

# JavaScript (ES6) For Framework

14. 변수도 이전 상황에 맞게 사용하자



“

var가 있는데 굳이  
let & const 나온 이유?

var의 문제점 : 함수 스코프

(함수가 아닌)  
블록에서 선언된 변수는  
전역 변수

var의 문제점 : 함수 스코프

전역변수가 너무 많아진다!

## var의 문제점 : 함수 스코프

```
if(true) {  
    var num = 10;    // if 문 안에서만 쓰기 위한 변수  
    console.log(num);  
}  
  
// 전역변수로 취급받음  
console.log(num);    // 10  
console.log(window.num);    // 10
```

var의 문제점 : 중복된 이름 선언 가능

의도치 않게 값의 변경이 일어날 가능성

var의 문제점 : 변수의 불변성 보장 X

유동적인 값을 담는 변수

VS

너무 중요해서 값이 바뀌면 안되는 변수

에 대한 구분이 없음

## var의 문제점 : 변수의 불변성 보장 X

```
var num = "이 값은 중요하니까 바꾸지 말아야지";  
  
if(true) {  
    var num = "아무 생각없이 값 변경";  
}  
  
console.log(num);    // 값이 바뀌어버림
```



var의 한계 보완 : let & const

let

중복 선언 X  
블록 레벨 스코프

var의 한계 보완 : let & const

**const**

중복 선언 X

블록 레벨 스코프

+ 값의 재할당 불가



“

결론!

앞으로 변수 선언은 `let` & `const`!



“

언제든 재할당이 필요한 경우는 **let!**

변할 일 없으면 기본적으로 **const!**

다음강의

## Section 15

### 템플릿 리터럴

Lecturer Github : <https://github.com/kangtegong/>

수업자료링크 : <https://github.com/kangtegong/develop-javascript>

Email : [tegongkang@gmail.com](mailto:tegongkang@gmail.com)