# 3주차 6차시

## 13장

#### 1. Lexical Environment란?

- 변수와 함수의 범위를 관리하는 구조적 측면이다.
- 코드가 실행될 때 새로운 Lexical Envirnment가 생성되어 외부 환경에 대한 링크를 통해 스코프 체인이 형성된다.
  - 변수와 함수는 이 체인 내에서 찾아지며, 가장 내부의 스코프에서 시작하여 원하는 변수가 찾아질 때까지 또는 전역 스코프에 도달할 때까지 외부로 진행된다.
- Lexical Environment는 2가지로 구성된다.
  - 1. Environment Record: 특정 컨텍스트 내의 변수와 함수 선언을 위한 저장 매커 니즘이다.
  - 2. 외부환경에 대한 참조: 변수를 찾기 위해 스코프 체인을 통한 외부 컨텍스트의 객체에 대한 링크를 제공한다.

#### 2. 아래 코드를 실행해 보시고 왜 저렇게 나오는지 설명하시오

```
var x = 1;
function foo() {
var x = 10;
bar();
}
function bar() {
console.log(x);
}
foo();
bar();
```

```
1
VM278:7 1
```

- foo() 에서 bar()를 호출한 경우 foo() 안의 x는 foo() 함수의 내부에서만 유효하기 때문에 전역변수 x가 출력되어 1이 출력된다.
- bar()를 호출한 경우 x는 전역변수 x밖에 없기 때문에 1이 출력된다.

### 14장

3주차 6차시

#### 1. 아래 콘솔로그에서 뭐가 나올까? 돌려봅시다.

```
var x = 'globaaaaal';
function banana(){
console.log(x); // 여기선 뭐가 나올까??
var x = 'locccccccccccccccccccccc;;
}
banana();
console.log(x);
```

```
undefined
VM274:7 globaaaaal
```

#### 1. 전역 객체란? 무엇인가?

a. 전역 객체란 전역 범위에 항상 존재하는 객체를 의미한다.

#### 2. 아래 키워드로 사용 억제법을 한 번 찾아보세요

#### • 즉시 실행 함수

- 。 즉시 실행 함수는 선언하면 그 즉시 실행된다.
- 모든 코드를 즉시 실행 함수로 감싸면 모든 변수가 즉시 실행 함수의 지역변수 가 된다.
- 따라서 내부 변수가 전역으로 저장되지 않아 전역 스코프의 오염을 줄일 수 있다.

#### • 네임스페이스 객체

전역의 네임스페이스 역할을 담당할 객체를 생성하고전역 변수처럼 사용하고 싶은 변수를 프로퍼티로 추가한다.

#### • 모듈 패턴

- 관련 있는 변수와 함수를 모아 즉시 실행함수로 감싸 하나의 모듈을 만든다.
- 모듈 패턴은 전역 변수의 억제와 동시에 캡슐화까지 구현 할 수 있다.

#### • ES6 모듈

- 。 ES6 모듈은 파일 자체의 독자적인 모듈 스코프를 제공한다.
- 。 따라서 전역 변수를 사용할 필요가 없다.

```
// 예시
```

3주차 6차시 2

```
<script type="module" src="파일명.mjs"></script>
```

3. 전역변수 + 지역변수 관련된 해석을 본인이 직접 최소 30줄 이상의 코드를 작성해서 서술해보세요.

```
var globalVariable = "Global";
function foo()
{
   var localVariable = "Local";
   var globalVariable = "Glocal?"; // 얘는 지역변수
   console.log(localVariable);
   console.log(globalVariable);
}
function func1()
   var localVariable = "Local";
   var globalVariable = "Glocal?";
   console.log(localVariable); // 지역변수 접근
   console.log(globalVariable); // 여기선 Glocal 출력
   func2();
function func2()
   console.log(localVariable); // 지역변수 접근 불가
   console.log(globalVariable); // 여기서는 Global 출력
}
foo();
func1();
console.log(globalVariable); // 전역변수
console.log(localVariable); // 지역변수 접근 불가
```

## 15장

1. 아래 콘솔로그의 결과값을 추론해보세요.

```
var x = 1;
var y = 1;
var x = 100;
var y;
console.log(x);
console.log(y);
```

- x는 선언된 것 중 가장 가까운 100이 출력 될 것이다
- y는 가장 가까운 선언에 값이 없으므로 그 다음 가까운 y값인 1이 출력 될 것이다.
- 2. 하단의 코드를 실행하고 에러가 난다면 왜 나는지 설명하시오.

```
let a = 1;
{
    let a = 2;
    let b = 3;
}
console.log(a);
console.log(b);
```

```
1
VM425:7 Uncaught ReferenceError: b is not defined
at <anonymous>:7:13
```

- let 키워드는 모든 코드 블록을 인정하는 블록 레벨 스코프를 따르므로 {} 안에 있는 a, b는 블록이 끝나면서 유효하지 않게 된다.
- 3. 지금까지 했던 코드를 기반으로 let, const 키워드를 적절히 사용해서 var 키워드 대신 써보세요.

```
let radius = 5;
let area = 0.0;
const pi = 3.14;

{
    let radius = 1;
    let area = 0.0;
    area = radius * radius * pi;
    console.log(area);
}

console.log(area);
```

3주차 6차시 4