

자바스크립트 (Javascript)

✓ 원리를 알면 IT가 맛있다

JavaScript for Beginners



chapter 03.

연산자

- 산술연산자
- 대입연산자
- 비교연산자 및 특징
- 논리연산자 및 특징
- 증감연산자
- 3항연산자
- 이외 연산자 및 특징

○ 산술 연산자

- 4칙 연산(+, -, *, /)과 나머지 값을 구하는 연산자(%)를 말한다.

[표 2-9] 산술 연산자의 종류

구분	연산자	의미
산술 연산자	+	더하기
	-	빼기
	*	곱하기
	/	나누기
	%	나머지 값 구하기

주의할 점은 문자열과 다른 데이터형이 + 연산을 하면 문자열이 연결된다.

예> var test = "ABCD" + 23; ➔ "ABCD23"

var test2 = "30" + 10; ➔ "3010"

○ 대입 연산자

- 특정한 상수 값이나 변수 값 또는 객체를 변수에 전달하여 기억시킬 때 사용하는 연산자이다.

[표 2-10] 대입 연산자의 종류

구분	연산자	의미
대입 연산자	=	연산자를 중심으로 오른쪽 변수값을 왼쪽 변수에 대입한다.
	+=	왼쪽 변수에 더하면서 대입한다.
	-=	왼쪽 변수값에서 빼면서 대입한다.
	*=	왼쪽 변수에 곱하면서 대입한다.
	/=	왼쪽 변수에 나누면서 대입한다.
	%=	왼쪽 변수에 나머지 값을 구하면서 대입한다.

○ 비교 연산자(관계 연산자)

- 변수나 상수의 값을 비교할 때 쓰이는 연산자로서 결과가 항상 true 또는 false인 논리값(boolean)이어야 한다.

[표 2-11] 비교 연산자의 종류

구분	연산자	의미
비교 연산자	>	크다.
	<	작다.
	>=	크거나 같다.
	<=	작거나 같다.
	==	피연산자들의 값이 같다.
	!=	피연산자들의 값이 같지 않다.

○ 비교 연산자

가. `==` , `!=`

equal 연산자로서 필요하면 형 변환하고 비교한다.

```
예> var w = 10;  
    var w2 = "10";  
    console.log( w == w2 ); // true
```

나. `===` , `!==`

identical 연산자로서 형 변환하지 않고 비교한다.

```
예> var w = 10;  
    var w2 = "10";  
    console.log( w === w2 ); // false
```

```
var x ;  
if( x === undefined ){ }
```

☐ 비교: 여기서는 보통의 균등 연산자(==) 보다는 엄격한 균등 연산자(===)를 사용해야 한다. 엄격 균등이 아닐 때 `x == undefined` 는 `x` 가 `null` 인지도 체크하기 때문이다. `null` 은 `undefined` 와 동일하지 않다.

○ 논리 연산자

- 기본적으로 true나 false인 논리 값을 가지고 연산하는 연산자이다.

[표 2-12] 논리 연산자의 종류 (1)

구분	연산자	의미	설명
논리 연산자	&&	and(논리곱)	주어진 조건들이 모두 true일 때만 true를 나타낸다.
		or(논리합)	주어진 조건들 중 하나라도 true이면 true를 나타낸다.
	!	not(부정)	true는 false로 false는 true로 나타낸다.

○ 주의할 점

자바스크립트에서는 다른 프로그래밍언어와 다르게 논리값만 논리연산자를 사용하지 않고 다른 데이터도 논리 연산자를 사용할 수 있다.

이유는 숫자나 문자열등이 논리값으로 자동 형 변환 될 수 있기 때문이다.
또한 최종적으로 반환되는 값은 true/false값이 아닌 좌측 또는 우측 피연산자의 최종 평가값이다.

좌객체 || 우객체

==> 좌객체가 참이면 우객체를 평가하지 않고 좌객체 값을 리턴 한다.
좌객체가 거짓이면 우객체 값을 리턴한다.

예> console.log (“123” || 0); ➔ “123”

좌객체 && 우객체

==> 좌객체가 참이면 우객체 값을 리턴한다.
좌객체가 거짓이면 좌객체 값을 리턴한다.

예> console.log(“123” && 0); ➔ 0

○ 활용 예

예> 어떤 함수가 인자를 받을 때 반드시 배열을 받기로 가정한 경우?

```
function x( a ){  
    var arr = a || [] ;  
    ...  
}  
x();  
x([1,2]);
```

예> 이벤트 처리시 IE와 다른 브라우저간의 처리방법이 다른 경우?

```
function eventHandle(e){  
    var e = e || window.event;  
    ..  
}  
eventHandle(event); // 비 IE  
eventHandle(); // IE
```

○ 증감 연산자

- 1씩 증가 또는 감소시키는 연산자이다.
- 주의할 점은 다른 연산자와 같이 사용시, ++ 또는 --와 같은 연산자가 변수 앞에 위치하느냐? (전치) 아니면 변수 뒤에 위치하느냐?(후치)에 따라서 결과값이 다르게 산출된다.

[표 2-17] 증감 연산자의 종류

구분	연산자	의미
증감 연산자	++	1씩 증가시킨다.
	--	1씩 감소시킨다.

○ 3항 연산자

- 하나의 조건을 정의하여 만족 시에는 '참값'을 반환하고 만족하지 못할 시에는 '거짓값'을 반환하여 단순 비교에 의해 변화를 유도하는 연산자이다.

[표 2-18] 조건 연산자의 종류

구분	연산자	의미	구성
조건 연산자	? :	제어문의 단일 비교문과 유사하다.	조건식 ? 참값 : 거짓값

예> `var k = (4 > 5) ? 100 : 200;`
`var k2 = (4 > 5) ? alert(1) : alert(2);`
`var k3= (4 > 5) ? (alert(1) , 100) : (alert(2) , 200);`
`var k4 = ("22") ? 100 : 200;`

□ 7) 연산자 우선 순위

[연산자 우선순위]

우선순위	연산자
높다	배열([], 괄호())
	증가 연산자(++), 감소 연산자(--), 단항 뺄셈(-), 반전(~), 부정(!)
	곱셈(*), 나눗셈(/), 나머지(%)
	더하기(+), 뺄셈(-), 문자열 결합(+)
	쉬프트(<<, >>, <<<)
	비교 (<, <=, >=, >)
	등가(==), 부등가(!=), 동치(===), 비동치(!==)
	AND (&)
	XOR (^)
	OR ()
	논리AND(&&)
	논리OR ()
	조건(?:)
낮다	대입 (=), 복합 대입 (+=, -= 등)
	coma(,)

[기타 연산자]

연산자	개요
,(콤마)	좌우의 식을 계속해서 실행
delete	객체의 프로퍼티나 배열의 요소를 삭제
instanceof	객체가 지정된 클래스의 인스턴스인지를 판정
new	새로운 인스턴스를 생성
typeof	오퍼랜드의 데이터형을 취득

- 연산자 종류와 특징
- 산술 연산자
- 대입 연산자
- 비교 연산자
- 논리 연산자
- 증감 연산자
- 3항 연산자
- 이외 연산자



Thank you
