



Chap6 데이터 타입

6-1 숫자 타입 (Number)

6-2 문자열 타입

6-3 템플릿 리터럴

6-4 불리언 타입

6-5 undefined 타입

6-6 null 타입

6-7 심벌 타입

6-8 객체 타입

6-9 데이터 타입의 필요성

6-1 숫자 타입 (Number)

- 정수, 실수 구분없이 모든 수를 실수로 처리하며, 정수만 표현하기 위한 데이터 타입이 별도로 존재하지 않는다.

```
var integer = 10; //정수
var double = 10.12; // 실수
var negative = -20; // 음의정수
```

=> 결국에는 모두 숫자 타입

- 정수, 실수, 2진수, 8진수, 16진수 리터럴은 모두 메모리에 배정밀도 64비트 부동소수점 형식의 2진수로 저장된다.
- 2진수, 8진수, 16진수를 표현하기 위한 데이터 타입이 존재하지 않아 이를 참조하려면 모두 10진수로 해석된다.

```
var binary=0b1000001; //2진수
var octal = 0o101; //8진수
var hex=0x41; //16진수
```

//모두 10진수로 해석됨.

```
console.log(binary); //65
console.log(octal ); //65
console.log(hex); //65
console.log(binary===octal); //true
console.log(octal===hex); //true
```

- 숫자타입은 세가지 특별한 값도 표현할 수 있다.

```
Infinity //양의 무한대
-Infinity //음의 무한대
NaN //산술 연산 불가(not-a-number)
```

▼ 배정밀도 64비트 부동소수점

- 숫자를 컴퓨터 메모리에 저장하는 방식을 규정한 표준
- 배정밀도 64비트 부동소수점 형식의 구조
 1. 부호 비트 (Sign bit): 1비트
 2. 지수부 (Exponent): 11비트
 3. 가수부 (Fraction or Mantissa): 52비트

6-2 문자열 타입

- 텍스트 데이터를 나타내는 데 사용
- 0개 이상의 16비트 유니코드 문자의 집합으로 전 세계 대부분의 문자를 표현할 수 있다.
- 사용방법 : 작은따옴표, 큰따옴표, 백틱으로 텍스트를 감싼다. (가장 일반적인 표기버은 작은따옴표로 감싸기)

6-3 템플릿 리터럴

- ES6부터 도입되었다.
- 멀티라인 문자열, 표현식 삽입, 태그드 템플릿 등 편리한 문자열 처리 기능을 제공
 - ▼ 멀티라인 문자열
 - 이스케이프 시퀀스 사용하기

이스케이프 시퀀스	의미
\0	Null
\b	백스페이스
\f	폼 피드Form Feed: 프린터로 출력할 경우 다음 페이지의 시작 지점으로 이동한다.
\n	개행LF, Line Feed: 다음 행으로 이동
\r	개행CR, Carriage Return: 커서를 처음으로 이동
\t	탭(수평)
\v	탭(수직)
\uXXXX	유니코드. 예를 들어 '\u0041'은 'A', '\uD55C'는 '한', '\u{1F600}'는 😄이다.
\'	작은따옴표
\"	큰따옴표
\\	백슬래시

- 백틱안에 적기
예시코드

```
var template=`<ul
  <li><a href="#">Home</a></li>
</ul>`;

console.log(template);
```

출력결과

```
<ul
  <li><a href="#">Home</a></li>
</ul>
```

▼ 표현식 삽입

- \${ }으로 표현식을 감싸 사용하면 된다. 이때 표현식의 평가 결과가 문자열이 아니라도 문자열로 타입이 강제 변환되어 삽입된다.

```
var first='Ung-mo';
var last='Lee';
console.log(`My name is ${first} &{last}.`); //My name is Ung-mo & Lee.
```

즉 +연산자로 길게 코드를 입력하지 않아도 된다는 것이다.

- 런타임에 일반 문자열로 변환되어 처리
- 백틱을 사용해 표현함

6-4 불리언 타입

- 논리적 참, 거짓을 나타내는 true와 false 값을 가짐.

6-5 undefined 타입

- undefined타입의 값은 undefined가 유일하다.
- 개발자가 의도적으로 할당하기 위한 값이 아닌 자바스크립트 엔진이 변수를 초기화 할 때 사용하는 값.

▼ undefined와 null 구별

null은 변수에 값이 없다는 것을 명시하고 싶을 때 쓰는 것이고 undefiend는 변수를 초기화하는데 사용되는 값이다. 즉 null은 개발자가 임의로 넣을 수 있는 값이고 undefined는 아니다.

▼ 선언(declaration)과 정의(definition)

ECMAScript 사양에서 변수는 '선언한다'라고 표현하고 함수는 '정의한다'라고 표현한다.

6-6 null 타입

- null 타입은 null이 유일하다.
- 변수에 값이 없다는 것을 명시하는데 사용
- 변수에 null을 할당하는 것은 변수가 이전에 참조하던 값을 더 이상 참조하지 않겠다는 의미. 즉 가비지 콜렉션 수행

```
var foo='Lee';  
foo=null; // 가비지 컬렉션 수행
```

- 함수가 유효한 값을 반환할 수 없는 경우 명시적으로 null을 반환하기도 함

6-7 심벌 타입

- ES6에 추가된 7번째 타입으로 변경 불가능한 원시 타입의 값
- 다른 값과 중복되지 않는 유일무이한 값

추후 33장에서 깊게 살펴 볼 예정

6-8 객체 타입

11장에서 설명 예정

6-9 데이터 타입의 필요성

- 값을 저장할 때 확보해야 하는 메모리 공간의 크기를 결정하기 위해
- 값을 참조할 때 한번에 읽어 들여야 할 메모리 공간의 크기를 결정하기 위해
- 메모리에서 읽어 들인 2진수를 어떻게 해석할지 결정하기 위해