

## 22장 this

### 22.1 this 키워드

- This를 통해 자신이 속한 객체를 가리키는 식별자를 참조할 수 있다.
- this는 자신이 속한 객체 또는 자신이 생성할 인스턴스를 가리키는 자기 참조 변수(self-referencing variable)이다.
- 함수 호출시 arguments 객체와 this가 암묵적으로 함수 내부에 전달하고 이를 지역변수처럼 사용한다.
- this는 함수 호출 방식에 의존하여 동적으로 결정한다.
- \*\* this 바인딩**

- 자바나 C++의 클래스 기반 언어에서 this는 언제나 클래스가 생성하는 인스턴스를 가리킨다.
- 하지만 JS에서는 동적으로 결정된다.
- Strict mode 역시 this 바인딩에 영향을 준다.
- this는 객체의 프로퍼티나 메서드를 참조하기 위한 자기 참조 변수이므로 일반적으로 객체의 메서드 내부 또는 생성자 함수 내부에서만 의미가 있다.
- 따라서 일반 함수 내부에서 this를 사용할 필요가 없다.
- Strict mode에서 함수내부 this는 undefined

### 0. 머릿말

- \*\* 렉시컬 스코프와 this 바인딩은 결정 시기가 다르다.
- 함수호출의 4가지 방식
  - 일반 함수 호출
  - 메서드 호출
  - 생성자 함수 호출
  - Apply / call / bind 메서드 간접호출

### 1. 일반 함수 호출

- 기본적으로 this에 전역 객체(global object)가 바인딩
- 중첩함수까지 this바인딩이 같다 — 콜백함수 또한 마찬가지
- 어떠한 함수라도 일반 함수로 호출되면 this에 전역 객체 바인딩
- 메서드 내부에 정의한 중첩 함수 또는 콜백함수(보조 함수)가 일반함수로 호출될 때 메서드 내의 중첩 함수 또는 콜백 함수의 this가 인스턴스 객체 말고 전역 객체를 바인딩하는 문제점

- This 바인딩을 일치시키기 위한 방법
  - That 변수 사용
  - Apply, call, bind 메서드 사용
  - 화살표 함수 사용

### 22.2 함수 호출 방식과 this 바인딩

### 2. 메서드 호출

- 메서드 내부의 this에는 메서드를 호출한 객체, 즉 메서드를 호출할 때 메서드 이름 앞의 마침표(.) 연산자 앞에 기술한 객체가 바인딩 된다.

- 주의할 것은 메서드 내부의 this는 메서드를 소유한 객체가 아닌 메서드를 호출한 객체에 바인딩된다는 것

### 3. 생성자 함수 호출

- 생성자 함수 내부의 this에는 생성자 함수가 미래에 생성할 인스턴스가 반영된다.

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

- 이들 메서드는 모든 함수가 상속받아 사용가능
- Apply 와 call 메서드
  - 호출할 함수에 인수를 전달하는 방식만 다를 뿐 this로 사용할 객체를 전달하면서 함수를 호출한다.
- bind 메서드
  - 위 두 개의 메서드와는 달리 함수를 호출하지 않고 this로 사용할 객체만 전달한다.
  - 메서드의 this와 메서드 내부의 중첩 함수 또는 콜백 함수의 this가 불일치하는 문제를 해결하기 위해 유용하게 사용한다.

### 총정리

- 함수 호출 방식
  - 일반 함수 호출
  - 메서드 호출
  - 생성자 함수 호출
  - Function.prototype.apply / call / bind 메서드에 의한 간접 호출

### This 바인딩

- 전역 객체
- 메서드를 호출한 객체
- 생성자 함수가 미래에 생성할 인스턴스
- 3개의 각 메서드에 첫번째 인수로 전달한 객체