Javascript

JavaScript was originally developed by Brendan Eich at Netscape under the name Mocha, later renamed LiveScript, and finally renamed JavaScript. JavaScript is most often used in web pages on the client side (web browser). Despite its name, JavaScript has very little in common with the Java programming language. The little that they have in common includes the syntax of the C programming language and some naming conventions and names, such as Date, Math, and String.

Embedding JavaScript

There are two ways of inserting JavaScript into a web document. The first is to place thecode within a script element as follows.

```
<script>
document.write("Hello World");
</script>
```

The other, more common method is to include the code in an external file and thenlink to that file using the src attribute of the script element.

```
<script src="mycode.js"></script>
```

Java Script example programme:

```
<!DOCTYPE HTML>
<html lang="en-us">
<head>
<meta charset="utf-8">
<title>My First JavaScript Program</title>
<script>
function showMessage() {
alert("Welcome to Javascript Learning!");
}
</script>
</head>
<body>
<button type="button" onclick="showMessage()">Show
Message </button>
</body>
</html>
```

Comments

```
// this is a single line

/* this is
a multiline */
```

A comment in a program is a note or documentation that a programmer writes for himself or other programmers who will look at his code. The computer ignores comments.

Variables

Variables are containers used for storing data, such as numbers or strings, so that theycan be used multiple times in a script.

A variable has a name, type, value, and scope.

Declaring a Variable:

```
Var name;
```

Assigning a Value to a Variable:

```
name =“Krishna”;
```

Dynamic Typing:

```
var myType = "Hi"; //string
myType = 1.5;  //number
```

Variable scope

```
var myVar = "global";
/*Global variable, canbe used
throughout the programme. */
function checkscope( ) {
/* a local variable will be visible
only within a function.  */
var myVar = "local";
document.write(myVar);
}
```

A variable namemust start with an uppercase or lowercase letter, theunderscore (_), or the dollar sign ($). The rest of the namemust be made of letters, digits (0-9), the underscore, or thedollar sign. A variable name cannot include spaces. Avariable name should be meaningful and should correctlydescribe what the variable stores. You should not use keywords i.e (if, loop, setup, else etc…)as variable or functionnames in your programs.

```
var _myVar32; // allowed
var 32Var; // incorrect (starts with number)
var my Var; // incorrect (contains space)
var var@32; // incorrect (contains special character)
var var; // incorrect (reserved keyword)
```

Data Types

Number: Comprises integer and floating-point numbers.
Boolean: Comprises the logical values true or false.
String: Comprises a sequence of alphanumeric characters. For example, “Hello, World!”,“What is your name?” or “KA30V2Y556.”

Operators

Mathematical Operators (Arithmetic Operators)	Comparison Operators	Boolean Operators (Logical Operators)
= assignment + addition - subtraction * multiplication / division % modulus ++ increment -- decrement	== equal to != not equal to <less than > greater than <= less than or equal to >= greater than or equal to	&&// Boolean AND || // Boolean OR ! // Boolean NOT **Bitwise Operators** & // bitwise AND | // bitwise OR ^ // bitwise XOR ~ // bitwise INVERT var << n // shift left by n bits var >> n //shift right by n bits

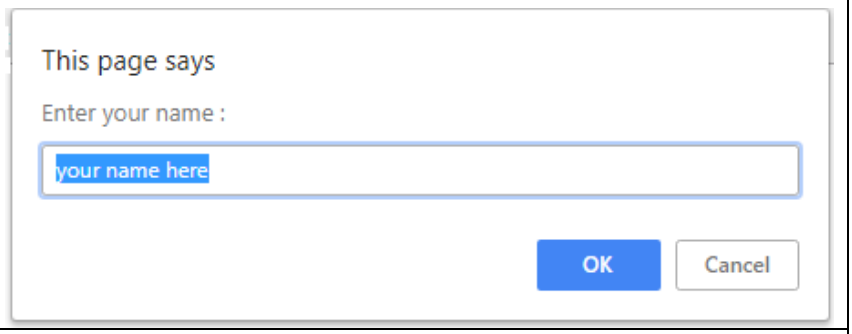
Compound Operators

```
+= compound addition
-= compound subtraction
*= compound multiplication
/= compound division
&= compound bitwise AND
|= compound bitwise OR
```

Order of Operations

B	Brackets	$10 \times (4 + 2) = 10 \times 6 = 60$
O	Order	$5 + 2^2 = 5 + 4 = 9$
D	Division	$10 + 6 \div 2 = 10 + 3 = 13$
M	Multiplication	$10 - 4 \times 2 = 10 - 8 = 2$
A	Addition	$10 \times 4 + 7 = 40 + 7 = 47$
S	Subtraction	$10 \div 2 - 3 = 5 - 3 = 2$

Useful Java Script Functions and methods-1

alert(message);	gives a warning message to the users (alert dialog box displays) alert("Thanks for your input!""); 
confirm(message);	A confirmation dialog box is mostly used to take user's consent on any option. `var result = confirm("Do you want to continue ?");` 
prompt(text, [default]);	The prompt dialog box is very useful when you want to pop-up a text box to get user input. `var result = prompt("Enter your name : ", "your name here");` 
`console.log ();`	

`document.write(message);`	`Writes the message into the document.`
	
	`document.write("Hello World");`

	`document.write("<h1>Hello World</h1>");`
`document.writeln(message)`	`Writes the message and advances to a new line.`

`+ operator on strings (Concatenating Strings)`	*Concatenation* refers to the act of combining two text strings into one longer text string.
	
	`var name = "Krishna";`
	
	`document.write("<h1>Welcome" + name + "</h1>");`

`typeof operand`	The *typeof* operator returns a string to identify the type of its operand (i.e., a variable, string, keyword, or object). Ex. `typeof(24.6)` `typeof("Hello")`
`parseInt();`	*Converts string into a numeric value.*
`parseFloat();`	*parseFloat accepts a string as an argument which it converts to a float*
`number.toFixed( [digits]);`	`var num=182.5698;` `alert(num.toFixed(2));`
`Math.ceil(x) ;`	This method returns the smallest integer greater than or equal to a number.
`Math.floor(x) ;`	This method returns the largest integer less than or equal to a number.
`Math.round(x);`	This method returns the value of a number rounded to the nearest integer.
`Math.pow(base, exponent );`	This method returns the base to the exponent power, that is, **base**$^{\text{exponent}}$.
`Math.random ( );`	This method returns a random number between 0 (inclusive) and 1 (exclusive).
`Math.sqrt ( x );`	This method returns the square root of a number. If the value of a number is negative, sqrt returns NaN.
`Date`	
`Date();`	Javascript **Date()** method returns today's date and time. `Wed Mar 25 2015 15:00:57 GMT+0530 (India Standard Time)`
`Date.getDate()`	`These methods return the day of the month (range:`

	1-31).
	
	var dt = new Date();
	alert(dt.getDay());

Date.getDay()	These methods return the day of the week (range: 0–6)
getFullYear()	These methods return a fourdigit number representing the year
getHours()	These methods return the current hour (range: 0–23)
getMilliseconds()	These methods return the number of milliseconds elapsed in range 0–999
getMinutes()	These methods return the number of minutes in range 0–59
getMonth()	These methods return the number corresponding to the current month (range 0–11)
getSeconds()	These methods return the number of seconds elapsed on specified date (range 0–59)

Control statements (Decision Making)

if statement

Syntax

```
if (condition)
{
Block of
statements;
}
```

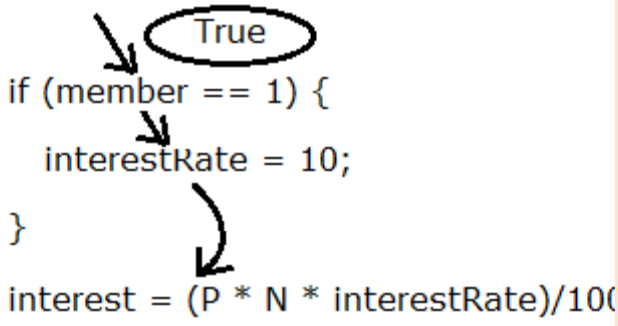
Example1

```
interestRate = 12;
if (member == 1) {
interestRate = 10;
}
```

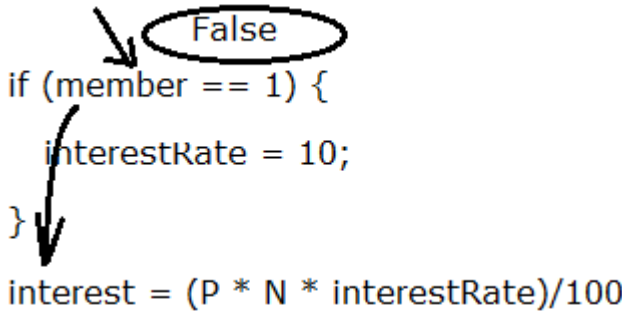
Example2

```
if(lightLevel> 900){
alert("Light level is: ");
alert(lightLevel);
}
```

Instructions execution sequence for condition True.



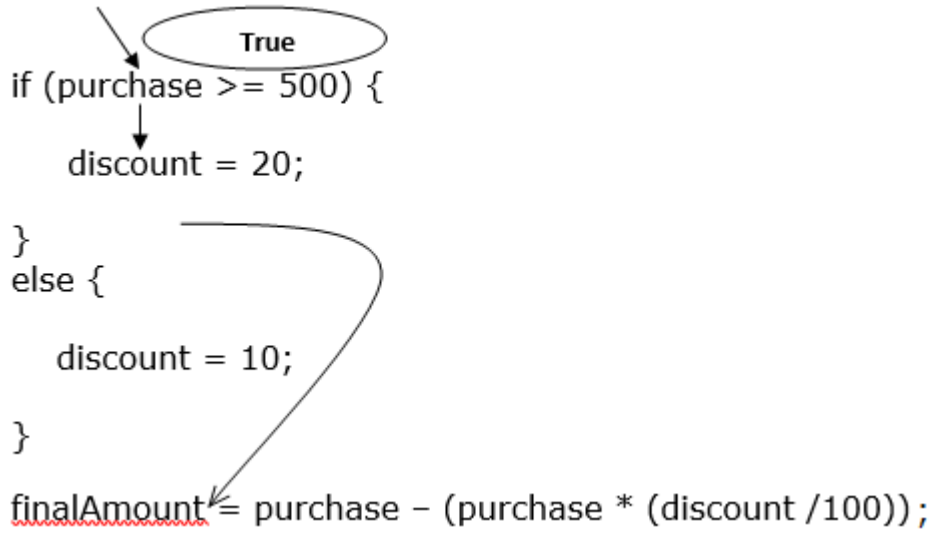
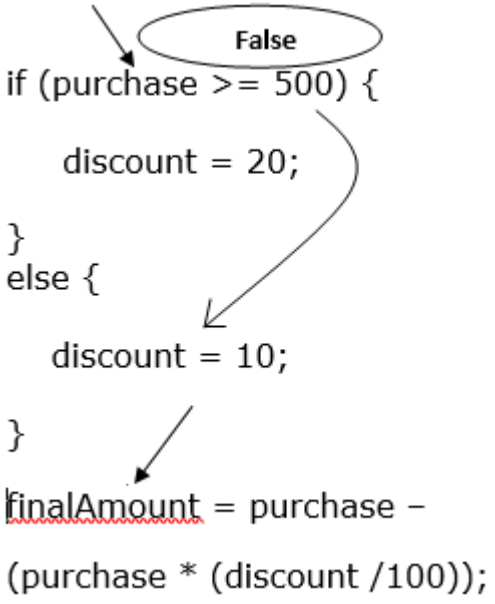
Instructions execution sequence for condition False.



if …else statement

Syntax	Example1	Example2

if (condition)
{
Block of statements;
}
else
{
Block of statements;
}
	if (purchase >= 500) {
	discount = 20;
	}
	else {
	discount = 10;
	}
		if (number % 2 == 0) {
		alert(number + "is Even.");
		}else{
		Alert(number + "is odd.");
		}

Instructions execution sequence for condition True.	Instructions execution sequence for condition False.
	

if … else if …else statement

Syntax

```
if (condition)
{
Block of statements;
}
else if (condition)
{
Block of statements;
}
else if (condition)
{
Block of statements;
}
….
else {
Block of statements;
}
```

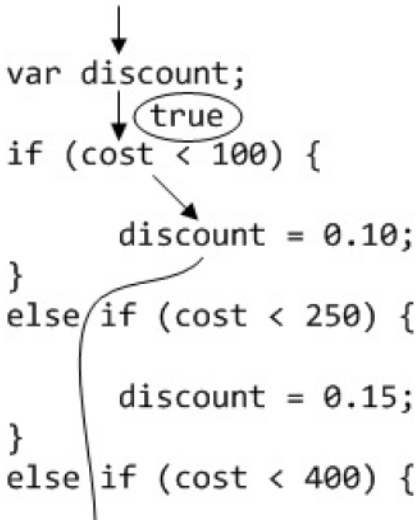
Example1

```
var discount;
if (cost < 100) {
    discount = 0.10;
}
else if (cost < 250) {
    discount = 0.15;
}
else if (cost < 400) {
    discount = 0.18;
}
else {
    discount = 0.20;
}
cost *= (1 - discount);
```

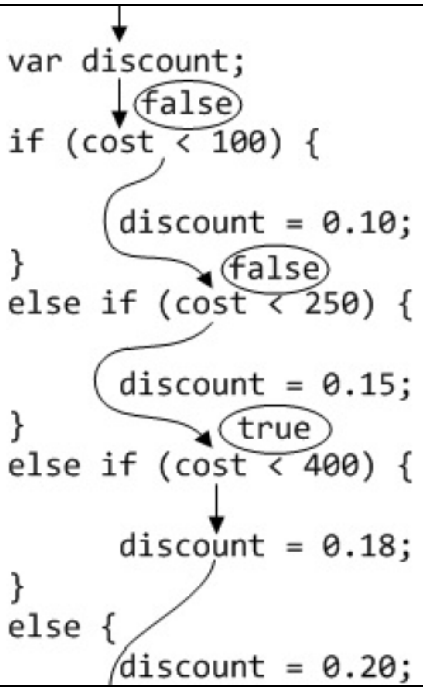
Example2

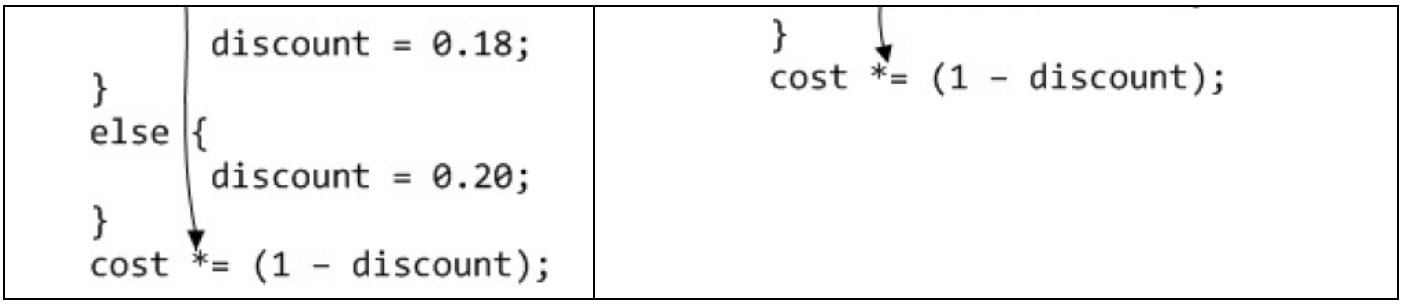
```
if (score >= 90)
alert ("A");
else
if (score >= 80)
alert("B");
else
if (score >= 70)
alert("C");
else
if (score >= 60)
alert("D");
else
alert("F");
```

Example1: Instructions execution sequence.



Example 2: Instructions execution sequence.

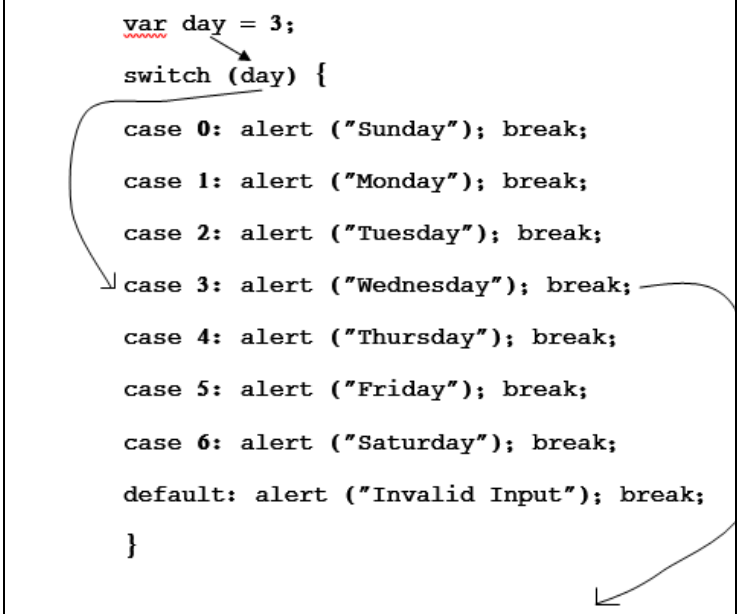
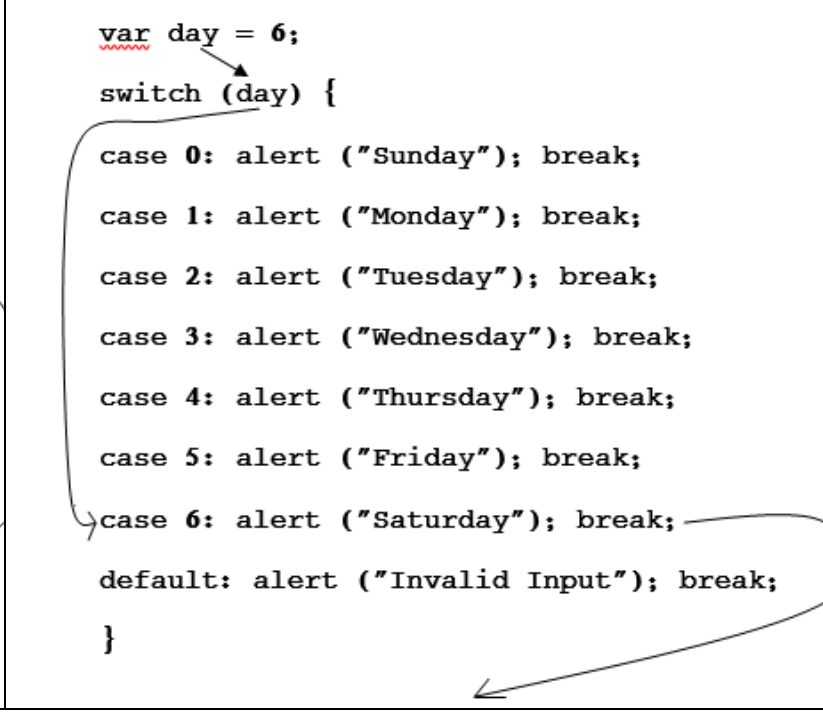




Switch Case

Syntex	**Example**

switch (variable) {
case constValue1: statements;
break;
case constValue2: statements;
break;
…
default: statements; break;
}
	switch (day) {
	case 0: alert ("Sunday"); break;
	case 1: alert("Monday"); break;
	case 2: alert("Tuesday"); break;
	case 3: alert("Wednesday"); break;
	case 4: alert("Thursday"); break;
	case 5: alert("Friday"); break;
	case 6: alert("Saturday"); break;
	default:alert("Invalid Input"); break;
	}

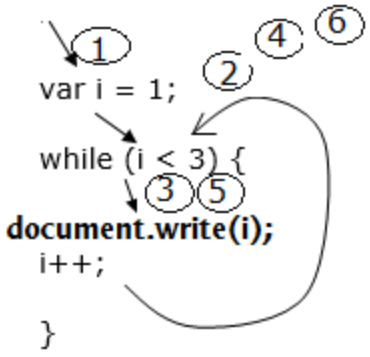
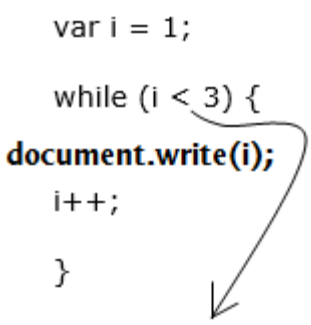
Example1: Instructions execution sequence.	Example2: Instructions execution sequence.
	

Loops (Repeat a sequence of statements)

While loop

while loop syntax:	Example:

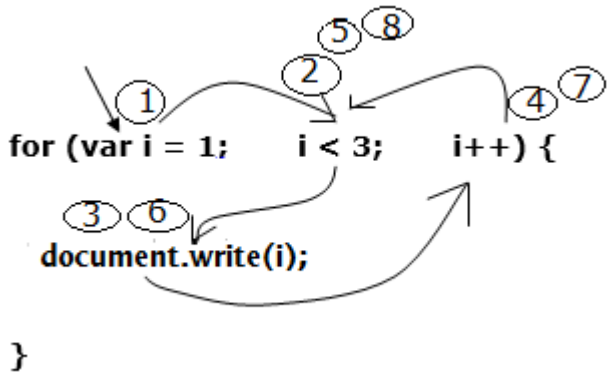
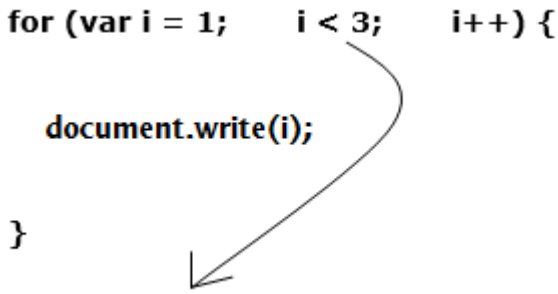
while (condition) {
//statements;
}
	var i = 1;
	while (i < 3) {
	Serial.println(i);
	i++;
	}

Instructions execution sequence for condition True.	Instructions execution sequence for condition False.
	

For loop

For loop syntax:	**Example:**

`for (initialize; condition; increment or decrement)`
`{`
`// statement block`
`}`
	`for (var i = 1; i <= 9; i++) {`
	`document.write(i);`
	`}`

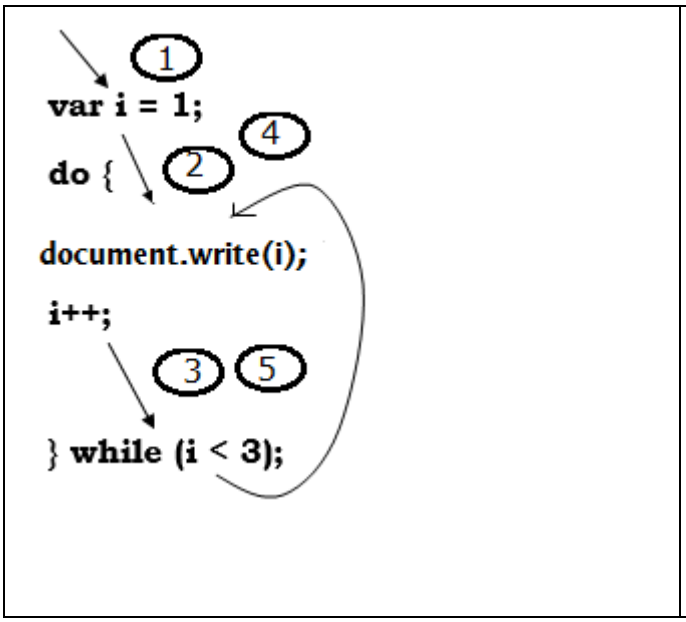
Instructions execution sequence for condition True.	**Instructions execution sequence for condition False.**
	

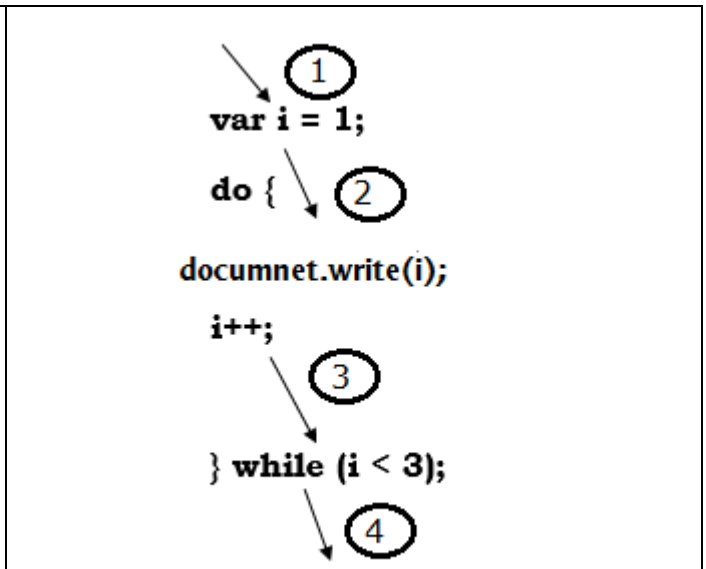
Do-While loop

Do-while loop syntax:	**Example:**

`do{`
`//statements;`
`} while (condition);`
	`var i = 1;`
	`do {`
	`document.write(i);`
	`i++;`
	`} while (i < 3);`

Instructions execution sequence for condition True.	**Instructions execution sequence for condition False.**

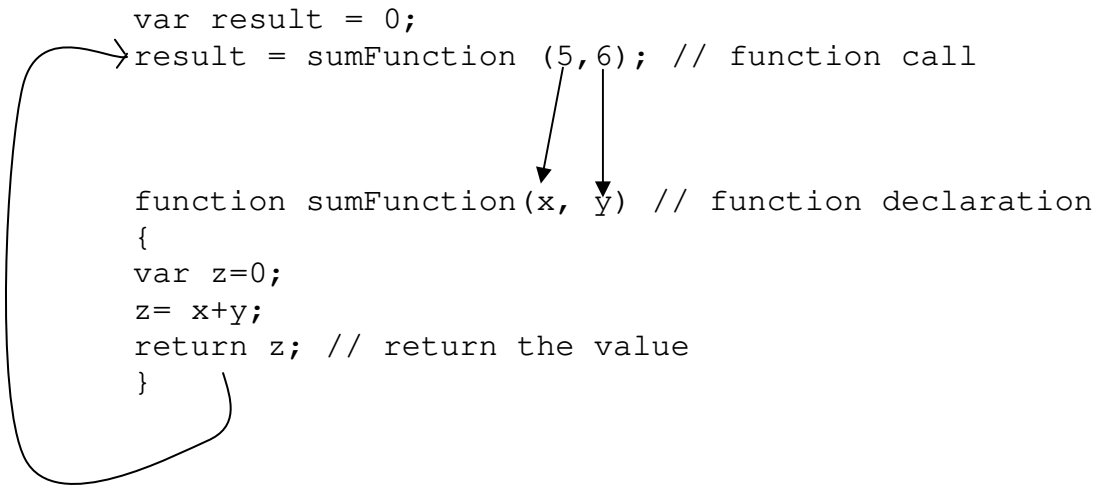

1
var i = 1;
4
do {
2
document.write(i);
i++;
3
5
} while (i < 3);


1
var i = 1;
do {
2
documnet.write(i);
i++;
3
} while (i < 3);
4

Functions (divide programs into parts)

Functions are reusable code blocks that will only execute when called. They allowdevelopers to divide their scripts into smaller parts that are easier to understand and reuse.

Sum calculation function



Function Syntax:

```
returnTypex = functionName (argument1, argument2);

returnTypefunctionName (parameter1, parameter2){
                                    statements;
                                     --
                                     --
return result;

 }
```

Arrays (a collection of similar variables)

An array is a data structure used for storing a collection of values. JavaScript arrays can be grouped into three kinds: numeric, associative, and multidimensional.

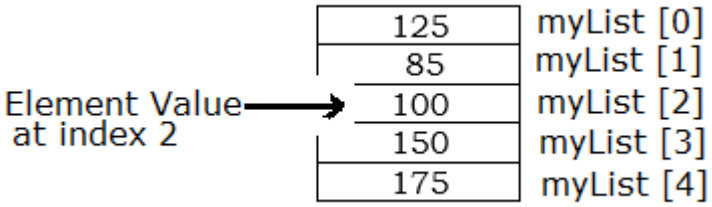
Array declaration:

```
var arrayName = new Array(number of elements);

var myList[] = new Array();
```

Array declaration with assigned values:

```
var myList = new Array(125,85,100,150,175);
```



Assign a value to the indexed position:

```
myList [3] = 65;
```

To retrieve a value from the array:

```
var x = myList [3];
```

Demonstrating arrays usage:

```
var months = new Array();
months[0] = "January";
months[1] = "February";
months[2] = "March";
months[3] = "April";
months[4] = "May";
months[5] = "June";
months[6] = "July";
months[7] = "August";
months[8] = "September";
months[9] = "October";
months[10] = "November";
months[11] = "December";
for (var i = 0; i < 12; i++) {
document.write (months[i] + "<br>");
}
```

Associative Arrays:

The associative array uses a key string to identify an element instead of a numeric index.To create one, first declare an empty array and then assign values to the desired keys.

```
var user = new Array();
user["name"] = "Krishna";
user["age"] = 13;
user["class"] = 8;

document.write(user[“name”] +  is  + user[“age”]);
```

Multidimensional Arrays

Arrays can be made multidimensional by adding arrays as elements to another array.

```
var m = [ ["00","01"], ["10","11"] ];
```

Multidimensional arrays can have any number of dimensions, but more than twodimensions are rarely needed. For each extra dimension another set of square bracketsis added.

```
document.write(m[1][1]); // "11"
```

Events, Forms and DOM

Events - inline	
Events: link	Click
Events: button	<input type="button" value="Click" onClick="alert('Hello world!');">
Events: mouse	<img src="before-pic.jpg" onMouseover="src='after-pic.jpg'">
Events: fields	Email: <input type="text" size="30" onFocus="this.style.backgroundColor='yellow';">
Adding –Event Listeners	

document.onload = createEventLiseners();
function createEventLiseners(){
document.getElementById("resetForm").addEventListener("click", resetForm, false);
document.getElementById("submitForm").addEventListener("click", submitForm, false);
};	

Reading Forms : Using form name, field name	
<script>	

```
function displayValue(){
alert("text box value is: " + myForm.userName.value );
};

</script>
</head>
<body>
<form name="myForm">
<input type="text" name="userName" size="50">
<input type="button" value="Show"  onClick="displayValue()" >
<input type="submit" value="Clear">
<input type="button" value="Reset"  onClick="reset()" >
</form>
```

Reading field values by id

```
function checkAddress(fieldId) {
var val = document.getElementById(email).value;
if val === "") {
alert("Email address required.");

}
}
<form onSubmit="checkAddress('email');">
Email:
<input type="text" id="email">
<input type="submit" value="Submit">
</form>
```

Setting field values

```
function fillCity() {
var cityName;
var zipEntered = document.getElementById("zip").value;
switch (zipEntered) {
case "581400" :
cityName = "Kaiga";
break;
case "68114" :
cityName = "Omaha";
break;
case "53212" :
```

```
cityName = "Milwaukee";
}
document.getElementById("city").value = cityName;
 }
<form>
ZIP:<br>
<input type="text" id="zip" onBlur="fillCity();"><br>
City:<br>
<input type="text" id="city">
</form>
```

Form Validation and Auto focus on error

```
<form>
<label>Email:</label>
<input type="text" id="email" onblur="checkAddress1()";>
<input type="submit" value="Submit">
</form>
function checkAddress() {
var val = document.getElementById("email").value;
if (val == "") {
document.getElementById("email").style.background = "red";
document.getElementById("email").focus() = "red";
}
}
```

Reading and setting paragraph text (.innerHTML for paragraphs, div, html tags)

```
<p id="slowLoris">Slow lorises are <a href="javascript:void(0);" onClick="expandLoris();"><em>Click for
more.</em></a></p>
function expandLoris() {
var expandedParagraph = "Slow lorises are a group of several species”;
document.getElementById("slowLoris").innerHTML = expandedParagraph;
}
```

Manipulating images and text – using class name:

```
<img src="blobfish.jpg" id="ugly" onClick="makeInvisible();">
function makeInvisible() {
   document.getElementById("ugly").className = "hidden";
 }
//for text
function makeBig() {
document.getElementById("p1").className += " big";
```

}

Swapping images:

Inline:<img src="before-pic.jpg" onMouseover="src='after-pic.jpg'">

Function1:

```
<img src="before-pic.jpg" id="before" onMouseover="swapPic();">
function swapPic() {
document.getElementById("before").src = "after-pic.jpg";
}
```

Function2:

```
<img src="before-pic.jpg" id="before" onMouseover="swapPic(id,'after-pic.jpg');">
function swapPic(eId, newPic) {
document.getElementById(eId).src = newPic;
 }
```

Setting styles

```
document.getElementById("p1").className += " big";
document.getElementById("p1").style.fontSize = "2em";
document.getElementById("pic99").style.cssFloat = "left";
document.getElementById("div9").style.visibility = "hidden";
document.getElementById("mainPic").style.margin = "0 10px 0 10px;";
document.getElementById("div9"). style.background = "yellow";
```

Reading Properties:

```
var m = document.getElementById("mainPic").style.margin;
var m = document.getComputedStyle("mainPic").margin;
```

Browser control:

Getting URL:
```
var weAreAt = window.location.href;
var theDomain = window.location.hostname;
var thePath = window.location.pathname;
var theAnchor = window.location.hash;
```

Setting URL:
```
window.location.href = "http://www.amazon.com/about.html";
window.location.assign("http://www.amazon.com/about.html ");
window.location.replace("http://www.amazon.com/about.html ");
```

Window reload:

```
window.location.reload(true);
window.location.reload(false);
window.location.reload();
```

Browser control: Forward and reverse

```
history.back();
history.forward();
```

history.go(-3); history.go(2);
Browser control: Filling the window with content

var newWindow = window.open('', "Message", "left=250, width=750, height=800 status=no, menubar=no, toolbar=no, location=no");
var windowContent = " Message <img src= 'test.jpg'> Alert message to user. ";
newWindow.document.write(windowContent);

Browser control: Open new html document

<script>
function openWindow(){
var newWindow = window.open('test.html', "Add record", "left=250, width=750, height=800 status=no, menubar=no, toolbar=no, location=no");
}
</script>
<button onclick="openWindow();">message</button>

Browser control: Close window
<button onclick="window.close();">Close</button>
Browser control: Print document
<button onclick="window.print();">Print</button>