타입스크립트 #4 this

서론 : 이번 강의가 마지막 강의가 될 듯 합니다. 여기까지 따라오느라 고생하신 여러분 진심으로 수고 많으셨습니다.

This ? :

This 키워드는 함수가 호출 될 때 지정되는 변수입니다. 꽤나 어려운 내용 덕분에 바로 따라가기란 어려울 것으로 예상되니, this키워드에 대한 설명과 함께 타입스크립트의 this 설명을 이어 나가도록 하겠습니다.

This를 이것이다 ! 라고 일률 적으로 설명하기는 어렵습니다. 왜냐하면 this는 arrow function 에서의 this와 함수 선언식에서 가리키는 this가 다르기 때문입니다. 다음과 같은 예제를 함께 보도록 하겠습니다.

const dummy = {

declaration : function(){

console.log(this);

},

arrow : () =>{

console.log(this)

}

}

dummy.declaration();

dummy.arrow();

declation 메서드와 arrow 메서드는 둘다, this키워드를 출력하도록 되어 있습니다. 두개의 함수는 모양만 봐서는 차이가 존재하지 않습니다. 이 두개의 this는 과연 같은 this를 가리키게 될까요? 정답은 X입니다. 함수 선언식으로 작성된 함수는 dummy객체를 가리키고 있고, 함수 표현식으로 작성된 함수는 빈 객체를 가리키고 있습니다. 실은 이 객체는 globalThis 키워드 같은 것이 아니라, module.exports를 가리키고 있습니다. 한번 확인 해볼까요 ?

expression : () =>{

console.log(this);

console.log(this === exports)

}

결과는 놀랍게도 true가 나옵니다. 이번 강의에서는 node.js의 내부를 뜯어 보는 것이 아니기 때문에 이런게 있구나라고 알고 컴공 수업시간에 갑작스레 손을 들어 아는 척을 해보도록 합시다.

그렇다면 이 두 함수 작성법에 따른 this는 어떻게 지정되는 가를 한번 알아보도록 합시다.

먼저 함수 선언식의 this키워드는 “이 함수를 호출 하는 객체를 this 키워드에 할당 하게 됩니다. ”

const dummy = {

declaration : function(){

console.log(this);

}

}

const declaration = dummy.declaration;

declaration();

예제를 통해서 알아봅시다. 다음과 같이 dummy객체를 변경하고 dummy객체의 declaration함수를 전역에 할당 한 후 호출 하면, global 객체가 this에 할당 된 것을 알 수 있습니다. 조금 더 예제를 변경 해보도록 하겠습니다 .

Dummy\_2 객체를 새로 만들고 dummy객체와 동일하게 declaration을 만들고 이에 dummy의 declarartion함수를 할당 해주도록 하겠습니다.

const dummy = {

declaration : function(){

console.log(this);

}

}

const dummy\_2 = {

name : "dummy2",

declaration : dummy.declaration

}

dummy\_2.declaration()

전역 스코프에서 declaration함수를 변수에 할당 한 후 호출 한 것과 같이 “함수를 호출 하는 객체”를 this 키워드에 할당 하는 모습을 볼 수 있습니다.

그렇다면 이번에는 화살표 함수로 작성된 함수에서 this키워드는 무엇을 가리키고 있는 것일까요 ? 바로 “상위 스코프의 this” 이를 영어로 lexical scope라고 합니다.

다음과 같은 예제를 작성 하도록 하겠습니다.

function Factory(name){

function Method(){

return () => console.log(this.name)

}

return {

name,

Method

}

}

const dummy = new Factory("lee");

const dummy\_2 = dummy.Method()

dummy\_2()

function 키워드를 사용한 함수 선언식과 같이 전역스코프에 있는 dummy\_2 변수에 sayMyname을 할당하였지만, dummy 변수의 name을 그대로 사용하는 모습을 볼 수 있습니다. 그래서 정리를 하자면 Function키워드를 통하여 작성된 함수 내에서 this는 “실행 컨텍스트에 따라 this가 동적으로 할당된다.” Arrow function을 사용된 함수 내에서의 this는 “작성된 객체의 상위 스코프의 this를 정적으로 할당한다” 입니다. 그래서 보통 react를 사용 할 때 function 키워드를 사용한 component 메서드를 써 element에 이벤트 콜백으로 사용하기 보다, 정적으로 this가 bind된 arrow function을 주로 사용 합니다.

Typescript this 사용법 :