2024\_1학기 프론트엔드프레임워크 3/11

**모듈:** 다른 javascript 소스코드 파일에 구현된 함수, 클래스, 객체 등을 가져와서 사용하기 위한 문법

A.js 파일에 구현된 함수/클래스/객체 등을 B.js 파일에서 사용할 수 있으려면,

A.js 에서 제공(export)해야 하고 B.js 에서 수입(import) 해야 한다.

"type": "module" > ES6 module 문법을 사용하려면, package.json 파일에 써야한다.

import { average, sum } from './math1.js'

// math1js 모듈에서 export하는 average, sum 함수를 import 한다

export { average, sum, PI }; > 여러 개(3개)로 export 한것

export default { average, sum, PI } > 하나로 묶은 것

import Math from './math2.js'

// export default 된 것은 위와 같은 변수 한 개로 import 할 수 있다

console.log(Math.average(3, 4, 5, 6));

console.log(Math.PI);

> 하나씩 export 가능

> import \* as Math from './math2.js' //이렇게 import 가능

**함수와 화살표 함수**

**메소드와 this**

let rectangle1 = {

    width: 4,

    height: 5,

    method1() {

        let area = this.width \* this.height; // rectangle1의 객체가리킴

        console.log(area);

    }

};

rectangle1.method1();

**inner함수:** 다른 함수(메소드) 내부에 정의된 함수.

**화살표 함수**

let getArea = () => this.width \* this.height;

let getArea2 = getArea.bind(rectangle1); //###.함수() 형태로 호출

화살표 함수 본문의 this는, 화살표 함수 소스코드를 둘러 싸고 있는 객체를 가르킨다

### function 키워드 문법 함수

function 키워드 문법으로 만든 함수 본문의 this는 무엇을 가르키는가?

let func2 = func1.bind(obj)

위와 같은 코드로 obj 객체에 bind된 func2 본문의 this는 obj 객체를 가르킨다.

obj.func(...)

위와 같은 코드로 호출된 함수 본문의 this는 obj 객체를 가르킨다.

func(...)

위와 같은 코드로 호출된 함수가 bind 되지도 않았다면,

이 함수 본문의 this는 어떤 객체를 가르키지 않는다.