

22. this

▼ this 값의 결정 시기

this는 함수 선언 시 정적으로 결정되는 것이 아니라, 함수가 어떻게 호출되었는지에 따라 this에 바인딩할 객체가 동적으로 결정된다.


▼ 함수 호출 방식과 this 바인딩

1. 일반함수

- 전역 객체가 바인딩된다.
- 위치와 상관없이 일반함수로 출된다면, 전역 객체가 바인딩된다.

```
const obj = {  
  foo() {  
    function bar() {  
      console.log(this);  
    }  
  
    bar(); // 일반 함수로 호출되었으므로, window가 나온다.  
  }  
}
```

2. 메서드

-  앞에 기술된 객체가 바인딩된다.
- 그러나 일반 변수에 할당되어 호출되면 일반함수로 호출된 것이므로, 전역객체가 바인딩된다.

3. 생성자 함수

- 미래에 생성할 인스턴스가 바인딩된다.

4. apply/call/bind 메서드에 대한 간접 호출

- 인수로 전달된 객체에 바인딩 된다.

```
const obj = {  
  name: 'kim',  
  sayHi() {  
    console.log(this.name);  
  }  
}  
  
const obj = {  
  name: 'kim',  
  sayHi() {  
    console.log(obj.name);  
  }  
}
```

▼ 객체 내부에 this를 적는 것과 객체 이름을 적는 것의 차이

객체 이름을 적는 것은 전역 스코프에서 해당 식별자를 검색해야 한다.

그러나 this를 적는다면, 현재 객체 내에서 프로퍼티를 찾는다는 의미이므로 시간과 비용을 절약할 수 있다.