

자바스크립트 (Javascript)

✓ 원리를 알면 IT가 맛있다

JavaScript for Beginners

chapter 02.

식별자와 데이터형

- 식별자의 개념 및 예약어에 관하여 학습한다.
- 자바스크립트에서 사용 가능한 데이터형에 관하여 학습한다.
- 변수의 개념 및 사용법에 관하여 학습한다.
- 데이터 형변환
- typeof 연산자를 이용한 데이터형 검사

□ 1) 식별자(identifier)

○ 식별자는 자바스크립트 코드 내에서 사용되는 각각의 단어를 의미한다.

○ 식별자 종류 2가지

a. 시스템 정의 식별자

자바스크립트가 필요에 의해서 먼저 정의한 식별자로서 보통 ‘예약어’, ‘키워드’라고 부른다.

[JavaScript에서의 예약어]

break	case	catch	continue	default	delete
do	else	finally	for	function	if
in	instanceof	new	return	switch	this
throw	try	typeofvar	void	while	with

b. 사용자 정의 식별자

자바스크립트 개발자가 필요에 의해서 정의한 식별자로서 변수, 함수, 생성자, 객체 정의시 사용할 이름을 의미한다.

○ 사용자 정의 식별자 명명규칙

- 첫 문자는 반드시 영문자, `_`, `$` 문자로 시작되어야 한다. 그 다음부터는 숫자와 영문자를 혼합해서 사용 가능하다.
- 자바스크립트는 대소문자를 구별한다.
- 예약어는 사용 불가

[권장되는 표기법]

기술 방법	개요	예
camelCase 기법	앞 단어 첫 문자는 소문자, 그 이후의 단어의 첫 문자는 대문자	lastName
Pascal 기법	모든 단어의 첫 문자는 대문자	LastName
언더스코프 기법	모든 단어의 첫 문자는 소문자, 단어 간은 '_'로 연결	last_name

- 데이터형은 자바스크립트 언어가 처리할 수 있는 데이터 종류를 의미한다.
- 자바스크립트의 데이터형 (Data Type) 2가지
 - 기본(원시) 데이터형 (primitive data type : PDT)
 - 수치 데이터 : 정수와 실수 (“number” 로 관리)
 - 문자 데이터 : 문자와 문자열 (“string” 로 관리)
 - “문자” 또는 ‘문자’ 로 표현.
Wn, Wt, WW, W'. W” (escape 문자)
 - 논리 데이터 : true 또는 false (“boolean” 로 관리)
 - undefined : 정의되지 않음. (변수에 초기화되지 않은 상태)
 - null : 객체 없음 (초기화는 된 상태) - Infinite (양의 무한대)
 - 참조 데이터형 (reference data type : RDT)
 - 객체 형(배열 포함) 과 함수형.
 - 객체표현방법은 {p1:10 , p2:20}
 - 배열표현방법은 [1 ,2 , 3 , 4]

□ 2) 자바스크립트의 데이터형(Data Type)

분류	데이터형	개요
기본형	수치형(number)	$\pm 4.94065645841246544 \times 10^{-324} \sim \pm 1.79769313486231570 \times 10^{308}$
	문자열형(string)	작은 따옴표/큰 따옴표로 감싸인 0 개 이상의 문자 집합
	논리형(boolean)	true(참)/false(거짓)
	특수형(null/undefined)	값이 미 정의된 것을 나타냄
참조형	배열(array)	데이터의 집합(각 요소에는 인덱스 번호로 접근 가능)
	객체(object)	데이터의 집합(각 요소에는 이름으로 접근 가능)
	함수(function)	일련의 처리(절차)의 집합

- 프로그램에서 사용하는 데이터(리터럴)를 저장하기 위한 용도로 사용된다.
- 저장된 데이터는 언제든지 변경이 가능하기 때문에 ‘변경이 가능한 수’ 즉, 변수라고 부른다.
변경이 불가능한 수는 ‘상수’라고 부른다.
- 기본형 데이터를 저장하면 ‘기본형 변수’ 라고 하고, 참조형 데이터를 저장하면 ‘참조형 변수’라고 한다.

○ 변수 사용 방법

가. 변수 선언

`var 변수명;` // 파싱단계에서는 undefined 값이 할당된다.

나. 초기화

`변수명 = 값;` //런타임시에 지정된 값으로 할당된다.

○ 특징

- 데이터형을 지정하지 않는다.

(실행단계에서 데이터형이 지정되기 때문에 데이터 종류에 상관없이 저장가능하다.)

- 함수단위로 **scope**가 정해진다. (블록단위 **scope**가 아니다.)

`var` 를 지정하지 않으면 전역scope(전역객체:window)에 동적으로 변수가 정의된다.

(즉, `var`가 없으면 변수가 정의되는 것은 파싱 단계가 아니고 런타임에 동적 정의됨)

- 데이터 타입을 철저하게 검사하지 않기 때문에 데이터 타입 변환이 자유롭다.
즉, 변수에 할당되는 값에 따라서 변수의 타입이 결정된다.
(실행시 변수에 저장되는 데이터형에 의해서 결정된다.)

예>

```
var test = 10; // number 타입
```

```
test = “홍길동”; // string 타입
```

```
test = true; // boolean 타입
```

□ 3 변수 (variables)

- * 함수 단위로 scope가 결정된다. (블록 단위 scope가 아니다)

```
function outputNumbers(count) {  
  for (var i = 0; i < count; i++) {  
    console.log(i);  
  }  
  console.log(">>" + i); // 5  
}
```

outputNumbers(5);

```
/* 익명 함수를 통해 블록 스코프를 흉내 내서 이 문제를 해결할 수 있습니다.  
   이를 '즉시 호출 함수'라고 부르기도 합니다  
   이 테크닉은 전역 스코프에 추가되는 변수나 함수의 수를 제한하는 용도로 자주 사용됩니다  
*/  
  
function outputNumbers(count) {  
  (function() {  
    for (var i = 0; i < count; i++) {  
      console.log(i);  
    }  
  })();  
  
  console.log(">>" + i); // Error!!  
  
}
```

outputNumbers(5);

즉시 호출 함수

- 데이터가 필요에 의해서 자동으로 형변환 된다.

가. *,/,- 사용하는 경우 (+제외)

예> var test = "100" * 2; ➔ 200

나. 불린값으로 변환

다음과 같은 5가지 데이터는 필요에 의해서 false 값 으로 변환된다. 역으로 5가지 이외의 데이터는 필요에 의해서 true 값으로 변환된다.

- 0

- ''

- NaN

- null

- undefined

```
예> var n = 0;
    if(n) {} // false 로 동작
    var name = "홍길동";
    if( name ) { } //true로 동작
```

○ 실행중에 변수에 설정된 데이터형을 검사하기 위한 방법이다.

문법:

typeof (변수) 또는 typeof 변수

결과는 문자열로 출력된다.

```
예> var name = "홍길동"; ➔ if( typeof (name) == 'string' ) {}  
    var age = 20; ➔ if( typeof age == 'number' ) {}  
    var array = [1,2]; ➔ if( typeof array == 'object' ) {}  
    var obj = {}; ➔ if( typeof obj == 'object' ) {}  
    var a = false; ➔ if( typeof a == 'boolean' ) {}  
    var b = null; ➔ if( typeof b == 'object' ) {}  
    var c; ➔ if( typeof c == 'undefined' ) {}  
    var k = function(){}; ➔ if( typeof k == 'function' ) {}
```

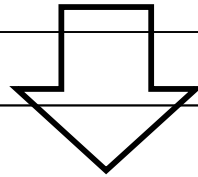
□ 6 / 변수의 유효범위와 끌어 올림(hoisting)

- 자바스크립트에서 함수의 유효범위는 이 함수 안에서 선언된 모든 변수가 함수 전체에 걸쳐 유효하다는 의미이다.

즉, 변수가 선언되기 전에도 유효하다. 이런 자바스크립트의 특징을 비공식적으로 ‘끌어올림’이라고 부른다.

=> 자바스크립트 코드는 함수안에 있는 모든 변수를 함수 맨 꼭대기로 ‘끌어올린’것 처럼 동작한다. 선언만 되고 초기화는 아직 안됨.

```
var scope = "global";  
function f(){  
    console.log( scope ); // undefined, 같은 이름의 전역이 지역에 감춰짐.  
    var scope = "local";  
    console.log(scope); // local
```



```
function f(){  
    var scope;  
    console.log( scope );  
    scope = "local";  
    console.log(scope);
```

- 식별자 개념 및 종류
- 예약어
- 자바스크립트의 데이터형 종류
- 변수 사용법 및 특징
- 데이터 형변환
- typeof 연산자를 이용한 데이터형 검사



Thank you
