

[1] 조사 정리 제출:

- javascript Hoisting (var, let, const)
- javascript 함수(function) 선언 구조

java Script Hoisting과 Scope개념이란?

JavaScript Hoisting(호이스팅)

-함수 안에 있는 선언들을 모두 끌어올려서 해당 함수 유효 범위의 최상단에 선언하는 것을 말한다.

자바스크립트는 함수가 실행되기 전 함수안에 필요한 변수를 모두 유효범위의 최상단에 선언합니다.

이후 자바스크립트 Parser가 함수 실행 전 해당 함수를 한번 확인하고 함수 안에 존재하는 정보를 기억하고 있다가 실행합니다.

여기서 유효범위는 {} 범위를 말하는 것이며 함수 내에서 아래쪽에 존재하는 내용 중 필요한 값들을 끌어올리게 되는 과정입니다.

그러나 실제 메모리상의 변화는 없기때문에 실제 코드도 위로올라간다 라고 말할 수 없습니다.

호이스팅의 대상으로는 var변수와 함수의 선언문이 있으며 선언부만 위로 올라가고 할당 부분은 끌어올라가지 않는다.

let, const 변수선언과 함수표현식에서는 호이스팅이 발생하지 않는다.

호이스팅은 함수 선언문과 함수 표현식 두가지에서도 차이점을 보인다

var foo2 = func(); 함수 표현식 (foo2만 호이스팅 발생)

function foo() // 함수 선언문 (호이스팅 발생)

```
{  
}
```

이어서 자바스크립트의 변수 var, let, const 에 대해 알아보자.

var는 기본적으로 function scope를 가지게 되고 let, const는 block scope를 가지게 된다.

변수의 유효범위

먼저 var는 기본적으로 function scope를 가지게되고
let, const는 block scope를 가지게된다.

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
```

자바스크립트 함수 선언 구조는 3가지로 나뉘어있다.

함수 표현식.

함수 선언식.

앞서 함수의 표현식과 선언식은 위 호이스팅 설명에서 나타낸 것과 동일하며

호이스팅 발생의 차이를 보이고 있다는 것을 알고있기 때문에 넘어가겠습니다.