

var vs let vs const 차이점

👤 배정	
▼ 상태	8월
🕒 속성	@2021년 8월 11일 오후 8:47
▼ 언어	

var, let, const는 무엇일까?

우선, 자바스크립트는 변수를 사용할때 변수 이름을 선언하고, 그 변수 이름에 값을 할당하여 사용을 한다.이런 변수 이름을 선언할 때 var,let,const 같은 키워드를 붙여 선언한다.

ex)

```
var food = '고기';  
let food = '고기';  
const food = '고기';
```

여기서 var는 기존부터 있었던 키워드이고, let과 const는 ES6 문법이다.과연 차이는 무엇일까?

차이점은?

1. 재할당과 재선언의 가능여부

재할당이란?

이미 값을 할당한 변수에 새로운 값을 할당하는 것을 말한다.

재선언이란?

이미 선언한 변수의 이름을 똑같이 선언하여 값을 덮어쓰는 것을 말한다.

var : 재할당과 재선언이 가능하다.

재할당

```
var food = '고기';  
food = '채소';  
console.log(food); // 채소
```

재선언

```
var food = '고기';  
var food = '채소';  
console.log(food); // 채소
```

let : 재할당은 가능하나 재선언은 불가하다.

재할당

```
let food = '고기';  
food = '채소';  
console.log(food); // 채소
```

재선언

```
let food = '고기';  
let food = '채소';  
console.log(food); // SyntaxError: Identifier 'food' has already been declared.
```

const : 재할당도 재선언도 불가하다.

재할당

```
const food = '고기';  
food = '채소';  
console.log(food); // TypeError: Assignment to constant variable.
```

재선언

```
const food = '고기';  
const food = '채소';  
console.log(food); // SyntaxError: Identifier 'food' has already been declared.
```

위의 비교 예시처럼 선언을 덮어 쓰기위해 var로 재선언을 해버리면 코드의 가독성을 저해하고 예상치 못한 버그를 만날 수 있다.하지만 let과 const는 재선언이 불가하기 때문에 재선

언으로 인한 단점을 예방할 수 있다.


2. 범위

var는 함수 범위 스코프다.

var는 함수 내부에서 선언되면 그 함수 내부에서 사용이 가능하다.

```
function blockTest() {  
  for (var i = 0; i < 3; i++) {}  
  console.log(i)  
}  
  
blockTest() // 3
```

let과 const는 블록 범위 스코프다.

let과 const는 함수 내부에서도  이렇게 생긴 블록 내에서만 사용이 가능하다.

```
function blockTest() {  
  for (let i = 0; i < 3; i++) {}  
  console.log(i)  
}  
  
blockTest() // ReferenceError: i is not defined
```

```
function blockTest() {  
  for (const i = 0; i < 3; i++) {}  
  console.log(i)  
}  
  
blockTest() // TypeError: Assignment to constant variable.
```

요즘 var는 거의 쓰이지 않는다!

그럼 let과 const는 언제 쓰일까?

const를 기본으로 사용하는 것이 좋다.

아까 설명했듯이 재선언 재할당 모두 불가능해서 에러를 방지할 수 있기 때문에 안전하다.

const를 사용할 수 없을땐 let을 사용하자.

예를 들어 반복되는 작업이 있거나, 카운트를 세어야 하는 경우에는 재할당이 가능한 let을 사용하는게 바람직하다.