

22장. this

22.1 this 키워드

▼ 메서드 특징

메서드 = 자신이 속한 객체의 상태로 프로퍼티를 참조하고 변경할 수 있어야 한다.

→ 먼저 자신이 속한 객체를 가리키는 식별자를 참조할 수 있어야 한다.

▼ 객체 리터럴 방식으로 생성한 객체 경우

메서드 내부에서 메서드 자신이 속한 객체를 가리키는 식별자를 재귀적으로 참조 가능

→ 하지만 자기 자신이 속한 객체를 재귀적으로 참조하는 방식은 X

▼ 생성자 함수 방식으로 인스턴스를 생성하는 경우

자신이 속한 객체 또는 자신이 생성할 인스턴스를 가리키는 특수한 식별자 필요 = this

▼ this란?

자신이 속한 객체 또는 자신이 생성할 인스턴스를 가리키는 자기 참조 변수

→ this를 통해 자신이 속한 객체 또는 자신이 생성할 인스턴스의 프로퍼티나 메서드를 참조 가능

this가 가리키는 값(this 바인딩)은 함수 호출 방식에 의해 동적으로 결정된다.

▼ this 바인딩이란?

식별자와 값을 연결하는 과정

▼ 객체 리터럴과 생성자 함수 내부에서 this가 가리키는 값

객체 리터럴: 메서드를 호출한 객체

생성자 함수: 생성자 함수가 생성할 인스턴스

strict mode(엄격 모드)가 적용된 일반 함수 내부의 this는 undefined가 바인딩 된다.

22.2 함수 호출 방식과 this 바인딩

▼ 렉시컬 스코프와 this 바인딩 결정 시기

렉시컬 스코프: 함수 정의가 평가되어 **함수 객체가 생성되는 시점**

this 바인딩: **함수 호출 시점에 결정**

▼ 함수 호출하는 방식

▼ 1. 일반 함수 호출

기본적으로 this에는 전역 객체가 바인딩된다.

→ 일반 함수로 호출하면 함수 내부의 this에는 전역 객체가 바인딩된다.

But, 일반함수에서 this는 의미 X

```
function foo() {  
  console.log(this) // window  
  function bar() {  
    console.log(this) // window  
  }  
}
```

콜백 함수가 일반 함수로 호출된다면?

⇒ 콜백 함수 내부의 this = 전역 객체 바인딩

▼ setTimeout 함수란?

두 번째 인수를 전달한 시간만큼 대기한 후,

첫 번째 인수로 전달한 콜백 함수를 호출하는 타이머 함수

▼ 2. 메서드 호출

메서드를 호출할 때 메서드 이름 앞의 마침표(.) 연산자 앞에 기술한 객체가 바인딩된다.

!!주의: 메서드 내부의 this ≠ 메서드를 소유한 객체

= **메서드를 호출한 객체**에 바인딩된다.

메서드는 객체에 포함된 것이 아닌, 독립적으로 존재하는 별도의 객체이다.

따라서, 메서드 내부의 this는 프로퍼티로 메서드를 가리키고 있는 객체와 관계없고, 메서드를 호출한 객체에 바인딩된다.

▼ 3. 생성자 함수 호출

생성자 함수 내부의 this == 생성자 함수가 (미래에) 생성할 인스턴스가 바인딩된다.

```
function Circle(radius) {
  this.radius = radius;
  this.getDiameter = function () {
    return 2 * this.radius;
  };
}

const Circle1 = new Circle(5);
console.log(circle1.getDiameter()); // 10

const Circle2 = Circle(5); // 생성자 함수 x -> 일반 함수의 호출
console.log(circle2); // undefined
```

▼ 4. Function.prototype.apply/call/bind 메서드에 의한 간접 호출

apply와 call 메서드의 본질적인 기능 = 함수를 호출하는 것

함수를 호출하면 첫 번째 인수로 전달한 **특정 객체를 호출한 함수의 this에 바인딩**한다.

▼ apply 메서드와 call 메서드 차이점

apply 메서드 = 호출할 함수의 인수를 배열로 묶어 전달

call 메서드 = 호출할 함수의 인수를 쉼표로 구분한 리스트 형식으로 전달

인수를 전달하는 방식의 차이가 있다.

하지만 this로 사용할 객체를 전달하면서 함수를 호출하는 점에서는 동일하다.

▼ bind 메서드 특징

apply와 call 메서드와 달리 함수를 호출하지 않는다.

첫 번째 인수로 전달한 값으로 this 바인딩이 교체된 함수를 새롭게 생성해 반환

→ 메서드의 this와 내부의 중첩 함수 또는 콜백 함수의 this가 불일치하는 문제를 해결하기 위해 사용