

# JAVASCRIPT-NOTES-PART-2

## FUNCTION:

A function is a subprogram designed to perform a particular task.

Create function in javascript:

```
function name(parameters){  
    statements  
}
```

Calling of function:

```
name(parameters);
```

## JavaScript variables :

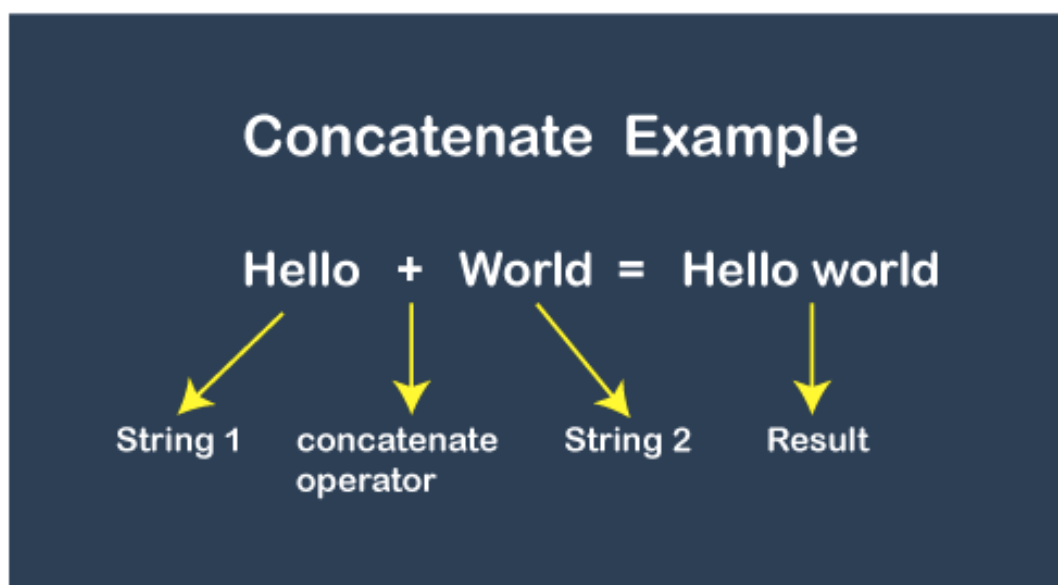
```
var x = 10;
```

### variables:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter.
- Do not add space in between var name use underscores or camelcase (demo\_var / demoVar)
- Names are case sensitive (y and Y are different variables)
- Reserved words (like JavaScript keywords) cannot be used as names.

## CONCATENATION IN JAVASCRIPT

Joining multiple strings together is known as concatenation. The concatenation operator. The concatenation operator (+) concatenates two or more string values together and return another string which is the union of the two operand strings.



## GETELEMENTBYID

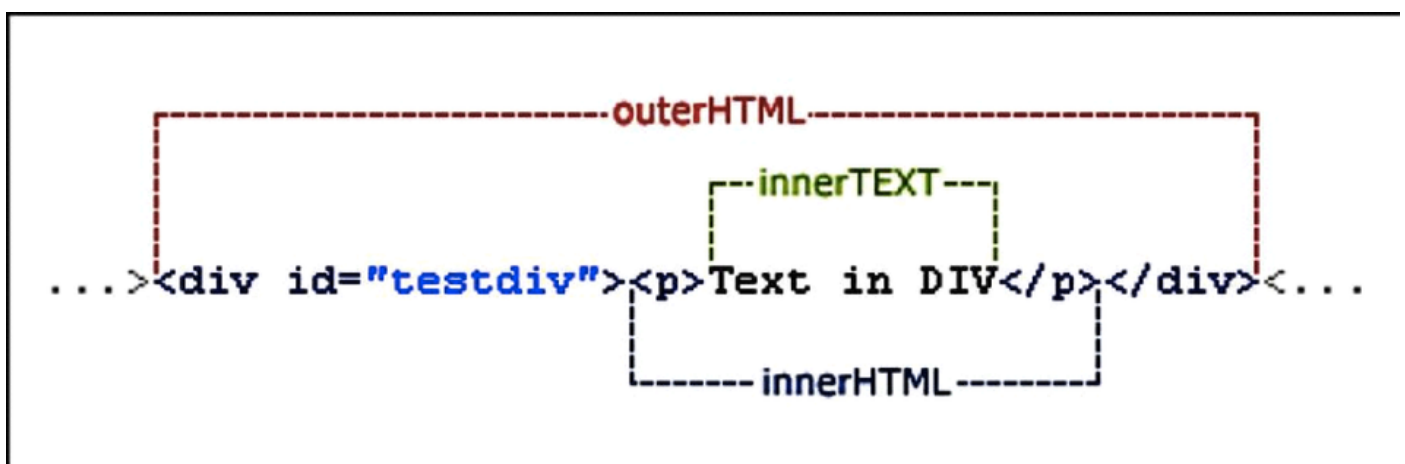
The `getElementById()` method returns an element with a specified value. The `getElementById()` method returns null if the element does not exist. The `getElementById()` method is one of the most common methods in the HTML DOM. It is used almost every time you want to read or edit an HTML element.

## HTML DOM PROPERTY:

### INNERHTML:

The innerHTML property sets or returns the HTML content (inner HTML) of an element.

The innerHTML property can be used to write the dynamic html on the html document. It is used mostly in the web pages to generate the dynamic html such as registration form, comment form, links etc.



- **STYLE.CSSPROPERTY:**

returns or set the css property.

- **CLASSNAME:**

to replace the className or push the new className

# TYPES OF JAVASCRIPT:

## 1. Inline Script :

Write javascript with the tag like:

```
<button onclick="alert(123);">Button 1</button>
```

## 2. Internal Script :

Write javascript on html page using script tag:

```
<script> alert(123); </script>
```

## 3. External Script :

Create new file with js extension and include in html file using script tag:

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

## Comments in Javascript:

For single line: // your script

for multiline: /\* your script \*/

## Global Variables –

A global variable has a global scope which means it can be defined anywhere in your JavaScript code.

## Local Variables

A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

**<SCRIPT>**

**<!--**

**VAR MYVAR = "GLOBAL"; // DECLARE A GLOBAL VARIABLE**

**FUNCTION CHECKSCOPE( ) {**

**VAR MYVAR = "LOCAL"; // DECLARE A LOCAL VARIABLE**

**DOCUMENT.WRITE(MYVAR);**

**}**

**//-->**

**</SCRIPT>**

## Arithmetic Operators

Operators	Meaning	Example	Result
+	Addition	4+2	6
-	Subtraction	4-2	2
*	Multiplication	4*2	8
/	Division	4/2	2
%	Modulus operator to get remainder in integer division	5%2	1
++	Increment	A = 10; A++	11
--	Decrement	A = 10; A--	9

## Logical Operators

Operator	Meaning	Example	Result
&&	Logical and	(5<2)&&(5>3)	False
	Logical or	(5<2)   (5>3)	True
!	Logical not	!(5<2)	True

## Assignment Operators

Operator	Example	Equivalent Expression
=	$m = 10$	$m = 10$
+=	$m += 10$	$m = m + 10$
-=	$m -= 10$	$m = m - 10$
*=	$m *= 10$	$m = m * 10$
/=	$m /=$	$m = m/10$
%=	$m \% = 10$	$m = m\%10$

## Comparison Operators

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
!=	Not equal to	5!=2	True
===	Equal value and same type	5 === 5	True
		5 === "5"	False
!==	Not Equal value or Not same type	5 !== 5	False
		5 !== "5"	True

## THERE ARE TWO TYPES OF COERCION IN JAVASCRIPT:

1) IMPLICIT COERCION: TYPE CONVERSION IS DONE BY JAVASCRIPT IS IMPLICIT COERCION.

2) EXPLICIT COERCION: TYPE CONVERSION IS DONE IN CODE USING THE INBUILT FUNCTIONS  
LIKE NUMBER(), STRING(), BOOLEAN(), ETC IS EXPLICIT COERCION.

## IF...ELSE IF... STATEMENT

The if...else if... statement is an advanced form of if...else that allows JavaScript to make a correct decision out of several conditions.

```
<SCRIPT TYPE = "TEXT/JAVASCRIPT">
  <!--
    VAR BOOK = "MATHS";
    IF( BOOK == "HISTORY" ) {
      DOCUMENT.WRITE("<B>HISTORY BOOK</B>");
    } ELSE IF( BOOK == "MATHS" ) {
      DOCUMENT.WRITE("<B>MATHS BOOK</B>");
    } ELSE IF( BOOK == "ECONOMICS" ) {
      DOCUMENT.WRITE("<B>ECONOMICS BOOK</B>");
    } ELSE {
      DOCUMENT.WRITE("<B>UNKNOWN BOOK</B>");
    }
  //-->
</SCRIPT>
```

## IF...ELSE STATEMENT

The 'if...else' statement is the next form of control statement that allows JavaScript to execute statements in a more controlled way.

```

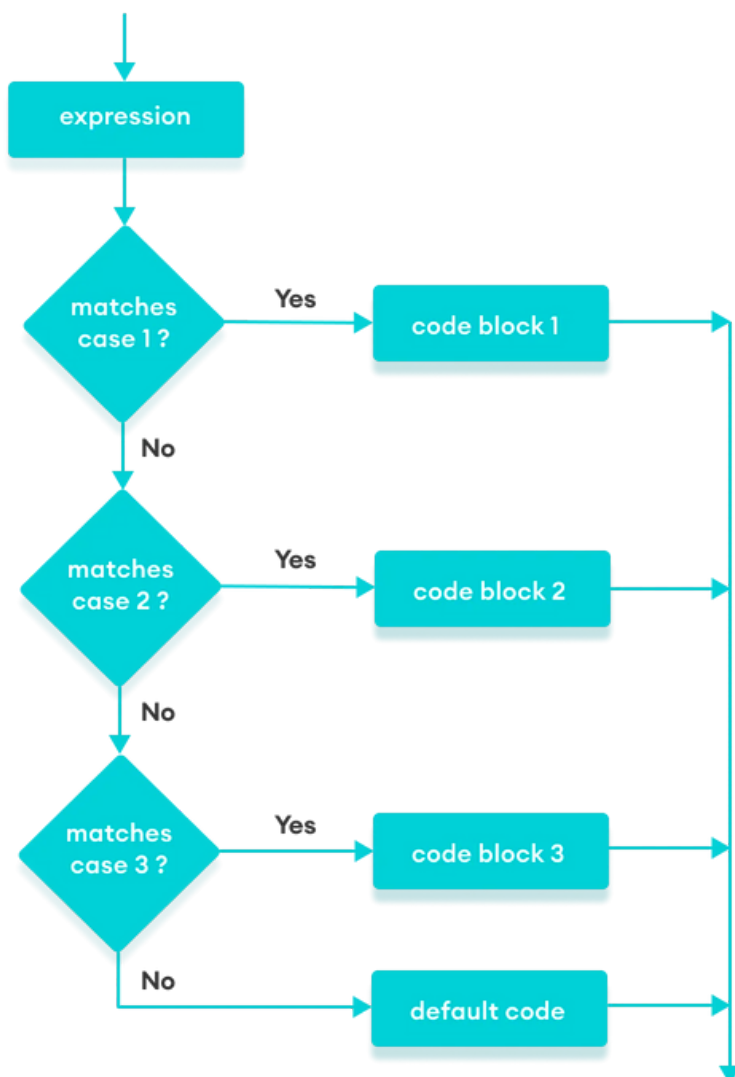
<SCRIPT TYPE = "TEXT/JAVASCRIPT">
  <!--
    VAR AGE = 15;

    IF( AGE > 18 ) {
      DOCUMENT.WRITE("<B>QUALIFIES FOR DRIVING</B>");
    } ELSE {
      DOCUMENT.WRITE("<B>DOES NOT QUALIFY FOR DRIVING</B>");
    }
  //-->
</SCRIPT>

```

THE SWITCH STATEMENT IS USED TO PERFORM DIFFERENT ACTIONS BASED ON DIFFERENT CONDITIONS.

- THE SWITCH EXPRESSION IS EVALUATED ONCE.
- THE VALUE OF THE EXPRESSION IS COMPARED WITH THE VALUES OF EACH CASE.
- IF THERE IS A MATCH, THE ASSOCIATED BLOCK OF CODE IS EXECUTED.
- IF THERE IS NO MATCH, THE DEFAULT CODE BLOCK IS EXECUTED.



```

<html>
<body>
<p> Switch case started </p>
<script type="text/javascript">

var grade='A';

switch(grade)
{
case 'A' : document.write("job " + grade + "<br/> ");
break;
case 'B' : document.write("job " + grade + " <br/> ");
break;
case 'C' : document.write("job " + grade + " <br/> ");
break;
case 'D' : document.write("job " + grade + " <br/> ");
break;
case 'E' : document.write("job " + grade + " <br/> ");
break;
case 'F' : document.write("job " + grade + " <br/> ");
break;
case 'G' : document.write("job " + grade + " <br/> ");
break;

default : document.write("Good job<br/>");

}

</script>
<p> Switch case stopped </p>
</body>
</html>

```