

Ch7 연산자

연산자는 하나 이상의 표현식을 대상으로 산술, 할당, 비교, 논리, 타입, 지수 연산 등을 수행해 하나의 값을 만든다.
연산의 대상을 **피연산자**라고 한다.

산술 연산자

산술 연산자는 피연산자를 대상으로 수학적 계산을 수행해 새로운 숫자 값을 만든다. 산술 연산이 불가능할 경우 **NaN**을 반환한다.

이항 산술 연산자

이항 산술 연산자는 2개의 피연산자를 산술 연산하여 숫자 값을 만든다.
모든 이항 산술 연산자는 피연산자의 값을 변경하는 부수 효과가 없다.

이항 산술 연산자

이항 산술 연산자	의미	부수 효과
+	덧셈	X
-	뺄셈	X
*	곱셈	X
/	나눗셈	X
%	나머지	X

단항 산술 연산자

단항 산술 연산자는 1개의 피연산자를 산술 연산하여 숫자 값을 만든다.
증가/감소 연산을 하면 피연산자의 값을 변경하는 **암묵적 할당**이 이뤄진다.

단항 산술 연산자

단항 산술 연산자	의미	부수 효과
++	증가	O
--	감소	O
+	어떠한 효과 없음. 음수를 양수로 반전하지 않음	X
-	양수를 음수로, 음수를 양수로 반전한 값을 반환함	X

- 피연산자 앞에 위치한 전위 증가/감소 연산자는 먼저 피연산자의 값을 증가/감소시킨 후, 다른 연산을 수행한다.
- 피연산자 뒤에 위치한 후위 증가/감소 연산자는 먼저 다른 연산을 수행한 후, 피연산자의 값을 증가/감소시킨다.

```
var x = 5, result;

// 선할당 후증가(postfix increment operator)
result = x++;
console.log(result, x); // 5 6

// 선증가 후할당(prefix increment operator)
result = ++x;
console.log(result, x); // 7 7○○
```

```
// 선할당 후감소(postfix decrement operator)
result = x--;
console.log(result, x); // 7 6

// 선감소 후할당 (prefix decrement operator)
result = --x;
console.log(result, x); // 5 5
```

+연산자는 피연산자 중 하나 이상이 문자열인 경우 **문자열 연결 연산자**로 동작한다.

- 암묵적 타입 변환 (타입 강제 변환)이 일어남

```
//true를 1로 타입 변환
1 + true; // -> 2

//null은 0으로 타입 변환
1 + null; // -> 1

//undefined는 숫자로 타입 변환하지 않음
1 + undefined; // -> NaN
```

할당 연산자

할당 연산자는 우항에 있는 피연산자의 평가 결과를 좌항에 있는 변수에 할당한다.

좌항의 변수에 값을 할당하므로 변수 값이 변하는 부수 효과가 있다.

→ =, +=, -=, *=, /=, %=

비교 연산자

비교 연산자는 좌항과 우항의 피연산자를 비교한 다음 그 결과를 불리언 값으로 반환한다.

== (동등 비교) : 값이 같음

=== (일치 비교) : 값과 타입이 같음

!= (부동등 비교) : 값이 다름

!== (불일치 비교) : 값과 타입이 같음

! NaN 주의

```
//NaN은 자신과 일치하지 않는 유일한 값이다.
NaN === NaN; // false
```

삼항 조건 연산자

삼항 조건 연산자는 조건식의 평가 결과에 따라 반환할 값을 결정한다.

조건식 ? 조건식이 true 일 때 반환할 값 : 조건식이 false 일 때 반환할 값

삼항 조건 연산자와 if ... else문의 차이점

삼항 조건 연산자는 값처럼 사용할 수 있지만, if ... else 문은 표현식이 아닌 문으로 if ... else 문은 값처럼 사용할 수 없다.

논리 연산자

논리 연산자는 우향과 좌향의 피연산자를 논리 연산한다.

`||` → 논리합(OR)

`&&` → 논리곱(AND)

`!` → 부정(NOT)

```
// 논리합(||) 연산자
true || true;    // -> true
true || false;   // -> true
false || true;   // -> true
false || false;  // -> false

// 논리곱(&&) 연산자
true && true;     // -> true
true && false;    // -> false
false && true;    // -> false
false && false;   // -> false

// 논리 부정(!) 연산자
!true;           // -> false
!false;          // -> true
```

선험 연산자

선험 연산자는 왼쪽 피연산자부터 차례대로 평가하고 마지막 피연산자의 평가가 끝나면 마지막 피연산자의 평가 결과를 반환한다.

그룹 연산자

소괄호로 피연산자를 감싸는 그룹 연산자는 자신의 피연산자인 표현식을 가장 먼저 평가한다.

typeof 연산자

피연산자의 데이터 타입을 문자열로 반환한다.

string, number, boolean, undefined, symbol, object, function 중 하나를 반환한다.

```
typeof ''          // -> "string"
typeof 1           // -> "number"
typeof NaN         // -> "number"
typeof true        // -> "boolean"
typeof undefined   // -> "undefined"
typeof Symbol()    // -> "symbol"
typeof null        // -> "object"
typeof []          // -> "object"
typeof {}          // -> "object"
typeof new Date()  // -> "object"
typeof /test/gi    // -> "object"
typeof function () {} // -> "function"
```

값이 null 타입인지 확인할 때는 typeof 가 아닌 일치 연산자(===) 를 사용한다.

```
var foo = null;
```

```
typeof foo === null; // -> false
foo === null;        // -> true
```

선언하지 않은 식별자를 typeof 연산자로 연산하면 ReferenceError 가 아닌 undefined 를 반환한다.

```
// undeclared 식별자를 선언한 적이 없다.
typeof undeclared; // -> undefined
```

지수 연산자

ES7 에서 도입된 지수 연산자는 좌항의 피연산자를 밑으로, 우항의 피연산자를 지수로 거듭 제공하여 숫자 값을 반환한다.

음수를 거듭제곱의 밑으로 사용해 계산하려면 다음과 같이 괄호로 묶어야 한다.

```
-5 ** 2;
// SyntaxError: Unary operator used immediately before exponentiation expression.
// Parenthesis must be used to disambiguate operator precedence

(-5) ** 2; // -> 25

//지수 연산자는 이항 연산자 중에서 우선순위가 가장 높다.
2 * 5 ** 2; // -> 50
```

연산자 우선순위

연산자 우선순위를 모두 기억하기 어렵기 때문에 연산자 우선순위가 가장 높은 그룹 연산자를 사용하여 우선순위를 명시적으로 조절하는 것을 권장한다.

연산자 결합 순서

결합 순서	연산자
좌항 → 우항	+, -, /, %, <, <=, >, >=, &&, , ., [], (), ??, ?, in, instanceof
우항 → 좌항	++, --, 할당 연산자(=, +=, -=, ...), !x, +x, -x, ++x, --x, typeof, delete, ? ... : ..., **