15 or 20");
}
</script>
</body>
</html>
```

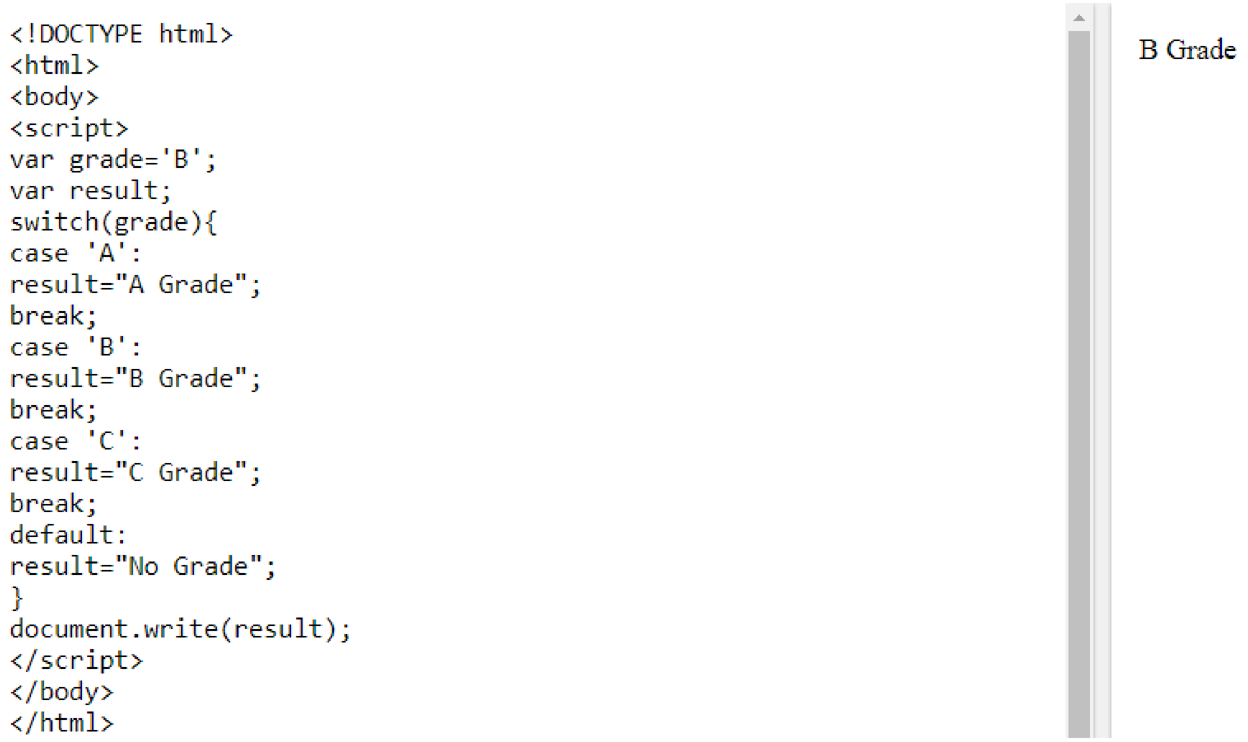
```
a is equal to 20
```

JavaScript Switch

The **JavaScript switch statement** is used *to execute one code from multiple expressions*. It is just like else if statement that we have learned in previous page. But it is convenient than *if..else..if* because it can be used with numbers, characters etc.The syntax of JavaScript switch statement is given below.



```
switch(expression){
case value1:
 code to be executed;
 break;
case value2:
 code to be executed;
 break;
......

default:
 code to be executed if above values are not matched;
}
```



```
<!DOCTYPE html>
<html>
<body>
<script>
var grade='B';
var result;
switch(grade){
case 'A':
result="A Grade";
break;
case 'B':
result="B Grade";
break;
case 'C':
result="C Grade";
break;
default:
result="No Grade";
}
document.write(result);
</script>
</body>
</html>
```

```
B Grade
```

JavaScript Functions

JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.

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

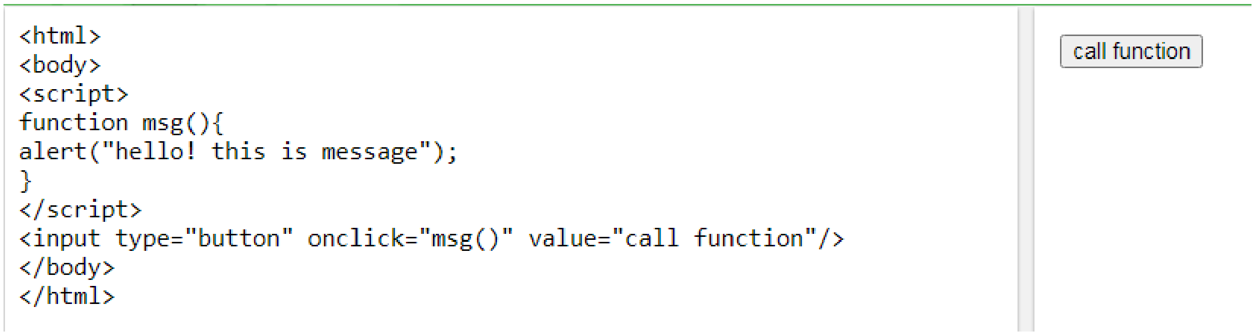
The syntax of declaring function is given below.

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

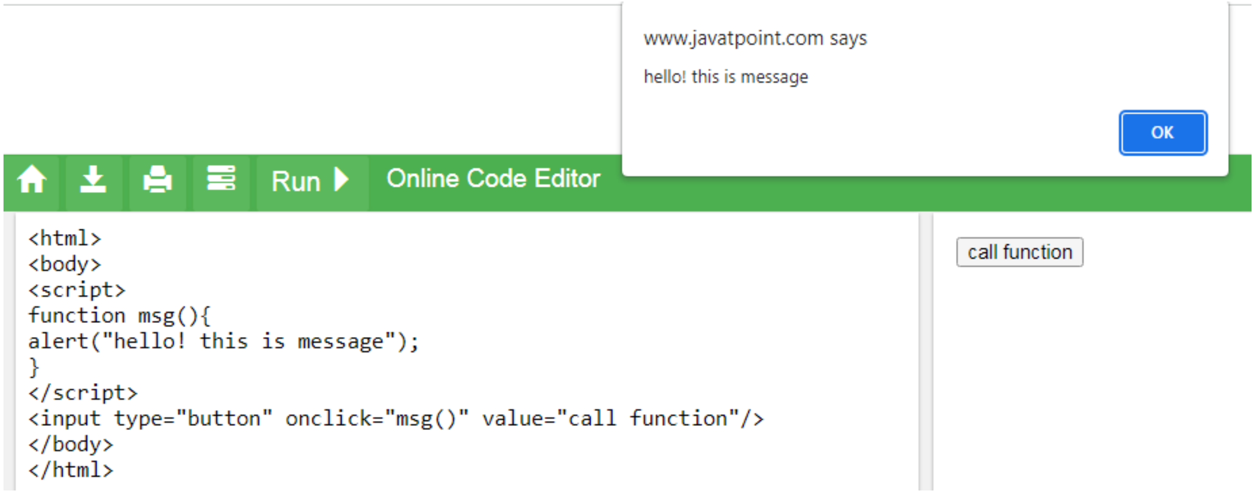
JavaScript Functions can have 0 or more arguments.

JavaScript Function Example

Let’s see the simple example of function in JavaScript that does not has arguments.



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



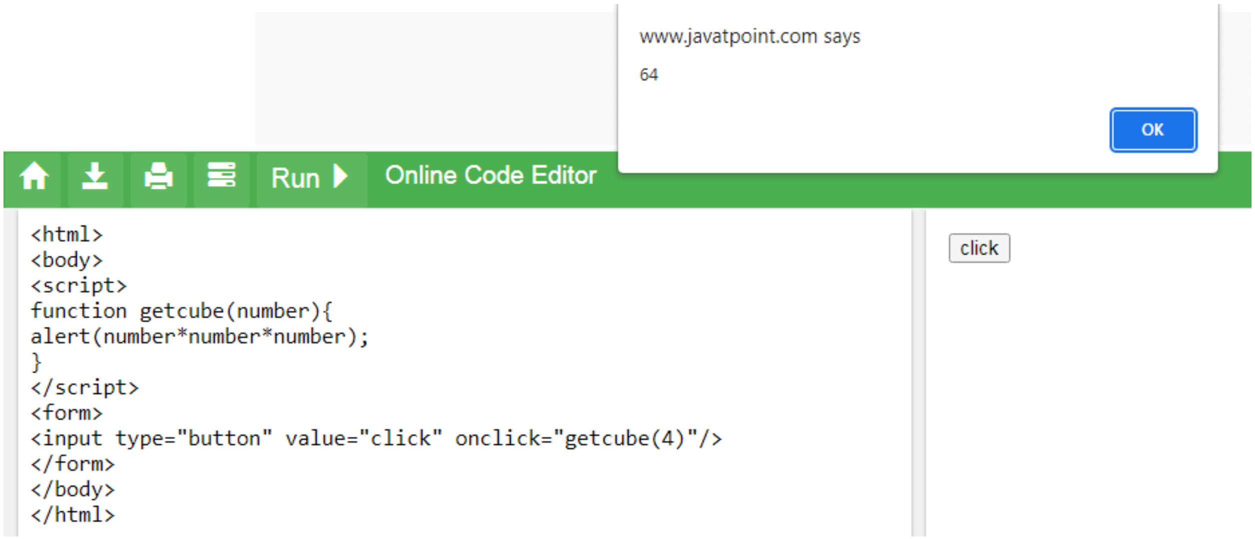
www.javatpoint.com says

hello! this is message

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

JavaScript Function Arguments

We can call function by passing arguments. Let’s see the example of function that has one argument.



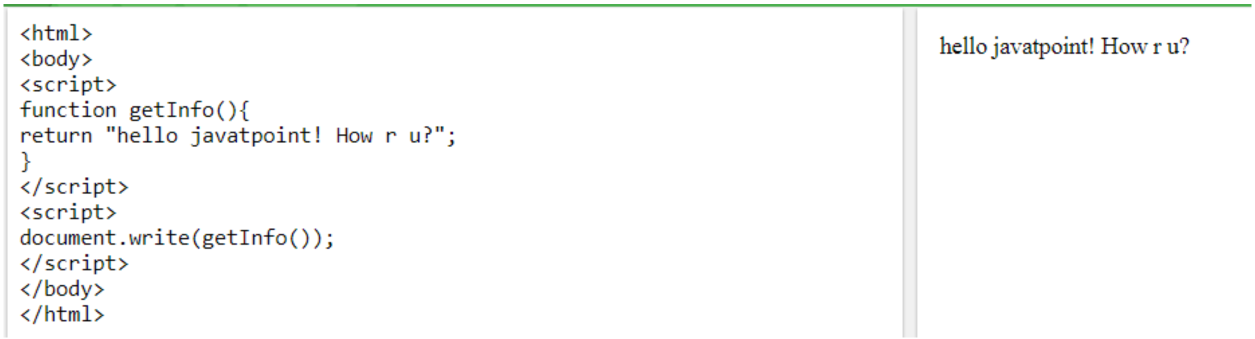
www.javatpoint.com says

64

```
<html>
<body>
<script>
function getcube(number){
alert(number*number*number);
}
</script>
<form>
<input type="button" value="click" onclick="getcube(4)"/>
</form>
</body>
</html>
```

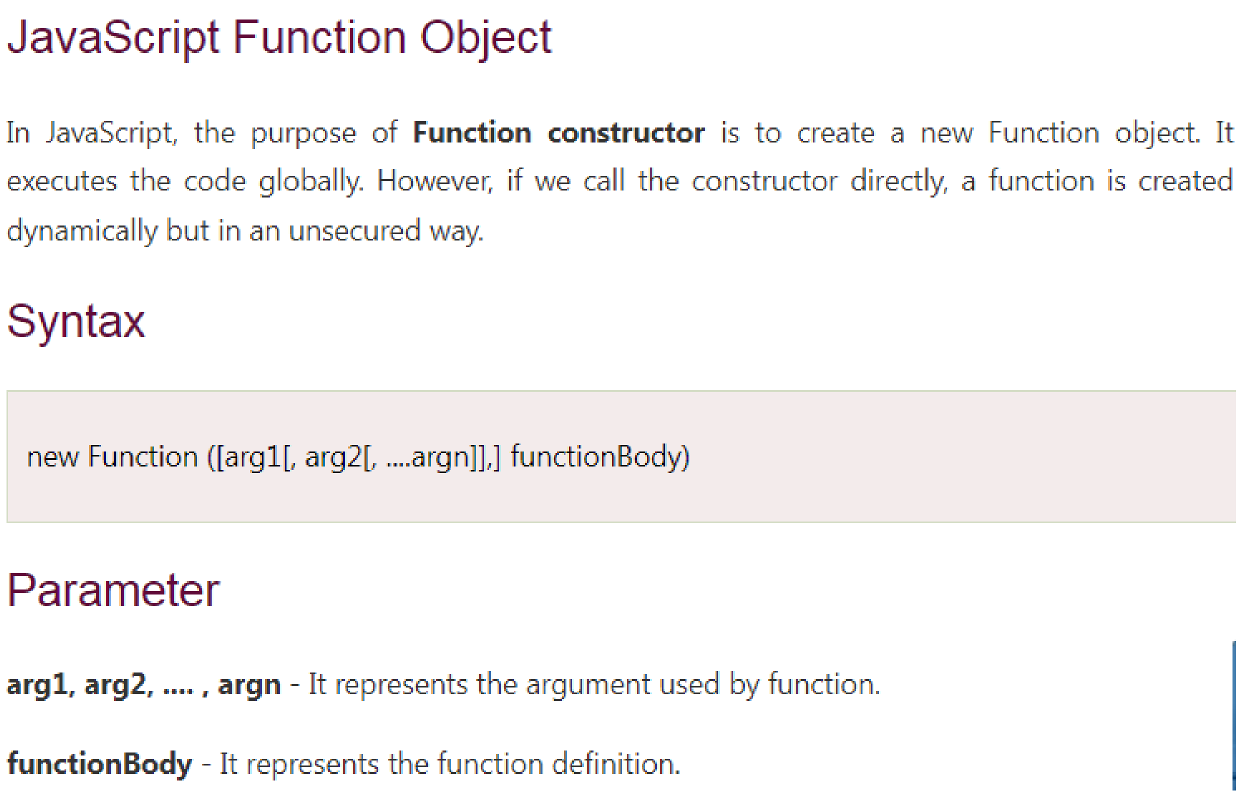
Function with Return Value

We can call function that returns a value and use it in our program. Let’s see the example of function that returns value.



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

hello javatpoint! How r u?



JavaScript Function Object

In JavaScript, the purpose of **Function constructor** is to create a new Function object. It executes the code globally. However, if we call the constructor directly, a function is created dynamically but in an unsecured way.

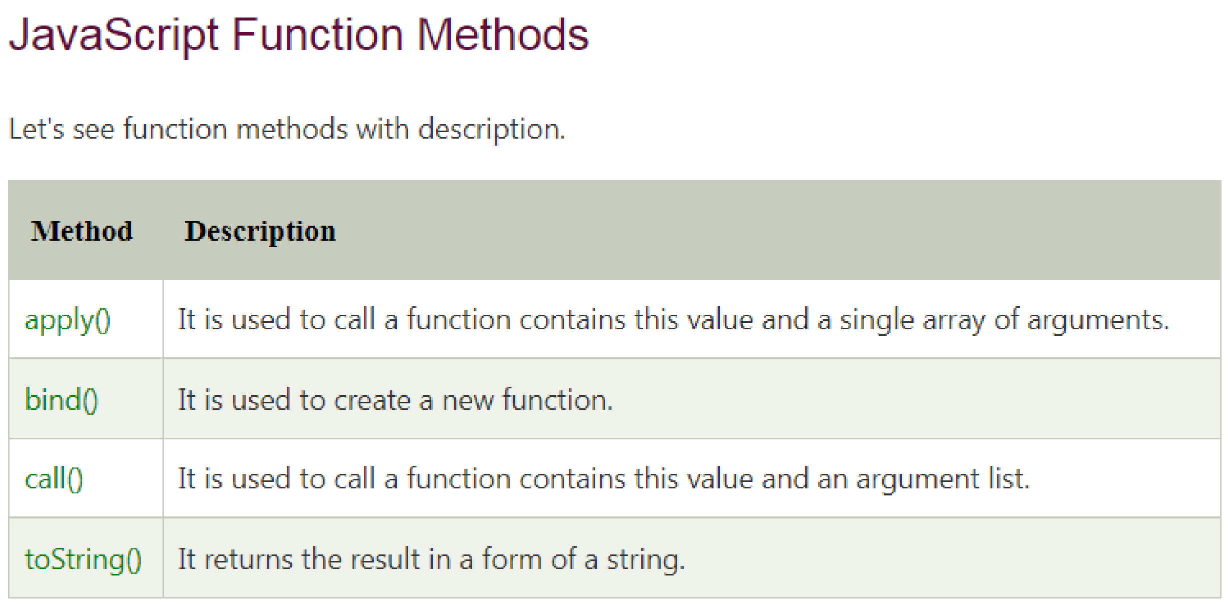
Syntax

```
new Function ([arg1[, arg2[, ....argn]],] functionBody)
```

Parameter

arg1, arg2, ... , argn - It represents the argument used by function.

functionBody - It represents the function definition.



JavaScript Function Methods

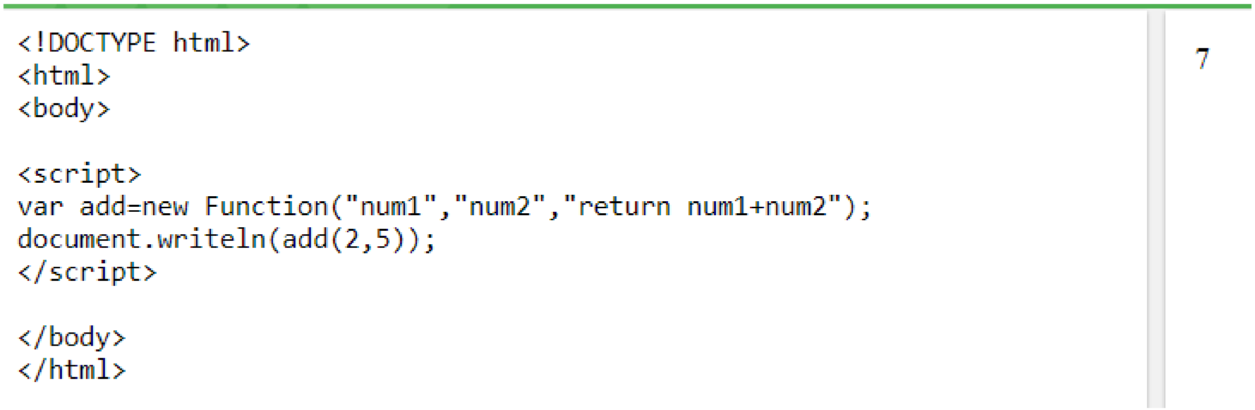
Let's see function methods with description.

Method	Description
apply()	It is used to call a function contains this value and a single array of arguments.
bind()	It is used to create a new function.
call()	It is used to call a function contains this value and an argument list.
toString()	It returns the result in a form of a string.

JavaScript Function Object Examples

Example 1

Let's see an example to display the sum of given numbers.

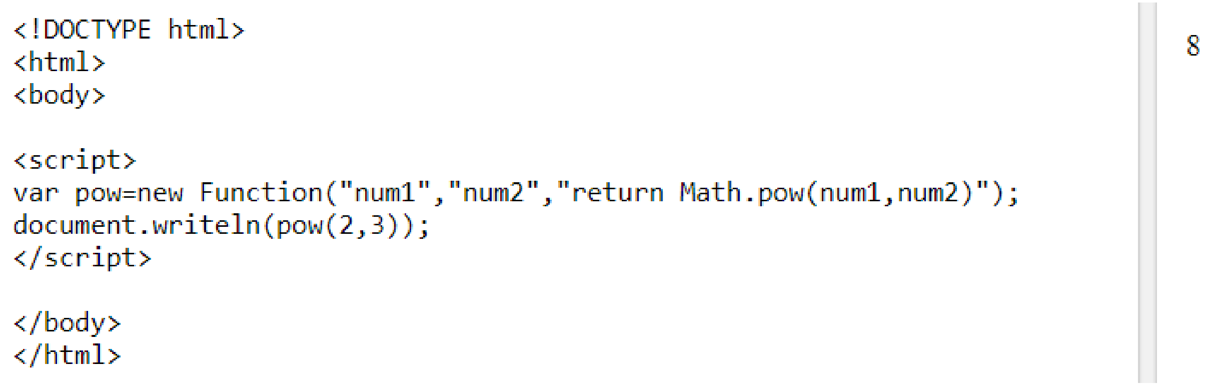


```
<!DOCTYPE html>
<html>
<body>

<script>
var add=new Function("num1","num2","return num1+num2");
document.writeln(add(2,5));
</script>

</body>
</html>
```

```
7
```

Example 2

Let's see an example to display the power of provided value.



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

<script>
var pow=new Function("num1","num2","return Math.pow(num1,num2)");
document.writeln(pow(2,3));
</script>

</body>
</html>
```

```
8
```