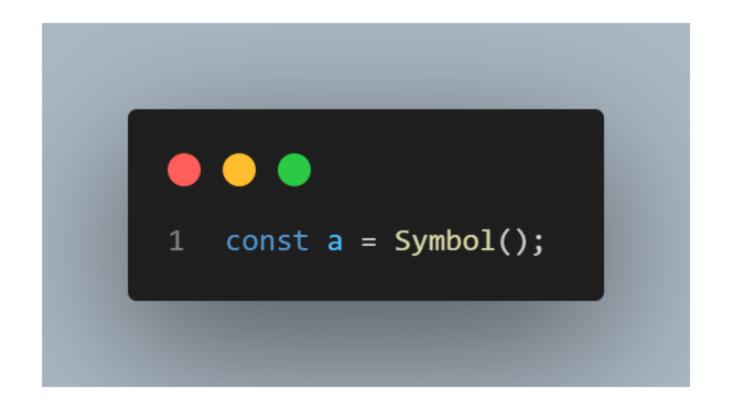
l 심볼형 (Symbol)

이은주

심볼(Symbol)이란?



심볼을 생성할 때는 Symbol() 함수 호출

```
1 const a = new Symbol();
```

```
Uncaught TypeError: Symbol is <u>VM72:1</u> not a constructor at new Symbol (⟨anonymous⟩) at ⟨anonymous⟩:1:11
```

new 연산자를 사용하면 TypeError가 발생

심볼(Symbol)

```
const a = Symbol();
const b = Symbol();
undefined

a
Symbol()
b
Symbol()
```

Symbol()은 유일한 식별자이기 때 문에 둘의 값이 완전히 다름

심볼(Symbol)

```
const a = Symbol('a');
const b = Symbol('a');
undefined
a
Symbol(a)
b
Symbol(a)
```

생성된 심볼은 항상 유일하며, 같은 심볼을 생성하려고 하면 항상 새로운 심볼이 반환

자동 형 변환

```
const a = Symbol('a');
alert(a);

Duncaught TypeError: Cannot VM430:2
convert a Symbol value to a string
at <anonymous>:2:1
```

심볼은 자동 형 변환 X
-> 자바스크립트에서 언어 차원의 보호장치를 마련해 심볼형이 다른형으로 변환 되지 않게 막아줌

```
1 const a = Symbol('id');
2 alert(a.toString())
```

```
이 페이지 내용:
Symbol(id)
확인
```

자동 형 변환

```
const a = Symbol('a');
alert(a);

Duncaught TypeError: Cannot VM430:2
convert a Symbol value to a string
at ⟨anonymous⟩:2:1
```

심볼은 자동 형 변환 X
-> 자바스크립트에서 언어 차원의 보호장치를 마련해 심볼형이 다른형으로 변환 되지 않게 막아줌

```
1 const a = Symbol('id');
2 alert(a.description)
```

```
이 페이지 내용:
id
확인
```

객체 프로퍼티

```
1 const id = Symbol('id');
2
3 const user = {
4     name: 'eunjoo',
5     [id]: 123
6 }
```

```
> user
< ▶ {name: 'eunjoo', Symbol(id): 123}
>
```

객체 프로퍼티

```
const id = Symbol('id')

const user = {
    name: 'eunjoo',
    [id] : 123
};
for (let key in user){
    console.log(key);
}

name
```

심볼로 만들어진 키는 for in문, Object.keys(), Object.getOwnPropertyNames()와 같은 메소드에 반환 X

심볼로 선언된 키 값 보고싶다 -> Object,getOwnPropertySymbols()

Symbol.for, Symbol.keyFor

```
const s1 = Symbol.for('foo');
const s2 = Symbol.for('foo');
console.log(s1 === s2);
```

```
const a = Symbol.for('abc')
console.log(Symbol.keyFor(a))
abc
VM66
```