**JavaScript**

It’s popular because it’s already built in to your browser (Firefox, Explorer,)

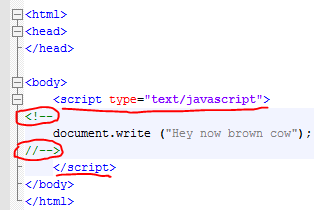
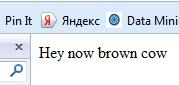
There are many scripting languages on the internet.

So we need explicitly tell the browser which scripting language we are going to use.

# 1. Beginner JavaScript Tutorial - 1 - Introduction to JavaScript

To start using JavaScript we need to add <script> </script> tags to the body. Everything that is in between those tags will be executed according to the JavaScript language.

For, example we don’t see “document.write” on the screen as it’s a command in the JavaScript language that was recognized by the browser.



If we think that some people may use older browsers that do not understand Java Script we may put a safety check.

Everything in between those signs older browsers will consider comment, while new browsers execute as a JavaScript program.

<!--

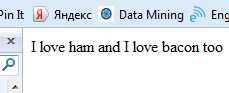
document.write ("Hey now brown cow");

//-->

# 2. Beginner JavaScript Tutorial - 2 - Comments and Statements

Adding one-line and multiple line comments

Every statement (command) has a semicolon at the end (;)

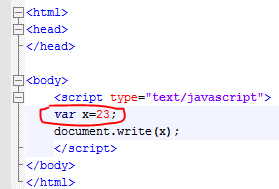
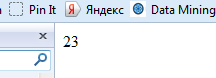


# 3. Beginner JavaScript Tutorial - 3 - Variables

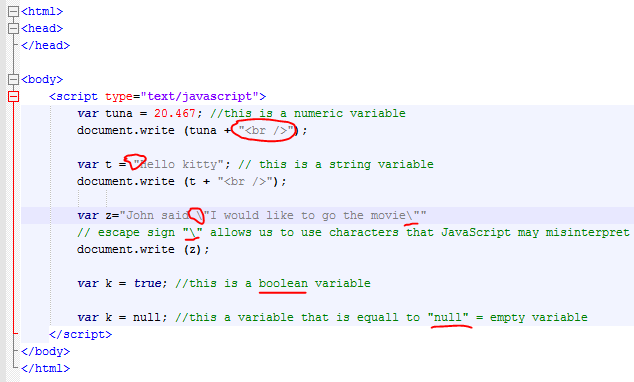
To define a variable we need to use “var”

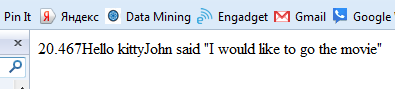
var x = 23;

* Variables are CASE SENSITIVE (X not equal to x)
* We need to start a variable name with a letter or underscore “\_”



# 4. Beginner JavaScript Tutorial - 4 - Different Types of Variables





# 5. Beginner JavaScript Tutorial - 5 - Using Variables with Strings

<html>

<head>

</head>

<body>

<script type="text/javascript">

var name = "Bucky";

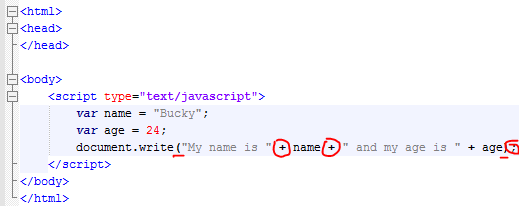
var age = 24;

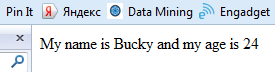
document.write("My name is " + name + " and my age is " + age);

</script>

</body>

</html>





# 6. Beginner JavaScript Tutorial - 6 - Functions

We can create a function and then call (execute) this function.

Functions have to be created inside of the JavaScript <script>.

To call a function you need to write the name of the function with () inside of the JavaScript.

Also the function may be called from the <body> but from, for example, forms.

If you write just “funky()” in the body it will be printed out on the screen as text.

In this example the function is called when a user presses a button on the screen.

<html>

<head>

</head>

<body>

<script type="text/javascript">

function funky (){

alert ("Ouch!") };

</script>

<form>

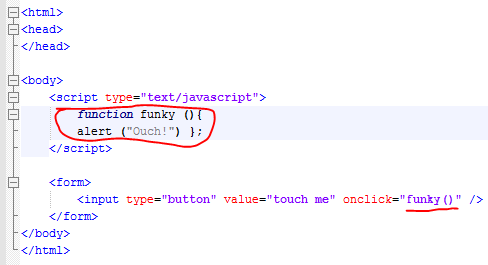
<input type="button" value="touch me" onclick="funky()" />

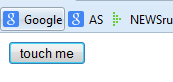
</form>

</body>

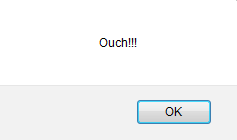
</html>

Alert -- creates a pop-up message on the screen.





Press the button and -->



# 7. Beginner JavaScript Tutorial - 7 - Using Parameters with Functions

We define a parameter for the function “love”. This parameter will be used by the alert function.

<html>

<head> </head>

<body>

<script type="text/javascript">

function love(x){alert("I love " + x);}

love("bacon");

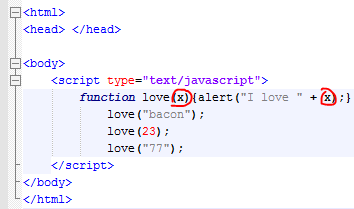
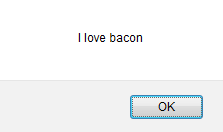
love(23);

love("77");

</script>

</body>

</html>



# 8. Beginner JavaScript Tutorial - 8 - Functions with Multiple Parameters

We can assign several parameters to a function.

To call this function we need to get all the parameters.

<html>

<head> </head>

<body>

<script type="text/javascript">

function better(one, two) {

document.write(one + " is better than " + two + "<br />"); }

better("day", "night");

better("rich", "poor");

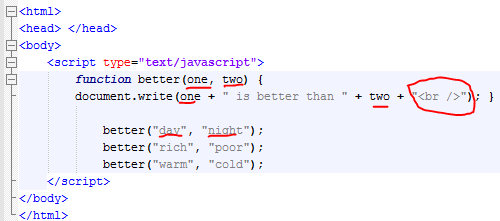
better("warm", "cold");

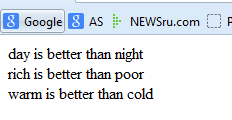
</script>

</body>

</html>

We can include a regular XHTML tag in “” to use it.





# 9. Beginner JavaScript Tutorial - 9 - The return Statement

<html>

<head> </head>

<body>

<script type="text/javascript">

function addNumbers (a,b) {

var z = a + b;

return z;}

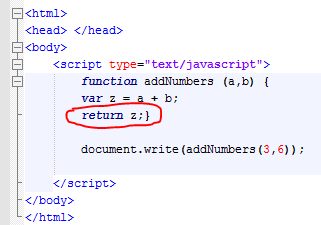
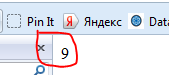
document.write(addNumbers(3,6));

</script>

</body>

</html>

We need to use the “return” function to print the value of z. Or we’ll get “undefined” on the screen.



# 10. Beginner JavaScript Tutorial - 10 - Calling a Function From Another Function

<html>

<head>

</head>

<body>

<script type="text/javascript">

function doFirst(){

document.write("I am the first function!");

}

function doSecond(){

document.write("222222222!");

}

function start(){

doFirst();

doSecond();

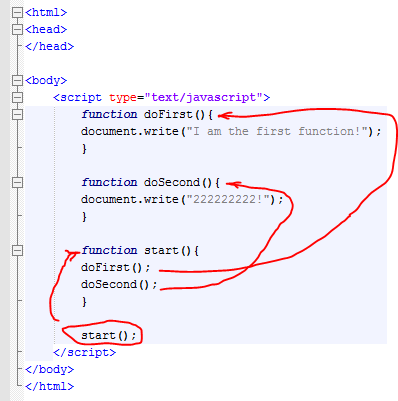
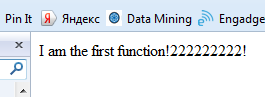
}

**start();**

</script>

</body>

</html>



# 11. Beginner JavaScript Tutorial - 11 - Global & Local Variables

Global variable – that was created outside of the function’s body

Local variable – that was created inside if the function

<html>

<head> </head>

<body>

<script type="text/javascript">

var girl = "Kelsey"; //Global variable outside the function

function spit(){

document.write(girl);

}

spit(); //calling the function

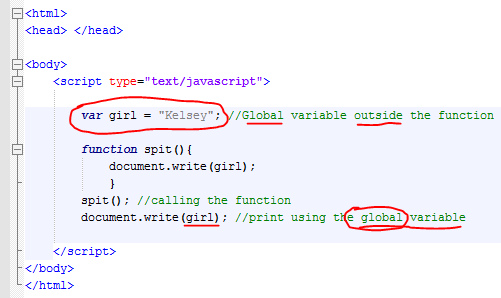
document.write(girl); //print using the global variable

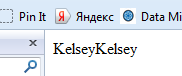
</script>

</body>

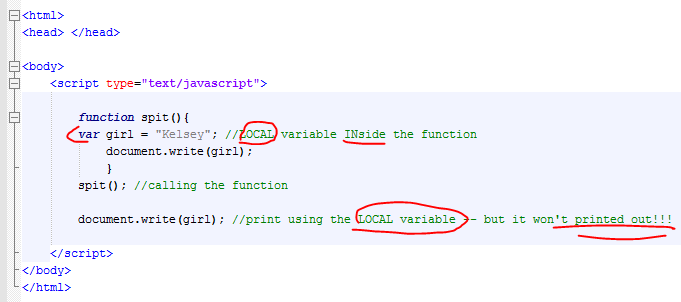
</html>

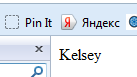
Global Variable



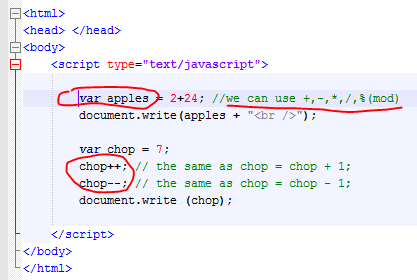


Local Variable





# 12. Beginner JavaScript Tutorial - 12 - Math Operators



# 13. Beginner JavaScript Tutorial - 13 - Assignment Operators

<html>

<head> </head>

<body>

<script type="text/javascript">

var bucky = 24;

bucky +=73; //adding 73 to the variable bucky

bucky -=15;

bucky \*=3;

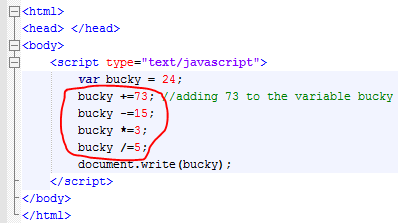
bucky /=5;

document.write(bucky);

</script>

</body>

</html>



# 14. Beginner JavaScript Tutorial - 14 - if Statement

<html>

<head> </head>

<body>

<script type="text/javascript">

var apples = 24;

var hotdog = 52;

if(apples == hotdog){

document.write("Checking the comparison");

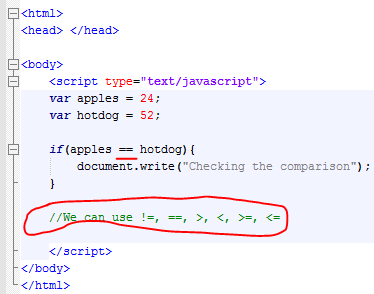
}

//We can use !=, ==, >, <, >=, <=

</script>

</body>

</html>



# 15. Beginner JavaScript Tutorial - 15 - if/else Statement

<html>

<head> </head>

<body>

<script type="text/javascript">

var blue = 12;

var red = 12;

**if(blue==red)**{

document.write("Variables are equal");

}**else**{

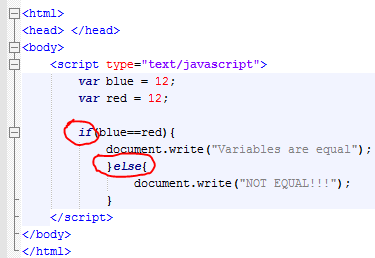
document.write("NOT EQUAL!!!");

}

</script>

</body>

</html>



# 16. Beginner JavaScript Tutorial - 16 - Nesting and Fridays!

The “if” statement is inside of another “if” statement

<html>

<head> </head>

<body>

<script type="text/javascript">

var firstName = "bucky1";

var lastName = "roberts1";

**if** (firstName == "bucky"){

**if** (lastName == "roberts"){

document.write("Hello Bucky!!!");

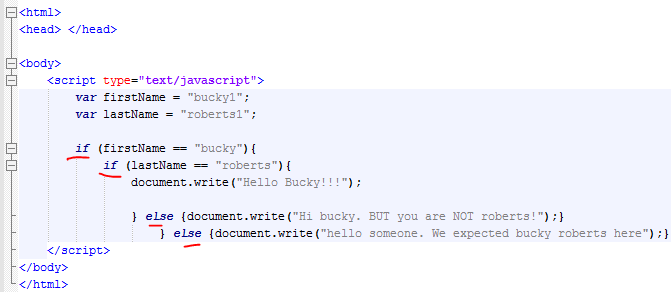
} **else** {document.write("Hi bucky. BUT you are NOT roberts!");}

} **else** {document.write("hello someone. We expected bucky roberts here");}

</script>

</body>

</html>



# 17. Beginner JavaScript Tutorial - 17 - Complex Conditions

Testing more than one condition.

Alternative way to nesting IF statements.

<html>

<head>

</head>

<body>

<script type="text/javascript">

var first = "bucky";

var last = "roberts";

if ( (first == "bucky") && (last == "roberts") ){

document.write("Hey!!! How are you doing?");

}

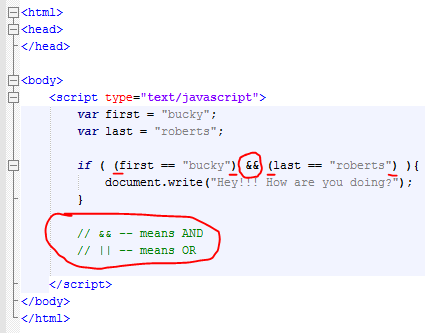
// && -- means AND

// || -- means OR

</script>

</body>

</html>



# 18. Beginner JavaScript Tutorial - 18 - switch

Switch statement takes a variable and tests it for all possible scenarios described before.

<html>

<head> </head>

<body>

<script type="text/javascript">

var girl = "Natalie";

**switch**(girl){

case("Natalie"):

document.write("You must like the Garden State movie");

break;

case("Ashley"):

document.write("What's up?");

break;

default:

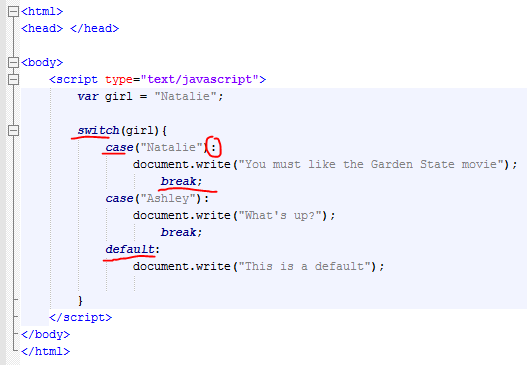
document.write("This is a default");

}

</script>

</body>

</html>



* Break helps not to search through other options once the matching option is found.
* After CASE there has to be a “:” NOT “;”.
* Default condition – if nothing matches it will be performed .

# 19. Beginner JavaScript Tutorial - 19 - for Loop

<html>

<body>

<script type="text/javascript">

for (x=0; x<10; x++){

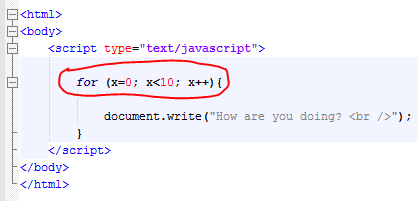
document.write("How are you doing? <br />");

}

</script>

</body>

</html>



# Beginner JavaScript Tutorial - 20 - while Loop

Hey Bucky, there's a DIFFERENCE between FOR and WHILE loop : for = DETERMINATE times, while = UNDETERMINATE times.

example:﻿ if you need to validate a password, you can not use FOR loop. With while loop you can validate infinite time's until the password can be right. (is just an example to explain the while loop)

something like that: var key=""; while (key!="password"){ key=prompt("wrong !!!"); } alert("finnaly!!!");

<html>

<body>

<script type="text/javascript">

var x = 1;

while(x<10){

document.write(x + " How do you like it? <br />");

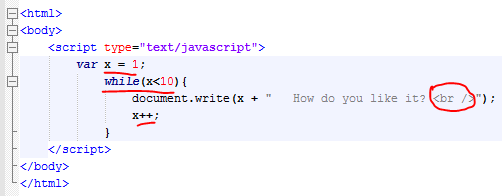
x++;

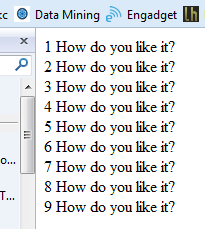
}

</script>

</body>

</html>





# 21. Beginner JavaScript Tutorial - 21 - do while

“Do While” works the same way as While loop does. But it first runs the statement and then check the condition.

<html>

<head> </head>

<body>

<script type="text/javascript">

var x=5;

**do** {

document.write(x + " Hi there!" + "<br />");

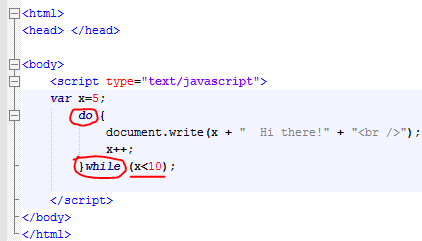
x++;

}**while** (x<10);

</script>

</body>

</html>



# 22. Beginner JavaScript Tutorial - 22 - Event Handlers

Event handlers can be inserted as JavaScript but in regular html code.

For example, in forms.

* Event handlers are “built in” functions. Your browser will automatically know that it’s JavaScript.
* Insert your statement using single quotes
* You can use several statements
* The command onClick is an event handler.

<html>

<head> </head>

<body>

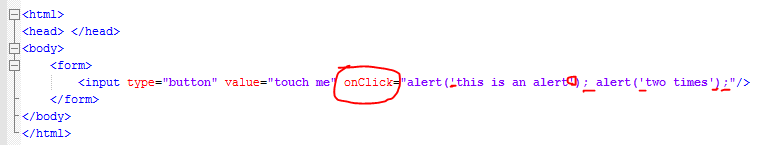
<form>

<input type="button" value="touch me" **onClick**="alert('this is an alert'); alert('two times');"/>

</form>

</body>

</html>



# 23. Beginner JavaScript Tutorial - 23 - onMouseOver & onLoad

**onMosueOver** – run an action when you hover your mouse over the link

<html>

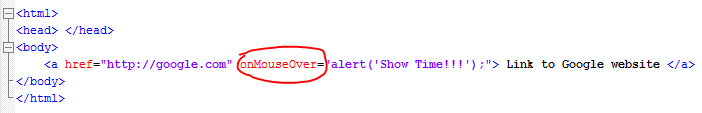
<head> </head>

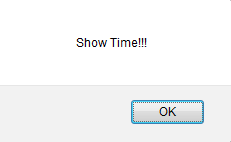
<body>

<a href="http://google.com" onMouseOver="alert('Show Time!!!');"> Link to Google website </a>

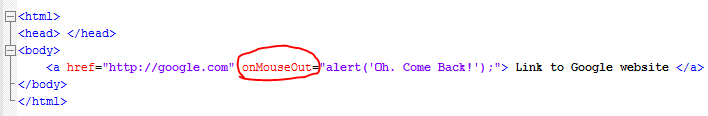
</body>

</html>





**onMouseOut** – run an action if you hover over the link, don’t click it, and move your mouse away.



**Event handlers for the <body>**

Whenever your webpage is downloading implement it.

<html>

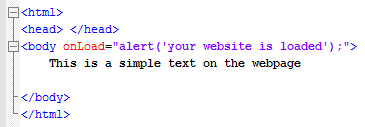
<head> </head>

<body **onLoad**="alert('your website is loaded');">

This is a simple text on the webpage

</body>

</html>



**onUnload -- R**un an event when you exit the webpage

<html>

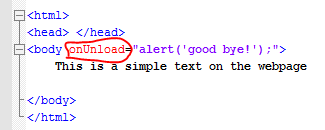
<head> </head>

<body **onUnload**="alert('good bye!');">

This is a simple text on the webpage

</body>

</html>



# 24. Beginner JavaScript Tutorial - 24 - Objects

Objects have their own properties and methods.

**If we need to describe a person those would be properties:**

Bucky:

* Eye: brown,
* Height: 170
* Weight: 180
* Car: blue

etc.

<html>

<head>

</head>

<body>

<script type="text/javascript">

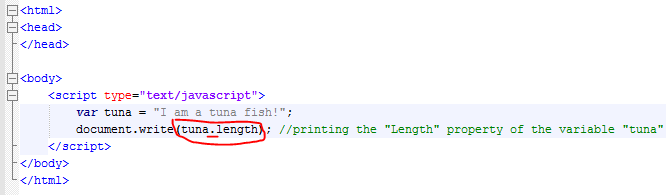
var tuna = "I am a tuna fish!";

document.write(**tuna.length**); //printing the "Length" property of the variable "tuna"

</script>

</body>

</html>



**Methods** are the sings that object does (Bucky):

* Drive a car
* Eat hotpockets
* Make video tutorials

document.write()

the “document” is an object itself.

“write” – is a method.

Object “Bucky”:

* Bucky.maketutorials()
* Bucky.driveacar()

There are some object that are already built in JavaScript.

In order to use them we need to mention the object name . and then mention the properties or actions.

# 25. Beginner JavaScript Tutorial - 25 - Creating Our Own Objects

Create a constructor function in the head

Creating and describing objects

If you need to print it out it should be in the body.

Creating a blueprint function (a constructor function) “person” that will have “age” and “name” as properties.

After we created a blueprint we can create many different objects (“bucky”, “taylor”).

<html>

<head>

<script type="text/javascript">

function person (name, age){

this.name = name;

this.age = age;

}

var bucky = new person ("Bucky Roberts", 24);

var taylor = new person ("Taylor Swift", 20);

</script>

</head>

<body>

<script type="text/javascript">

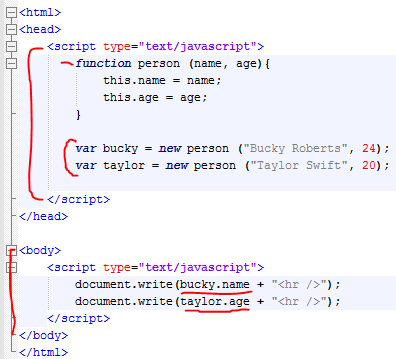
document.write(bucky.name + "<hr />");

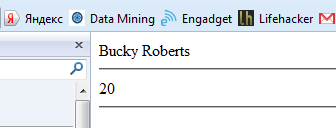
document.write(taylor.age + "<hr />");

</script>

</body>

</html>





# 26. Beginner JavaScript Tutorial - 26 - Object Initializers

Another way to create objects.

Not using a constructor function.

But using object initializers.

If you need to create one or two objects of the same type it’s better to use this method.

When we use object initializer we do not need to use “var”.

<!doctype html>

<html>

<head>

<script type="text/javascript">

bucky = {name: "Bucky Roberts", age: 24};

taylor = {name: "Taylor Swift", age: 20};

</script>

</head>

<body>

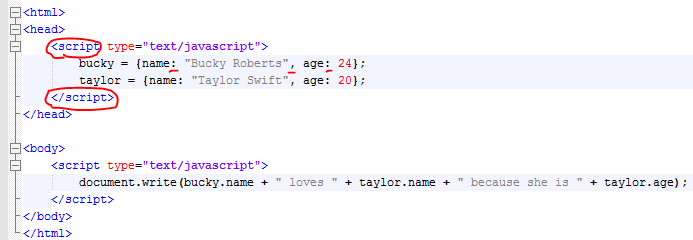
<script type="text/javascript">

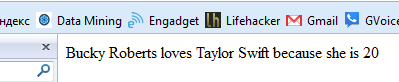
document.write(bucky.name + " loves " + taylor.name + " because she is " + taylor.age);

</script>

</body>

</html>





# 27. Beginner JavaScript Tutorial - 27 - Adding Methods to Our Objects

First, create a constructor function with all the properties

<html>

<head>

<script type="text/javascript">

function people (name, age){

this.name = name;

this.age = age;

this.yearsUntilRetire = yearsLeft;

}

//do not iclude (). That tells JavaScript that this method(yearsLeft) is a part of the object (people)

//because it has no paranthesis ()

function yearsLeft(){

var numYears = 65 - this.age;

return numYears;

}

var natalie = new people ("Natalie Portman", 28);

var bucky = new people ("Bucky Roberts", 24);

</script>

</head>

<body>

<script type="text/javascript">

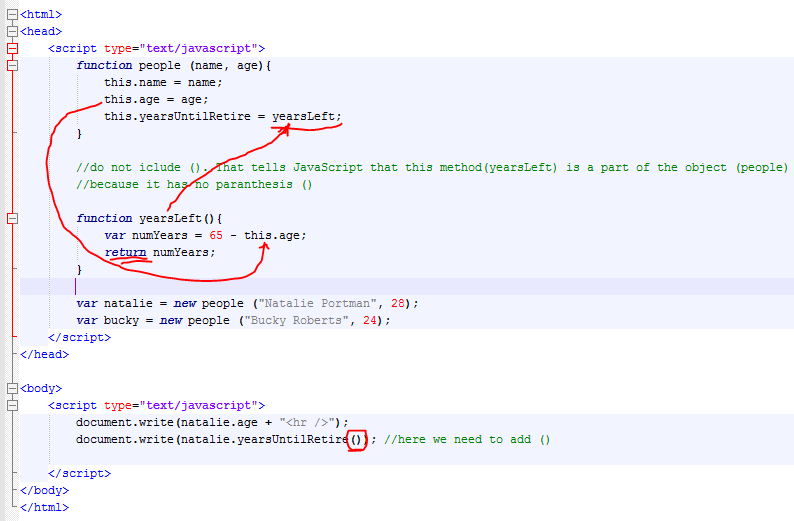
document.write(natalie.age + "<hr />");

document.write(natalie.yearsUntilRetire()); //here we need to add ()

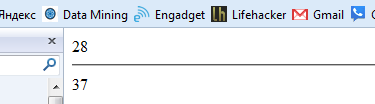
</script>

</body>

</html>



Adding the function without parameters



# 28. Beginner JavaScript Tutorial - 28 - Arrays

So, we can keep may values at one variable.

Creating an array is similar to creating a new object.

<html>

<head> </head>

<body>

<script type="text/javascript">

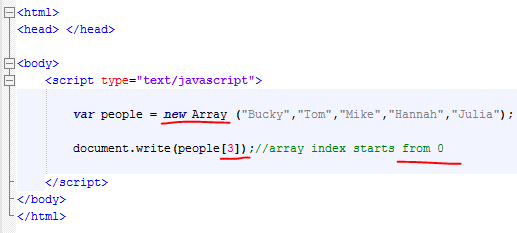
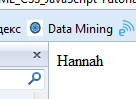
var people = new Array ("Bucky","Tom","Mike","Hannah","Julia");

document.write(people[3]);//array index starts from 0

</script>

</body>

</html>



# 29. Beginner JavaScript Tutorial - 29 - Other Ways to Create Arrays

<html>

<head> </head>

<body>

<script type="text/javascript">

var things = new Array (3);

things[0] = "jersey shore";

things[1] = "brian regan";

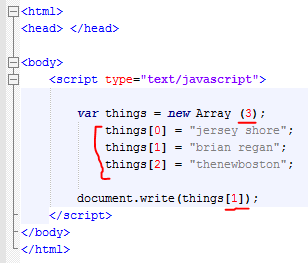
things[2] = "thenewboston";

document.write(things[1]);

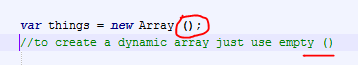
</script>

</body>

</html>



Creating a dynamic array. If we don’t know how many items it will finally store.



# 30. Beginner JavaScript Tutorial - 30 - Array Properties and Methods

Arrays are objects.

And every object has PROPERTIES such as “length”

And METHODS such as “concat” (takes two arrays and sticks them together)

<html>

<head> </head>

<body>

<script type="text/javascript">

var dudes = new Array ("Robert","John","Bob");

var chicks = new Array ("Julia","Rebecca","Hannah");

document.write(dudes.length + "<hr />"); //printing the Dudes Array length

var people = dudes.concat(chicks);// adding two arrays together

for (i=0; i<people.length; i++){

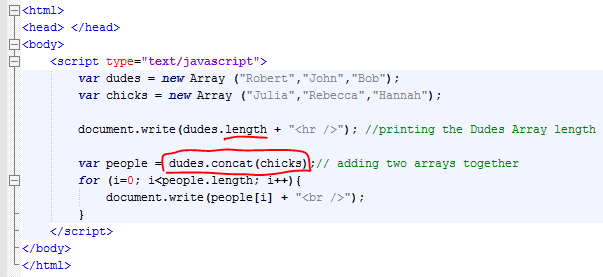
document.write(people[i] + "<br />");

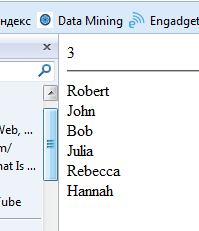
}

</script>

</body>

</html>





# 31. [Beginner JavaScript Tutorial - 31 - join and pop](http://www.youtube.com/watch?v=rhnqA9kmFRE&feature=edu&list=PL46F0A159EC02DF82)

**Join** method takes an array and converts it to a string.

POP – removes the last element from an array.

<html>

<head> </head>

<body>

<script type="text/javascript">

var movies = new Array ("Avatar", "Goodwill Hunting", "Vanilla Sky", "Fight Club");

//join creates a string from an array

var string1 = movies.join();//()by default no parameters -- means items will be separated by ","

document.write(string1 + "<br />");

var string2 = movies.join(" - ");////items will be separated by " - "

document.write(string2 + "<hr />");

document.write(movies[3] + "<br />");

movies.pop();// the last element in the array was deleted

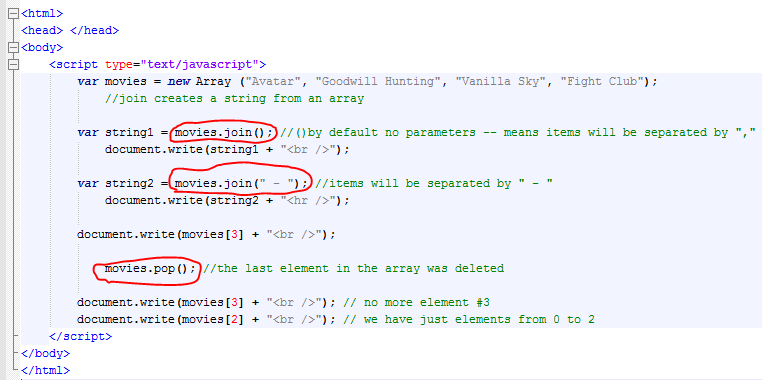
document.write(movies[3] + "<br />"); // no more element #3

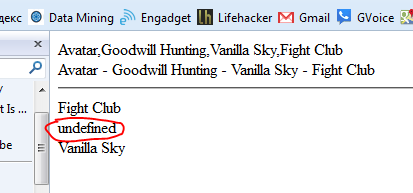
document.write(movies[2] + "<br />"); // we have just elements from 0 to 2

</script>

</body>

</html>





# 32. Beginner JavaScript Tutorial - 32 - reverse, push, sort

* **REVERSE** – reverse the order of items in the array (first becomes last and so forth)
* **PUSH** -- ads one or more elements to your array
* **SORT** -- sorts items in the alphabetical order

<html>

<head> </head>

<body>

<script type="text/javascript">

var bp = new Array ("Head", "Shoulders","Knees","Toes");

var string1 = bp.join(" + ");

document.write(string1 + "<br />");

bp.reverse(); //reverse the order of items in the array

var string1 = bp.join(" + ");

document.write(string1 + "<br />");

bp.push("Tongue","Liver"); //adds one or more elements to the end of the array

var string1 = bp.join(" + ");

document.write(string1 + "<br />");

bp.sort(); //sort items in the alphabetical order

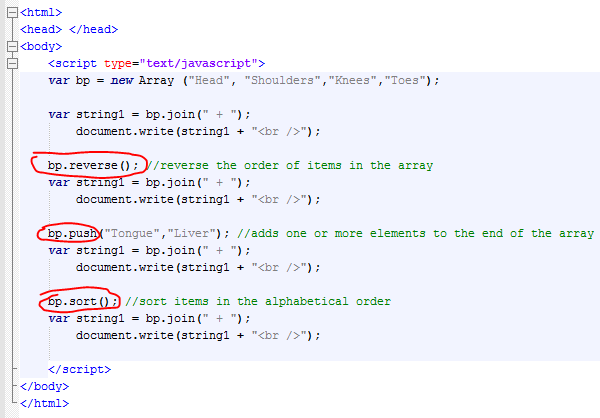
var string1 = bp.join(" + ");

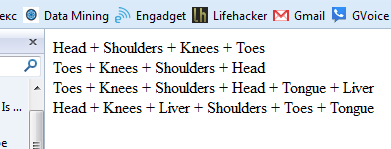
document.write(string1 + "<br />");

</script>

</body>

</html>





# 33. Beginner JavaScript Tutorial - 33 - Add Array Elements Using a Loop

<html>

<head></head>

<body>

<script type="text/javascript">

//the PROMPT function is built in JavaScript

//it takes two parameters: prompting message and the default value

//if we do not want to use default value we still HAVE to use ""

var pie = prompt ("Enter your name:","");

document.write("Hello " + pie);

var misc = new Array(3);

for (i=0; i<3; i++){

misc[i] = prompt ("Enter you item:","");

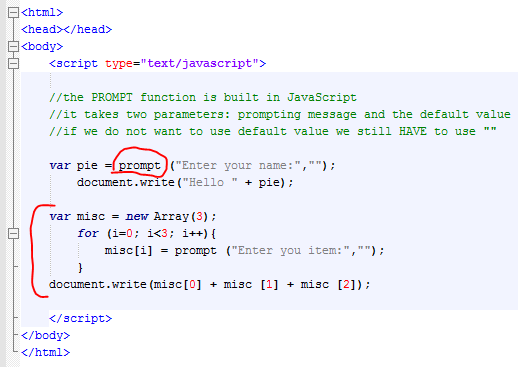
}

document.write(misc[0] + misc [1] + misc [2]);

</script>

</body>

</html>



# 34. Beginner JavaScript Tutorial - 34 - Cool Technique to Print Array Elements

<html>

<head> </head>

<body>

<script type="text/javascript">

var food = new Array ("Apple","Orange","Tuna","Bacon","Ham");

for (i=0; i<food.length; i++){

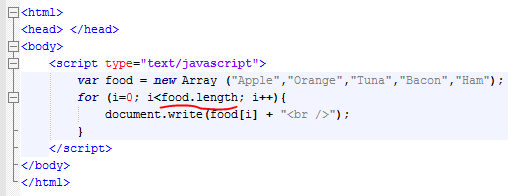
document.write(food[i] + "<br />");

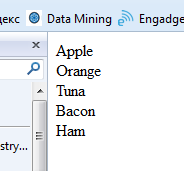
}

</script>

</body>

</html>





# 35. Beginner JavaScript Tutorial - 35 - Associative Arrays

<html>

<head> </head>

<body>

<script type="text/javascript">

var bucky = new Array ();

bucky ["color"] = "blue";

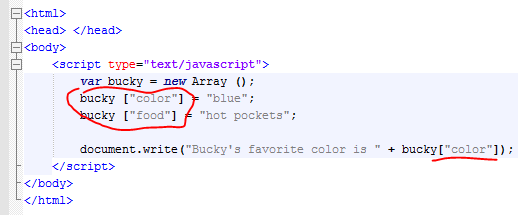
bucky ["food"] = "hot pockets";

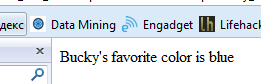
document.write("Bucky's favorite color is " + bucky["color"]);

</script>

</body>

</html>





# 36. Beginner JavaScript Tutorial - 36 - Math Objects

The Math Object is built in.

Look on-line for the list of all built in methods

<html>

<head> </head>

<body>

<script type="text/javascript">

document.write("The value of PI is " + Math.PI + "<br />");

document.write("The value of E is " + Math.E + "<hr />");

//let's find the square root fo a number

var x = prompt ("Enter your number","");

var y = Math.sqrt(x);

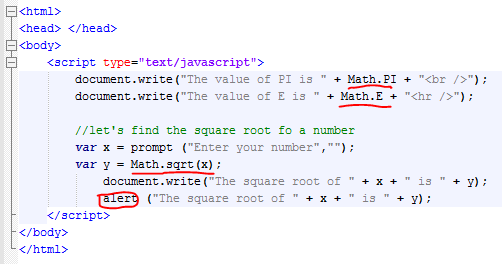
document.write("The square root of " + x + " is " + y);

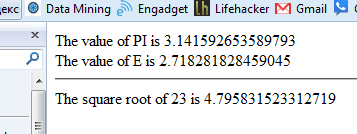
alert ("The square root of " + x + " is " + y);

</script>

</body>

</html>





# 37. Beginner JavaScript Tutorial - 37 - Date Objects

<html>

<head> </head>

<body>

<script type="text/javascript">

//function doSomething (){document.write("Tuna");}

// setInterval("doSomething()",1000);

function printTime(){

var now = new Date();// writes current date and time to the variable "now"

var hour = now.getHours();

var min = now.getMinutes();

var sec = now.getSeconds();

document.write(hour+":"+min+":"+sec+"<br />");

}

//printTime();

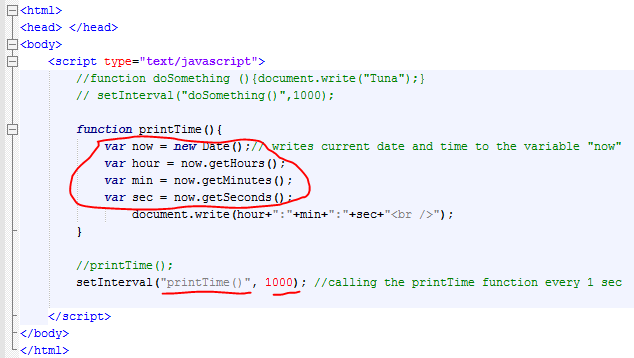
setInterval("printTime()", 1000); //calling the printTime function every 1 sec

</script>

</body>

</html>

**Works in Chrome only**



# 38. Beginner JavaScript Tutorial - 38 - Accessing Forms

JavaScript automatically puts forms into the Forms Array

In our example we created just one form. It will be put in the Form Array with the index [0].

<html>

<head> </head>

<body>

<form>

Username: <input type="text"/>

Password: <input type="password"/>

<input type="submit" value="Submit!"/>

</form>

<script type="text/javascript">

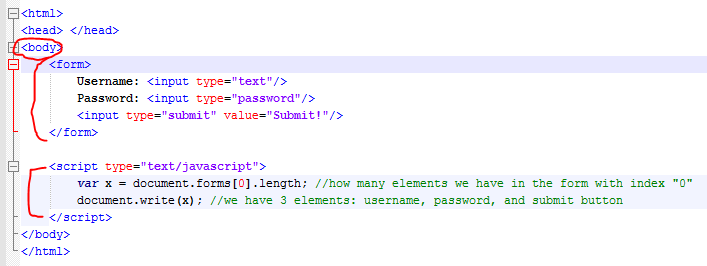
var x = document.forms[0].length; //how many elements we have in the form with index "0"

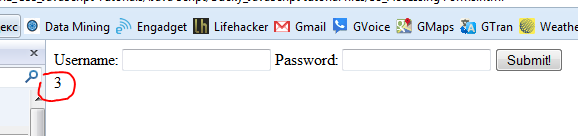
document.write(x); //we have 3 elements: username, password, and submit button

</script>

</body>

</html>





# 39. Beginner JavaScript Tutorial - 39 - Accessing Form Elements

<html>

<head> </head>

<body>

<form name="buckysForm">

Username: <input type="text" name="UN"/>

Password: <input type="password" name="MYPASS"/>

<input type="submit" value="Submit!"/>

</form>

<script type="text/javascript">

var x = document.forms[0].elements[0].name; //one way to access the elements value

document.write(x + "<br />");

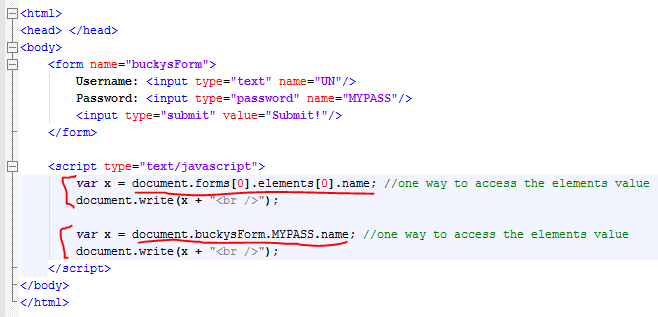
var x = document.buckysForm.MYPASS.name; //one way to access the elements value

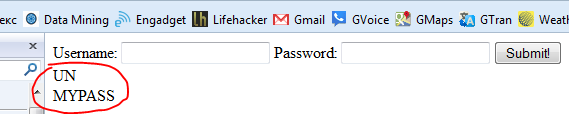
document.write(x + "<br />");

</script>

</body>

</html>





# 40. Beginner JavaScript Tutorial - 40 - Simple Form Validation

<html>

<head>

<script type="text/javascript">

function validator(){

if (document.buckysForm.thebox.checked)

alert("Yes, it's checked!");

else

alert("No, please check the checkbox");

}

</script>

</head>

<body>

<form name="buckysForm">

<input type="checkbox" name="thebox" />

<input type="button" value="Press Me!" onClick="validator()" />

</form>

</body>

</html>



