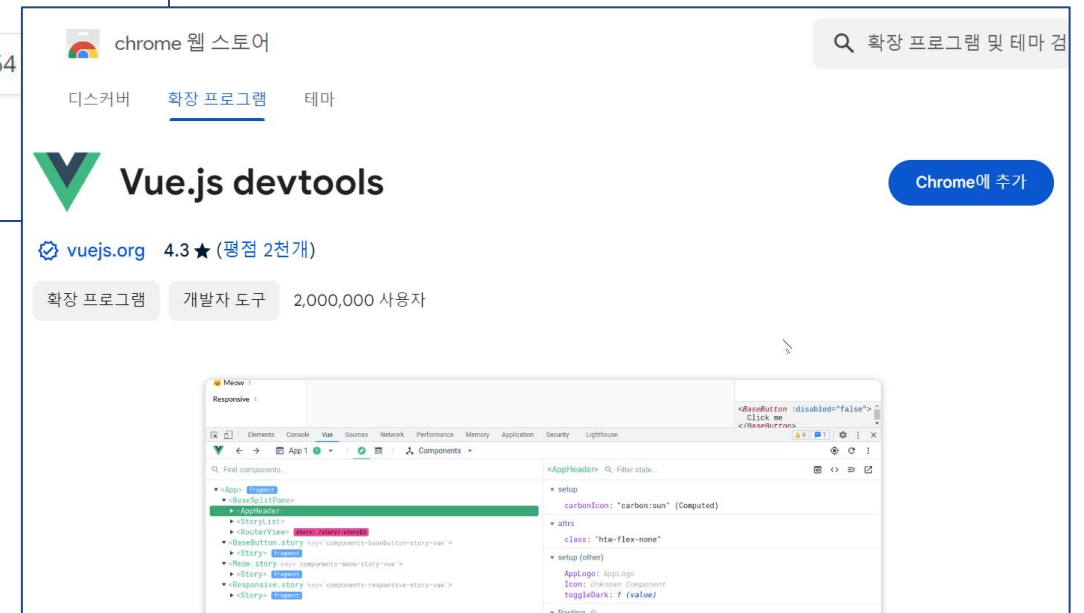
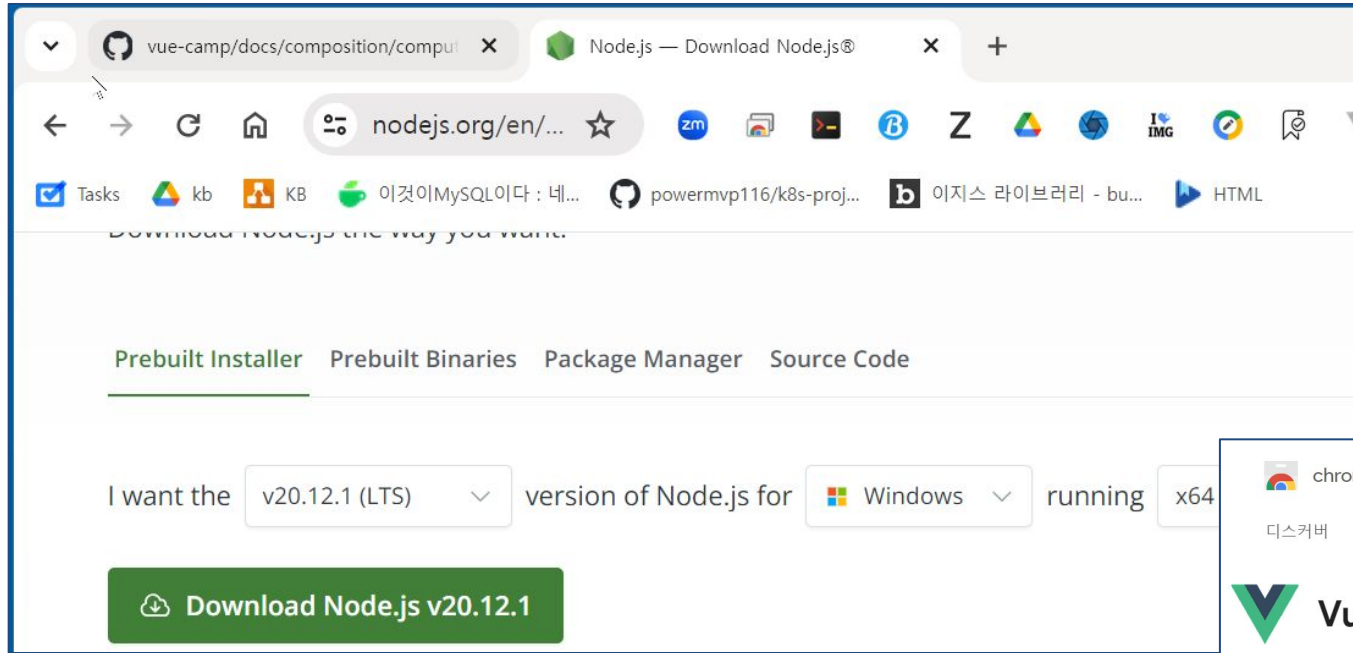


2024년 상반기 K-디지털 트레이닝

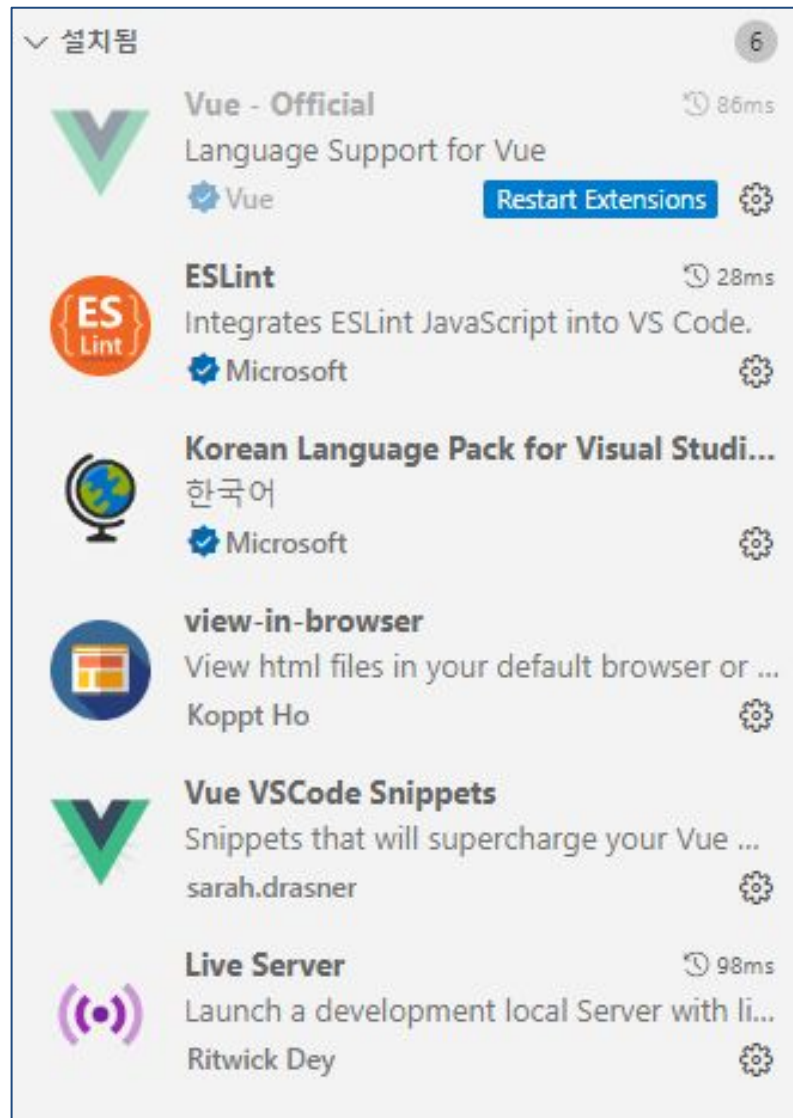
ES6

[KB] IT's Your Life

- nodeJS와 vueJS 크롬 확장 플러그인 설치하시오.



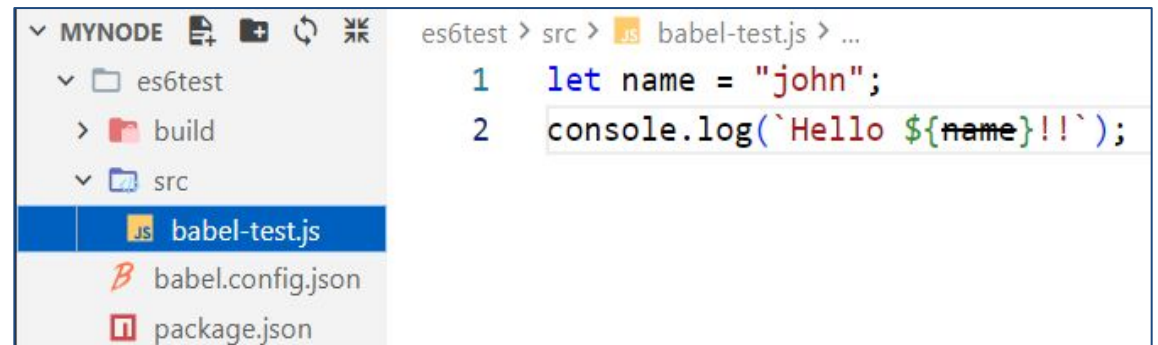
- vsCode에 플러그인 설치하시오.



- 다음 es6test 프로젝트를 생성하고, babel을 설정하고 테스트하시오.

```
1  {
2    "name": "test",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1",
8      "build": "babel src -d build"
9    },
10   "author": "",
11   "license": "ISC",
12   "devDependencies": {
13     "@babel/cli": "^7.24.5",
14     "@babel/core": "^7.24.5",
15     "@babel/preset-env": "^7.24.5",
16     "core-js": "^3.37.0"
17   }
18 }
19
```

- npm init
- npm install --save-dev @babel/core @babel/cli @babel/preset-env core-js
- babel.config.json
- src/babel-test.js
- package.json에
"build": "babel src -d build"
- npm run build 또는 npx babel src -d build
- node ../build/babel-test.js



```
es6test > src > babel-test.js > ...
1  let name = "john";
2  console.log(`Hello ${name}!!`);
```

Q2 - 가변 파라미터

- 다음과 같이 가변적인 입력값을 허용할 수 있는 함수를 정의하여, 프로젝트를 빌드하고 다음과 같이 출력되도록 실행하시오.

```
addContact("홍길동", "010-222-3331")
addContact("이몽룡", "010-222-3331", "02-3422-9900", "서울시")
addContact("이몽룡", "010-222-3331", "02-3422-9900", "서울시", "hero@email.com", "감자", "고구마")
```

```
name >> 홍길동
home >> 없음
address >> 없음
email >> 없음
favoritesFoods >>
-----
name >> 이몽룡
home >> 02-3422-9900
address >> 서울시
email >> 없음
favoritesFoods >>
-----
name >> 이몽룡
home >> 02-3422-9900
address >> 서울시
email >> hero@email.com
favoritesFoods >> 감자,고구마
```

- 프로젝트 초기화
`npm init`
- 소스코드 작성
`src/test.js`
- 소스코드 실행
`node src/test.js`

Q3 - 구조 분할 할당

- `addContact3()` 함수를 정의하시오.
- 입력하지 않으면 `email`은 “이메일없음”, `age`는 0으로 설정

```
name>> 지수
phone>> 010-222-3333
email>> hera
next_age>> 101
-----
name2>> 지수
tel>> SKT 010-222-3333
contact_email>> hera@email.com
next_age>> 101
```

함수내 출력

리턴값 구조 분할 할당
후 출력

```
let {name2, tel, contact_email, next_age} = addContact3("지수", "010-222-3333", "hera", 100);
```

Q4 - 화살표 함수

- 다음을 화살표 함수로 변경하고, 각각 호출하시오.

1

```
const test1 = function(a,b) {  
  return a+b;  
}
```

2

```
function test2(params) {  
  console.log(params + "님 환영합니다.!");  
}
```

3

```
function test3() {  
  return "ok";  
}
```

300

hong님 환영합니다.!

OK

- 다음의 **this**는 함수가 포함관계에 있을 때 객체가 전달되지 않는다. 해결책 2가지를 제시하고 프로그래밍 하여 값을 비교해보시오.
- 교안ppt 참고

```
var obj = { result:0 };
obj.add = function(x,y) {
  function inner() {
    this.result = x+y;
  }
  inner();
}
obj.add(3,4)

console.log(obj)           // { result: 0 }
console.log(result)        // 7
```


Q6 - 객체 리터럴

- 쇼핑물의 물건을 객체 리터럴로 표현하시오.
- 예시)
 - 속성 : 물건이름, 원가격, 할인가격, 1개당가격, 도착여부, 평점, 적립금
 - 메서드 : 익명 함수 형태(할인을 하다), 일반 함수 형태(물건 속성값을 프린트 하다.)

```
할인을 하는 중입니다 .
name>> 에너자이저
total_price>> 5000
discount_price>> 4700
one_price>> 2350
arrive_date>> 내일(수)도착보장
rate>> 50
stack_money>> 188
```



에너자이저 2입

6% 5,000원
4,700 원
(1개입당 2,350원)

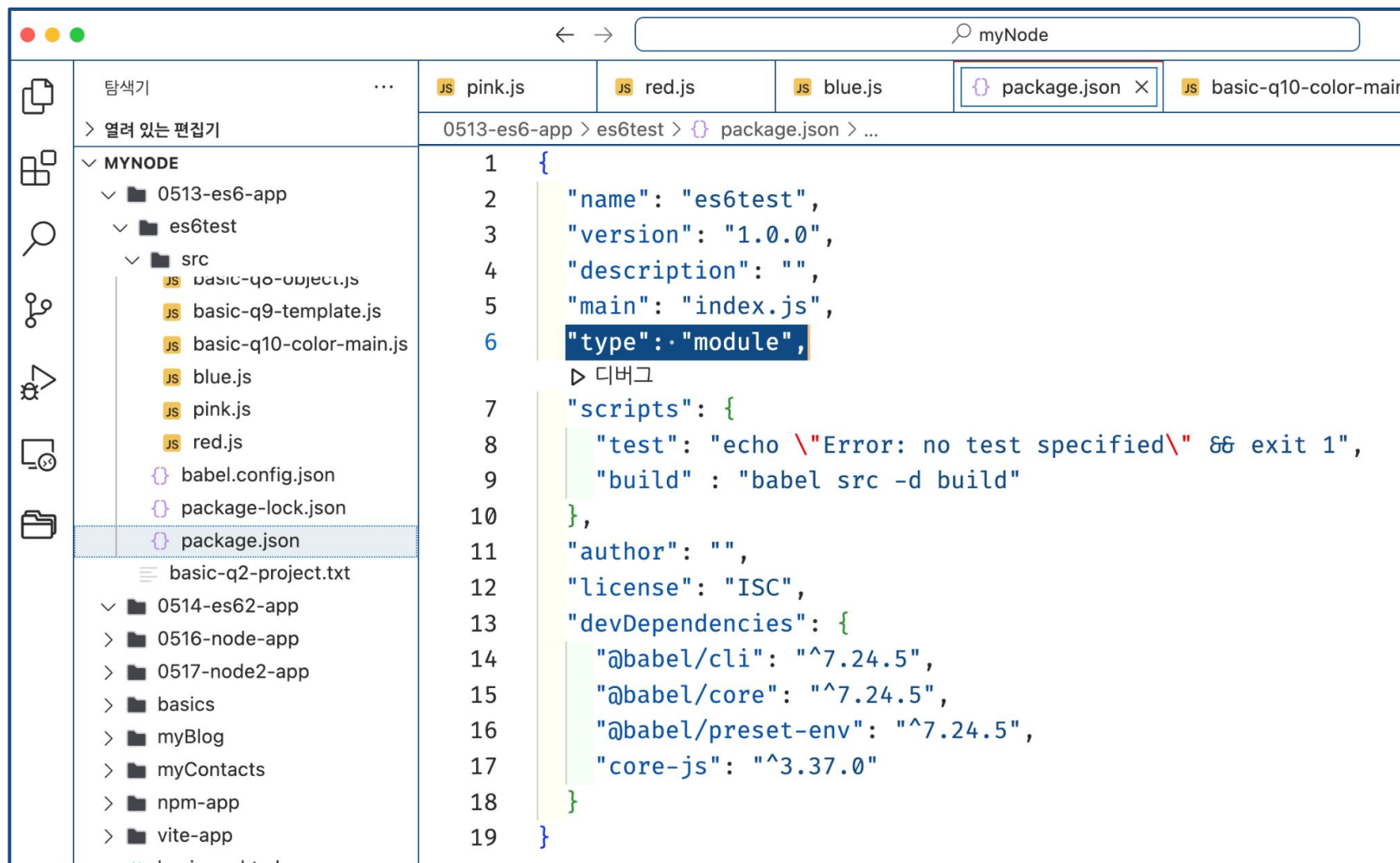
 로켓배송

내일(수) 도착 보장

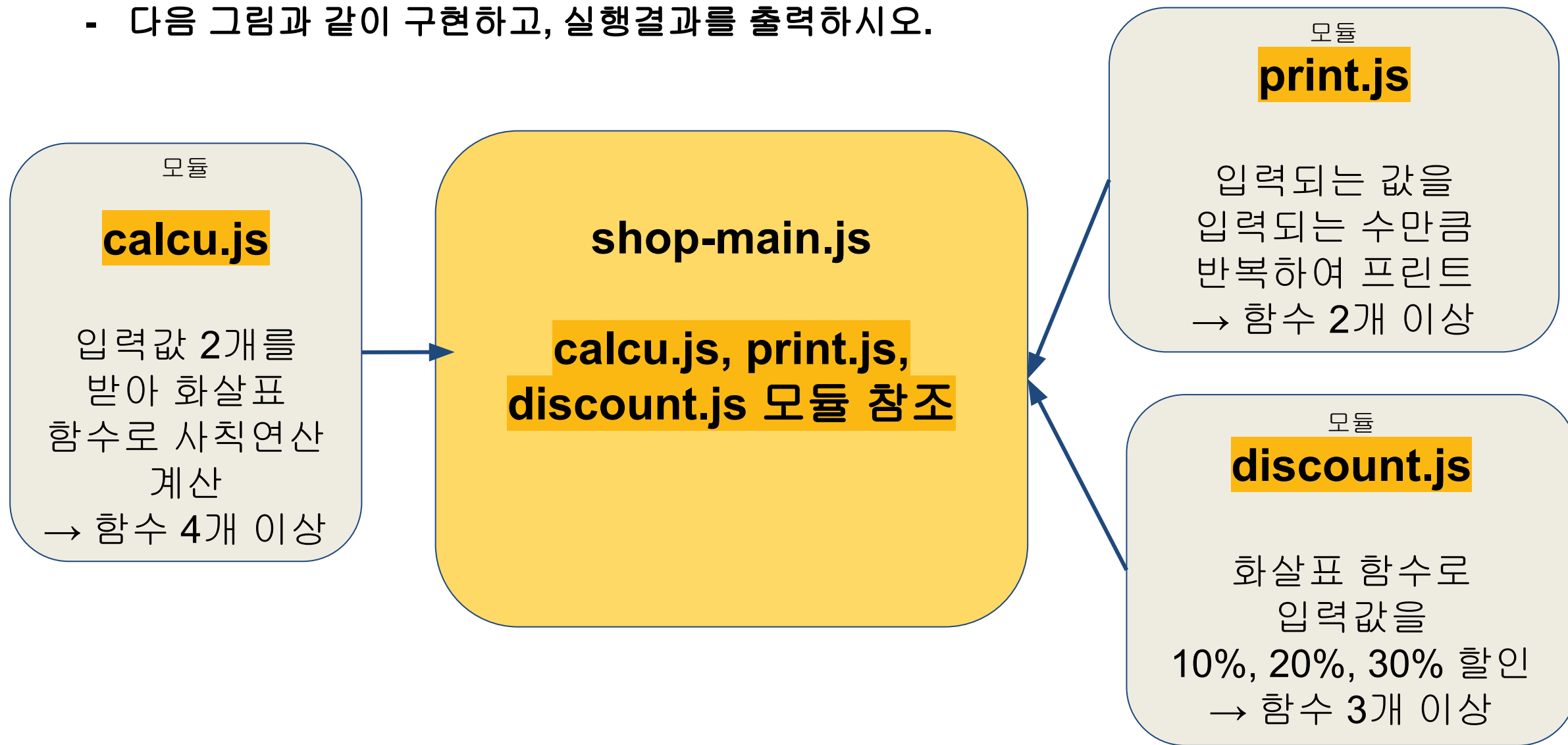
★★★★★ (50)

 최대 188원 적립

- import문으로 모듈을 사용하고자 하는 경우에는 다음과 같이 설정해준다.



- 다음 그림과 같이 구현하고, 실행결과를 출력하시오.



덧셈 : $10 + 5 = 15$

뺄셈 : $10 - 5 = 5$

곱셈 : $10 * 5 = 50$

나눗셈 : $10 / 5 = 2$

Hello, world!

Hello, world!

Hello, world!

Original Price: ₩10,000

Price after 10% discount: ₩9,000

Price after 20% discount: ₩8,000

Price after 30% discount: ₩7,000

