Item 49.

콜백에서 this에 대한 타입 제공하기

'23. 7. 29. (SAT) 신현호(@SWARVY)

this

원래 JS 에서 이상한 일이 일어났을 때
이[redacted]를 찍으면 대충 맞는다

역시나 이번 이야기도
JS 만악의 근원에서 출발한다

자바스크립트를 가장 혼란스럽게 만드는 존재

this가 window를 가르키는 경우

단독사용한 this

```
> this
< ▶ Window {parent: Window, opener: null, top: Window, le
  ngth: 0, frames: Window, …}
```

함수안에서 쓴 this

```
1  function myFunction() {
2    return this;
3  }
4  console.log(myFunction()); //Window
```

this가 undefined인 경우

함수안에서 사용한 this (strict mode 적용)

```
"use strict";

var num = 0;
function addNum() {
  this.num = 100; //ERROR! Cannot set property 'num' of undefined
  num++;
}

addNum();
```

함수 내에 default binding이 없어서 undefined를 반환하게 됩니다

나머지 this의 경우...

메서드 안에서 사용한 this

```
var person = {
  firstName: 'John',
  lastName: 'Doe',
  fullName: function () {
    return this.firstName + ' ' + this.lastName;
  },
};

person.fullName(); //"John Doe"
```

해당 메서드를 호출한 객체로 바인딩됩니다

이벤트 핸들러 안에서 사용한 this

```
var btn = document.querySelector('#btn')
btn.addEventListener('click', function () {
  console.log(this); //#btn
});
```

이벤트를 받는 html요소를 가르킵니다

생성자 안에서 사용한 this

```
function Person(name) {
  this.name = name;
}

var kim = new Person('kim');
var lee = new Person('lee');

console.log(kim.name); //kim
console.log(lee.name); //lee
```

생성자 함수가 생성하는 객체로 this가 바인딩됩니다

이외 등등등...

나를 화나게 하는 this에게서 도망가는 가장 쉬운방법

화살함수를 쓰세요

```
var Person = function (name, age) {
  this.name = name;
  this.age = age;
  this.say = function () {
    console.log(this); // Person {name: "Nana", age: 28}

    setTimeout(function () {
      console.log(this); // Window
      console.log(this.name + ' is ' + this.age + ' years old');
    }, 100);
  };
};
var me = new Person('Nana', 28);

me.say(); //global is undefined years old
```

작성중인 라이브러리에 this가 있다면..?

```
function addKeyListener(
  el: HTMLElement,
  fn: (this: HTMLElement, e: KeyboardEvent) => void
) {
  el.addEventListener('keydown', e => {
    fn.call(el, e);
  });
}
```

콜백함수의 매개변수에 this를 추가하고,
콜백 함수를 call로 호출해서 사용

작성중인 라이브러리에 this가 있다면..?

```
function addKeyListener(
  el: HTMLElement,
  fn: (this: HTMLElement, e: KeyboardEvent) => void
) {
  el.addEventListener('keydown', e => {
    fn(e);
// ~~~~ 'void' 형식의 'this' 컨텍스트를
//         메서드의 'HTMLElement' 형식 'this'에 할당할 수 없습니다.
  });
}
```

this의 바인딩을 체크해주기때문에
혹시 모를 실수 방지 가능

```
declare let el: HTMLElement;
addKeyListener(el, function(e) {
  this.innerHTML;  // 정상, "this"는 HTMLElement 타입
});
```

라이브러리 사용자의 콜백 함수에서
this 참조 가능 / 타입안전성도 얻을 수 있어요

작성중인 라이브러리에 this가 있다면..?

```
class Foo {
  registerHandler(el: HTMLElement) {
    addKeyListener(el, e => {
      this.innerHTML;
        // ~~~~~~~~ 'Foo' 유형에 'innerHTML' 속성이 없습니다.
    });
  }
}
```

라이브러리 사용자가 콜백을 화살표 함수로 작성하고
this를 참조하려고 하면 TS가 문제를 잡아냅니다