

# 10. 약타입 언어

## 약타입 언어와 강타입 언어

(weak typed language, strong typed language)

javascript는 약타입 언어이다.

약타입 언어는 변수의 자료형(type)을 엄격하게 구분하지 않는다.

java 언어는 강타입 언어이다.

java는 변수의 자료형을 엄격하게 구분한다.

```
let a = 3, b = "3";  
  
console.log(a == b);  
console.log(a != b);
```

위 코드에서 변수 a 값은 숫자 3이고, 변수 b 값은 문자열 "3"이다.

javascript는 숫자 타입(number type)과 문자열 타입(string type)을 엄격하게 구분하지 않기 때문에, 숫자 3과 문자열 "3"이 같은지 비교하면, 같다고 나온다.  
a == b의 결과는 true이다.



The screenshot shows a dark-themed browser developer tools console window. At the top, it says 'Console' and has 'Clear' and 'X' buttons. Below that, the word 'true' is displayed in green, followed by a horizontal line, and then the word 'false' is displayed in green again.

## ==== 연산자, != 연산자

==== 연산자는 == 연산자와 기능이 같고,

`!==` 연산자는 `!=` 연산자와 기능이 같다.

`==` 연산자나 `!=` 연산자로 비교할 때는 타입을 엄격하게 비교하지 않지만,  
`====` 연산자나 `!==` 연산자로 비교할 때는 타입을 엄격하게 비교한다.

```
let a = 3, b = "3";  
  
console.log(a === b);  
console.log(a !== b);
```

a와 b는 타입이 다르기 때문에,  
`a === b`의 결과는 `false`이고, `a !== b`의 결과는 `true`이다.

## typeof 연산자

`typeof` 연산자를 이용하여, 값의 타입을 확인할 수 있다.

```
let a = 3, b = "4", c = true;  
  
console.log(typeof a);  
console.log(typeof b);  
console.log(typeof c);
```

위 예제에서 변수 `a`의 타입은 `number`이고, `typeof a` 표현식의 값은 `"number"`이다.

위 예제에서 변수 `b`의 타입은 `string`이고, `typeof b` 표현식의 값은 `"string"`이다.

위 예제에서 변수 `c`의 타입은 `boolean`이고, `typeof c` 표현식의 값은 `"boolean"`이다.

```
Console Clear X
"number"
"string"
"boolean"
```

```
let a = 3;
console.log(typeof a == "number");

a = "hello";
console.log(typeof a == "string");

a = true;
console.log(typeof a == "boolean");
```

강타입 언어인 java 에서는, int 변수에 int 값만 대입할 수 있다

약타입 언어인 javascript 에서는, 어떤 변수에 number, string, boolean 값을 번갈아 대입해도 된다

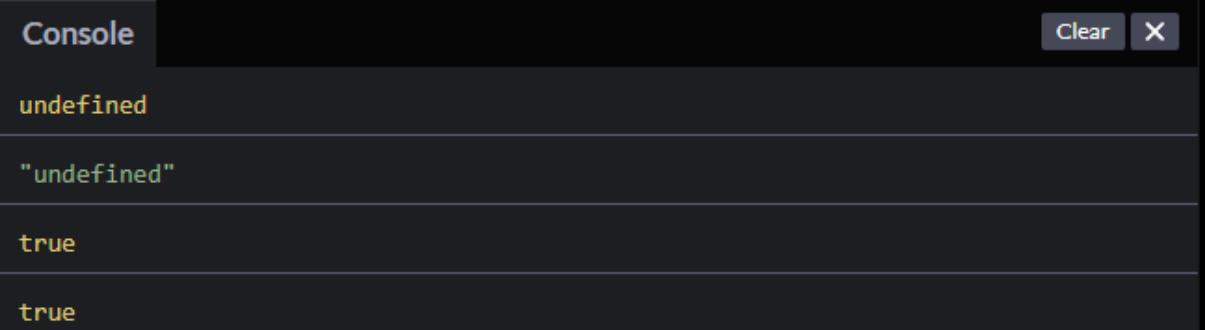
```
Console Clear X
true
true
true
```

## undefined

javascript 언어에 undefined 키워드가 있다.  
아직 값이 대입되지 않은 변수의 값은 undefined 이다.

```
let a;  
  
console.log(a);  
console.log(typeof a);  
console.log(a === undefined);  
console.log(typeof a == "undefined");
```

변수 a에 아무 값도 대입되지 않았기 때문에, a의 값은 undefined이고, a의 타입도 undefined이다.



```
Console Clear X  
undefined  
"undefined"  
true  
true
```

## null

javascript에도 java처럼 null 키워드가 있다.

null 값의 의미 = 의도적으로 값을 설정하지 않았음을 나타내기 위해 사용

- null 값을 잘 다루는 것은 프로그래밍에서 오류를 방지하고, 데이터 처리의 정확성을 높이는 데 중요

```
let s = null;  
  
console.log(s);  
console.log(s === null);
```

변수에 null 값을 대입할 수 있고, 변수 값이 null인지 비교할 수 있다.

The screenshot shows a dark-themed browser developer tools interface. At the top, there's a header bar with the word "Console" on the left and "Clear" and "X" buttons on the right. Below the header, the main area displays two lines of text: "null" in yellow and "true" in green. These colors are typically used by many JavaScript environments to highlight different types of values.

## undefined vs null

undefined 와 null 은 둘 다 변수 값이 없음을 의미한다.

즉 undefined 와 null 의 의미는 거의 비슷하다.

약간의 차이점은 다음과 같다.

    |   null       : 값이 있고, 그 값이 null 이다.

    |   undefined : 값이 없다.