JavaScript Syntax

Web Authoring and Design

Benjamin Kenwright

Milestone Dates

- Demonstrate Coursework 1
  - Friday (15th December)
  - 10 Minutes Each

- Coursework 2 (Group Project)
  - Website (XMAS)
  - Javascript Game (xmas theme)
  - Team 2-3 People
  - 29th December (Friday)
  - Demonstrate/Present Group Work

Group Names

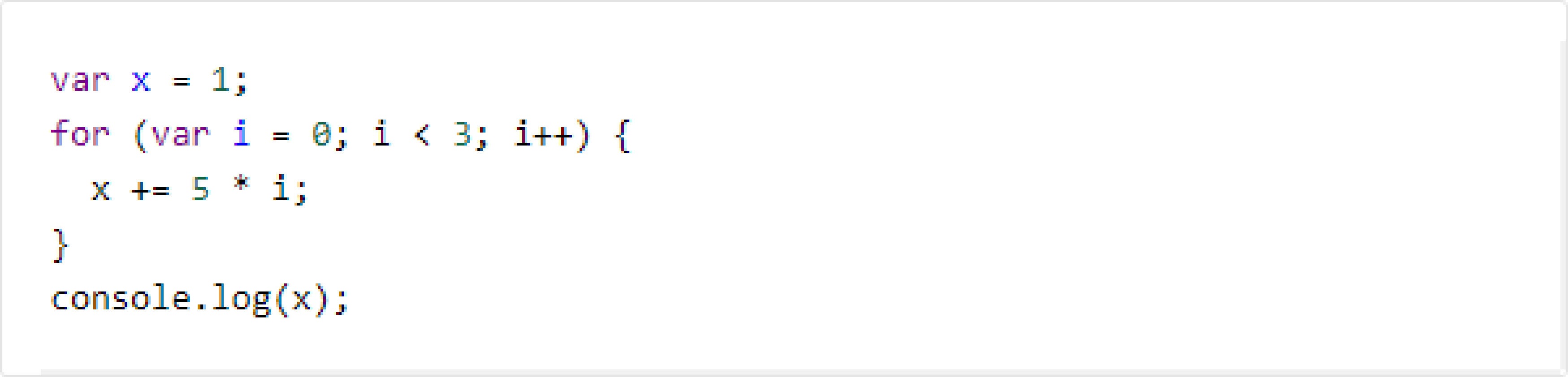
- Let me know who your groups are
- 2-3 People

Outline

- Review
- Fundamental Syntax Javascript
- Summary
- Review/Discussion

Revision Question

- What will the following code print?



```
var x = 1;
for (var i = 0; i < 3; i++) {
  x += 5 * i;
}
console.log(x);
```

a) 1

b) 10

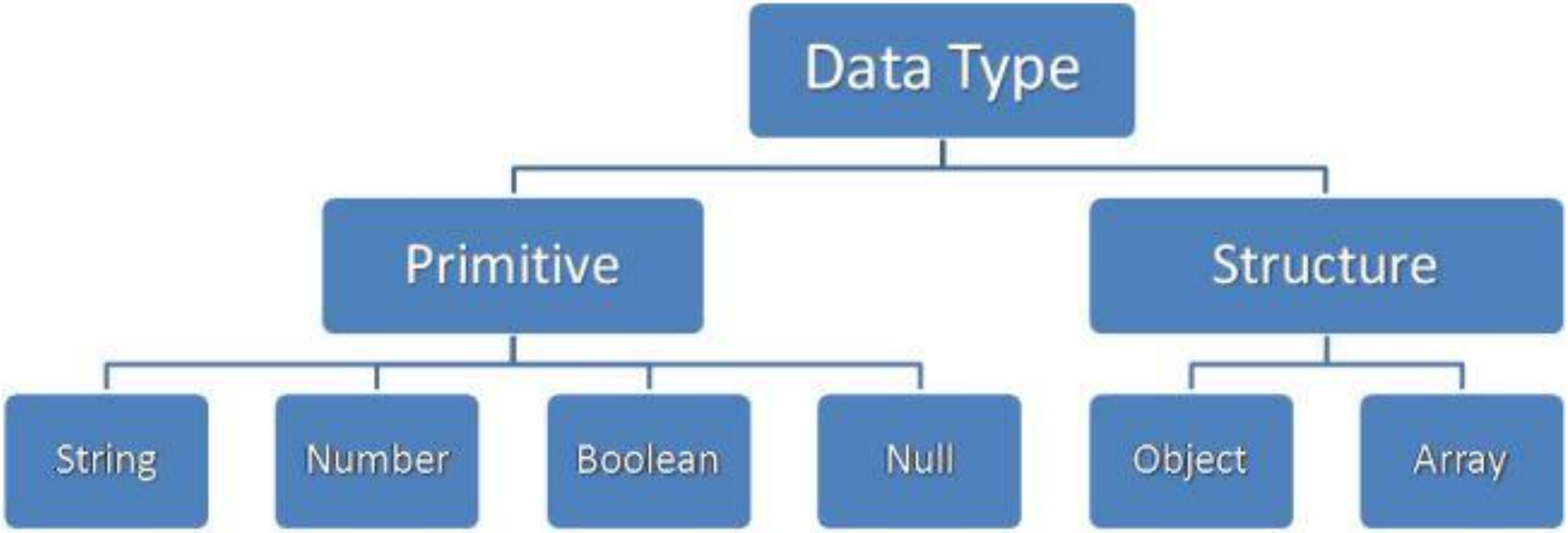
c) 16

d) 31

Answer

- c) 16

JavaScript Data Types



Primary Data Types (Primitive)

- String
- Number
- Boolean

Composite Data Types (reference)

Object

Array

Special Data Types

Null

Undefined

Overview of Types

String : A strings of any characters enclosed in quotes is a string primitive.

Number : A number by itself is a number primitive, all numbers in JavaScript are 64-bit floating-point numbers.

Boolean : It will store Variables or expressions which are either true or false.

Object: A collections of properties and methods. In Javascript, anything that is not a primitive is an Object.

Arrays : Arrays are regular objects, both objects and arrays can have properties and methods.

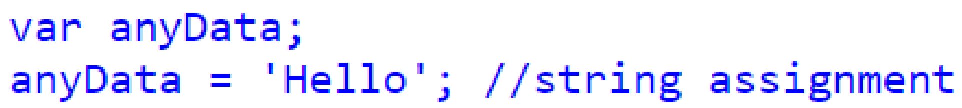
Null : It has only one value in JavaScript: null.

Undefined : It treated as a variable that has been declared, but has never had a value assigned to it.

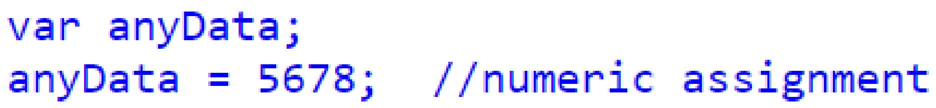
Var in Javascript

```
var anyData;
```

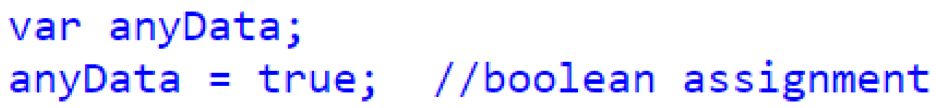
Example var



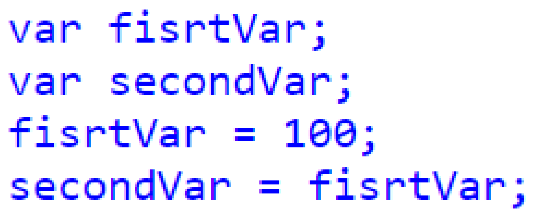
```
var anyData;
anyData = 'Hello'; //string assignment
```



```
var anyData;
anyData = 5678;  //numeric assignment
```



```
var anyData;
anyData = true;  //boolean assignment
```

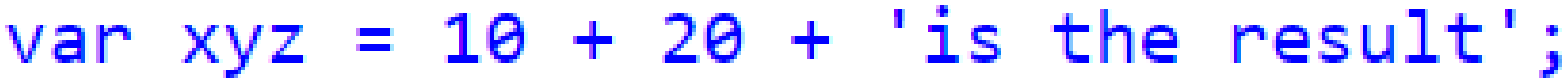


```
var fisrtVar;
var secondVar;
fisrtVar = 100;
secondVar = fisrtVar;
```

Question

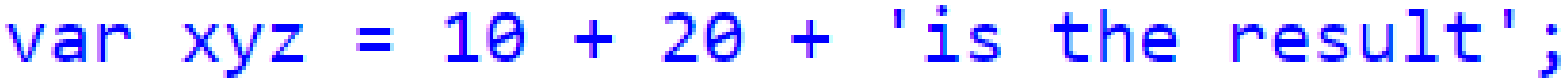
Combining Data Types

What would be in variable xyz?

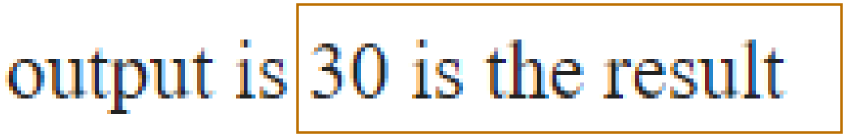


```
var xyz = 10 + 20 + 'is the result';
```

Answer



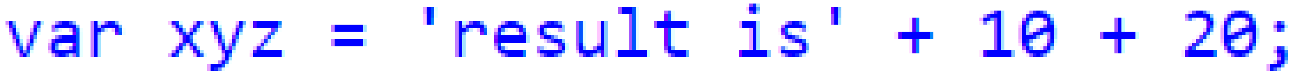
```
var xyz = 10 + 20 + 'is the result';
```



```
output is 30 is the result
```

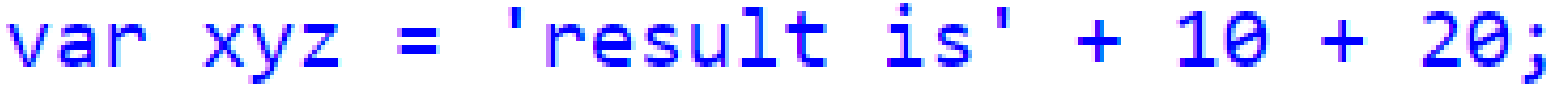
Question

What would be in variable xyz this time?

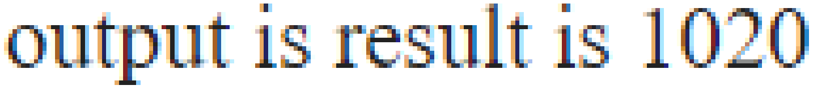


```
var xyz = 'result is' + 10 + 20;
```

Answer



```
var xyz = 'result is' + 10 + 20;
```



output is result is 1020

Scope of JavaScript Variables

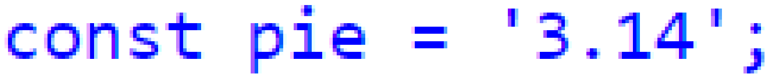
The Javascript variable has only **two** scopes, Global and Local.

When you declare a variable within your function , its scope is local, that means you can use that variable only inside that function.

If a variable declare outside the function, its scope is Global.

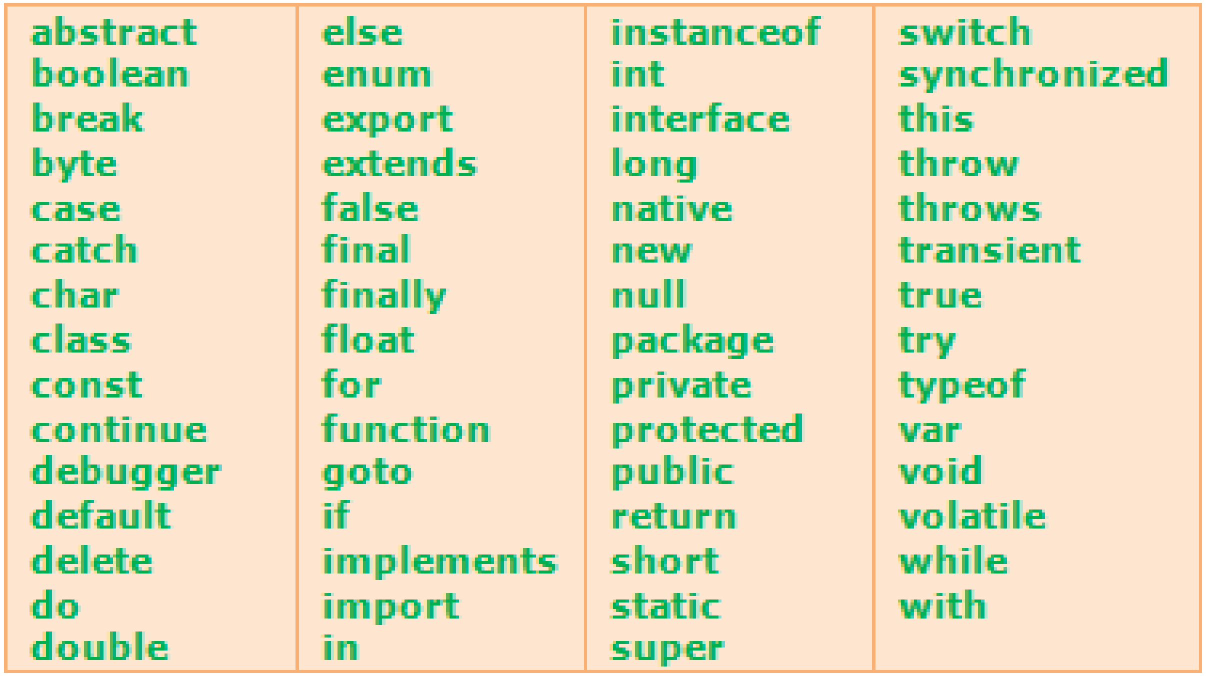
Constants

Javascript you can define read-only Constant variables with const keyword



```
const pie = '3.14';
```

JavaScript keywords



abstract
boolean
break
byte
case
catch
char
class
const
continue
debugger
default
delete
do
double
else
enum
export
extends
false
final
finally
float
for
function
goto
if
implements
import
in
instanceof
int
interface
long
native
new
null
package
private
protected
public
return
short
static
super
switch
synchronized
this
throw
throws
transient
true
try
typeof
var
void
volatile
while
with

Question

- How many Javascript keywords could you write down?

Question

- How many scope levels does Javascript have?

a) one

b) two

c) three

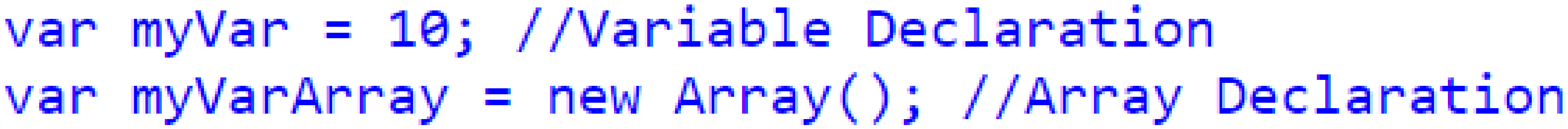
d) unlimited

Answer

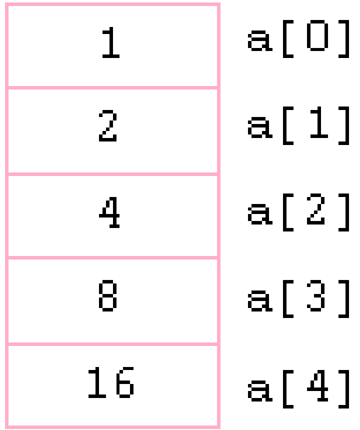
b) two

The Javascript variable has only two scopes, Global and Local.

JavaScript Array Object



```
var myVar = 10; //Variable Declaration
var myVarArray = new Array(); //Array Declaration
```

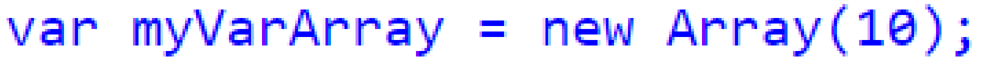


1	a[0]
2	a[1]
4	a[2]
8	a[3]
16	a[4]

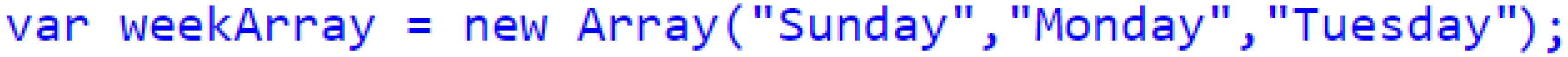
Constructors

Already know the number of items that would store in an array, you can pass the number to the constructor of an Array Object

Directly initialize the Array elements when declaring an Array Object



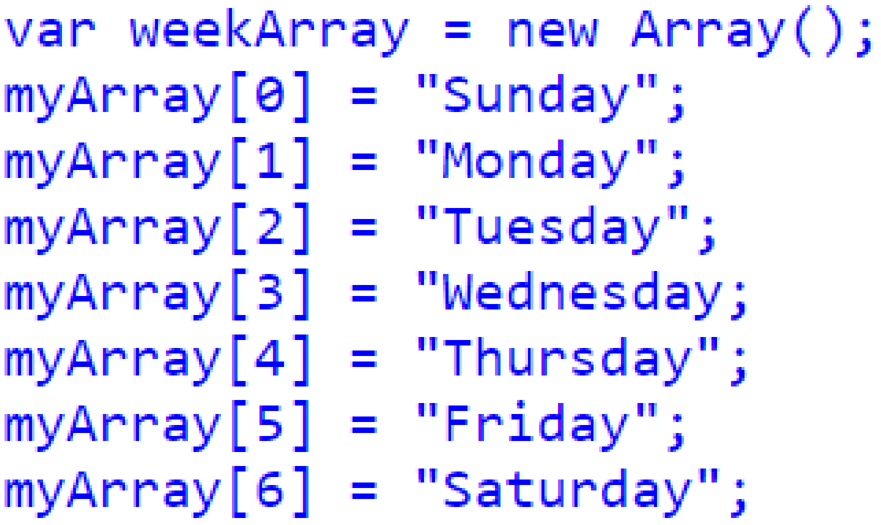
```
var myVarArray = new Array(10);
```



```
var weekArray = new Array("Sunday","Monday","Tuesday");
```

From 0...

JavaScript array indexes start from 0 (zero) and continue as 1,2,3 etc..

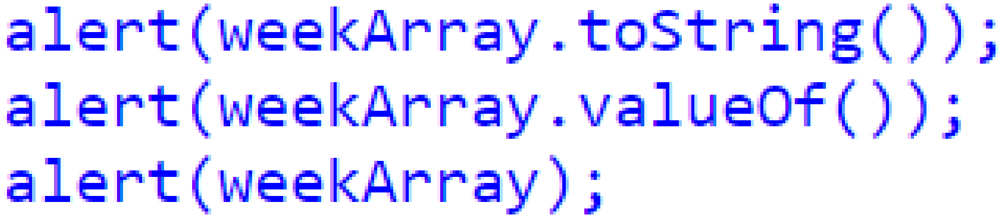


```
var weekArray = new Array();
myArray[0] = "Sunday";
myArray[1] = "Monday";
myArray[2] = "Tuesday";
myArray[3] = "Wednesday;
myArray[4] = "Thursday";
myArray[5] = "Friday";
myArray[6] = "Saturday";
```

Displaying

Displaying Javascript Array elements

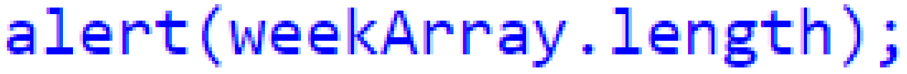
You can display Array elements in several ways. The following methods will show all elements from an Array.



```
alert(weekArray.toString());
alert(weekArray.valueOf());
alert(weekArray);
```

Javascript Array Length

Length property of Javascript Array return the number of elements contains in an Array. Javascript Array index starts from 0 so the length of an array is equal to the last index + 1



```
alert(weekArray.length);
```

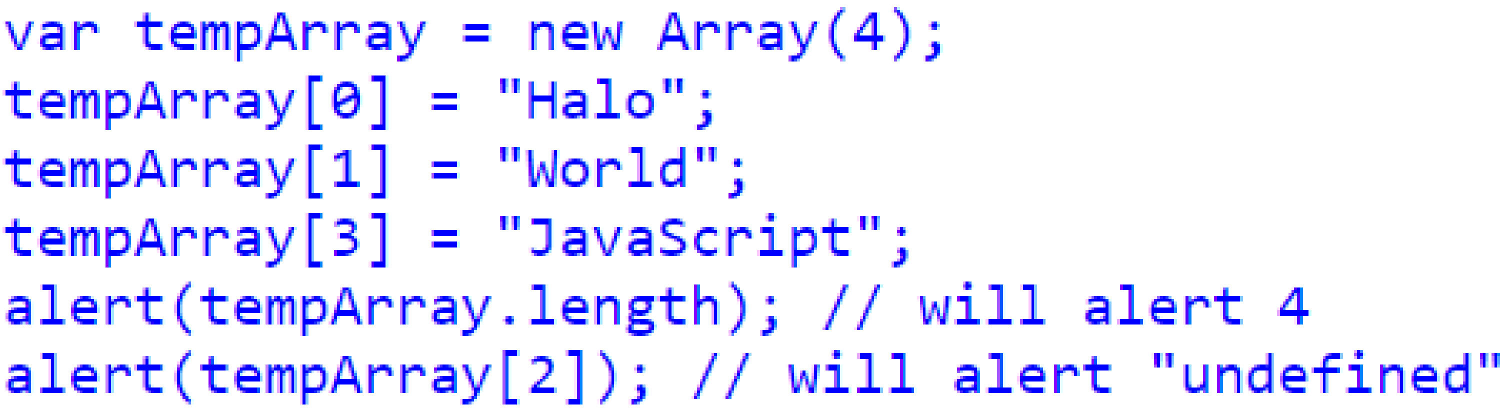
Arrays Undefined Values

Arrays are automatically growing and dynamically sized to accommodate any type of data that is added to them

Add or remove elements from an array, the length of the array will change as needed

Declare an Array with constructor, each slot will be set to "undefined"

Example



```
var tempArray = new Array(4);
tempArray[0] = "Halo";
tempArray[1] = "World";
tempArray[3] = "JavaScript";
alert(tempArray.length); // will alert 4
alert(tempArray[2]); // will alert "undefined"
```

Other Array Methods

Array.push()

Array.contains()

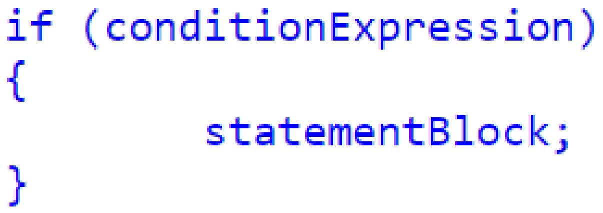
Array.sort()

Array.splice()

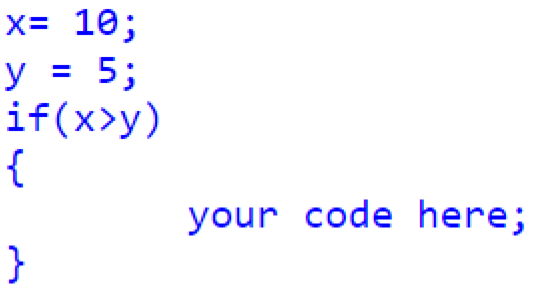
Array.slice()

Array.join()

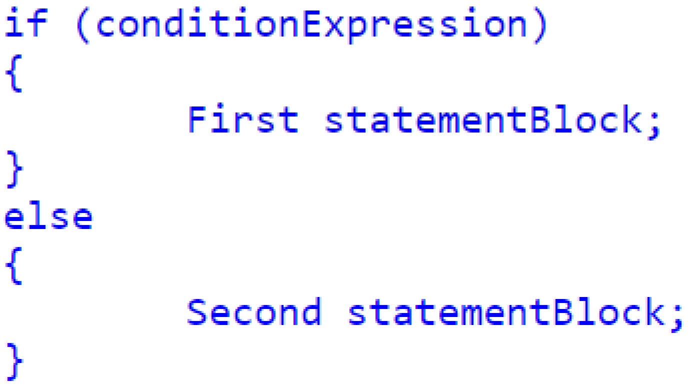
Conditional Statements



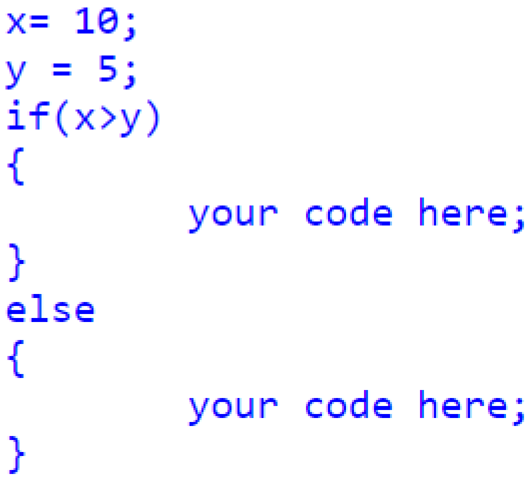
```
if (conditionExpression)
{
         statementBlock;
}
```



```
x= 10;
y = 5;
if(x>y)
{
        your code here;
}
```

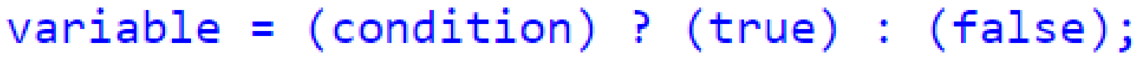


```
if (conditionExpression)
{
        First statementBlock;
}
else
{
        Second statementBlock;
}
```

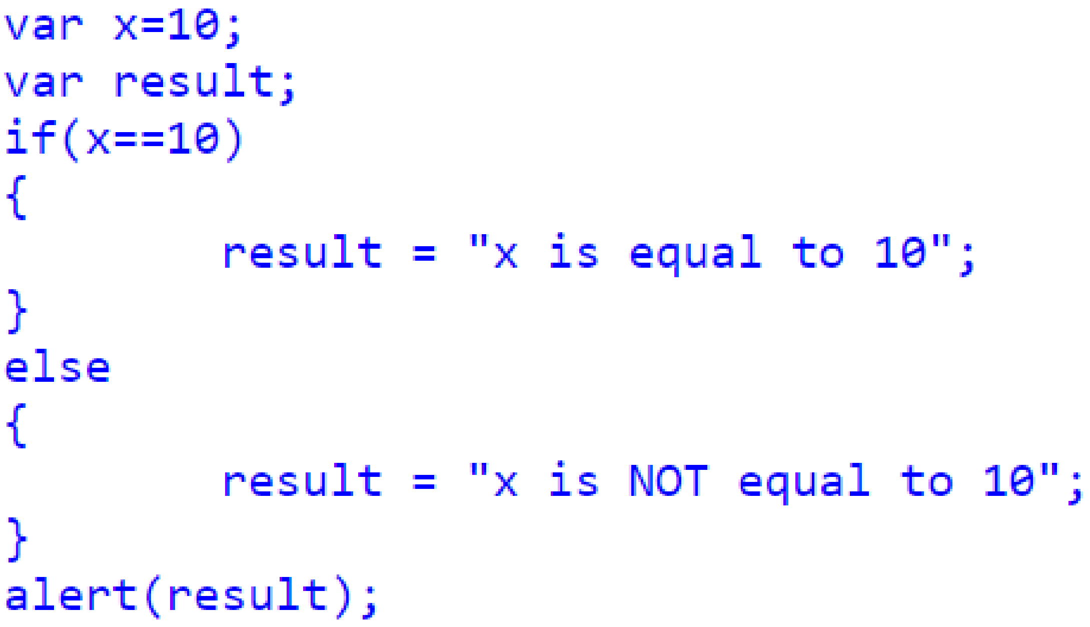


```
x= 10;
y = 5;
if(x>y)
{
        your code here;
}
else
{
        your code here;
}
```

Alternative if..else Syntax (? Ternary Operator)

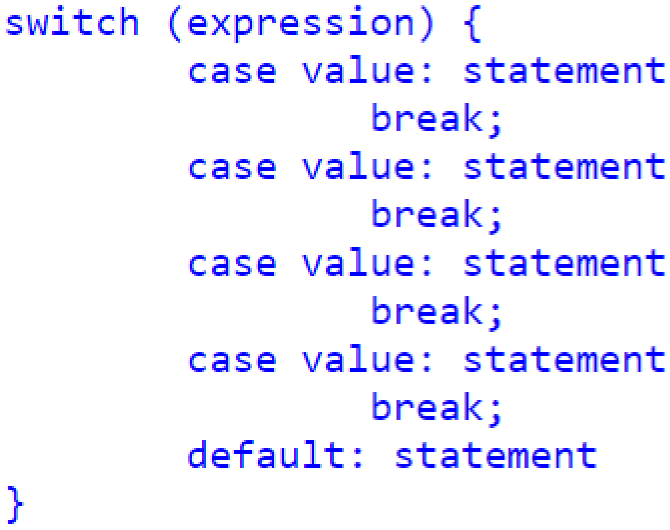


```
variable = (condition) ? (true) : (false);
```

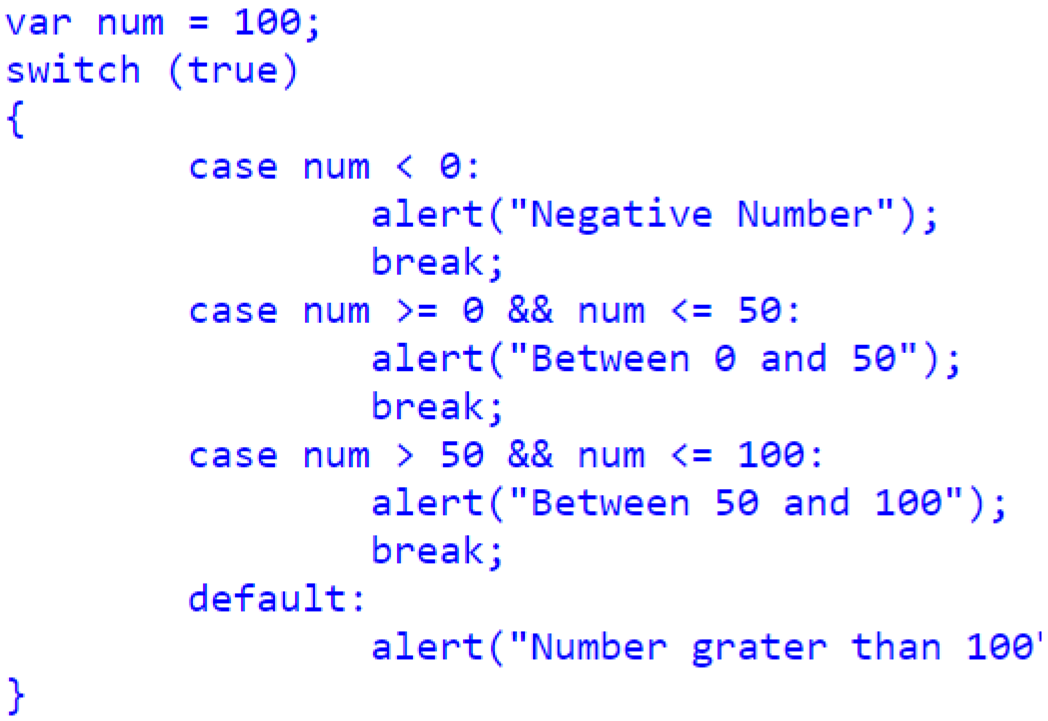


```
var x=10;
var result;
if(x==10)
{
        result = "x is equal to 10";
}
else
{
        result = "x is NOT equal to 10";
}
alert(result);
```

Switch Statements

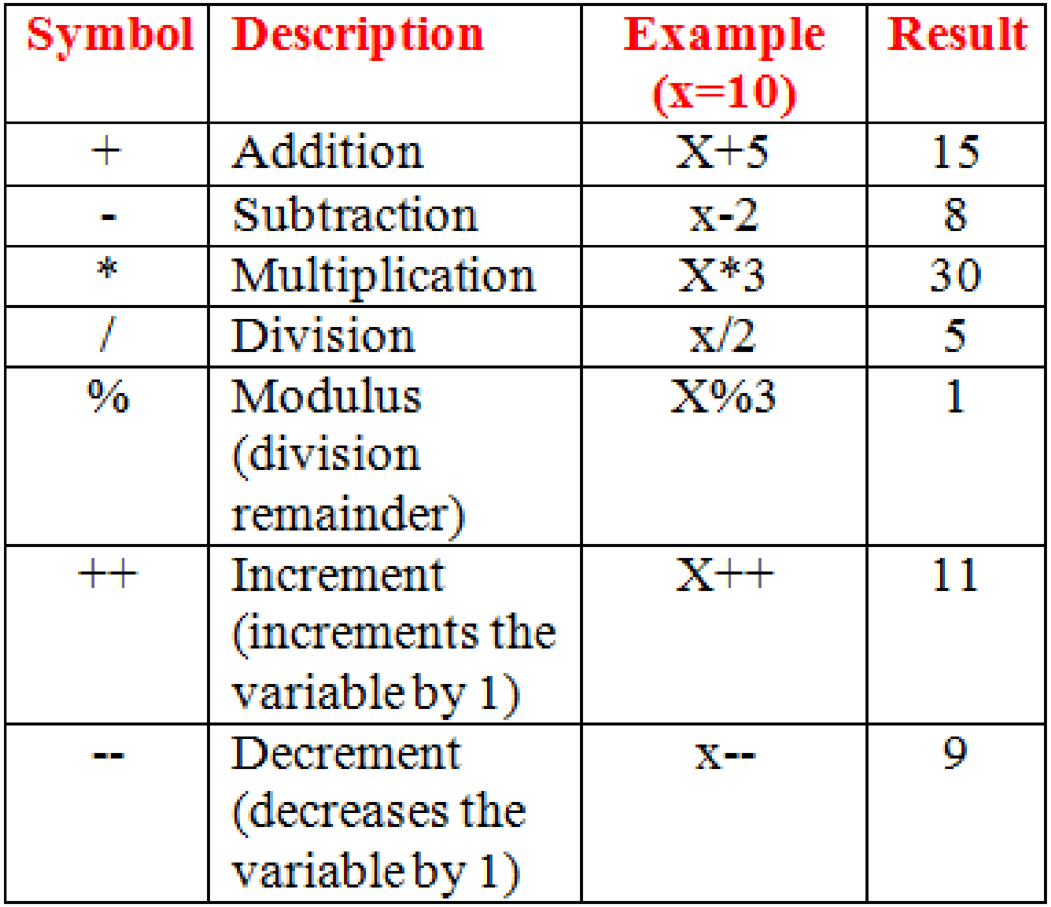


```
switch (expression) {
        case value: statement
                break;
        case value: statement
                break;
        case value: statement
                break;
        case value: statement
                break;
        default: statement
}
```



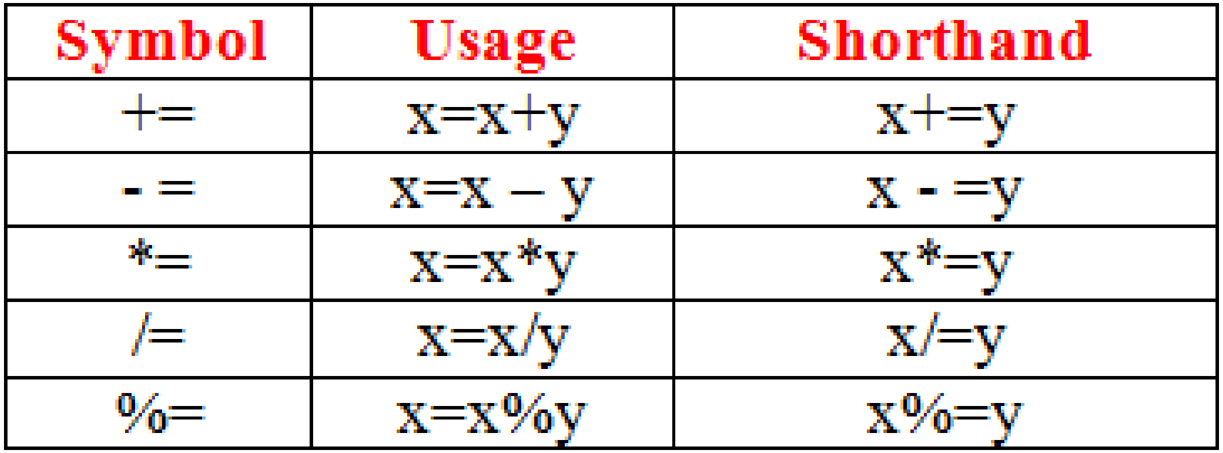
```
var num = 100;
switch (true)
{
        case num < 0:
                alert("Negative Number");
                break;
        case num >= 0 && num <= 50:
                alert("Between 0 and 50");
                break;
        case num > 50 && num <= 100:
                alert("Between 50 and 100");
                break;
        default:
                alert("Number grater than 100'
}
```

Arithmetic Operators



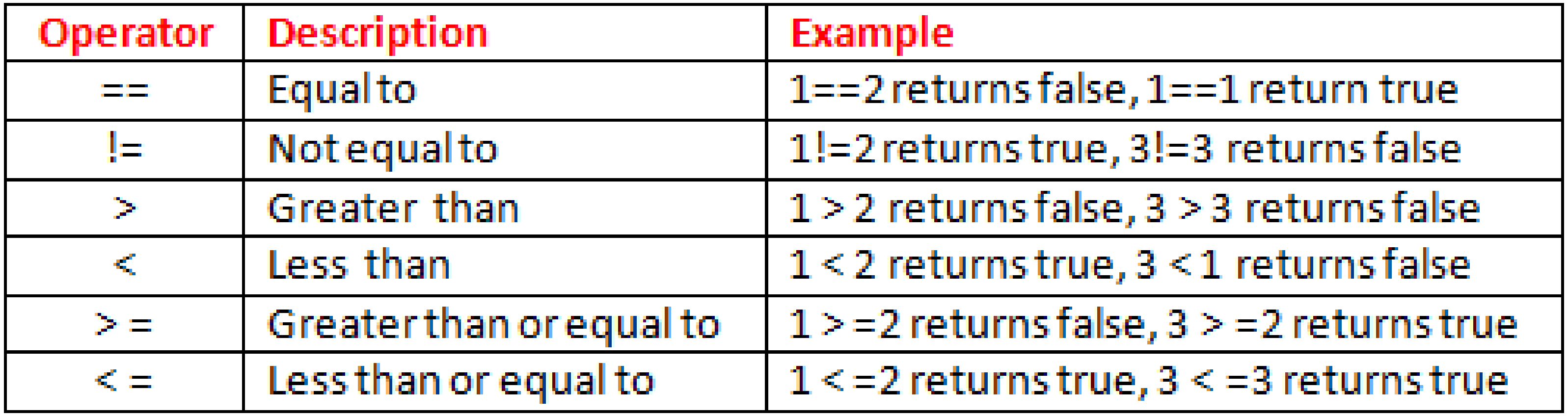
Symbol	Description	Example (x=10)	Result
+	Addition	X+5	15
-	Subtraction	x-2	8
*	Multiplication	X*3	30
/	Division	x/2	5
%	Modulus (division remainder)	X%3	1
++	Increment (increments the variable by 1)	X++	11
--	Decrement (decreases the variable by 1)	x--	9

Assignment Operators



Symbol	Usage	Shorthand
+=	x=x+y	x+=y
- =	x=x – y	x - =y
*=	x=x*y	x*=y
/=	x=x/y	x/=y
%=	x=x%y	x%=y

Comparison Operators



Operator	Description	Example
==	Equal to	1==2 returns false, 1==1 return true
!=	Not equal to	1!=2 returns true, 3!=3 returns false
>	Greater than	1 > 2 returns false, 3 > 3 returns false
<	Less than	1 < 2 returns true, 3 < 1 returns false
> =	Greater than or equal to	1 > =2 returns false, 3 > =2 returns true
< =	Less than or equal to	1 < =2 returns true, 3 < =3 returns true

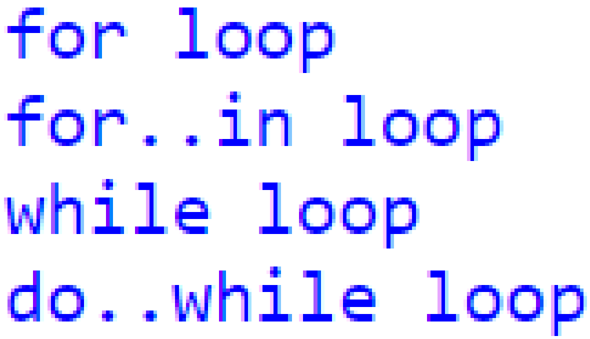
Logical Operators

AND (&&) : Allows you to check if both of two conditions are met

OR (||) : Allows you to check if one of two conditions are met

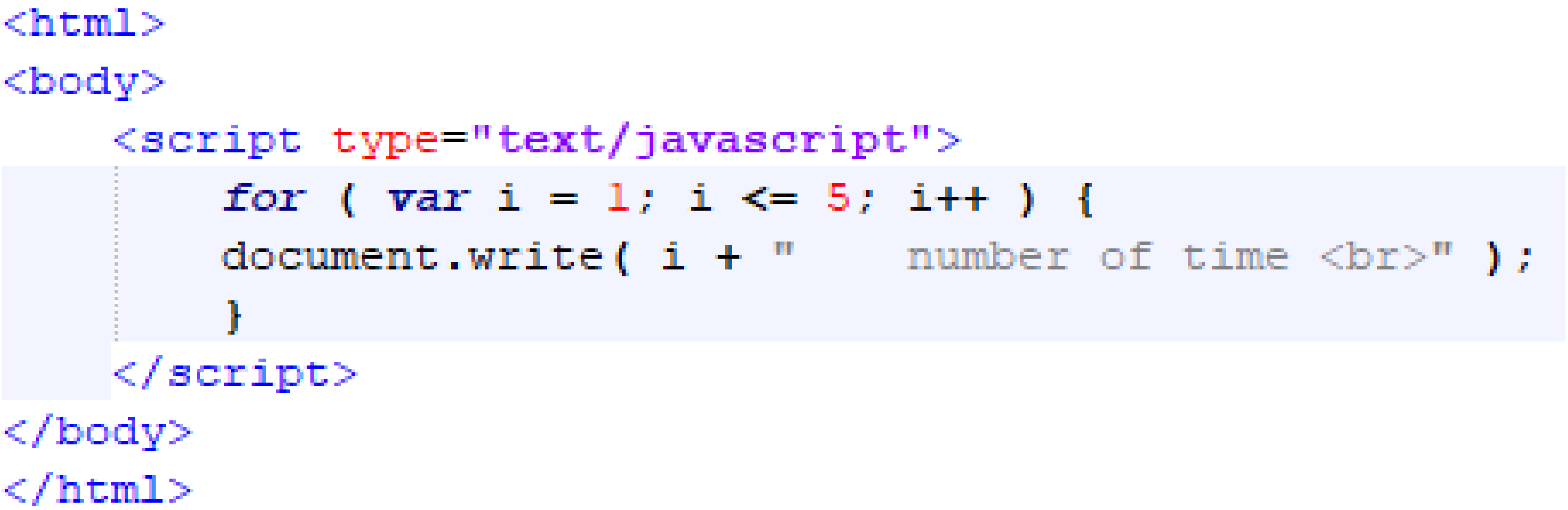
NOT (!) : Allows you to check if something is not the case

Loops in JavaScript



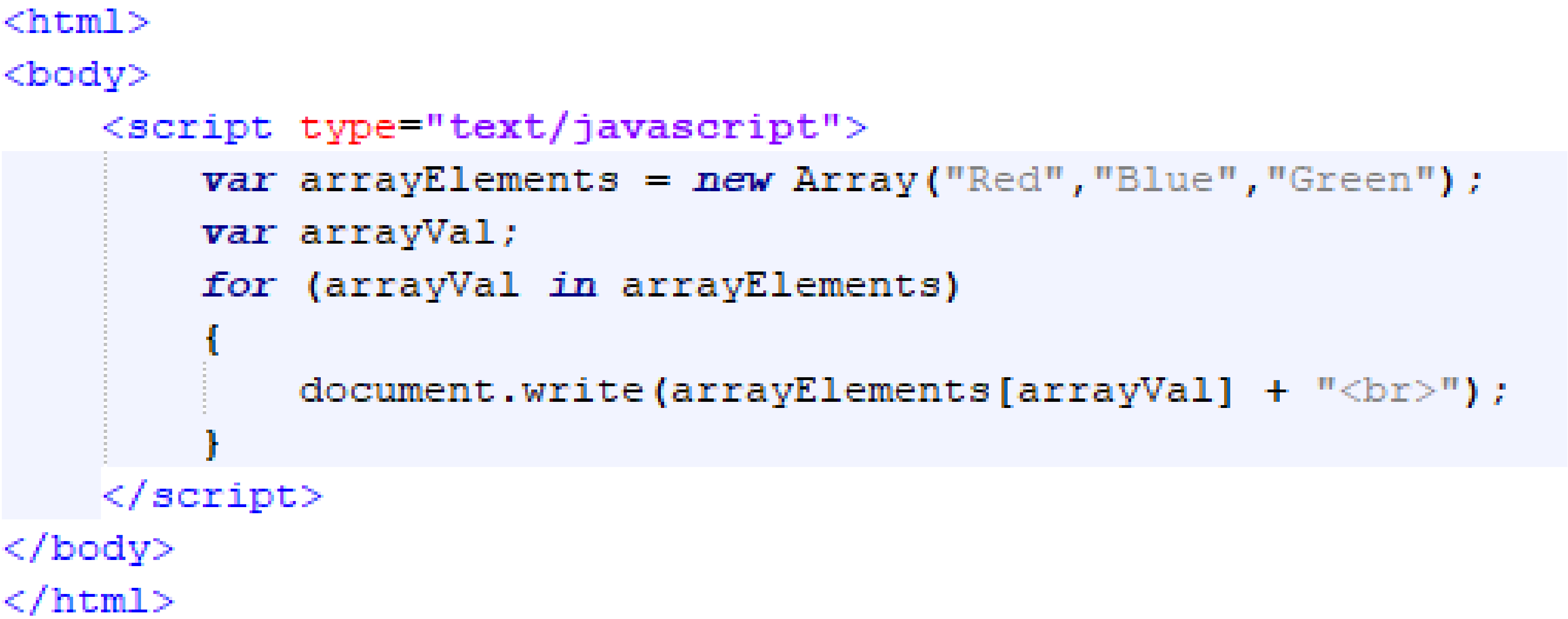
```
for loop
for..in loop
while loop
do..while loop
```

Javascript for loop



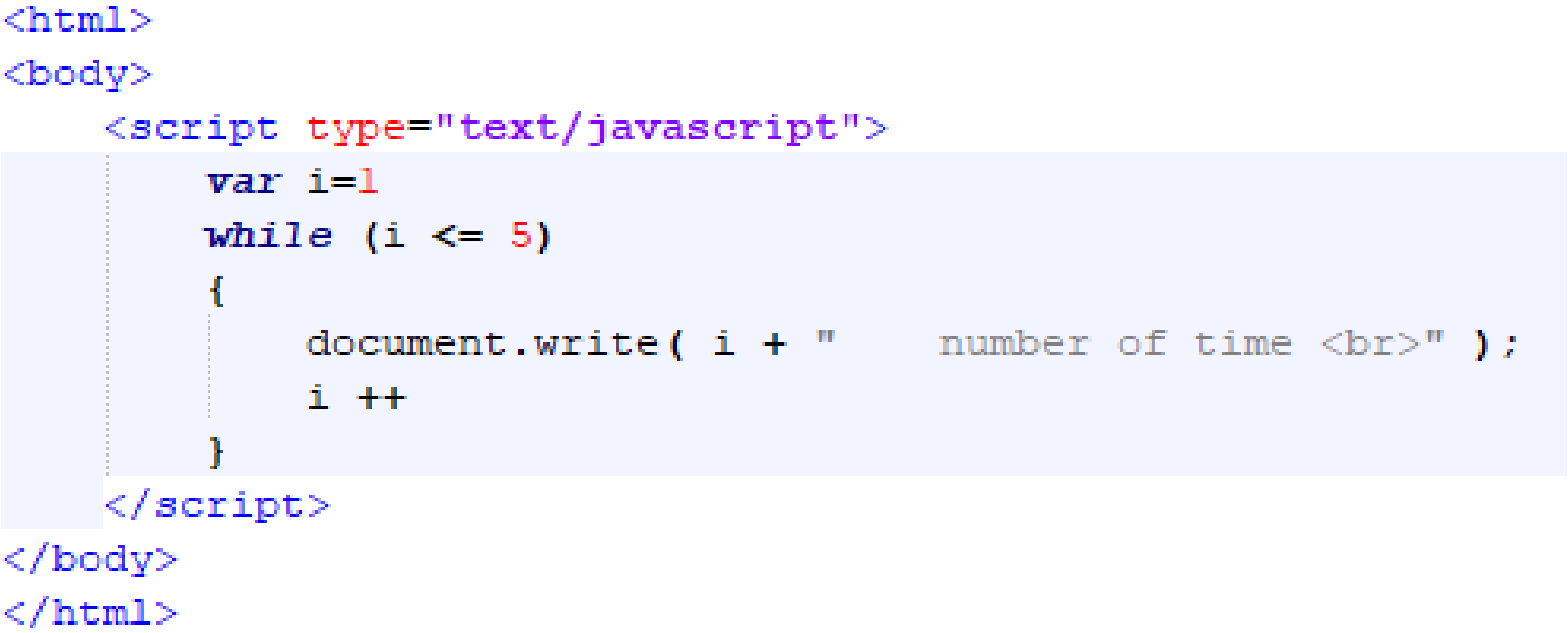
```
<html>
<body>
    <script type="text/javascript">
        for  ( var i = 1; i <= 5; i++ ) {
        document.write( i + "    number of time <br>" );
        }
    </script>
</body>
</html>
```

Javascript for..in Loop



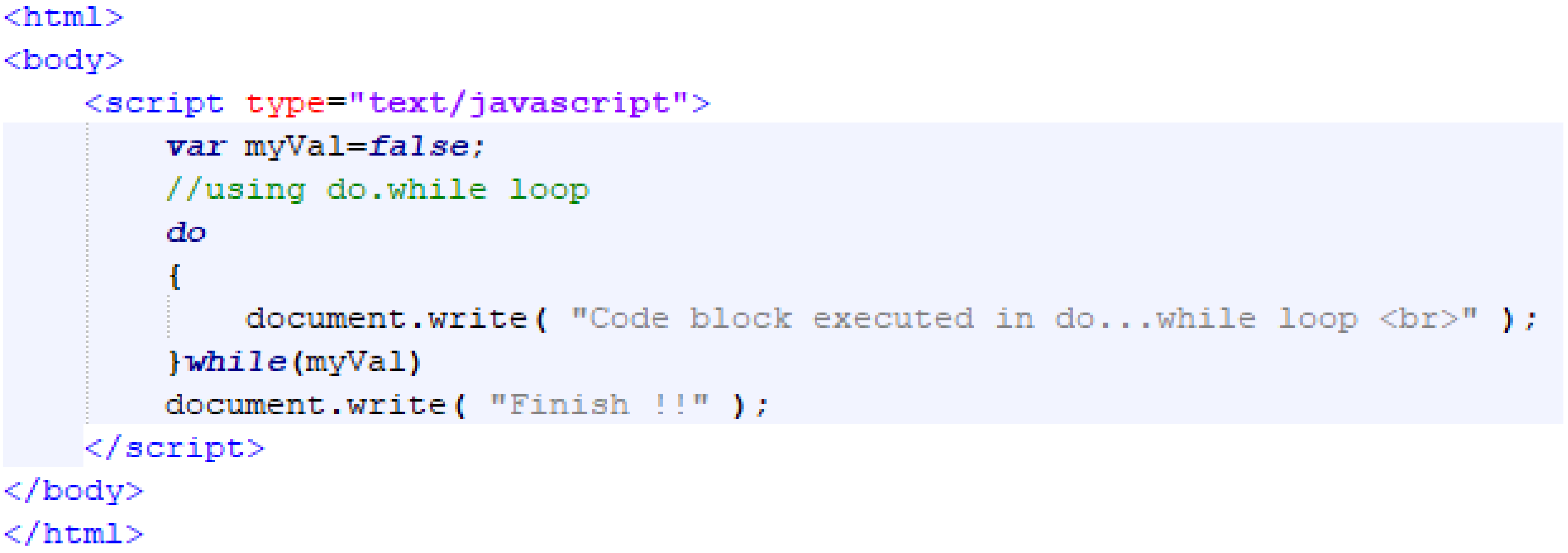
```
<html>
<body>
    <script type="text/javascript">
        var arrayElements = new Array("Red","Blue","Green");
        var arrayVal;
        for (arrayVal in arrayElements)
        {
            document.write(arrayElements[arrayVal] + "<br>");
        }
    </script>
</body>
</html>
```

Javascript while Loop



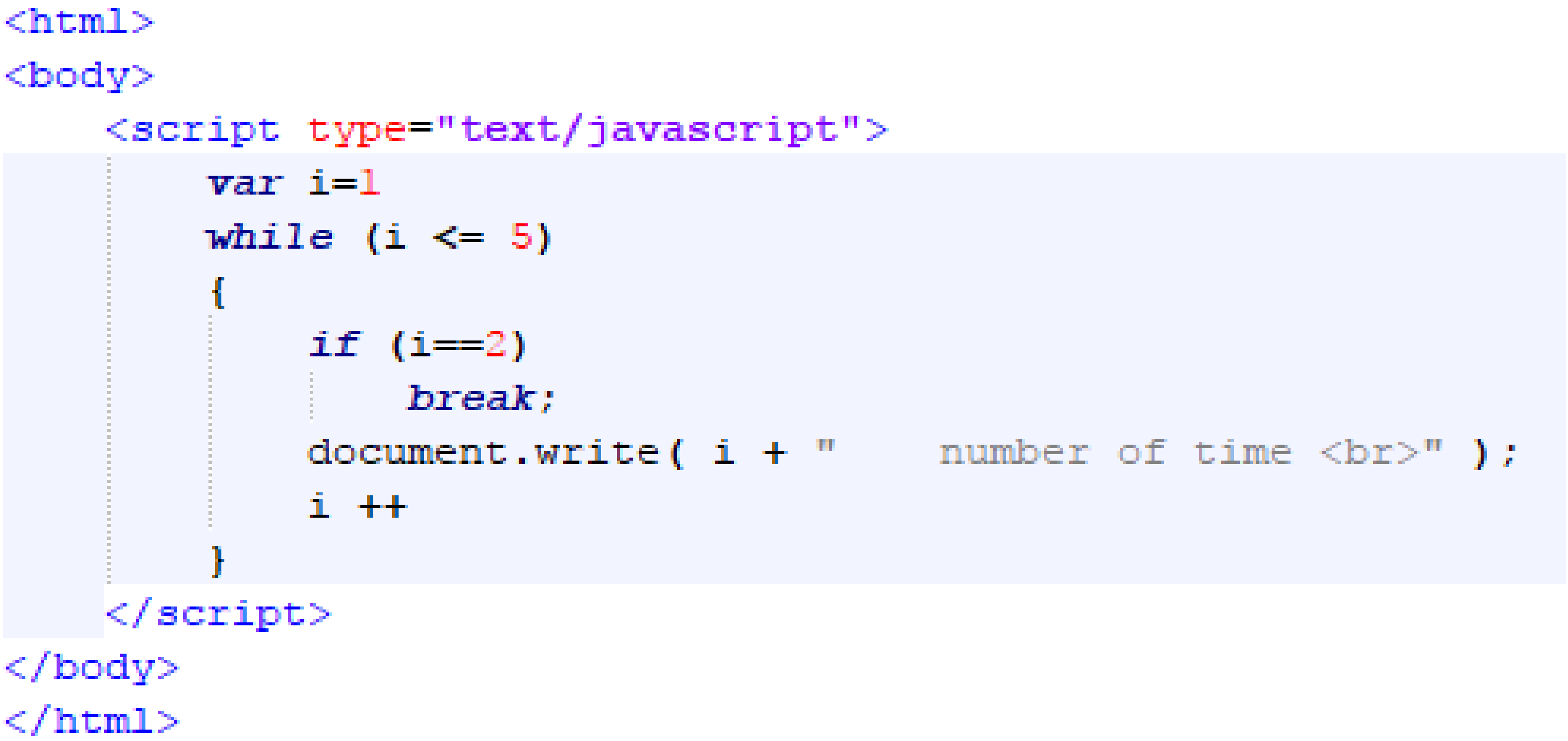
```
<html>
<body>
    <script type="text/javascript">
        var i=1
        while (i <= 5)
        {
            document.write( i + "    number of time <br>" );
            i ++
        }
    </script>
</body>
</html>
```

Javascript do..while loop



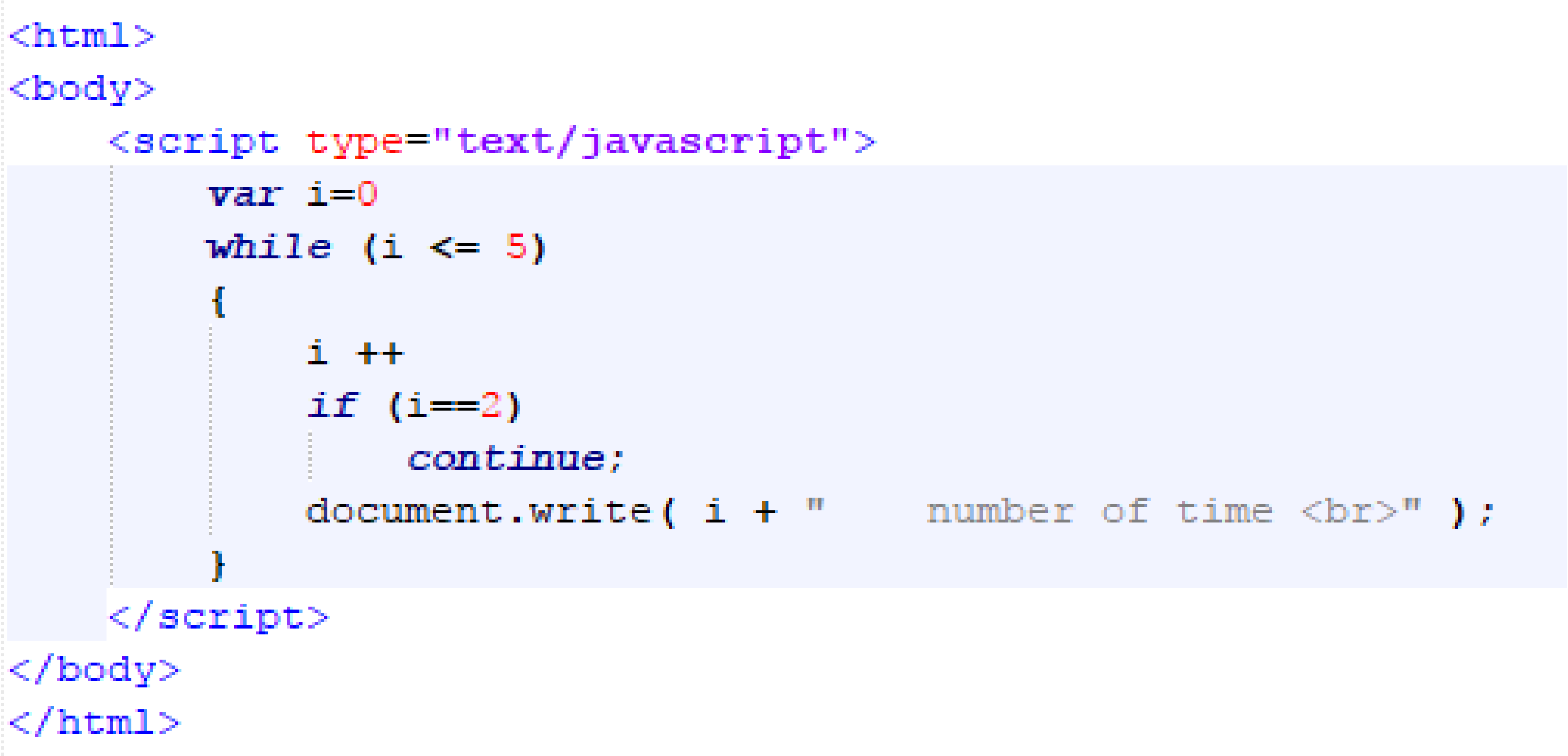
```
<html>
<body>
    <script type="text/javascript">
        var myVal=false;
        //using do.while loop
        do
        {
            document.write( "Code block executed in do...while loop <br>" );
        }while(myVal)
        document.write( "Finish !!" );
    </script>
</body>
</html>
```

Javascript Break statement



```
<html>
<body>
    <script type="text/javascript">
        var i=1
        while (i <= 5)
        {
            if (i==2)
                break;
            document.write( i + "    number of time <br>" );
            i ++
        }
    </script>
</body>
</html>
```

Continue statement



```
<html>
<body>
    <script type="text/javascript">
        var i=0
        while (i <= 5)
        {
            i ++
            if (i==2)
                continue;
            document.write( i + "    number of time <br>" );
        }
    </script>
</body>
</html>
```

This Week

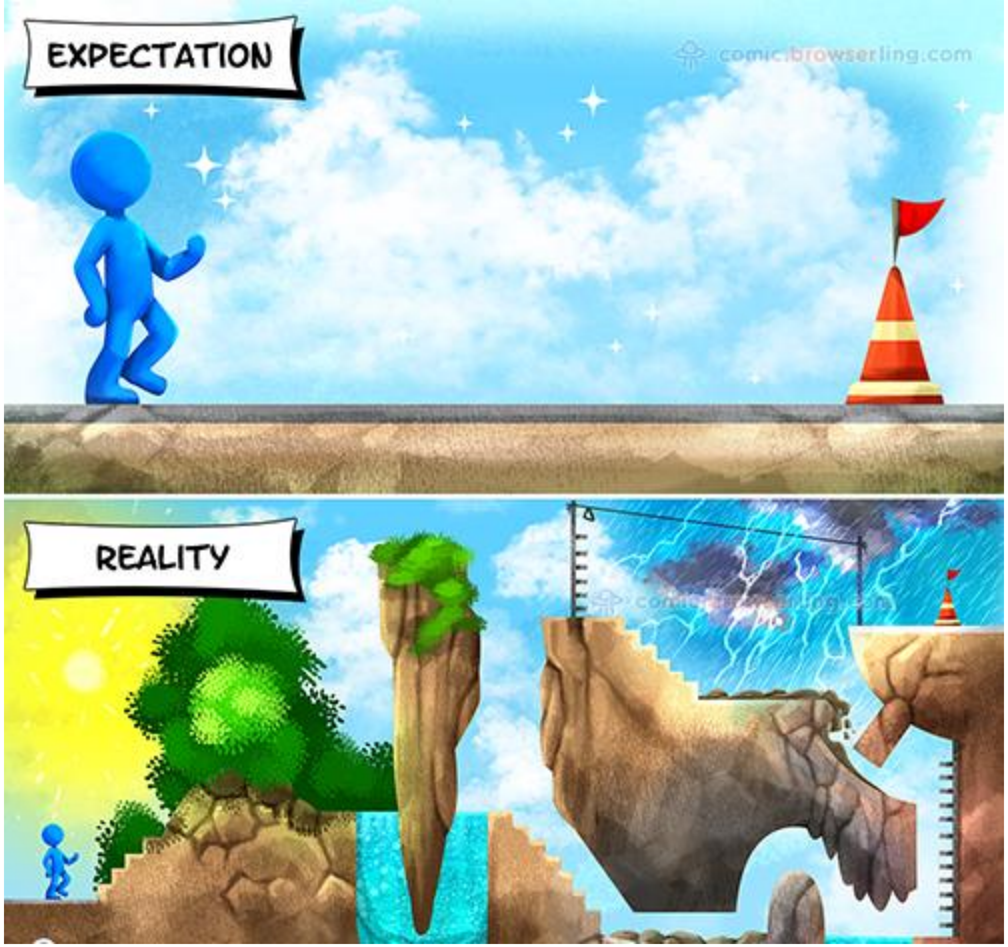
- Review Slides
- Read Associated Chapters
- Work through Javascript Examples
  - Update GitHub Account/Webpage
- **Start Group Project**

Summary

- Overview of Javascript Syntax
- Examples
- Hands-On/Practical

Questions/Discussion

Learning JavaScript

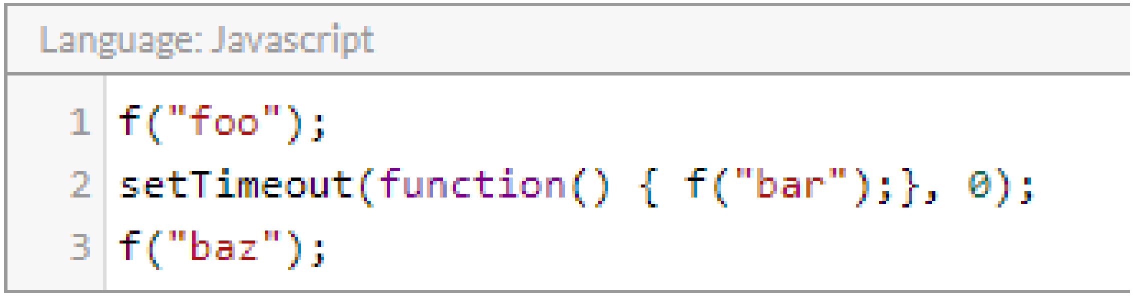


Questions Next Week

Questions to Investigate for Next Week

Question

- What order does f receive its arguments?



```javascript
f("foo");
setTimeout(function() { f("bar");}, 0);
f("baz");
```

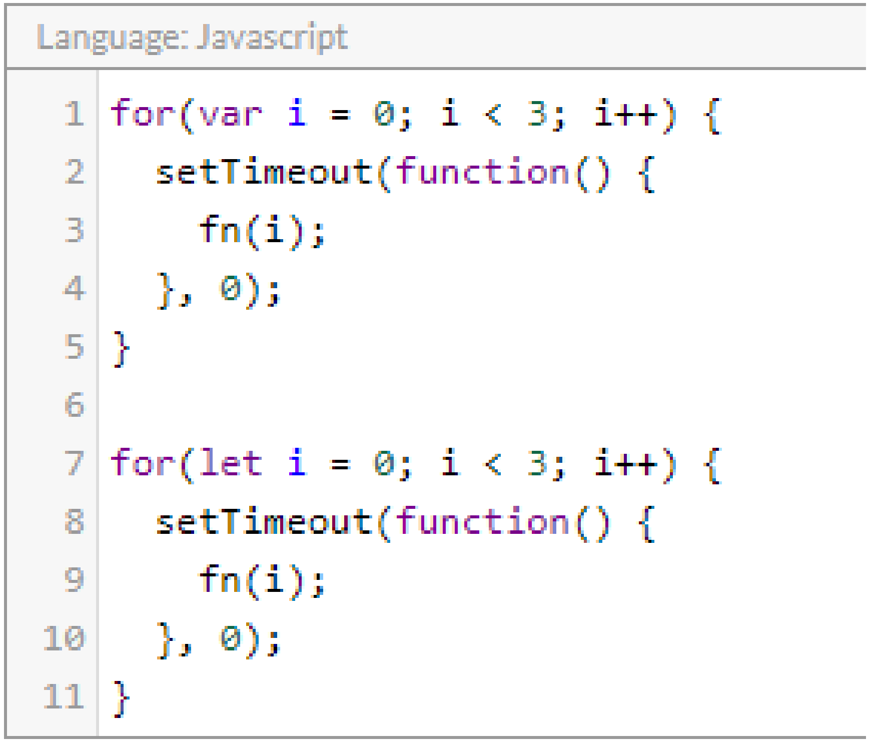
a) foo bar baz
b) foo baz bar
c) bar foo baz
d) foo baz

Question

- Which of the following is correct about the expression 1==“1.0” in Javascript?

a) It returns false. The two values have different types.
b) It returns true. The two values are converted to strings and are equal.
c) It returns false. The two values are converted to strings and are not equal.
d) It returns true. The two values are converted to numbers and are equal.

What arguments will fn receive, in order?



```javascript
for(var i = 0; i < 3; i++) {
  setTimeout(function() {
    fn(i);
  }, 0);
}

for(let i = 0; i < 3; i++) {
  setTimeout(function() {
    fn(i);
  }, 0);
}
```

a) 3, 3, 3, 3, 3, 3

b) 3, 3, 3, 0, 1, 2

c) 0, 1, 2, 3, 3, 3

d) 0, 1, 2, 0, 1, 2