

7/27 ~ 7/31 4주차 - 3

Javascript var - let - const 의 차이

먼저 Hoisting과 Scope의 개념이 필요하다.


- Hoisting에 대한 정리

<https://velog.io/@marcus/Javascript-Hoisting>

- Scope에 대한 정리

자바스크립트 스코프

안녕하세요 Marcus입니다. 오늘은 자바스크립트에서 중요한 스코프(scope)에 대해서 소개해드리겠습니다. 스코프란? 자바스크립트에서 스코프란 어떤 변수들에 접근할 수 있는지를 정의합니다. 스코프엔 두 가지 종류가 있습니다. 전역 스코프와 지역 스코프로 나뉩니다.

 <https://velog.io/@marcus/%EC%9E%90%EB%B0%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8-%EC%8A%A4%EC%BD%94%ED%94%84>

JS

UNDERSTANDING JAVASCRIPT SCOPE

1. 변수 값의 변환

var를 사용하면 변수 선언의 경우 할당되는 값이 유동적으로 변경될 수 있는 단점을 가지고 있다.

```
var name = "Marcus";
console.log(name);

var name = "Jogeonsang";
console.log(name);

output: Marcus
output: Jogeonsang
```

name이라는 변수를 2번 선언했는데도 에러가 나오지 않고 각기 다른 값이 출력되는 걸 볼 수 있다.

하지만 ES6 업데이트 이후로 추가된 변수 선언 방식인 let과 const는 var와 같은 선언 방식을 막고 있다.

```
let name = "Marcus";
console.log(name);

let name = "Jogeonsang";
console.log(name);
output: Identifier 'name' has already been declared
```

위와 같이 let을 사용했을 경우에는 name이 이미 선언되었다는 에러 메시지가 나오는 걸 볼 수 있다.

위에 코드에는 let만 케이스로 집어 넣었지만 const도 마찬가지로 변수 재할당이 안된다는 특징을 가지고 있다.

let과 const 의 차이

- let과 const의 차이점은 변수의 Immutable 여부이다.
- let은 변수에 재할당이 가능하다
- const는 변수 재선언, 재할당 모두 불가능

```
// let
let testCase = 'let' // output: let
let testCase = 'let2' // output: Uncaught SyntaxError: Identifier 'testCase' has already been declared
testCase = 'let3' // output: let3
```

```
// let
let testCase = 'let' // output: let
let testCase = 'let2' // output: Uncaught SyntaxError: Identifier 'testCase' has already been declared
testCase = 'let3' // output: let3
```

2. 변수의 유효범위

먼저 var는 기본적으로 function scope를 가지게 된다.

let,const 는 block scope를 가지게 됨.

var

```
var foo = "This is String.";
if(typeof foo === 'string'){
  var result = true;
} else {
  var result = false;
}

console.log(result);    // result : true
```

let과 const

```
var foo = "This is String.";
if(typeof foo === 'string'){
  const result = true;
} else {
  const result = false;
}

console.log(result);    // result : result is not defined
```