연산자

연산자

- 산술 연산자 : +, -, *, /, %, ++, --
- 대입 연산자 : =, +=, -=, *=, /=, %=, &=, |=, ^=
- 비교 연산자 : ==, !=, <, <=, >, >=, ===, !==
- 논리 연산자 : &&, ||, !
- 비트 연산자 : &, |, ^, <<, >>

연산자

- 연산자
  - ==, === : 동등, 일치 비고
  - in : 프로퍼티 존재 확인
  - delete : 프로퍼티 값 무효화
  - typeof : 타입 반환
  - void : undefined 반환
  - instanceof : 객체 타입 확인
  - ||, && : OR, AND

동등, 일치 연산자

- 동등 연산자(==)
  - 값의 평가가 동등(equal)
  - 참조 타입 : 동일 참조 비교
  - 형 변환 수행

    1 == “1”
- 일치 연산자(===)
  - 값이 동등하고 타입 일치 비교
  - 참조 타입 : 동일 참조 비교

    1 === “1”
- 예제 코드 : 02.Operator/equal.js

in 연산자

- in 연산자
  - 객체 내 프로퍼티 존재 확인

```
var iu = {
    name: "IU",
    best: "좋은날"
};

console.log("name" in iu);
console.log("friend" in iu);
```

delete 연산자

- delete 연산자
  - 객체 내 프로퍼티 제거

```
var iu = {
    name: "IU",
    best: "좋은날"
};
delete iu.best;
```

delete 연산자

- delete 연산자
  - var로 선언된 변수 제거 불가

```
var bar = "1234";

delete bar
```

객체 인스턴스

- 밸류 타입 리터럴과
- 숫자

```
var num1 = 1
var num2 = new Number(1);

typeof num1 // number
typeof num2 // object
```

- 문자

```
var str1 = 'Hello';
var str2 = new String(‘Hello’);

typeof str1 // string

typeof str2 // object
```

객체 인스턴스

- 클래스 객체와 리터럴
- 배열

```
var array1 = [1, 2, 3];
var array2 = new Array(1, 2, 3);
```

- 타입

```
typeof array1 // object
typeof array2 // object
```

instanceof 연산자

- 레퍼런스 타입 객체와 타입 비교

```
var obj1 = new Number(1);
obj1 instanceof Number // true
var obj2 = [1, 2, 3];
obj2 instanceof Array // true
var obj3 = new Array(1, 2, 3);
obj3 instanceof Array // true
```

instanceof 연산자

- 객체와 객체 생성자 비교
  - 객체 생성자

```
function Clazz() {
}
```

  - 객체와 instanceof 검사

```
var c = new Clazz();
c instanceof Clazz     //true
c instanceof Object    //true
```

instanceof 연산자

- 객체와 클래스 비교
  - 클래스

```
class MyClass {
}
```

  - 객체와 instanceof 검사

```
var obj = new MyClass();

obj instanceof MyClass    //true
```

논리 연산자

- OR 연산자(||)
  - 좌변의 boolean값 true => 좌변값 반환
  - 좌변의 boolean값이 false => 우변값 반환

`var val = input || ‘default';`	`var val;`
	`if ( input ) {`
	`  val = 'default';`
	`}`

- AND 연산자 (&&)
  - 좌변의 boolean값 true => 우변값 반환

new 연산자

- new 연산자
- 생성자 함수에서 객체 생성

```
var MyClass = function() {
}
var obj = new MyClass();
```

실습

- 02.Operator
  - equal.js
  - in_delete.js
  - instanceof.js

제어문

반복

- for loop

```
for (var i = 0 ; i < array.length ; i++) {
   var item = array[i]; console.log(i, item);
}
```

- for-in : 객체와 사용

```
for (var prop in obj) {
   console.log('property name : ', prop + " value : " + obj[prop]);
}
```

- for-in : 배열과 사용

```
for (var item in array) {
   console.log('index : ', item, ' element : ‘,array[item]);
}
```

switch

```
switch (표현식) {
  case 값#1:
    문장 #1;
    break;
  case 값#2:
    문장 #2;
    break;
  default:
    문장 #3;
}
```