

33장 7번째 데이터 타입 Symbol

33.1 심벌이란 ?

ES6에서 도입된 7번째 데이터 타입으로
변경 불가능한 원시 타입의 값이다.

다른 값과 절대 중복되지 않는
유일무이한 값이다.

프로퍼티 키로 사용할
수 있는 값이다.

33.2 심벌 값의 생성

1. Symbol 함수

심벌 함수로만 생성이 가능하다. 다른 원시값처럼
리터럴 표기법으로 생성할 수 없다.

New 연산자와 쓰이지 않는다.

문자열을 인수로
전달할 수 있다.

래퍼 객체 생성

암묵적 타입 변환 X
(불리언 타입으로는 변환)

2. Symbol.for / Symbol.keyFor 메서드

Symbol.for 메서드는 인수로 전달받은 문자열을 키로 사용하여
키와 심벌 값의 쌍들이 저장되어 있는 전역 심벌 레지스트리에서
해당 키와 일치하는 심벌 값을 검색한다.

검색 성공시 => 새로운 심벌 값 생성
X, 검색된 심벌 값을 반환

검색 실패시 => 새로운 심벌 값 생성하여 Symbol.for
메서드의 인수로 전달된 키로 전역 심벌 레지스트리에
저장한 후, 생성된 심벌 값 반환

Symbol.keyFor 메서드를 사용하면 전역 심벌
레지스트리에 저장된 심벌 값의 키를 추출할 수 있다.

그냥 심벌 메서드로 생성한
심벌 값은 추출이 불가

33.3 심벌과 상수

객체 내 심벌을 값으로 사용하면 변경/중복될
가능성이 없이 안전하게 사용이 가능하다.

JavaScript에서의
enum 구현

Object.freeze로 객체를 만든 후 객체 내
값을 심벌값으로 사용하면 된다.

33.4 심벌과 프로퍼티 키

객체의 프로퍼티 키로 사용이 가능하며
Symbol.for 메서드를 사용한다.

호출시 대괄호를
이용해야 한다.

다른 프로퍼티 키와 절대 충돌하지 않는다!

33.5 심벌과 프로퍼티 은닉

심벌 값으로 프로퍼티 키로 사용하여 생성한 프로퍼티는
for...in 문이나 Object.keys,
Object.getOwnPropertyNames 메서드로 찾을 수
없다.

즉, 은닉이 가능하다.

But, 완전히 숨길 수 없다. ES6에 도입된
Object.getOwnPropertySymbols
메서드를 사용하면 찾을 수 있기때문

33.6 심벌과 표준 빌트인 객체 확장

일반적으로 표준 빌트인 객체에 사용자
정의 메서드를 직접 추가하여 확장하는
것을 권장하지 않는다.

미래에 표준 사양으로 추가될 메서드와 이름이
중복되어 충돌이 날 수 있기 때문

불가피하게 꼭 추가해야 된다면 심벌값을
이용하면 충돌을 방지하여 확장할 수 있다.

33.7 Well-known Symbol

자바스크립트가 기본 제공하는 빌트인 심벌
값을 ECMAScript 사양에서는
Well-known-Symbol 이라 부른다.

이는 자바스크립트 엔진의 내부
알고리즘에 사용된다.

Symbol.iterator 메서드