



JavaScript – hoisting, let, var, const

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

Contents

- ▶ Hoisting (호이스팅)
- ▶ var
- ▶ let
- ▶ const
- ▶ 분석

호이스팅

- ▶ Hoisting
- ▶ 변수 끌어 올림 - 변수들을 범위의 최상단으로 끌어올리는 것

```
console.log(x);    // undefined  
var x;
```



```
var x;  
console.log(x);    // undefined
```

var

- ▶ 같은 이름을 가진 변수를 여러 개 선언 가능 → 중복 선언
- ▶ Global context에 포함됨
- ▶ 전역 변수, 지역 변수의 차이가 불확실

```
for (var i = 0; i < 5; i++) {  
    console.log(i);  
}  
console.log(i);
```

let

- ▶ ES6에서 도입
- ▶ 블록 유효 범위를 가짐 – Script 또는 함수(블록)의 context에 포함됨
- ▶ 중복 선언 안됨
- ▶ hoisting이 안됨

```
console.log(x);    // ReferenceError
```

```
let x;
```

const

- ▶ ES6에서 도입
- ▶ 블록 유효 범위를 가짐
- ▶ 변하지 않는 값을 선언할 때
- ▶ 반드시 초기화 해야 함

```
<script>
var v0 = 'v0';
let l0 = 'l0';
console.log(v0, l0);
function fn1(){
    v0 = 'fn1-v0';
    l0 = 'fn1-l0';
    console.log(v0, l0);
}
fn1();
console.log(v0, l0);
```

```
function fn2(){
    var v0 = 'fn2-v0';
    let l0 = 'fn2-l0';
    console.log(v0, l0);
}
fn2();
console.log(v0, l0);
```

```
{
    var v0 = 'block-v0';
    let l0 = 'block-l0';
    console.log(v0, l0);
}
console.log(v0, l0);
```

```
var i = 'I';  
for(var i=0; i<1; i++) {  
    console.log(i);  
}  
console.log(i);
```

```
let j = 'J';  
for(let j=0; j<1; j++){  
    console.log(j);  
}  
console.log(j);
```


Source

- ▶ hoisting - <https://www.youtube.com/watch?v=fETYLCU2YYc>
- ▶ var, let, const - <https://www.youtube.com/watch?v=61iolhWgQt0>

분석

- ▶ 전역에 있는 var, let
- ▶ 함수 안에 있는 var, let
- ▶ 블록 안에 있는 var, let

	var	let	const
global scope	yes	no	no
script scope	no	yes	yes
function local scope	yes	yes	yes
block scope	no	yes	yes
재선언	yes	no	no
재할당	yes	yes	no