02

면접에서 단골로 나올만한 질문

02-b. Scope 와 Hoisting

변수와 함수의 호이스팅은 다른 각도에서 봐야 한다.

JavaScript Basic

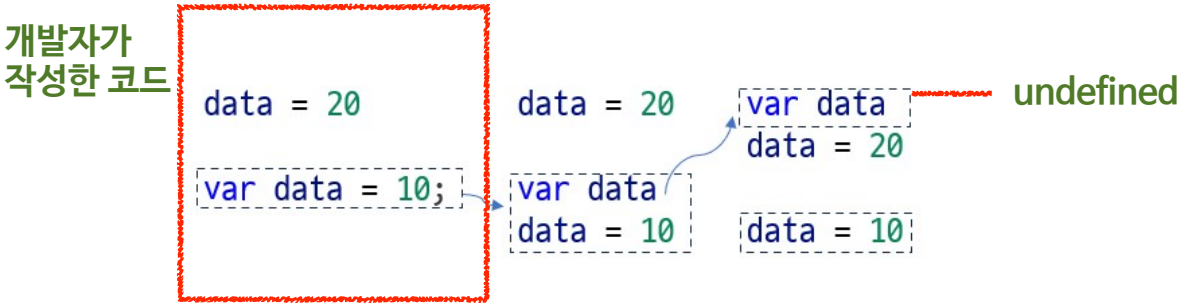
변수와 호이스팅

- 호이스팅(Hoisting)은 '무언가를 끌어올린다' 라는 의미의 단어로 아래에 선언된 변수를 위에서 사용하게 해주는 기법입니다.
- 코드는 위에서 아래로 흐르면서 실행됨으로 코드 위치만 보자면 변수가 아랫부분에 선언되어 있다면 윗 부분이 실행될때는 변수가 선언되지 않은 상태가 됨으로 에러가 발생되어야 하는데 실행시에 아래에 작성한 변수 선언 부분을 자동으로 위로 끌어올려서 에러 없이 코드가 진행되게 해주는 기법입니다.

- 변수를 선언하는 방법은 var, let, const 예약어중 하나를 이용합니다.
- 이중 호이스팅을 지원하는 변수 선언은 var 입니다.

변수와 호이스팅

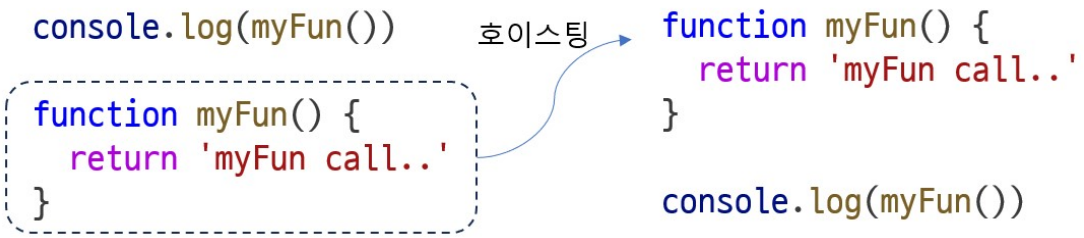
- 코드 아래에 선언된 변수가 호이스팅이 되면 어디선가 값 대입이 되기 전까지는 undefined 상태가 됩니다.



- 변수 선언을 let 과 const 로도 선언할 수 있는데 let, const 로 선언된 변수는 호이스팅이 지원되지 않습니다.

함수와 호이스팅

- 변수와 마찬가지로 함수도 코드의 윗 부분에서 함수가 선언되고 그 하위 어디선가 선언된 함수를 호출해서 이용하는 것이 일반적입니다.
- 그런데 함수의 선언위치와 함수의 이용 위치가 변경되어 코드 아랫부분에 선언된 함수를 코드 윗 부분에서 호출할 수 있게 지원이 됩니다.
- 이를 함수의 호이스팅이라고 합니다.
- 함수의 호이스팅은 함수 선언식에서 제공되며 표현식 함수는 호이스팅이 지원되지 않습니다.



스코프

- 같은 영역에서 실행되는 코드를 { } 로 묶어서 개발합니다.
- 그럼으로 하나의 { } 내에 선언된 코드들은 동일 스코프에서 실행된다는 표현을 합니다.
- { } 에 의해 코드가 묶여야 하는 대표적인 것이 함수, for, if 등입니다.
- 함수를 선언하면서 function a() { } 형식으로 함수가 호출될 때 실행되어야할 코드를 { } 로 묶습니다.
- 이 { } 내의 코드들은 함수 스코프에서 실행되는 코드들입니다.

var : 함수 스코프만 지원
let, const : 다른 블록 레벨 스코프 지원

스코프

중복선언

- 중복 선언이란 같은 동일 이름으로 변수가 중복되어 선언되는 것을 의미합니다.
- 중복 선언은 스코프와 관련있으며 동일 스코프내에서 중복인지? 다른 스코프에서 중복 선언인지에 따라 다릅니다.

다른 스코프에서 중복 선언

- 스코프란 { } 로 묶이는 코드의 영역을 의미하며 어떤 스코프 내에서 선언된 변수는 그 스코프에만 영향을 미치게 됩니다.
- 그럼으로 다른 스코프에 동일 이름으로 변수가 중복되었다고 하더라도 두 변수는 개별 변수가 되며 상호 영향을 미치지 않게 됩니다.

다른 스코프에서 변수 중복 선언

```
1  let data = '홍길동'
2
3  const myFun = () => {
4    let data = '김길동'
5    console.log(`in myFum, data = ${data}`)
6  }
```

local variable 우선

이렇게 중복 선언을 정말 많이한다.
의미 있는 단어로 식별자 네이밍을 하려면 매번 새로운 단어를 쓸 순 없다.

스코프

동일 스코프에서 변수 중복 선언

- 동일 스코프내에서 이미 선언된 변수명과 동일한 변수를 다시 중복으로 선언하는 것은 var 로 선언된 변수는 가능하지만 let, const 로 선언된 변수는 불가능합니다.

동일 스코프에서 변수 중복 선언

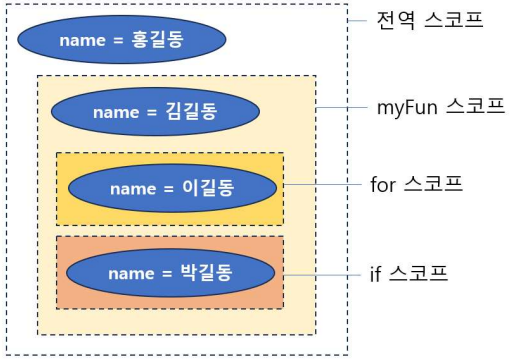
```
1   var data1 = 10
2   let data2 = 10
3   const data3 = 10
4
5   var data1 = '홍길동'
6   let data2 = '홍길동'//error
7   let data3 = '홍길동'//error
```

스코프

var 은 함수 스코프만 지원한다.

- 어떤 코드가 실행되는 영역을 스코프라고 하며 어떤 스코프에 선언된 변수는 그 스코프에서만 사용되고 다른 스코프에는 영향을 미치지 않게 하는 것이 기본입니다.

```
let name = '홍길동'
const myFun = () => {
  let name = '김길동'
  for(let i =0; i<1; i++){
    let name = '이길동'
  }
  if(true){
    let name = '박길동'
  }
}
```

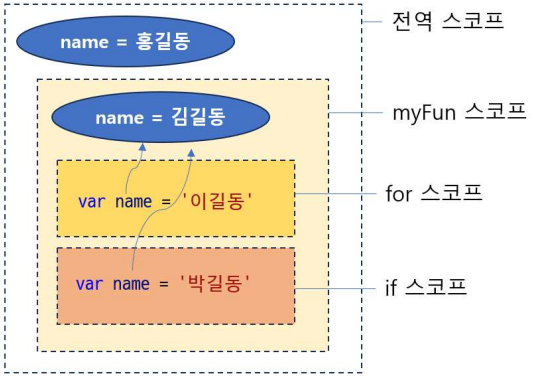


스코프

var 은 함수 스코프만 지원한다.

- var 로 선언한 변수는 함수 스코프만 지원하며 for, if 등의 다른 스코프는 지원하지 않습니다.

```
var name = '홍길동'
const myFun = () => {
  var name = '김길동'
  for(let i =0; i<1; i++){
    var name = '이길동'
  }
  if(true){
    var name = '박길동'
  }
}
```



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.

윌리엄 셰익스피어
William Shakespeare