

1. Scope란 ?

변수 선언과 할당, 함수 선언과 호출, 블록 실행, 전역 변수, 지역 변수, 블록 스코프, 전역 스코프

변수 선언과 할당은 어디서든 가능하지만, '변수 선언'과 '변수 호출'은 반드시 '변수 선언' 이후에 이루어져야 합니다.

이러한 변수 선언과 호출의 규칙을 정의하는 것이 바로 스코프(Scope)와 실행 컨텍스트(Execute Context)입니다.

스코프 Scope

스코프란 변수가 어디서 어디서 접근 가능한지를 정의하는 것입니다. 전역 스코프와 지역 스코프가 있습니다.

전역 스코프(Global Scope)와 지역 스코프(Local Scope)가 있습니다.

“ **전역 스코프(Global Scope)** : 어디서든 접근 가능한 스코프입니다. 전역 변수는 이 스코프에 속합니다. ”

“ **지역 스코프(Local Scope)** : 함수 내부나 블록 내에서만 접근 가능한 스코프입니다. ”

“ **블록 스코프(Block Scope)** : `{ }`로 감싸진 코드 블록 내에서만 접근 가능한 스코프입니다. (`let`, `const`) ”

“ **함수 스코프(Function Scope)** : 함수 내부에서만 접근 가능한 스코프입니다. (ES5 기준) ”

전역 스코프는 어디서든 접근 가능하며, 지역 스코프는 함수 내부에서만 접근 가능합니다. global scope는 전역 스코프를 의미합니다.

1. 全局作用域 (Global Scope)

在 JavaScript 中，全局作用域是指代码在浏览器或 Node.js 环境中运行时的默认作用域。

1) 变量声明：在全局作用域中声明的变量，可以在整个代码文件中访问。

```
let clock = "Wall Clock"; // 全局变量

function livingRoom() {
  console.log(clock); // "Wall Clock"
}

function kitchen() {
  console.log(clock); // "Wall Clock"
}

livingRoom();
kitchen();
```

2. 函数作用域 (Function Scope)

在 JavaScript 中，函数作用域是指代码在函数内部运行时的作用域。

2) 变量声明：在函数内部声明的变量，只能在函数内部访问。

3) 函数调用：在函数内部调用函数，只能在函数内部访问。

```
function livingRoom() {
  let sofa = "Sofa";
  console.log(sofa); // "Sofa"
}

livingRoom();
console.log(sofa); // Error: sofa is not defined
```

3. 块级作用域 (Block Scope)

在 JavaScript 中，块级作用域是指代码在块级语句（如 if、for、while 等）内部运行时的作用域。

if (true) {
 console.log("Remote Control");
}

console.log(remote); // Error: remote is not defined

```
if (true) {  
  let remote = "Remote Control";  
  console.log(remote); // "Remote Control"  
}  
  
console.log(remote); // Error: remote is not defined
```

4. 词法作用域 (Lexical Scope)

JavaScript 使用词法作用域，即变量在代码中的位置决定了其作用域。

例如，以下代码中，light 变量是在全局作用域中定义的，而 livingRoom 函数是在全局作用域中定义的。

因此，livingRoom 函数可以访问全局作用域中的 light 变量。

```
let light = "Ceiling Light";  
  
function livingRoom() {  
  console.log(light); // "Ceiling Light"  
}  
  
livingRoom();
```

5. 块级级作用域

ES6 引入了 let 和 const 关键字，它们创建的是块级级作用域的变量。

以下代码中，note 变量是在 livingRoom 函数作用域中定义的，因此只能在该函数作用域内访问。

```
function livingRoom() {  
  let note = "Notebook";  
  
  if (true) {  
    let pen = "Pen";  
    console.log(note); // "Notebook" (在 livingRoom 函数作用域内)  
  }  
}
```

```
console.log(pen); // "Pen"
```

```
}
```

```
console.log(pen); // Error: pen is not defined
```

```
}
```

```
livingRoom();
```

```
// pen: undefined
```

```
// pen: undefined
```

```
// pen: undefined
```

```
// pen: undefined
```

Revision #1

Created 20 April 2025 05:27:08 by DainChoi

Updated 20 April 2025 05:27:08 by DainChoi