**JavaScript**

**Part-2**

**JavaScript Statements**

### **Example**

var x, y, z;    // Statement 1
x = 5;          // Statement 2
y = 6;          // Statement 3
z = x + y;      // Statement 4

## **JavaScript Programs**

* A **computer program** is a list of "instructions" to be "executed" by a computer.
* In a programming language, these programming instructions are called **statements**.
* JavaScript statements are composed of:Values, Operators, Expressions, Keywords, and Comments.
* A **JavaScript program** is a list of programming **statements**.

## **Semicolons ;**

* Semicolons separate JavaScript statements.
* Add a semicolon at the end of each executable statement:

var a, b, c;     // Declare 3 variables
a = 5;           // Assign the value 5 to a
b = 6;           // Assign the value 6 to b
c = a + b;       // Assign the sum of a and b to c

## **JavaScript White Space**

* JavaScript ignores multiple spaces. You can add white space to your script to make it more readable.

The following lines are equivalent:

var person = "Hege";
var person="Hege";

## **JavaScript Code Blocks**

* JavaScript statements can be grouped together in code blocks, inside curly brackets {...}.
* The purpose of code blocks is to define statements to be executed together.
* One place you will find statements grouped together in blocks, is in JavaScript functions:

### **Example**

function myFunction() {
  document.getElementById("demo1").innerHTML = "Hello Dolly!";
  document.getElementById("demo2").innerHTML = "How are you?";
}

## **JavaScript Keywords**

* JavaScript statements often start with a **keyword** to identify the JavaScript action to be performed.
* Visit our Reserved Words reference to view a full list of [JavaScript keywords](https://www.w3schools.com/js/js_reserved.asp).
* Here is a list of some of the keywords you will learn about in this tutorial:

|  |  |
| --- | --- |
| **Keyword** | **Description** |
| break | Terminates a switch or a loop |
| continue | Jumps out of a loop and starts at the top |
| debugger | Stops the execution of JavaScript, and calls (if available) the debugging function |
| do ... while | Executes a block of statements, and repeats the block, while a condition is true |
| for | Marks a block of statements to be executed, as long as a condition is true |
| function | Declares a function |
| if ... else | Marks a block of statements to be executed, depending on a condition |
| return | Exits a function |
| switch | Marks a block of statements to be executed, depending on different cases |
| try ... catch | Implements error handling to a block of statements |
| var | Declares a variable |

# JavaScript Syntax

## **JavaScript Values**

* The JavaScript syntax defines two types of values: Fixed values and variable values.
* Fixed values are called **literals**. Variable values are called **variables**.

## **JavaScript Literals**

The most important rules for writing fixed values are:

**Numbers** are written with or without decimals:

10.50

1001

**Strings** are text, written within double or single quotes:

"John Doe"

'John Doe'

## **JavaScript Variables**

* In a programming language, **variables** are used to **store** data values.
* JavaScript uses the var keyword to **declare** variables.
* An **equal sign** is used to **assign values** to variables.
* In this example, x is defined as a variable. Then, x is assigned (given) the value 6:

var x;

x = 6;

## **JavaScript Operators**

JavaScript uses **arithmetic operators** ( + - \* / ) to **compute** values:

(5 + 6) \* 10

## **JavaScript Comments**

* Not all JavaScript statements are "executed".
* Code after double slashes // or between /\* and \*/ is treated as a **comment**.

## **JavaScript Identifiers**

* Identifiers are names.
* In JavaScript, identifiers are used to name variables (and keywords, and functions, and labels).
* The rules for legal names are much the same in most programming languages.
* In JavaScript, the first character must be a letter, or an underscore (\_), or a dollar sign ($).
* Subsequent characters may be letters, digits, underscores, or dollar signs.

## **JavaScript is Case Sensitive**

* All JavaScript identifiers are **case sensitive**.
* The variables lastName and lastname, are two different variables:

var lastname, lastName;
lastName = "Doe";
lastname = "Peterson";