JavaScript Variables

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

JavaScript 변수는 데이터 값을 저장하기 위한 컨테이너입니다.

하기 예제에서, x, y 및 z는 var 키워드로 선언한 변수들 입니다.

```
var x = 5;
vat y = 6;
var z = x + y;
```

상기 예제에서 다음을 기대할 수 있습니다.

- x는 값 5를 저장합니다.
- y는 값 6을 저장합니다.
- z는 값 11을 저장합니다.

Using let and const (ES6)

2015년 이전에는 var 키워드를 사용하는 것이 JavaScript 변수를 선언하는 유일한 방법이었습니다.

2015 버전의 JavaScript (ES6)에서는 const 키워드를 사용하여 재할당할 수 없는 변수를 정의하고, let 키워드를 사용하여 제한된 범위의 변수를 정의할 수 있습니다.

이러한 키워드 간의 차이점을 설명하는 것은 약간 복잡하고 이전 브라우저에서는 지원되지 않기 때문에, 본 튜토리얼의 첫 번째 부분에서는 var를 가장 자주 사용합니다.

Safari 10 및 Edge 14는 ES6를 완벽하게 지원하는 최초의 브라우저입니다.

Much Like Algebra

하기 예제에서, price1, price2, 그리고 total은 변수입니다.

```
var price1 = 5;
```

```
var price2 = 6;
var total = price1 + price2;
```

대수학과 마찬가지로 프로그래밍에서, 값을 유지하기 위해 price1과 같은 변수를 사용합니다.

대수학과 마찬가지로 프로그래밍에서, 표현식expressions에 변수를 사용합니다(total = price1 + price2).

상기 예제에서 합계를 11로 계산할 수 있습니다.

JavaScript 변수는 데이터 값을 저장하기 위한 컨테이너입니다.

JavaScript Identifier

모든 JavaScript **변수variables**는 **고유한 이름unique names**으로 *8식별되어야identified **합니다.**

이러한 고유한 이름을 식별자identifiers**라고 합니다.**

식별자는 짧은 이름 (예 : x 및 y)이거나 더 설명적인 이름 (age, sum, totalVolume) 일 수 있습니다.

변수 (고유 식별자) 이름을 구성하는 일반적인 규칙은 다음과 같습니다.

*** 이름에는 문자, 숫자, 밑줄 및 달러 기호가 포함될 수 있습니다. * 이름은 문자로 시작해야 합니다. * 이름은 $ 및 _로 시작할 수도 있지만, 본 튜토리얼에서는 사용하지 않습니다. * 이름은 대소문자를 구분합니다 (y와 Y는 다른 변수 임). * 예약어 (예 : JavaScript 키워드)는 이름으로 사용할 수 없습니다.**
JavaScript 식별자는 대소 문자를 구분합니다.

=====The Assignment Operator===== JavaScript에서 등호 (😃는 " 같음 " 연산자가 아니라 할당 **assignment연산자입니다.**

이것은 대수학과 다릅니다. 다음은 대수학에서 의미가 없습니다.
<code javascript> x = x + 5 </code>
그러나 JavaScript에서는 완벽하게 의미가 있습니다. 이것은 x + 5의 값을 x에 할당한다는 의미입니다.

(x + 5의 값을 계산하고 그 결과를 x에 넣습니다. x의 값은 5 씩 증가합니다.)
" 같음 " 연산자는 JavaScript에서 ==처럼 작성됩니다.

=====JavaScript Data Types==== JavaScript 변수는 100과 같은 숫자와 "John Doe"와 같은 텍스트 값을 포함할 수 있습니다.

프로그래밍에서, 텍스트 값을 텍스트 문자열text strings 이라고 합니다.

JavaScript는 다양한 유형의 데이터를 처리할 수 있지만, 지금은 숫자와 문자열만 생각하면 됩니다.

문자열은 큰 따옴표 또는 작은 따옴표 안에 작성됩니다. 숫자는 따옴표 없이 작성됩니다.

따옴표 안에 숫자를 넣으면, 텍스트 문자열로 처리됩니다.

<code javascript> var pi = 3.14; var person = "John Doe"; var answer = 'Yes I am!';
</code>

=====Declaring (Creating) JavaScript Variables===== JavaScript에서 변수를 만드는 것을 변수 “ 선언 ” 이라고 합니다.

var 키워드를 사용하여 JavaScript 변수를 선언합니다.

<code javascript> var carName; </code>
선언 후 변수에는 값이 없습니다 (기술적으로는, undefined 값을 가지고 있습니다)

변수에 값을 할당assign하기 위해 등호 (😃를 사용합니다.

<code javascript> carName = “Volvo”; </code>
변수를 선언할 때, 변수에 값을 할당할 수도 있습니다.

<code javascript> var carName = “Volvo”; </code>
하기 예제에서는, carName이라는 변수를 만들고, 여기에 “Volvo”값을 할당합니다.

그런 다음 id = “demo”로 HTML 단락 내부의 값을 “ 출력 ” 합니다.

<code javascript> <script> var carName = “Volvo” document.getElementById(“demo”).innerHTML = carName; </script> </code>
스크립트 시작 부분에 모든 변수를 선언하는 것이 좋은 프로그래밍 방법입니다.

=====One Statement, Many Variables===== 하나의 스테이트먼트에서 여러 변수를 선언할 수 있습니다.

var로 스테이트먼트를 시작하고, 쉼표comma로 변수를 구분합니다.

<code javascript> var person = “John Doe”, carName = “Volvo”, price = 200; </code>
선언은 여러 줄에 걸쳐있을 수 있습니다.

<code javascript> var person = “John Doe”, carName = “Volvo”, price = 200; </code>
=====Value = undefined===== 컴퓨터 프로그램에서, 변수는 종종 값이 없는 상태로 선언됩니다. 값은 계산해야 하는 항목이거나, 사용자 입력과 같이 나중에 제공될 항목일 수 있습니다.

값 없이 선언된 변수는 undefined 값을 갖습니다.

carName 변수는 하기의 표현식을 실행한 후 undefined 값을 갖습니다.

<code javascript> var carName; </code>
=====Re-Declaring JavaScript Variables===== JavaScript 변수를 다시 선언하면, 변수는 값이 손실되지 않습니다.

변수 carName은 하기의 표현식들을 실행한 후에도 여전히 “Volvo”값을 갖습니다.

<code javascript> var varName = “Volvo”; var carName; </code>
=====JavaScript Arithmetic===== 대수학과 마찬가지로, = 및 +와 같은 연산자를 사용하여 JavaScript 변수로 연산arithmetic을 수행할 수 있습니다.

<code javascript> var x = 5 + 2 + 3; </code>
문자열을 추가할 수도 있지만, 문자열은 <code javascript> var x = 5; vat y = 6; var z = x + y; </code>
상기 예제에서 다음을 기대할 수 있습니다.
*** x는 값 5를 저장합니다. * y는 값 6을 저장합니다. * z는 값 11을 저장합니다.**
=====Using let and const (ES6)===== 2015년 이전에는 var 키워드를 사용하는 것이

JavaScript 변수를 선언하는 유일한 방법이었습니다.

2015 버전의 JavaScript (ES6)에서는 `const` 키워드를 사용하여 재할당할 수 없는 변수를 정의하고, `let` 키워드를 사용하여 제한된 범위의 변수를 정의할 수 있습니다.

이러한 키워드 간의 차이점을 설명하는 것은 약간 복잡하고 이전 브라우저에서는 지원되지 않기 때문에, 본 튜토리얼의 첫 번째 부분에서는 var를 가장 자주 사용합니다.

Safari 10 및 Edge 14는 ES6를 완벽하게 지원하는 최초의 브라우저입니다.

=====Much Like Algebra===== 하기 예제에서, `price1`, `price2`, 그리고 `total`은 변수입니다.

<code javascript> var price1 = 5; var price2 = 6; var total = price1 + price2; </code>
대수학과 마찬가지로 프로그래밍에서, 값을 유지하기 위해 price1과 같은 변수를 사용합니다.

대수학과 마찬가지로 프로그래밍에서, 표현식expressions에 변수를 사용합니다(total = price1 + price2).

상기 예제에서 합계를 11로 계산할 수 있습니다.

JavaScript 변수는 데이터 값을 저장하기 위한 컨테이너입니다.

=====JavaScript Identifier===== 모든 JavaScript 변수variables**는** 고유한 이름unique names**으로 *8식별되어야**identified** 합니다.

이러한 고유한 이름을 **식별자identifiers**라고 합니다.

식별자는 짧은 이름 (예 : x 및 y)이거나 더 설명적인 이름 (age, sum, totalVolume) 일 수 있습니다.

변수 (고유 식별자) 이름을 구성하는 일반적인 규칙은 다음과 같습니다.

- 이름에는 문자, 숫자, 밑줄 및 달러 기호가 포함될 수 있습니다.
- 이름은 문자로 시작해야 합니다.
- 이름은 $ 및 _로 시작할 수도 있지만, 본 튜토리얼에서는 사용하지 않습니다.
- 이름은 대소문자를 구분합니다 (y와 Y는 다른 변수 임).
- 예약어 (예 : JavaScript 키워드)는 이름으로 사용할 수 없습니다.

JavaScript 식별자는 대소 문자를 구분합니다.

The Assignment Operator

JavaScript에서 등호 (😃는 " 같음 " 연산자가 아니라" **할당assignment**연산자입니다.

이것은 대수학과 다릅니다. 다음은 대수학에서 의미가 없습니다.

```
x = x + 5
```

그러나 JavaScript에서는 완벽하게 의미가 있습니다. 이것은 x + 5의 값을 x에 할당한다는 의미입니다.

(x + 5의 값을 계산하고 그 결과를 x에 넣습니다. x의 값은 5 씩 증가합니다.)

“ 같음 ” 연산자는 JavaScript에서 ==처럼 작성됩니다.

JavaScript Data Types

JavaScript 변수는 100과 같은 숫자와 “John Doe”와 같은 텍스트 값을 포함할 수 있습니다.

프로그래밍에서, 텍스트 값을 텍스트 문자열text strings 이라고 합니다.

JavaScript는 다양한 유형의 데이터를 처리할 수 있지만, 지금은 숫자와 문자열만 생각하면 됩니다.

문자열은 큰 따옴표 또는 작은 따옴표 안에 작성됩니다. 숫자는 따옴표 없이 작성됩니다.

따옴표 안에 숫자를 넣으면, 텍스트 문자열로 처리됩니다.

```
var pi = 3.14;
var person = "John Doe";
var answer = 'Yes I am!';
```

Declaring (Creating) JavaScript Variables

JavaScript에서 변수를 만드는 것을 변수 “ 선언 ” 이라고 합니다.

var 키워드를 사용하여 JavaScript 변수를 선언합니다.

```
var carName;
```

선언 후 변수에는 값이 없습니다 (기술적으로는, undefined 값을 가지고 있습니다)

변수에 값을 **할당**assign하기 위해 등호 (😃를 사용합니다.

```
carName = "Volvo";
```

변수를 선언할 때, 변수에 값을 할당할 수도 있습니다.

```
var carName = "Volvo";
```

하기 예제에서는, carName이라는 변수를 만들고, 여기에 “Volvo”값을 할당합니다.\
그런 다음 id = “demo”로 HTML 단락 내부의 값을 “ 출력 ” 합니다.

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

<script>
var carName = "Volvo"
document.getElementById("demo").innerHTML = carName;
</script>
```

스크립트 시작 부분에 모든 변수를 선언하는 것이 좋은 프로그래밍 방법입니다.

One Statement, Many Variables

하나의 스테이트먼트에서 여러 변수를 선언할 수 있습니다.

var로 스테이트먼트을 시작하고, **쉼표**[comma]로 변수를 구분합니다.

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

선언은 여러 줄에 걸쳐있을 수 있습니다.

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

Value = undefined

컴퓨터 프로그램에서, 변수는 종종 값이 없는 상태로 선언됩니다.
값은 계산해야 하는 항목이거나, 사용자 입력과 같이 나중에 제공될 항목일 수 있습니다.

값 없이 선언된 변수는 undefined 값을 갖습니다.

carName 변수는 하기의 표현식을 실행한 후 undefined 값을 갖습니다.

```
var carName;
```

Re-Declaring JavaScript Variables

JavaScript 변수를 다시 선언하면, 변수는 값이 손실되지 않습니다.

변수 carName은 하기의 표현식들을 실행한 후에도 여전히 “Volvo”값을 갖습니다.

```
var varName = "Volvo";
var carName;
```

JavaScript Arithmetic

대수학과 마찬가지로, = 및 +와 같은 연산자를 사용하여 JavaScript 변수로 연산[arithmetic]을 수행할 수 있습니다.

```
var x = 5 + 2 + 3;
```

문자열을 추가할 수도 있지만, 문자열은 연결됩니다.

```
var x = "John" + " " + "Doe";
```

```
var x = "5" + 2 + 3;
```

숫자를 따옴표로 묶으면 나머지 숫자는 문자열로 처리되고 연결됩니다.

```
var x = 2 + 3 + "5";
```

JavaScript Dollar Sign $

JavaScript 식별자 (이름)는 다음으로 시작해야 합니다.

- 문자 (A-Z 또는 a-z)
- 달러 기호 ($)
- 또는 밑줄 (_)

JavaScript는 달러 기호를 문자로 취급하므로, $를 포함하는 식별자는 유효한 변수 이름입니다.

```
var $$$ = "Hello World";
var $ = 2;
var $myMoney = 5;
```

달러 기호를 사용하는 것은 %5JavaScript에서 그다지 일반적이지 않지만, 전문 프로그래머는 종종 JavaScript 라이브러리의 __주 함수__^{main function}에 대한 별칭^{alias}으로 사용합니다.\\ \\ 예를 들어 JavaScript 라이브러리 jQuery에서, 주 함수 ''$''는 HTML 요소를 선택하는 데 사용됩니다.\\ jQuery에서 ''$(“p”);%%''는 “ 모든p 요소 선택 ” 을 의미합니다.

오션, Javascript, Variables

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_variables&rev=1617254684

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