JavaScript Syntax

- description : JavaScript Syntax
- author : 오션
- email : shlim@repia.com
- lastupdate : 2021-04-01

JavaScript 구문Syntax은 JavaScript 프로그램이 구성되는 방식인, 규칙 집합입니다.

```
var x, y, z;     // Declare Variables
x = 5; y = 6;    // Assign Values
z = x + y;       // Compute Values
```

JavaScript Values

JavaScript 구문은 두 가지 유형의 값values을 정의합니다.

- 고정 값 (Fixed values)
- 변수 값 (Variable values)

고정 값을 리터럴**Literals**이라고 합니다.

변수 값을 변수**Variables**라고 합니다.

JavaScript Literals

고정 값에 대한 가장 중요한 두 가지 구문 규칙은 다음과 같습니다.

1. **숫자Numbers**는 소수를 포함하거나 포함하지 않고 작성됩니다.

```
10.50

1001
```

2. **문자열 (스트링Strings)**은 큰 따옴표 또는 작은 따옴표로 묶인 텍스트입니다.

```
"John Doe"

'John Doe'
```

JavaScript Variables

프로그래밍 언어에서, **변수**variables는 데이터 값을 **저장**store하는 데 사용됩니다.

JavaScript는 var 키워드를 사용하여 변수를 **선언**declare**합니다.**

등호equal sign는 변수에 **값을 할당**assign values하는 데 사용됩니다.

하기 예제에서 x는 변수로 정의됩니다. 그런 다음 x에 값 6이 할당됩니다(주어집니다).

```
var x;

x = 6;
```

JavaScript Operators

JavaScript는 **산술 연산자**arithmetic operators (+ - * /)를 사용하여 값을 계산합니다.

```
(5 + 6) * 10
```

JavaScript는 **할당 연산자**assignment operator (=)를 사용하여 변수에 값을 **할당**assign합니다.

```
var x, y;
x = 5;
y = 6;
```

JavaScript Expressions

표현식expression은 값(values), 변수(variables) 및 연산자(operators)의 조합이며, 값을 계산합니다.

계산computation을 평가(evaluation)라고 합니다.

예를 들어, 5 * 10은 50으로 평가됩니다:

```
5 * 10
```

표현식에는 특정 변수 값도 포함될 수 있습니다.

```
x * 10
```

값은 숫자 및 문자열과 같은 다양한 유형이 될 수 있습니다.

예를 들어, “John”+ “”+ “Doe”는 “John Doe”로 평가됩니다.

JavaScript Keywords

JavaScript **키워드**keyword는 수행할 작업을 식별하는 데 사용됩니다.

`var` 키워드는 브라우저에게 변수를 만들도록 지시합니다:

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

JavaScript Comments

모든 JavaScript 스테이트먼트가 " 실행 " 되는 것은 아닙니다.

이중 슬래시double slashes `//` 뒤 또는 `/*`와 `*/` 사이의 코드는 **주석**comment으로 처리됩니다.

주석은 무시되며 실행되지 않습니다.

```
var x = 5;      // I will be executed

// var x = 6;   // I will NOT be executed
```

JavaScript Identifiers

식별자Identifier는 이름입니다.

JavaScript에서, 식별자는 변수 (및 키워드, 함수 및 라벨)의 이름을 지정하는 데 사용됩니다.

공식 이름에 대한 규칙은 대부분의 프로그래밍 언어에서 거의 동일합니다.

JavaScript에서, 첫 번째 철자는 문자, 밑줄 (_) 또는 달러 기호 ($) 이어야 합니다.

이어지는 철자는 문자, 숫자, 밑줄 또는 달러 기호 일 수 있습니다.

첫 번째 철자로 숫자를 사용할 수 없습니다.
이런 식으로 JavaScript는 식별자와 숫자를 쉽게 구별 할 수 있습니다.

JavaScript is Case Sensitive

모든 JavaScript 식별자는 **대소 문자를 구분**case sensitive합니다.

변수 `lastName` 과 `lastname` 는 두 개의 다른 변수입니다.

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

```
lastname = "Peterson";
```

JavaScript는 **VAR** 또는 **Var**을 키워드 **var**로 해석하지 않습니다.

JavaScript and Camel Case

역사적으로, 프로그래머는 여러 단어를 하나의 변수 이름으로 결합하는 다양한 방법을 사용했습니다.

하이픈Hyphens:

- first-name, last-name, master-card, inter-city
- JavaScript에서는 하이픈을 사용할 수 없습니다. 하이픈은 뺄셈subtractions을 위해 예약되어 있습니다.

밑줄Underscore:

- first_name, last_name, master_card, inter_city.

Upper Camel Case (Pascal Case):

- FirstName, LastName, MasterCard, InterCity.

Lower Camel Case:

- JavaScript 프로그래머는 소문자lowercase letter로 시작하는 카멜 케이스를 사용하는 경향이 있습니다.
- firstName, lastName, masterCard, interCity.

JavaScript Character Set

JavaScript는 **유니코드 (Unicode)** 문자 집합을 사용합니다.

유니코드는 세계의 거의 모든 문자characters, 구두점punctuations, 및 기호symbols를 포함합니다.

자세히 살펴 보려면 HTML Unicode (UTF-8) Reference를 살펴보십시오.

오션, Javascript, Syntax

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:javascript:javascript_note:js_syntax&rev=1617244922

Last update: **2022/03/10 19:52**