## JavaScript Syntax or Programs or Statements

A **computer program** is a list of "instructions" to be "executed" by the computer. 지시 목록

JavaScript is a **programming language**.

|  |  |
| --- | --- |
|  | In HTML, JavaScript programs can be executed by the web browser. |

JavaScript **syntax** is the set of rules, how JavaScript programs are constructed. 작성, 이루어지다.

In a programming language, these program instructions are called **statements**. 문장

# JavaScript Statements구문

This statement tells the browser to write "Hello Dolly." inside an HTML element with id="demo":

### Example

document.getElementById("demo").innerHTML = "Hello Dolly.";

## JavaScript Programs

Most JavaScript programs contain many JavaScript statements.

The statements are executed, one by one, in the same order as they are written.

In this example, x, y, and z is given values, and finally z is displayed:

### Example

var x = 5;  
var y = 6;  
var z = x + y;  
document.getElementById("demo").innerHTML = z;

## Semicolons ;

Semicolons separate JavaScript statements.

Add a semicolon at the end of each executable statement:

a = 5;  
b = 6;  
c = a + b;

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_statements_semicolon1)

When separated by semicolons, multiple statements on one line are allowed:

a = 5; b = 6; c = a + b;

On the web, you might see examples without semicolons.   
Ending statements with semicolon is not required, but highly recommended.

추천(주선하다)

## 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: 동등한

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

A good practice is to put spaces around operators ( = + - \* / ): 연습

var x = y + z;

## JavaScript Line Length and Line Breaks

For best readability, programmers often like to avoid code lines longer than 80 characters. 피하다

If a JavaScript statement does not fit on one line, the best place to break it, is after an operator: 끈다.

### Example

document.getElementById("demo").innerHTML =  
"Hello Dolly.";

## JavaScript Code Blocks

JavaScript statements can be grouped together in code blocks, inside curly brackets {...}. 곱슬한 내부중괄호

The purpose of code blocks is to define statements to be executed together.

One place you will find statements grouped together in blocks, are in JavaScript functions:

### Example

function myFunction() {  
    document.getElementById("demo").innerHTML = "Hello Dolly.";  
    document.getElementById("myDIV").innerHTML = "How are you?";  
}

## JavaScript Keywords

JavaScript statements often start with a **keyword** to identify the JavaScript action to be performed. 수행하다.

Here is a list of some of the keywords you will learn about in this tutorial:

JavaScript keywords are reserved words. Reserved words cannot be used as names for variables. (예약어, 식별자, 키워드)

|  |  |
| --- | --- |
| **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 Statements자바스크립트문장(구문)

JavaScript statements are composed of:

Values, Operators, Expressions, Keywords, and Comments.

1. JAVA 주석
2. 코멘트는 자바스크립트 내용을 설명하기 위해서 사용한다.
3. 자바스크립트 코드에는 영향을 주지 못한다.

## Single Line Comments

### Example

// Change heading:  
document.getElementById("myH").innerHTML = "My First Page";  
// Change paragraph:  
document.getElementById("myP").innerHTML = "My first paragraph.";

## Multi-line Comments

/\*  
The code below will change  
the heading with id = "myH"  
and the paragraph with id = "myP"  
in my web page:  
\*/  
document.getElementById("myH").innerHTML = "My First Page";  
document.getElementById("myP").innerHTML = "My first paragraph.";

1. /\* \*/ /\* \*/
2. /\* /\* \*/ \*/ 이것은 예러이다.

//document.getElementById("myH").innerHTML = "My First Page";  
document.getElementById("myP").innerHTML = "My first paragraph.";

/\*  
document.getElementById("myH").innerHTML = "My First Page";  
document.getElementById("myP").innerHTML = "My first paragraph.";  
\*/

다음을 실행 시켜보자.

<!DOCTYPE html>

<html>

<head>

<!-- 주석입니다. -->

<script>

// 주석은 코드의 실행에 아무 영향을 미치지 않습니다.

/\*

alert('Hello JavaScript');

alert('Hello JavaScript');

alert('Hello JavaScript');

\*/

</script>

</head>

<body>

<!-- <h1>HTML 주석입니다.</h1> -->

</body>

</html>

## 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**.

상수 변수 변하기 쉬운

## JavaScript Literals

The most important rules for writing fixed values are:

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

10.50  
  
1001

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_syntax_numbers)

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

"John Doe"  
  
'John Doe'

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_syntax_strings)

## JavaScript Variables

In a programming language, **variables** are used to **store** data values.

JavaScript uses the **var**keyword to **define** variables.

An **equal sign** is used to **assign values** to variables. = 할당하다.

In this example, x is defined as a variable. Then, x is assigned (given) the value 6:

var x;  
  
x = 6;

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_syntax_variables)

## One Statement, Many Variables

var person = "John Doe", carName = "Volvo", price = 200;

var person = "John Doe",  
carName = "Volvo",  
price = 200;

## Value = undefined 라는 값이 있다.

## var carName;

## <!DOCTYPE html>

## <html><body>

## <h1>JavaScript Variables</h1>

## <p>A variable declared without a value will have the value undefined.</p>

## <p id="demo"></p>

## <script>

## var carName;

## document.getElementById("demo").innerHTML = carName;

## </script>

## </body>

## </html>

## Re-Declaring JavaScript Variables

If you re-declare a JavaScript variable, it will not lose its value.

The variable carName will still have the value "Volvo" after the execution of these statements:

### Example

var carName = "Volvo";  
var carName;

var x = 5 + 2 + 3;

var x = "John" + " " + "Doe";

var x = "5" + 2 + 3;

var x = 2+3+"5";

## 다음예제를 풀어보자.

## http://www.w3schools.com/js/exercise.asp?filename=exercise\_variables1

. [Exercise 1 »](http://www.w3schools.com/js/exercise.asp?filename=exercise_variables1)  .

Create a variable called **carName**, assign the value "**Volvo**" to it, and display it.

Hint: Use the var keyword to create a variable.

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

// Create the variable here

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

var carName = "Volvo";

document.getElementById("demo").innerHTML = carName;

</script>

</body>

</html>

[Exercise 2 »](http://www.w3schools.com/js/exercise.asp?filename=exercise_variables2)

Create a variable called **number**, assign the value **50** to it, and display it.

Hint: Use the var keyword to create a variable.

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

// Create the variable here

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

var number = 50;

document.getElementById("demo").innerHTML = number;

</script>

</body>

</html>

[Exercise 3 »](http://www.w3schools.com/js/exercise.asp?filename=exercise_variables3)

The code below should display "Volvo" - Fix it.

**Important**: Variables are case sensitive! The variables carName and carname are not the same.

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

var carName = "Volvo";

document.getElementById("demo").innerHTML = carname;

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

var carName = "Volvo";

document.getElementById("demo").innerHTML = carName;

</script>

</body>

</html>

22

[Exercise 4 »](http://www.w3schools.com/js/exercise.asp?filename=exercise_variables4)

Display the sum of 5 + 10, using two variables x and y.

Hint: Use the var keyword to create variables.

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

// Create the variables here

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

var x = 5;

var y = 10;

document.getElementById("demo").innerHTML = x + y;

</script>

</body>

</html>

[Exercise 5 »](http://www.w3schools.com/js/exercise.asp?filename=exercise_variables5)

Create a third variable called **z**, assign x + y to it, and display it.

Hint: var z = x + y.

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

var x = 5;

var y = 10;

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

var x = 5;

var y = 10;

var z = x + y;

document.getElementById("demo").innerHTML = z;

</script>

</body>

</html>

[Exercise 6 »](http://www.w3schools.com/js/exercise.asp?filename=exercise_variables6)

Use a **single** var keyword to create three variables with the following values:   
firstName = "John"   
lastName = "Doe"   
age = 35

Hint: Use the var keyword and separate the variables by comma.  
Assign the specified value to each variable.

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

// Create the variables here

document.getElementById("demo").innerHTML =

firstName + " " + lastName + " is " + age;

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<p id="demo">Display the result here.</p>

<script>

var firstName = "John", lastName = "Doe", age = 35;

document.getElementById("demo").innerHTML =

firstName + " " + lastName + " is " + age;

</script>

</body>

</html>

## 숫자 관련 예제..

## <!DOCTYPE html>

## <html><head> <script>

## var num1 = 10;

## var num2 = 30.4;

## alert(10 + 4);

## alert(1.03 + 1.4);

## alert(10 / 4.0);//2.5

## alert(10 / 4); //2.5

## alert(num1 + num2);

## alert(num1 + 1.3);

## </script></head><body>

## </body>

## </html>

## 문자열관련 예제

## <script>

## alert('Hello JavaScript..!');

## alert("Hello JavaScript..!");

## //이스케이프

## alert('동해물과 백두산이\n마르고 닳도록');

## alert("This is \"string\"");

## alert('This is \'string\'');

## //문자열은""를 사용하자.

## alert('가나라다'+'마바사'+'아자차카'); //문자열 연산이 가능한다.

## </script>

## 숫자 + 문자 예제

## <!DOCTYPE html>

## <html>

## <head>

## <script>

## // 1번

## alert('52 + 273');

## // 2번

## alert(52 + 273);

## // 3번

## alert('52' + 273);

## // 4번

## alert(52 + '273');

## // 5번

## alert('52' + '273');

## alert('52 \* 273');

## alert(52 \* 273);

## alert('52' \* 273);//문자가 숫자로 변환

## alert(52 \* '273');

## alert('52' \* '273');

## </script>

## </head>

## <body>

## </body>

## </html>

## 그밖의 자료형 표시 방법에 대해서 확인해보자.

## <script language="javascript" type="text/javascript">

## // 변수 선언

## var a;

## // 변수 초기화(할당)

## a = "안녕<br />";

## // 변수 참조(사용)

## document.write(a);

## document.write(a);

## //자료형이 하나밖에 없다. 일반적으로 자동형변환 되지만 원하는 결과를 보장받고 싶으면 사용자가 형변환하여 사용하여야한다.

## // 변수를 선언과 동시에 초기화

## var intNum = 100; // 정수형

## var dblSu = 12.34; // 실수형

## var blnFlag = true; // 불린값(참 또는 거짓) : true/false

## var objNothing = null; // 널값 : 아무것도 아닌 값(빈값(Empty)과 다르다)

## var objthing = {}; // 값을 가지고 있는 객체

## //var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

## var strHello = "안녕하세요."; // 문자열 : 큰 따옴표 또는 작은 따옴표로 묶는다.

## var functionVar = function () { };

## //functionVar.toString = "hello";

## //functionVar.toString = function () { return "hello"; };

## //var cars = ["Saab", "Volvo", "BMW"];

## // 사용(참조)

## document.write( intNum + intNum + "<br />" ); // 200

## document.write( dblSu + intNum + "<br />" ); // 112.34

## document.write( blnFlag + "<br />" ); // true

## document.write(objNothing + "<br />"); // null

## document.write(objthing + "<br />"); // null

## document.write(strHello + "<br />"); // 안녕하세요.

## document.write(functionVar + "<br />"); // 안녕하세요.

## /\*

## --화면에는 문자열 밖에 찍을 수 없다 선언된 자료형들이 자동형 변환되어 찍힌 것이다.

## --어떻게 자동형변환되는가?

## --문자로 자동형변환할때는 toString()함수가 실행된다.\*/

## </script>

## <!DOCTYPE html>

## <html>

## <head>

## <script>

## var answer = "It's alright"; // Single quote inside double quotes

## var answer = "He is called 'Johnny'"; // Single quotes inside double quotes

## var answer = 'He is called "Johnny"'; // Double quotes inside single quotes

## // 변수를 선언합니다.

## var cup = 'Coffee';

## cup = 'green Tea';

## cup = 'Water';

## // 출력합니다.

## alert('Drink ' + cup + '..!');

## //-- 변수를 선언합니다.

## var favoriteFood = '김치 찌개';

## var favoriteFood = '라면';

## var favoriteFood = '냉면';

## // 출력합니다.

## alert(favoriteFood);

## //-- 변수를 선언합니다.

## var alert = 'red Alert';

## // 출력합니다.

## alert(alert);

## 

## </script>

## </head>

## <body>

## </body>

## </html>

## 자바같은 언어는 명확한 결과를 요구하지만 자바스크립트 같은 경우

## 브라우저마다 결과가 다를수 있으니 명확한값보다 이런값도 나올수

## 있다라고 생각하면된다.

## <!DOCTYPE html>

## <html>

## <head>

## <script>

## alert(0);

## alert(NaN);

## alert('');

## alert(null);

## alert(undefined);

## //자동 bool형 변환

## alert(!!0);

## alert(!!NaN);

## alert(!!'');

## alert(!!null);

## alert(!!undefined);

## alert(!!23);

## alert(!!'asdfs');

## alert(Boolean(0));

## alert(Boolean(NaN));

## alert(Boolean(''));

## alert(Boolean(null));

## alert(Boolean(undefined));

## </script>

## </head>

## <body>

## </body>

## </html>