JavaScript Syntax

JavaScript is a scripting language. It is a lightweight, but powerful, programming language.

Syntax definition: "The principles by which sentences are constructed in a language."

The sentences of a programming language are called computer statements, or just statements.

JavaScript Literals

In a programming language, literals are constant values like 3.14.

Number literals can be written with or without decimals, and with or without scientific notation (e):

```
3.14

1001

123e5
```

String literals can be written with double or single quotes:

```
"John Doe"

'John Doe'
```

JavaScript Variables

In a programming language, variables are containers for storing information (data).

The equal sign (=) assigns a value to a named variable (just like in normal algebra):

```
x = 5

length = 6
```

JavaScript Operators

JavaScript uses operators to compute values (just like algebra):

```
5 + 6

a * b
```

JavaScript can assign computed values to named variables (just like algebra):

```
x = 5 + 6

y = x * 10
```

Expressions like **5 + 6**, and **x * 10**, are called **expression literals**.

JavaScript Statements

In HTML, JavaScript statements are written as sequences of "commands" to the HTML browser.

Statements are separated by semicolons:

```
x = 5 + 6;
y = x * 10;
```

JavaScript Keywords

A JavaScript statement often starts with a **keyword**. The **var** keyword tells the browser to create a new variable:

```
var x = 5 + 6;
var y = x * 10;
```

JavaScript Comments

Not all JavaScript statements are "commands". Anything after double slashes **//** is ignored by the browser:

```
// I will not be executed
```

JavaScript Data Types

JavaScript variables can hold many types of data: numbers, text strings, arrays, objects and much more:

```
var length = 16;                                       // Number
assigned by a number literal
var lastName = "Johnson";                              // String
assigned by a string literal
var cars = ["Saab", "Volvo", "BMW"];                   //
Array  assigned by an array literal
```

```
var person = {firstName:John, lastName:Doe};          // Object
assigned by an object literal
```

JavaScript Functions

JavaScript statements written inside a function, can be invoked many times (reused):

Invoke a function = Call upon a function (ask for the code in the function to be executed).

```
function myFunction(a, b) {
    return a * b;                                     // returns the
product of a and b
}
```

JavaScript Identifiers

All programming languages must **identify** variables, functions, and objects, with **unique** names.

These unique names are called **identifiers**.

Identifier names can contain letters, digits, underscores, and dollar signs, but cannot begin with a number.

Reserved words (like JavaScript keywords) cannot be used as identifiers.

JavaScript is Case Sensitive

In JavaScript all identifiers are case sensitive.

The variables **lastName** and **lastname**, are two different variables.

The functions **myFunction** and **myfunction**, are two different functions.

JavaScript does not interpret **Var;** as **var**.