

# var, let, const 변수 & 호이스팅(Hoisting)

## 1. var

```
var human = 'white';
console.log(human); // white

var human = 'black';
console.log(human); // black
```

위의 예시와 같이 이미 선언된 변수라도 다시 재선언 할 수 있다. 몇줄 안되는 애플리케이션에서는 문제가 되지 않을 수 있다. 변수의 이름이 헤갈릴 일이 적기 때문. But 애플리케이션이 커지면 커질수록 변수 선언의 양이 자연스레 늘면서 어제 내가 어떤 이름의 변수를 선언했는지도 까먹게 된다. 그러면 위의 예시와 같이 이전에 선언한 변수에 다른 값이 선언되면서 의도치 않은 결과 값이 도출될 수 있다.

## 2. let

```
let human = 'white';
console.log(human); // white

let human = 'black';
console.log(human); // Uncaught SyntaxError: Identifier 'human' has already been declared

human = 'black';
console.log(human); // black
```

let는 변수 재선언이 불가능하다. 변수 재선언으로 인한 불상사를 방지 할 수 있다.

## 3. const

```
const human = 'white';
console.log(human); // white

const human = 'black';
console.log(human); // Uncaught SyntaxError: Identifier 'human' has already been declared

human = 'black';
console.log(human); // Uncaught TypeError: Assignment to constant variable.
```

const도 재선언이 불가능하다.

- 정리-

var    - 재선언 o, 재할당 o  
let    - 재선언 x, 재할당 o  
const - 재선언 x, 재할당 x

호이스팅(Hoisting)이란, var 선언문이나 function 선언문 등을 해당 스코프의 선두 즉, 맨 위로 옮긴 것처럼 동작하는 특성을 말한다. Javascript에서는 모든 선언(var, let, const, function, class)을 Hoisting한다.