

# 6장

- 데이터 타입
  - Js의 모든 값은 데이터 타입을 가진다
  - 7가지의 타입
    - 원시타입
      - 숫자, 문자열, 불리언, undefined, null, 심벌 타입
    - 객체타입
  - 개발자는 명확한 의도를 가지고 타입을 구별해서 값을 생성
  - Js 엔진은 타입을 구별해서 값을 취급
- 숫자 타입
  - 모든 수를 실수로 처리
  - Js는 숫자 타입은 정수만을 위한 타입은 없고 전부 다 실수로 처리
  - NaN 산술 연산 불가
    - Js는 값이 아닌 식별자로 해석
- 문자열 타입
  - 텍스트 데이터를 나타내는데 사용
  - 작은따옴표, 큰따옴표, 백틱으로 텍스트를 감싸서 사용
    - 키워드나 식별자 같은 토큰과 구분하기 위해서 사용
  - 원시 타입이며 변경 불가능한 값이다.
- 템플릿 리터널
  - 일반 문자열 내에서는 줄바꿈이 허용되지 않는다.
    - 백슬래시로 시작하는 이스케이프 시퀀스를 사용해야한다
  - 템플릿 리터널내에서는 줄바꿈이 허용된다.

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

```
</ul>
```

- 연사자 +를 사용하면 문자열 연결 연산자로 동작한다.
- 불리언 타입
  - 값은 true, false 뿐이다.
- undefined 타입
  - 변수를 참조했을 때 undefined가 반환된다면 초기화되지 않은 변수라는 것을 간파할 수 있다.
  - 개발자가 의도적으로 undefined를 할당한다면 본래 취지의 어긋나고 혼란을 줄 수 있다.
  - 만약 필요한 경우는 null을 할당한다.
- null 타입
  - 변수에 값이 없다는 것을 의도적으로 명시
  - 변수에 null을 할당하는 것은 변수가 이전에 참조하던 값을 더 이상 참조하지 않겠다는 의미.
- 심벌 타입
  - 변경 불가능한 원시 타입의 값.
  - 다른 값과 중복 되지 않는 유일무이한 값.
- 객체 타입
  - JS는 객체 기반의 언어
  - 이후에 자세히 알아보자
- 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
  - JS의 변수는 정적 타입 언어와 같이 미리 선언한 데이터 타입의 값만 할당할 수 있는 것이 아니다
  - 어떠한 데이터 타입의 값이라도 자유롭게 할당할 수 있다.

- JS의 변수는 선언이 아닌 할당에 의해 타입이 결정된다. 그리고 재할당에 의해 변수의 타입은 언제든지 동적으로 변할 수 있다.
  - 동적 타입
    - 동적 타입 언어는 유연성은 높지만 신뢰성은 떨어진다.
- 코드는 개발자를 위한 문서이기도하다. 따라서 사람이 이해할 수 있는 코드, 즉 가독성이 좋은 코드가 좋은 코드다.