

7장 연산자

정의

산술 연산자

이항 산술 연산자

단항 산술 연산자

문자열 연결 연산자

할당 연산자

비교 연산자

동일 일치 비교 연산자

대소 관계 비교 연산자

삼항 조건 연산자

논리 연산자

선험 연산자

그룹 연산자

typeof 연산자

지수 연산자

그 외 연산자

연산자 우선순위

결합 순서

정의

하나 이상의 표현식을 대상으로 산술, 할당, 비교, 논리, 타입, 지수 연산 등을 수행해 하나의 값을 만들며 연산의 대상을 피연산자라고 함

산술 연산자

이항 산술 연산자

- 모든 이항 산술 연산자는 피연산자의 값을 변경하는 부수 효과가 없음
 - 피연산자의 값을 변경하는 경우는 절대 없고 언제나 새로운 값을 생성함

이항 산술 연산자	의미	부수 효과
+	덧셈	X
-	뺄셈	X
*	곱셈	X

이항 산술 연산자	의미	부수 효과
/	나눗셈	X
%	나머지	X

단항 산술 연산자

- 1개의 피연산자를 산술 연산하여 숫자 값을 만들

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

- 단항 산술 연산자는 피연산자의 값을 변경함



아래의 코드를 실행하시면 됩니다.

```
var a = '1';

console.log(+a, typeof +a);
console.log(a, typeof a);

a = true;
console.log(+a, typeof +a);
console.log(a, typeof a);

a = false;
console.log(+a, typeof +a);
console.log(a, typeof a);

a = 'Hi';
console.log(+a, typeof +a);
console.log(a, typeof a);
```

문자열 연결 연산자

- + 연산자는 피연산자 중 하나 이상이 문자열인 경우 문자열 연결 연산자로 동작함



암묵적 타입 변환 또는 타입 강제 변환에 대해서 알아보시오

할당 연산자

할당 연산자	예	동일 표현	부수 효과
=	a = 1	a = 1	O
+=	a += 1	a = a + 1	O
-=	a -= 1	a = a - 5	O
*=	a *= 1	a = a * 1	O
/=	a /= 1	a = a / 5	O
%=	a %= 1	a = a % 5	O

```
var str = "shit get on the ";
str += "ground";
console.log(str); // shit get on the ground
```

- 할당문은 값으로 평가되는 표현식인 문으로서 할당된 값으로 평가됨

```
var a, b, c;
a = b = c = 0;
console.log(a,b,c);
```

비교 연산자

동일 일치 비교 연산자

비교 연산자	의미	사례	설명	부수 효과
==	동등 비교	x == y	x와 y의 값이 같음	X
===	일치 비교	x === y	x와 y의 값과 타입이 같음	X
!=	부동등 비교	x != y	x와 y의 값이 다름	X
!==	불일치 비교	x !== y	x와 y의 값과 타입이 다름	X

- **동등 비교** 연산자는 좌항과 우항의 피연산자를 비교할 때 먼저 암묵적 타입 변환을 통해 타입을 일치시킨 후 같은 값인지 비교함
- **일치 비교** 연산자는 좌항과 우항의 피연산자가 타입도 같고 값도 같은 경우에 한하여 true 값을 반환



아래의 비교가 뭐가 다른지 알아보시오.

```
5 == 5;
5 == '5';
// =====
5 === 5;
5 === '5';
// =====
'0' == '';
0 == '';
0 == '0';
// =====
false == 'false';
false == '0';
false == null;
false == undefined;
// ---
NaN === NaN
0 == -0
0 === -0
```

- Number.isNaN()
해당 파라미터가 NaN인지 판단해주는 함수
- Object.is()
예측 가능한 정확한 비교 결과를 반환



- 0 === 0 ;
Object.is(-0,0)

NaN === NaN;
Object.is(NaN,NaN);
의 결과가 왜 다를까?

대소 관계 비교 연산자

대소 관계 비교 연산자	예제	설명	부수 효과
>	X > Y	X가 Y보다 크다	X
<	X < Y	X가 Y보다 작다	X

대소 관계 비교 연산자	예제	설명	부수 효과
>=	X >= Y	X가 Y보다 크거나 작다	X
<=	X <= Y	X가 Y보다 작거나 같다	X

삼항 조건 연산자

조건식 ? TRUE 반환 값 : FALSE 반환 값

```
x === y ? 1 : 2;
var lol = 1 === 1 ? 'true' : 'false';
```

논리 연산자

논리 연산자	의미	부수 효과
	논리합	X
&&	논리곱	X
!	부정	X

실패 연산자

```
var a, b, c;
a = 1, b = 2, c = 3;
```

- 왼쪽 피연산자부터 차례대로 평가 후 결과 반환

그룹 연산자

- ()로 감싼 연산자들은 표현식을 가장 먼저 평가함

typeof 연산자

- typeof 연산자는 피연산자의 데이터 타입을 문자열로 반환함
- string, number, boolean, undefined, symbol, object, function으로 반환됨



위에 있는 반환 값을 다 나타내보시오.

예시)

```
typeof 1
```

지수 연산자

- ES7 도입된 지수연산자는 좌항의 피연산자를 밑으로 우항의 피연산자를 지수로 거듭제곱하여 숫자를 반환함

```
2 ** 2;
2 ** 2.5;
2 ** 0;
2 ** -2;
```

```
2 ** ( 3 ** 2 );
(-5) ** 2;
```

그 외 연산자

연산자	개요
?.	옵셔널 체이닝 연산자
??	null 병합 연산자
delete	프로퍼티 삭제
new	생성자 함수를 호출할 때 사용하여 인스턴스를 생성
instanceof	좌변이 객체가 우변의 생성자 함수와 연결된 인스턴스인지 판별
in	프로퍼티 존재 확인

연산자 우선순위

우선순위	연산자
1	()

우선순위	연산자
2	new(매개변수o), . , [], (), ?.
3	new(매개변수x)
4	++, —
5	!, +, -, ++, —, typeof, delete
6	**
7	*, / , %
8	+, -
9	<, ≤ , > , ≥, in, instanceof
10	==, !=, ===, !==
11	??
12	&&
13	
14	? ... : ...
15	할당 연산자
16	,

결합 순서

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