

ㅋ
ㅋ

08 Dec 2018 | javascript basic this

es6

binding이란?

프로젝트 경험이 거의 없었을 때는 this를 binding한다는 말조차 이해가 가지 않았었다. javascript기본서에서 call, apply, bind가 나오면 머리가 아팠다. binding이란 도대체 뭘까?

javascript의 함수는 각자 자신만의 this라는 것을 정의한다. 예를 들어 자기소개를 하는 함수를 만들기 위해 say()이라는 함수를 만든다고 하자.

```
const say = function() {
  console.log(this); // 여기서 this는 뭘까?
  console.log("Hello, my name is " + this.name);
};

say();
```

실행해보면

Window {postMessage: f, blur: f, focus: f, close: f, parent: Window, ...}	app.js:39
Hello, my name is	app.js:39

window 객체가 나타난다. 기본적으로 this 는 window 이기 때문이다. 사실 참 어려운데, 꼭 window 라고만 말할 수는 없다. this는 객체 내부, 객체 메서드 호출시, 생성자 new 호출시, 명시적 bind시에 따라 바뀌기 때문이다.

어찌되었든 우리는 say 함수에서 Window 객체를 사용하고 싶지 않다. 즉, this 를 그때 그때 알맞은 객체로 바꿔서 this 값에 따라 인사말이 할 것이다. 이 것이 this의 binding이다. 명시적으로 위의 this 를 Window가 아닌 다른 객체로 바꿔주는 함수가 call, apply, bind이다.

call과 apply

say 함수의 this를 변경하고 싶다면, 당연히 this를 대체할 객체가 있어야 한다. 코드를 조금 수정해서 아래와 같이 만들었다.

ㄱㄹ
ㄱㄷ

```
const obj = { name: 'Tom' };

const say = function(city) {
  console.log(`Hello, my name is ${this.name}, I live in ${city}`);
}

say("seoul"); // Hello, my name is , I live in seoul
say.call(obj, "seoul"); // Hello, myname is Tom, I live in soeol
say.apply(obj, ["seoul"]); // Hello, myname is Tom, I live in soeol
```

call 과 apply 는 함수를 호출하는 함수이다. 그러나 그냥 실행하는 것이 아니라 첫 번째 인자에 this 로 setting하고 싶은 객체를 넘겨주어 this 를 바꾸고나서 실행한다.

첫 번째 실행인 say("soeul") 의 경우는 say가 실행 될 때 this에 아무런 setting이 되어있지 않으므로 this는 window객체이다.

두 번째 실행인 say.call(obj, "seoul"); 의 경우와 세 번째 실행인 say.apply(obj, "seoul") 은 this를 obj로 변경시켰으므로 원하는 값이 나온다.

call과 apply의 유일한 차이점은, 첫 번째 인자(this를 대체할 값)를 제외하고, 실제 say에 필요한 parameter를 입력하는 방식이다. call과는 다르게 apply함수는 두 번째 인자부터 모두 배열에 넣어야 한다.

bind

```
const obj = { name: 'Tom' };

const say = function(city) {
  console.log(`Hello, my name is ${this.name}, I live in ${city}`);
}

const boundSay = say.bind(obj);
boundSay("seoul"); // Hello, my name is Tom, I live in seoul
```

bind 함수가 call , apply 와 다른 점은 함수를 실행하지 않는다는 점이다. 대신 bound함수를 리턴한다. 이 bound함수(boundSay)는 이제부터 this를 obj로 갖고 있기 때문에 나중에 사용해도 된다. bind에 사용하는 나머지 rest 파라미터는 call과 apply와 동일하다.

참고자료

- [How-to: call\(\) , apply\(\) and bind\(\) in JavaScript](#)

크리
크리

Related Posts

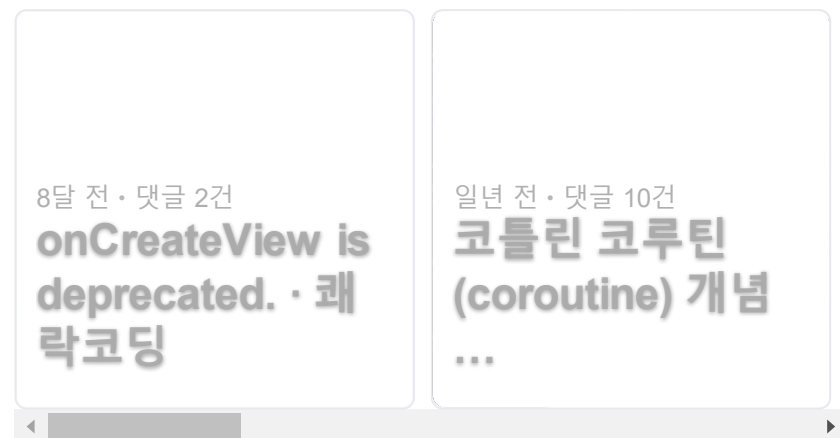
HackerRank - Organizing Containers of Balls 05 Oct 2020

이모티콘 컨테이너 UI 구현 (채팅, like 카카오톡) 30 Jul 2020

Kotlin exhaustive with sealed class and when 08 Jun 2020

Comments

HTTPS://WOOOOOOAK.GITHUB.IO/의 다른 댓글.



댓글 커뮤니티 개인정보 보호정책 로그인 ▾

추천 1 Tweet 공유 인기순 ▾