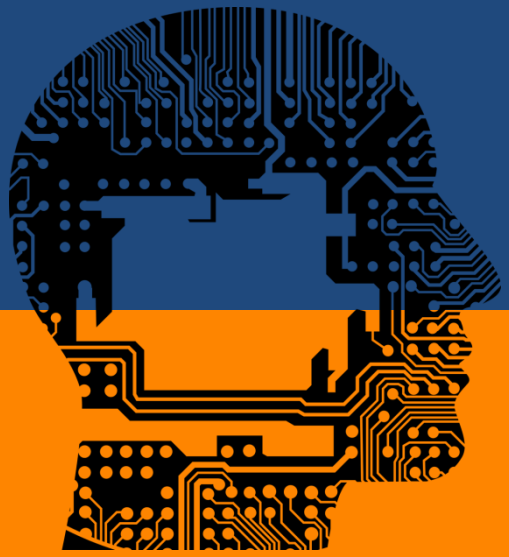
한국IT진흥부설
정보보호교육학원 아이섹

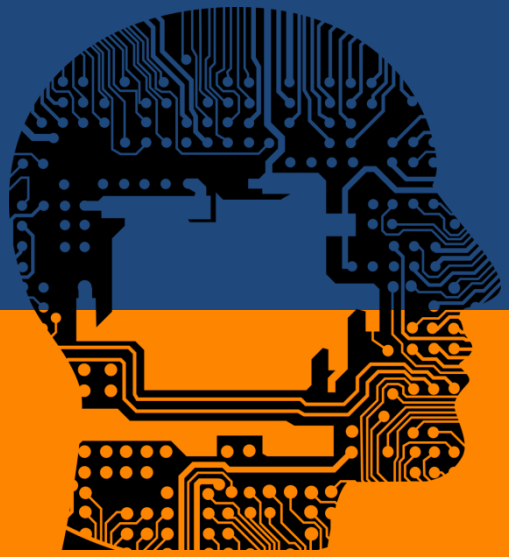


JAVA 웹 개발자 양성과정
Javascript

10강 - var의 문제점과 let, const

By SoonGu Hong

한국IT진흥부설
정보보호교육학원 아이섹



JAVA 웹 개발자 양성과정

Javascript

1. var 키워드 변수의 문제점

* 변수 중복 선언 허용

```
//1. 변수 중복 선언 허용

var x = 1;
var x = 100;

console.log(x); // 100
```

- var 키워드로 변수를 선언하면 동일한 이름으로 중복 선언시 var 키워드를 안 붙인 것처럼 동작합니다.

- 위와 같이 동일 이름 변수가 선언된 지 모르고 변수를 중복 선언하면 의도치 않게 변수의 값이 변경되는 부작용이 발생합니다.

* 블록 레벨 스코프를 지원하지 않음

```
//2. 블록 레벨 스코프를 지원하지 않음
var x = 1;

if (true) {
    var x = 10;
}
console.log(x);  // 10
```

- var 키워드 변수는 오로지 함수의 영역만을 지역 스코프로 인정합니다.
- 따라서 함수가 아닌 블록들에서는 모두 전역 변수로 일괄 적용됩니다.
- 이는 전역 변수를 남발할 가능성이 커지는 문제가 생깁니다.

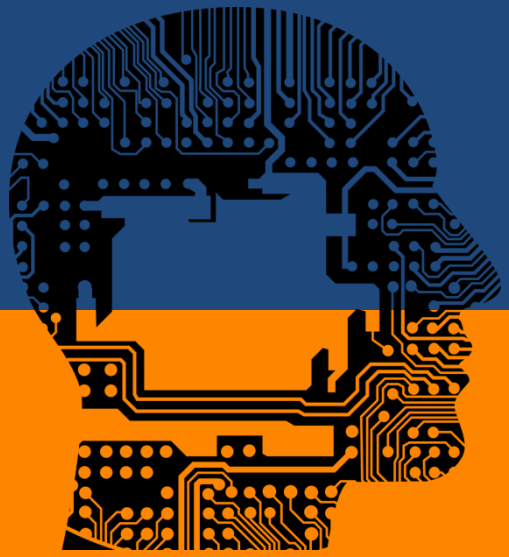
* 변수 호이스팅

```
//3. 변수 호이스팅
y = 100;
console.log(`y: ${y}`);

var y;
```

- var 키워드로 변수를 선언하면 변수 호이스팅에 의해 변수 선언문이 자동으로 맨 위로 끌어올려진 것처럼 동작합니다.
- 이는 프로그램의 흐름을 방해하여 코드의 가독성과 유지보수성을 현격하게 떨어뜨립니다.

한국IT진흥부설
정보보호교육학원 아이섹



JAVA 웹 개발자 양성과정

Javascript

2. let 키워드

* 변수 중복 선언 불가능

```
let x = 3;
let x = 4; // SyntaxError!
```

```
SyntaxError: Identifier 'x' has already been declared
```

- ES6에서는 var의 단점을 보완하기 위해 새로운 변수 선언 키워드 let과 const가 등장했습니다.
- let은 var와 달리 변수의 중복 선언을 허용하지 않고 에러를 발생시켜 안전성을 제공합니다.

* 블록 레벨 스코프를 지원

```
let y = 10;
if (true) {
    let y = 20;
    console.log(`if내부 y: ${y}`);
}
console.log(`if외부 y: ${y}`);
```

```
if내부 y: 20
if외부 y: 10
```

- let 키워드로 선언한 변수는 모든 코드 블록(함수, if문, for문, while문, try/catch문 등)을 각각의 지역 스코프로 인정하는 블록 레벨 스코프를 지원합니다.

- 이는 전역 변수의 사용을 제어할 수 있는 좋은 방법으로 쓰일 수 있습니다.

* 변수 호이스팅이 일어나지 않음

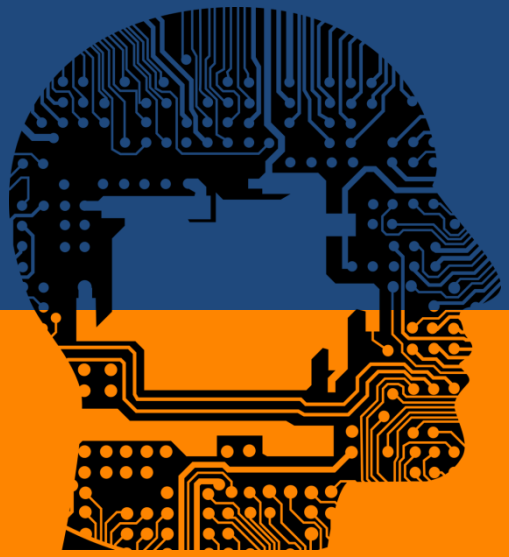
```
z = 10;
console.log(z); //ReferenceError

let z;
```

```
ReferenceError: Cannot access 'z' before initialization
```

- let 키워드로 선언된 변수는 '선언 단계'와 '초기화 단계'를 엄격하게 구분합니다.
- 따라서 암묵적으로 호이스팅이 일어나지 않아 안전한 코딩을 할 수 있게 도와줍니다.

한국IT진흥부설
정보보호교육학원 아이섹



JAVA 웹 개발자 양성과정

Javascript

2. const 키워드

* const는 반드시 선언과 동시에 초기화해야 한다.

```
const x; //SyntaxError
```

```
SyntaxError: Missing initializer in const declaration
```

- const 키워드는 상수를 선언하기 위해 사용합니다.
- const 키워드로 선언한 변수는 let과 마찬가지로 블록 레벨 스코프를 지원합니다.
- const는 let과 달리 반드시 초기에 값을 할당하는 초기화 작업을 필수로 해야 합니다.

* 상수란?

```
//세율
const TAX_RATE = 0.1;
//세전 가격
let preTaxPrice = 100;
//세후 가격
let afterTaxPrice = preTaxPrice + (preTaxPrice * TAX_RATE);
```

- 상수란 변수의 반대개념으로 값 변경이 금지된 변수를 말합니다.
- 상수는 값의 상태 유지와 가독성, 유지보수의 편의를 위해 적극적으로 사용해야 합니다.
- 상수 이름은 대문자로 지정하여 상수임을 관례적으로 알립니다.

* const 키워드와 객체

```
//객체와 const
const person = {
    name: 'Lee'
};
person.name = 'Park';
person.age = 30;
```

- const 키워드에 기본타입의 값을 할당한 경우 변경할 수 없지만 객체를 할당한 경우에는 프로퍼티를 변경할 수 있습니다.

* 정리

1. ES6부터는 var 키워드는 사용하지 마세요!

2. 재할당이 필요한 경우에만 한정하여 let을 사용하세요!
   이 때, 변수의 스코프는 최대한 좁게 만드세요.

3. 변경이 발생하지 않고 읽기 전용으로 사용하는 기본 타입의 값과 객체에는 const를 사용하세요.

감사합니다
THANK YOU