

6_JS(ES6)_PART_07

- 스코프
- 스코프 체인
- 호이스팅

1. 스코프

: 어떤 변수가 선언된 공간 또는 환경

: 어떤 변수의 스코프는 해당 변수가 접근 될 수 있는 범위

```
// 기본 코드

// strict mode 를 작성하는 이유
// : 자바스크립트가 원칙대로 맞음, 틀림을 엄격하게 판단을 원한다면
// strict mode 라는 키워드를 넣어서 사용한다
// (자주 사용하지 않지만 실습을 위해)

'strict mode'

function calcAge(birthYear) {
    const age = 2025 - birthYear;
    return age;
}

const firstName = 'su';
console.log(calcAge(1990));
// console.log(age); : 에러 발생==> scope와 연관있음
```

(1) 글로벌 스코프

- 함수나 블록 바깥에 선언된 함수
- 어디에서든지 접근 가능

(2) 함수 스코프

- 함수 안에서에서만 접근 가능한 변수
- 지역 스코프라고도 불림

(3) 블록 스코프(ES6)

- 블록({}) 안에서만 접근 가능
- let, const에만 적용
- 함수 블록 역시 블록 스코프

2. 스코프 체인

- `isAdult` 는 secondFunc 스코프에서 참조 가능
- `decade` 는 secondFunc 스코플에서 참조 불가능

3. ★★☆☆ 호이스팅

: 끌어올린다.

: 자바스크립트 스코프 안에서 변수를 최상단에서 선언

: var, let, const 변수는 모두 선언 시 호이스팅 적용

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

console.log(bar);
//error: uncaught referenceError: bar is not defined
let var;
```

- var : 변수 선언(1) 시 초기화 (1) 가 동시에 일어난다
=====> undefined
- let, const : 변수 선언(1) 과 초기화(5) 가 다른 시점에 일어난다.
=====> referenceError

👁 초기화가 되었다는건,

: 변수가 메모리 공간에 값을 할당할 수 있는 장소를 마련했다는 뜻

즉 어떤 메모리 공간을 참조 했고 그 메모리 공간을 만들어 놓았으니 값을 넣기만 하면 된다.

● TDZ 일시적 사각지대

선언단계 > 일시적 사각지대(TDZ) > 초기화 단계(let foo;) > 선언 단계(foo = 1)

`var` 는 변수 선언시 초기화가 동시에 일어나기 때문에 TDZ 공간이 존재하지 않는다 왜냐하면 스코프의 시작 지점부터 초기화까지 되었기 때문이다.

`let` 과 `const` 는 변수 선언시점과 초기화시점이 겹이 있다. 그래서 그 겹이 있는 부분을 일시적 사각지대 TDZ라고 부른다.

왜 사각지대라고 하는가?: 말 그대로, 그 공간에는 변수에 할당되는 메모리 공간이 아직 만들어지지 않았기 때문이다. 그래서 참조하려고 하면 에러가 발생하고 변수를 사용하려고 하면 에러가 나는 것이다. 그래서 그것을 사각지대라고 한다.

- 변수와 함수가 선언되는 부분에서 호이스팅의 적용 유무에 따라 선언 되기전에 호출하는 것이 가능한지 불가능한지에 대해서 간단한 코드를 통해 살펴볼 수 있다.

```
const addArrow = (a, b) => a + b;

console.log(numProducts);
if(!numProducts){
  deleteShoppingCart();
}

var numProducts = 10;

function deleteShoppingCart(){
  console.log("모든 상품 삭제")
}
```

	호이스팅?	초기값	스코프
함수 선언식	yes	실제 함수	블록
var 변수	yes	undefined	함수
let, const 변수	no	uninitialized	블록
함수 표현식, 화살표 함수			

공통) 함수 표현식, 화살표 함수 : var를 쓰는지, let, const를 쓰는지에 따라 다르다.

■ Summary

자바스크립트 스코프

스코프 체인, 호이스팅