

JavaScript | Website Design & Development®

Dilshard Ahamed | BSc (Hons) in Computing, *University of Wolverhampton*

JavaScript

Introduction
Basic Exercises



JavaScript

JavaScript is most popular and widely used programming language. It is being used by multinational companies around the world like..

NETFLIX

Walmart

PayPal™



Versions

JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997.

ECMAScript is the official name of the language.

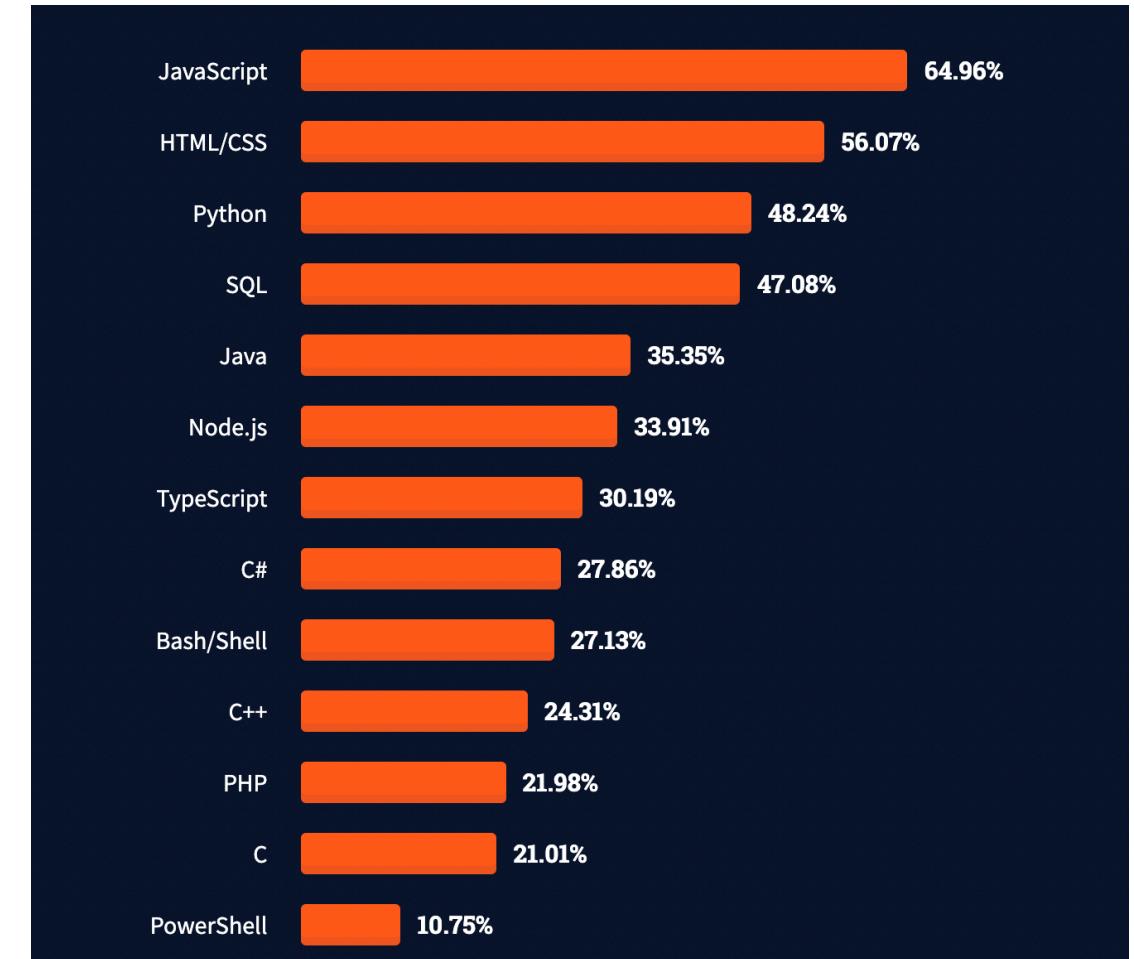
ECMAScript versions have been abbreviated to ES1, ES2, ES3, ES5, and ES6.

Since 2016 new versions are named by year (ECMAScript 2016 / 2017 / 2018).

- The Original JavaScript ES1 ES2 ES3 (1997-1999)
- The First Main Revision ES5 (2009)
- The Second Revision ES6 (2015)
- The Yearly Additions (2016, 2017, 2018) [for more](#)

Developer Survey Results 2019

Growing faster than any
other programming
languages



J O B

**FRONTEND
DEVELOPER**

**BACKEND
DEVELOPER**

**FULLSTACK
DEVELOPER**

Average salary of JavaScript
Developer **\$72,000/Year**





Where does JavaScript code run?

In every Browser we have JavaScript Engine inbuild



SpiderMonkey



Google V8



What can we do with JavaScript?

Earlier days we used JavaScript to built only interactive web pages

These days we can build...



Web/Mobile
APPLICATIONS



Real-time
NETWORKING APPS
Chats / Video Steaming apps

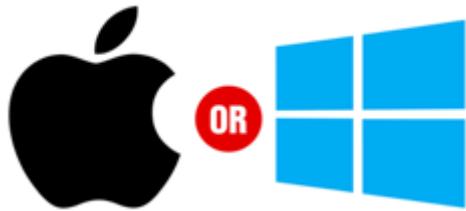


Games
ONLINE



Also JS can..

- JavaScript can change all the HTML elements & attributes in the page
- CSS styles in the page
- Remove existing HTML elements and attributes
- Add new HTML elements and attributes
- React to all existing HTML events in the page
- Create new HTML events in the page



Command + Option + I or Command + Shift + C
Ctrl + Shift + I or Ctrl + Shift + C

Try..

```
> console.log('Hello World');  
> 2 + 2  
> alert('Hii');
```

Open the Browser

Press Ctrl + Shift + I

Try...



Exercise 01

File | /Users/macbookpro/Documents...

This page says

Hii

Selected context only Autocomplete from history

Group similar Evaluate triggers user activation

Hello World Exercise-1.html:12

> 2+2

< 4

> alert('Hii');

>

OK

The screenshot shows a browser window titled 'Exercise 01'. An alert box is displayed with the text 'This page says' followed by 'Hii'. Below the alert, the developer tools are open, showing the 'Elements' tab with the 'Evaluate' section expanded. The 'Evaluate' input field contains the code 'alert("Hii")'; when run, it outputs 'Hii' in the results pane. Other visible options in the developer tools include 'Selected context only', 'Autocomplete from history', 'Group similar', and 'Evaluate triggers user activation'. The browser's address bar shows the file path '/Users/macbookpro/Documents...'.



A function called
`isLegalDrinkingAge()`
makes more sense than
`isOverEighteen()`
as the legal drinking age varies
from country to country,

Keep your `<script>` tag end of your `body`

Browser read files from top to bottom. So, if we put our JavaScript code on top of `<head>` browser will be busy on loading and executing it so content on the page will get delayed to render.

The code between `<script>` tag needs to talk to the elements of the web page, Example `show/hide` an element. So adding the script on end of the body section will be confident that all other elements rendered by the browser.

Keep your variable names in **simple** and **meaningful**

Good variable and function names should be easy to understand and tell you what is going on

`x1()`
`fe04`
`createNewMemberIfAgeOverTwentyOne`

// Comment as much as needed but not more

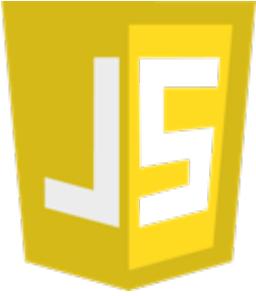
Comments are your messages to other developers and yourself, if you come back to your code after several months

Keep your `<script>` tag end of your `body`



Keep your variable names in **simple** and **meaningful**

// Comment as much as needed but not more



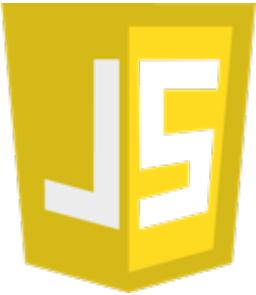
JavaScript Output

Writing into an HTML element, using
innerHTML.

document.write()

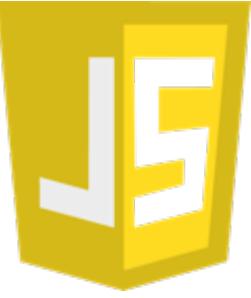
Alert box, using **window.alert()**

Browser console, using **console.log()**



Keywords

Keyword	Description
var	Declares a variable
let	Declares a block 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



Comments

● ● ●

```
/*
Multi-line Comments
The code below will change
the heading with id = "myH"
and the paragraph with id = "myP"
in my web page:
*/
```

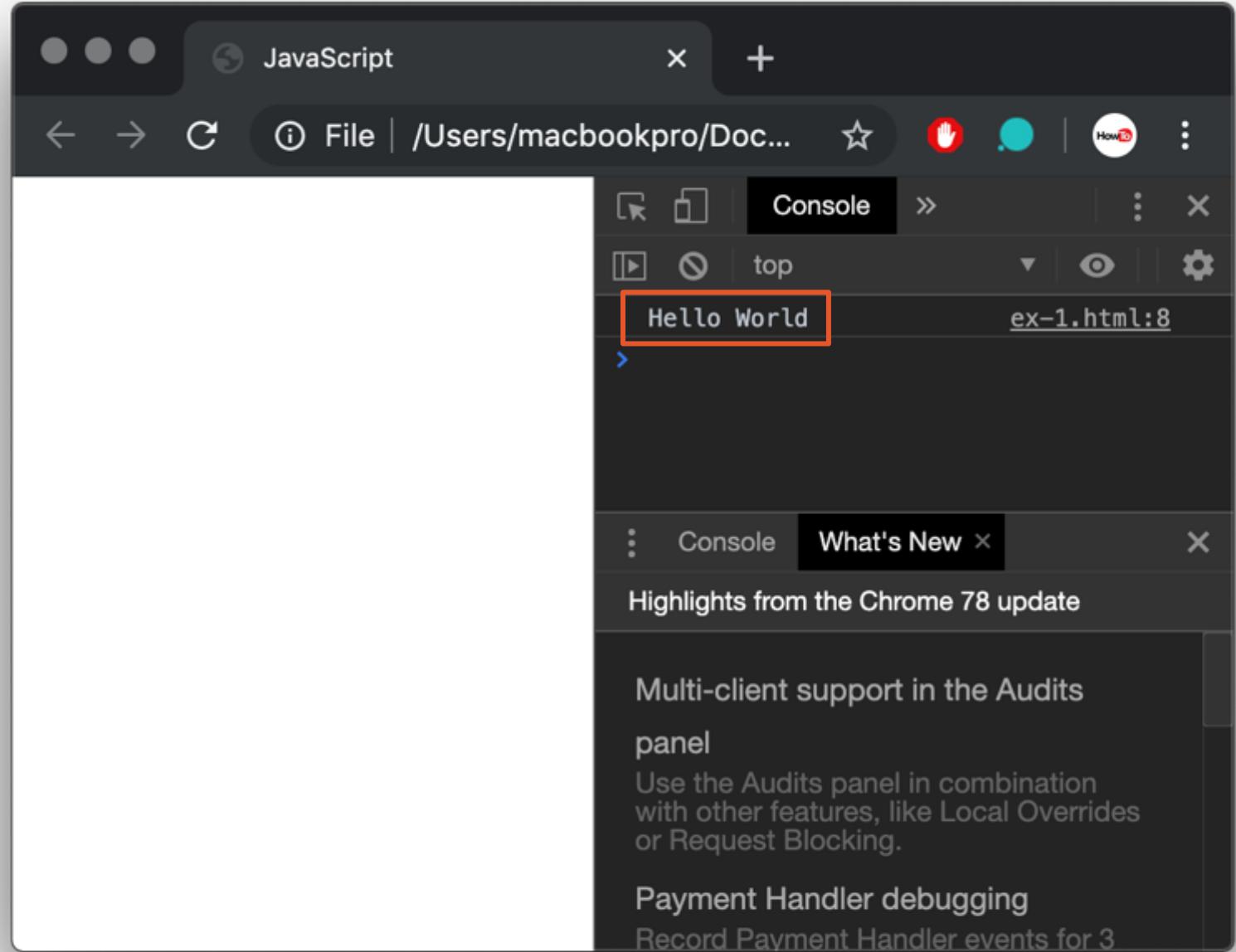
```
let x = 5;          // Declare x, give it the value of 5
```



Create a HTML document and add first JavaScript code

1. Create a HTML document with all the common code
2. Insert a <script> tag at the end of the body
3. Try to print “Hello World” at console log
4. Include a comment as “This is my first JavaScript code”
5. Check your browser Inspect -> Console for output







Separate JavaScript file from HTML Document

1. Create a HTML document with all common codes save it as **ex-2.html**
2. Create a another file and save it as **ex-2.js**

```
console.log('HND batch 62');
```

3. Reopen HTML file and import ex-2.js

```
<script src="ex-2.js"> </script>
```



The screenshot shows a code editor with two tabs open. The left tab is titled 'ex-2.html' and contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>JavaScript</title>
5 </head>
6 <body>
7 <script src="ex-2.js"></script>
8 </body>
9 </html>
```

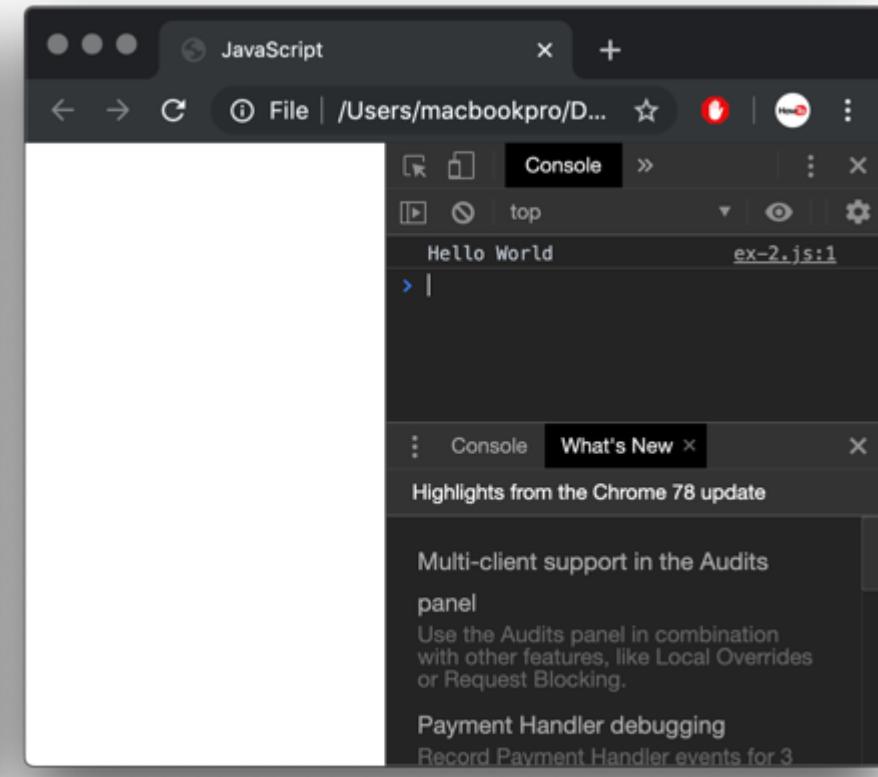
The right tab is titled 'ex-2.js' and contains the following JavaScript code:

```
1 console.log('Hello World');
```

Both tabs have a status bar at the bottom indicating 'Tab Size: 4'.

This screenshot shows the same code editor as the first one, but only the 'ex-2.js' tab is visible. It contains the same JavaScript code as the previous screenshot:1 console.log('Hello World');

The status bar at the bottom indicates 'Line 1, Column 28'.



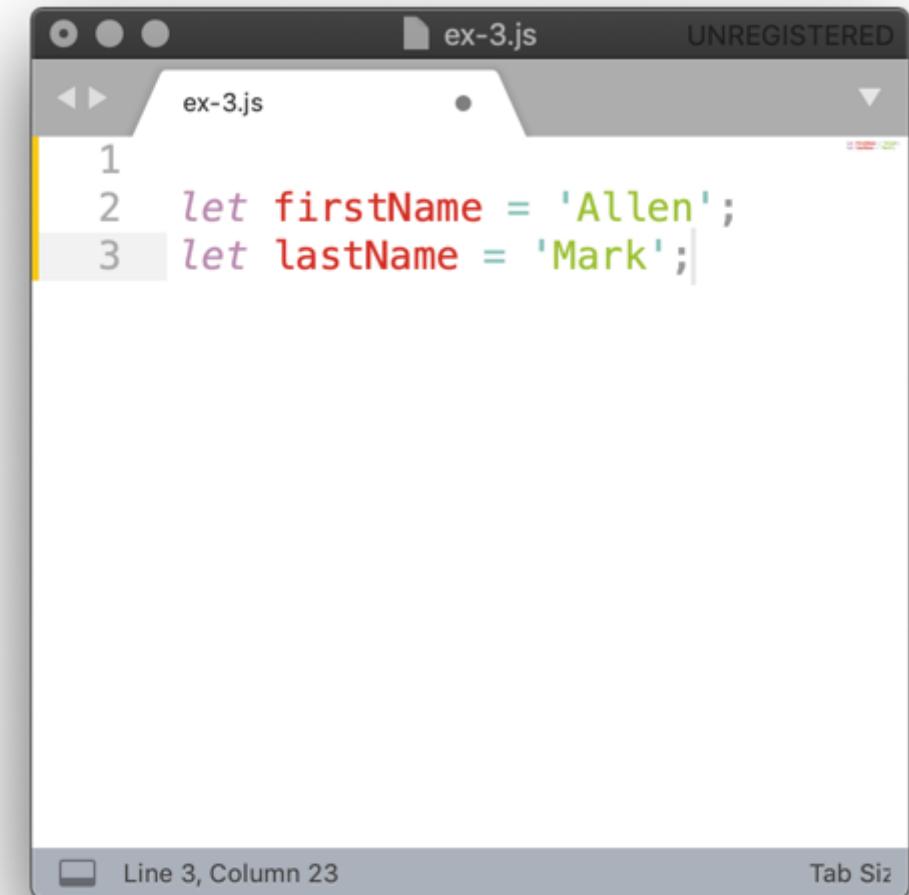


JavaScript - Variables

*We use variable to store data temporarily
In computer memory*

How to declare variables

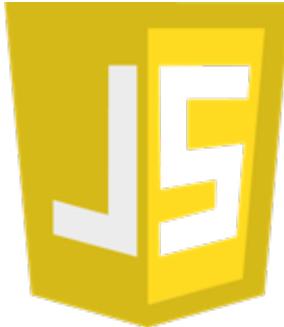
```
→ var marks1=89;  
    console.log(marks1);  
  
→ var or let + variable name + value
```



The screenshot shows a code editor window titled "ex-3.js" with the file status as "UNREGISTERED". The code in the editor is:

```
1  
2 let firstName = 'Allen';  
3 let lastName = 'Mark';
```

The status bar at the bottom of the editor indicates "Line 3, Column 23" and "Tab Siz".



JavaScript - Variables

JavaScript is a dynamic language

We have two types of languages

1. Static Language
2. Dynamic Language

Static Language

In static language when we declare a variable that cannot be changed



```
String name = 'Ahamed';  
String name = 123;
```

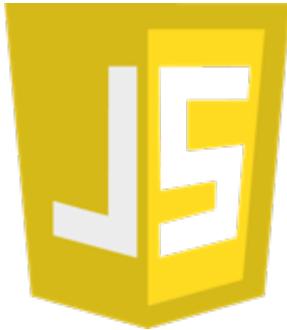
```
Console  
top  
> typeof name  
< "string"  
> name = 1;  
< 1  
> typeof name  
< "number"  
> |
```

Dynamic Language

In dynamic language the type of variable can be changed on run time

```
let nic = '123456789V';  
let nic = 25;
```





JavaScript - Variables

*We use variable to store data temporarily
In computer memory*



Rules & Regulations

- Variable cannot be a **reserved keyword**
e.g. if, else, let, var, true, false, typeof, return etc.
- Variables should be **meaningful**
- Variable **cannot** start with a **number**
e.g. 1name, 50color etc.
- Variable cannot contain **space** or **hyphen**
e.g. basic salary, basic-salary etc.
- Variable must be declared with **camelCase**
e.g. firstName, basicSalary, etc.
- Variable names are **case sensitive**
e.g. firstName, FirstName two different.
- Variable can be declared in **same row** with comma
e.g. let firstName, lastName;
Note : *but better to declare it in separate line*



JavaScript - Variables

JavaScript have two types of variables

Primitives / Value Types

- String
- Number
- Boolean
- undefined
- null

Reference Types

- Objects
- Array
- Functions



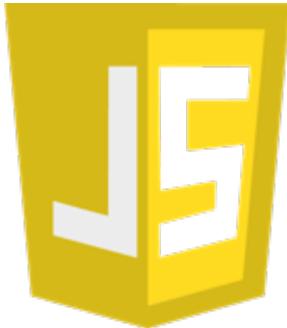
JavaScript – Variables (Primitives)

*We use variable to store data temporarily
In computer memory*

What kind of values can be assigned to a variable

```
let name = 'Ahamed'; //String  
let age = 25; //Number  
let isApproved = true; //Boolean  
let firstName = undefined; //undefined (Default)  
let lastName = null; //null (if we want to clear) (Object)
```

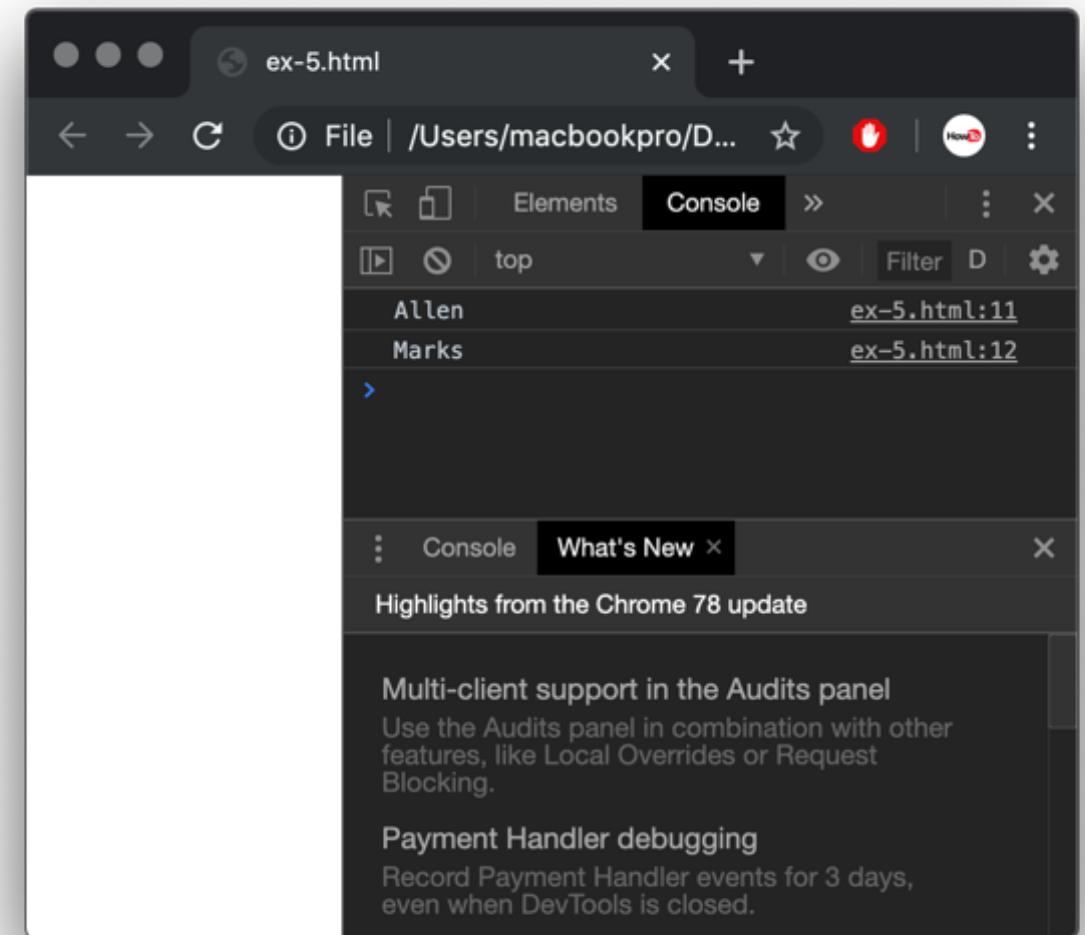
Try “typeof” to find data types with chrome.

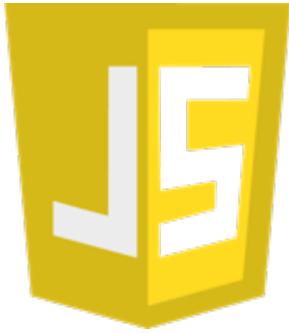


JavaScript - Program

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
<script>
    let firstName = 'Allen';
    let lastName = 'Marks';

    console.log(firstName);
    console.log(lastName);
</script>
</body>
</html>
```





JavaScript - Program

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
<script>
    let mk1 = 50;
    let mk2 = 50;
    let tot = mk1 + mk2;

    console.log('Marks 01 : ' + mk1);
    console.log('Marks 02 : ' + mk2);
    console.log('Total : ' + tot);
</script>
</body>
</html>
```

The screenshot shows the Google Chrome DevTools interface with the 'Console' tab selected. The page title is 'ex-5.html'. The console output displays three lines of text: 'Marks 01 :50', 'Marks 02 :50', and 'Total :100', each preceded by the timestamp 'ex-5.html:12', 'ex-5.html:13', and 'ex-5.html:14' respectively. Below the console, there is a 'Highlights from the Chrome 78 update' section with two items: 'Multi-client support in the Audits panel' and 'Payment Handler debugging'.

```
Makrs 01 :50
Makrs 02 :50
Total :100
```

Highlights from the Chrome 78 update

- Multi-client support in the Audits panel
- Use the Audits panel in combination with other features, like Local Overrides or Request Blocking.
- Payment Handler debugging
- Record Payment Handler events for 3 days, even when DevTools is closed.

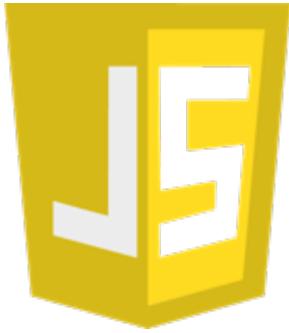


Create a JavaScript code to find total

1. Create a HTML document with all common codes save it as **ex-3.html**
2. Use external JavaScript file
3. Take below output in **console.log**

Name = Your name
Math = 98
Science = 88
English = 89
Total = ?
Average = ?





JavaScript – Objects (Reference)

Objects | Array | Functions

What object means

e.g. An object is like a person in real life, A person has **name, age, address**. These are properties that a personal has. The same concept we have here in JavaScript object. If we have multiple, related variables we put all inside an object.

```
<script>
let name = 'john';
let age = 25;

console.log(name);
console.log(age);
</script>
```

```
<script>
let person={
    name: 'john',
    age: 25,
    address: 'Colombo'
};

console.log(person);
</script>
```



```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript</title>
</head>
<body>
    <script>
        let person={
            name: 'john',
            age: 25,
        };
        console.log(person);
    </script>
</body>
</html>
```

A screenshot of the Chrome DevTools interface. The title bar shows 'JavaScript'. The main area is the 'Console' tab, which is highlighted. Below the tabs are icons for back, forward, and search. The console output shows:

```
ex-7-objects.html:12
▶ {name: "john", age: 25}
```

The bottom of the DevTools window shows a 'What's New' section with the following text:

Highlights from the Chrome 78 update

Multi-client support in the Audits panel
Use the Audits panel in



JavaScript – Objects (Reference)

Objects

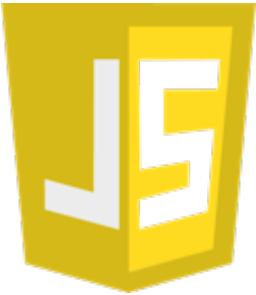
Object can be called and assigned in different ways

Dot Notation

```
person.name = 'John';
console.log(person.name);
```

Bracket Notation

```
person['name'] = 'Peter';
console.log(person.name);
```



JavaScript – Array (Reference)

Objects | Array | Functions

What and why

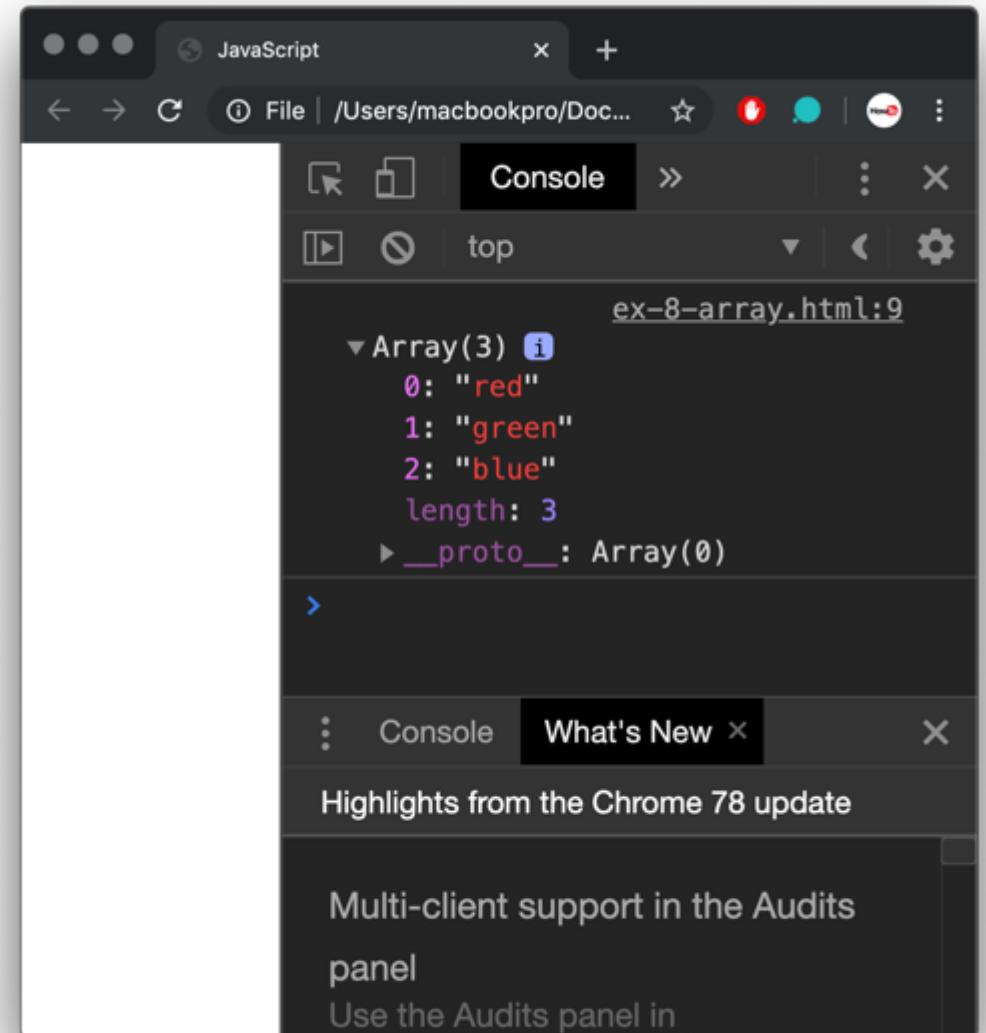
*Array is a data structure that we use to represent the **list of items***

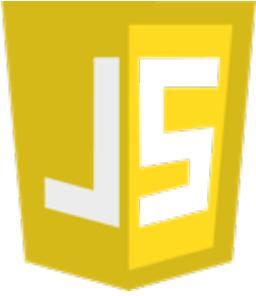
e.g. If we dealing with list of objects or list of colors that user has selected. We can use array to store them.

```
<script>

let selectedColors = ['red', 'green', 'blue'];
console.log(selectedColors);

</script>
```





JavaScript – Array length

Return number of elements in an array

`array.length`

```
var arr = new Array( 10, 20, 30 );
document.write("arr.length is : " +
arr.length);
```



Array

The screenshot shows the Chrome DevTools Console. A red box highlights the text "Index of an array". An arrow points from this box to the index "0" in the array object shown in the console.

```
Array(3) i
0: "red"
1: "green"
2: "blue"
length: 3
__proto__: Array(0)
```

The screenshot shows the Chrome DevTools Console. A red box highlights the word "red". An arrow points from this box to the index "0" in the code below.

```
red
> |
```

```
<script>
let selectedColors = ['red', 'green', 'blue'];
console.log(selectedColors[0]);
</script>
```

To access specific index



JavaScript – Function (Reference)

Objects | Array | Functions

What and why

*Function is set of statement that **perform a task** or **calculate and return a value***



Function

Declare a function

```
function greet(){  
    console.log('Hello World');  
}
```

Call a function

```
greet();
```

The screenshot shows a browser developer tools window titled "JavaScript". The console tab is selected, displaying the output "Hello World ex-9-function.html:9". The browser address bar shows the file path "/Users/macbookpro/Doc...". The bottom navigation bar includes "Console", "What's New", and a close button.



Function – Perform a task

Functions can have inputs. That inputs can change the behavior of the function

e.g. If we want to say “Hello John” instead of “Hello World”

*We have to add a **variable** to the function and use it in the output by concatenate*

```
function greet(name){  
  console.log('Hello ' + name);  
}
```

*Call a function by passing a value for name, So **john** is an argument, **Name** is a parameter of the **green function***

```
greet('john');
```

Functions can be reused.. NEXT Slide



Function – Perform a task

A function can be used many times by passing different values, So this same function can be used with two different values

```
function greet(name){  
    console.log('Hello ' + name);  
}
```

Value 01

```
greet('john');
```

Value 02

```
greet('Peter');
```

The screenshot shows a browser's developer tools open to the 'Console' tab. The console window has a dark background with light-colored text. It displays two log entries:

```
Hello john ex-10-function.html:9  
Hello Peter ex-10-function.html:9
```

Below the console window, the browser's main interface is visible, showing a dark header bar with tabs and a search bar.



Function – Perform a task

A function can have many parameters by adding comma ,

```
function greet(fname, lname){  
    console.log('Hello ' + fname + ' ' + lname);  
}
```

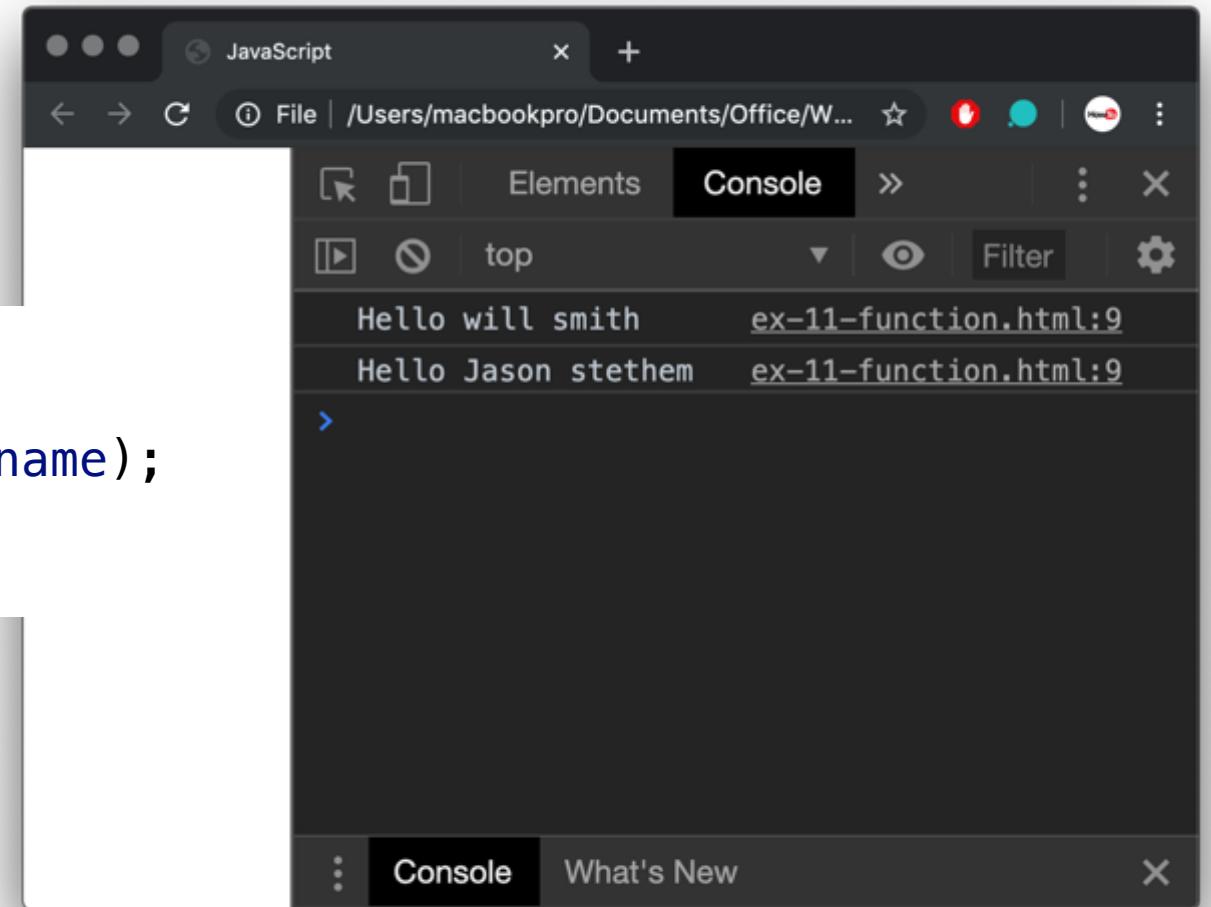
So we have to pass two values

```
greet('will', 'smith');
```

Second set of values

```
greet('Jason', 'stethem');
```

Note : If we missed passing second value *lname* will show as *undefined*





Function – Calculate

A function can have many parameters by adding comma,

```
function total(num1,num2){  
    let total = num1 + num2;  
    console.log(total);  
}
```

We have to pass two numbers

```
total(50,60);
```

The screenshot shows a browser's developer tools with the "Console" tab selected. The status bar at the top indicates it's a "JavaScript" tab. The URL bar shows "File | /Users/macbookpro/Do...". The console output area displays the following:

```
110 ex-13-function.html:10  
>
```

This indicates that the code was run at line 10 of the file "ex-13-function.html". The output "110" is shown in green, which typically represents a number or a primitive value in the console.

Note : If we missed passing one value, it will show as **undefined**



Common HTML Events

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an 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

[Fore more](#)



Create a JavaScript code to find total

1. Create a HTML document with inputs (number) for each and every field names as shown below
2. Convert and calculate the subject marks and store inside a variable
3. Display all the data including subject marks total & Average in the body

Marks 01

Marks 02

Total

Student Marks Cal

Name = Your name

Math = 98

Science = 88

English = 89

Total = ?

Average = ?



Employee Salary Calculation

Employee No : Enter by user EM3434

Employee Name : Enter by user YourName

Basic Salary : Enter by user 30k – 100k

Allowance : Enter by user 8000 – 15000

EPF : Enter by user < 5000

Loan : Enter by user < 3000

Gross Salary : BS + All ?

Deduction : EPF + Loan ?

Net Salary : GS – Ded ?

Bonus : ?

Bonus

IF NS >= 100000, 30% From BS

IF NS >= 80000, 20% From BS

IF NS >= 60000, 10% From BS

IF NS >= 40000, 8% From BS

IF NS >= 20000, 5% From BS

Else, 2% From BS

When GS button has been clicked change the GrossSalary text into Green color





Create a calculator

Marks 01

Marks 02

Total

```
<form name="formtotal" onsubmit="return(total())">
    <input type="number" name="txtmk1" placeholder="Marks 01"> <br>
    <input type="number" name="txtmk2" placeholder="Marks 02"> <br>
    <input type="submit" value="Total">
</form>

<script>
    function total(){
        let m1 = parseInt(document.formtotal.txtmk1.value);
        let m2 = parseInt(document.formtotal.txtmk2.value);

        let tot = m1 + m2;

        document.write("Total : "+tot);
    }
</script>
```



HTML Events with form validation

```
<form action="" name="myForm" onsubmit="return(validate());">
    <input type = "text" name = "EMail" />
    <input type = "submit" value = "Submit" />
</form>

function validate() {

    if( document.myForm.EMail.value == "" ) {
        alert( "Please provide your Email!" ); //Error msg
        document.myForm.EMail.focus() ; //Cursor selection
        return false;
    }
    return( true );
}
```



HTML Events with form validation

```
<form name="myForm" onsubmit="return validateForm()" method="post">
    <input type="text" name="fname">
    <input type="submit" value="Submit">
</form>
```

```
function validateForm() {
    let x = document.forms["myForm"]["fname"].value;
    if (x == "") {
        alert("Name must be filled out");
        return false;
    }
}
```



IF Condition

```
var age = 20;  
  
if( age > 18 ) {  
    document.write("<b>Qualifies for driving</b>");  
}  
else {  
    document.write("<b>Does not qualify for driving</b>");  
}
```



Switch Case

```
var grade = 'A';

document.write("Entering switch block<br />");
switch (grade)

{
    case 'A': document.write("Good job<br />");
    break;
    case 'B': document.write("Pretty good<br />");
    break;
    default:  document.write("Unknown grade<br />")

}
document.write("Exiting switch block");
```



While Loop

```
var count = 0;  
document.write("Starting Loop ");  
  
while (count < 10) {  
    document.write("Current Count : " + count + "<br />");  
    count++;  
}  
  
document.write("Loop stopped!");
```



For Loop

```
var count;  
  
document.write("Starting Loop" + "<br />");  
  
for(count = 0; count < 10; count++) {  
    document.write("Current Count : " + count );  
    document.write("<br />");  
}  
  
document.write("Loop stopped!");
```



Loop handling

```
var x = 1;  
  
document.write("Entering the loop<br /> ");  
  
while (x < 20) {  
    if (x == 5) {  
        break; // breaks out of loop completely  
    }  
    x = x + 1;  
    document.write( x + "<br />");  
}  
  
document.write("Exiting the loop!<br /> ");
```



Page redirecting

```
<form>
    <input type = "button" value = "Redirect Me" onclick = "Redirect();" />
</form>

function Redirect() {
    window.location = "https://www.bcas.lk";
}
```

onclick = "window.print()"



Show hide password



```
<form action="" onsubmit="return(f1())">  
    User Name <br>  
    <input type="text" name="" id="txtun"><br>  
    Password <br>  
    <input type="text" name="" id="txtpass"><br> <br>  
    <input type="submit" value="show/hide password" id="btnpass">  
</form>
```



```
<script>  
    function f1(){  
        var pass = document.getElementById("txtpass");  
        if(pass.type == "text"){  
            pass.type = "password";  
        }else{  
            pass.type = "text";  
        }  
        return false;  
    }  
</script>
```

User Name

Password

show/hide password



Changing HTML content

```
<p id="myP">I am a paragraph.</p>

<p>The content of "myP" is:</p>
<p id="demo"></p>

<script>
let html = document.getElementById("myP").innerHTML;
document.getElementById("demo").innerHTML = html;
</script>
```

```
<p id="demo">Click to delete.</p>
<button onclick="myFunction()">Delete</button>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "";
}
</script>
```

```
<p id="demo" onclick="myFunction()">Click me to change.</p>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "I have changed!";
}
</script>
```



getElement...

Finding HTML Elements

document.getElementById(<i>id</i>)	Find an element by element id
document.getElementsByTagName(<i>name</i>)	Find elements by tag name
document.getElementsByClassName(<i>name</i>)	Find elements by class name

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = "Hello";
</script>
```

```
<p id="demo"></p>
<script>
document.getElementsByTagName("p")[0].innerHTML = "Hello";
</script>
```

```
<p class="test"></p>
<p class="test"></p>
<script>
document.getElementsByClassName("test")[0].innerHTML = "Hello";
</script>
```



getElement...

```

<script>
document.getElementById("image").src = "pic_mountain.jpg";
</script>
```

```
<input type="text" id="myText" value="Hello">
<script>
document.getElementById("myText") .value = "Have a nice day!";
</script>
```



Changing CSS

```
document.getElementById(id).style.property = new style
```

```
<p id="demo">sample</p>
<script>
document.getElementById("demo").style.color = "red";
</script>
```

```
<p id="demo">sample</p>
<script>
document.getElementById("demo").style.fontSize = "40px ";
document.getElementById("demo").style.display = "none ";
</script>
```

.style.border
.style.color
.style.fontSize
.style.margin
.style.padding

...

For more Style Object

https://www.w3schools.com/jsref/dom_obj_style.asp



Trigger function

```
<button id="demo">Click me1</button>
<script>
document.getElementById("demo").addEventListener("click", myFunction);

function myFunction(){
    alert("Clicked!");
}
</script>
```

```
document.addEventListener("mouseover", myFunction);
document.addEventListener("click", someOtherFunction);
document.addEventListener("mouseout", someOtherFunction);
```



Trigger function

```
<button onclick="removeHandler()">Remove</button>

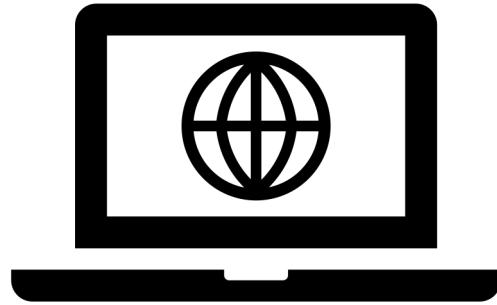
<p id="demo"></p>

<script>
document.addEventListener("mousemove", myFunction);

function myFunction() {
  document.getElementById("demo").innerHTML = Math.random();
}

function removeHandler() {
  document.removeEventListener("mousemove", myFunction);
}
</script>
```

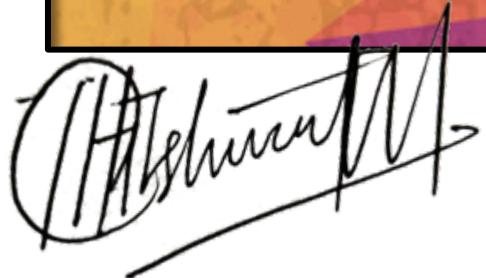
```
document.addEventListener("mouseover", myFunction);
document.addEventListener("click", someOtherFunction);
document.addEventListener("mouseout", someOtherFunction);
```



Q1 - Complete until JS Strings

www.w3schools.com

Course Pack 03 | Package 01



HowTo[®]



w3schools.com

BCAS

BCAS
CAMPUS
BRITISH COLLEGE OF APPLIED STUDIES