11. 원시 값과 객체의 비교

▼ 값의 할당 시, 새로운 메모리를 사용하는 이유

원시 값은 불변성을 가지므로 변경할 수 없다.

그러므로 재할당을 통해서 할당된 값을 교체한다.

▼ 문자열

유사 배열 객체

Symbol.iterator를 가지므로 순회 가능.

문자열은 객체와 같이 인덱스로 프로퍼티 값에 접근할 수 있고 length프로퍼티를 가진다.

그러나 인덱스로 프로퍼티에 접근해서 값을 변경하는 것은 불가능.

→ 객체 내부 프로퍼티의 writable이 false이기 때문.

▼ 비트 not 연산자를 사용한 코드 축약

~ (bitwies not 연산자): 피연산자를 32비트 정수로 바꾼 후 모든 비트를 반전한다. 즉, ~n → -(n+1)

```
//예시

( ~2 ); // -3, -(2+1)과 같음
( ~1 ); // -2, -(1+1)과 같음
( ~0 ); // -1, -(0+1)과 같음
( ~-1 ); // 0, -(-1+1)과 같음
```

여기서 0이 falsy값인 걸 이용.

```
if(~str.indexOf('target')) {
  // 실행할 코드
}
```