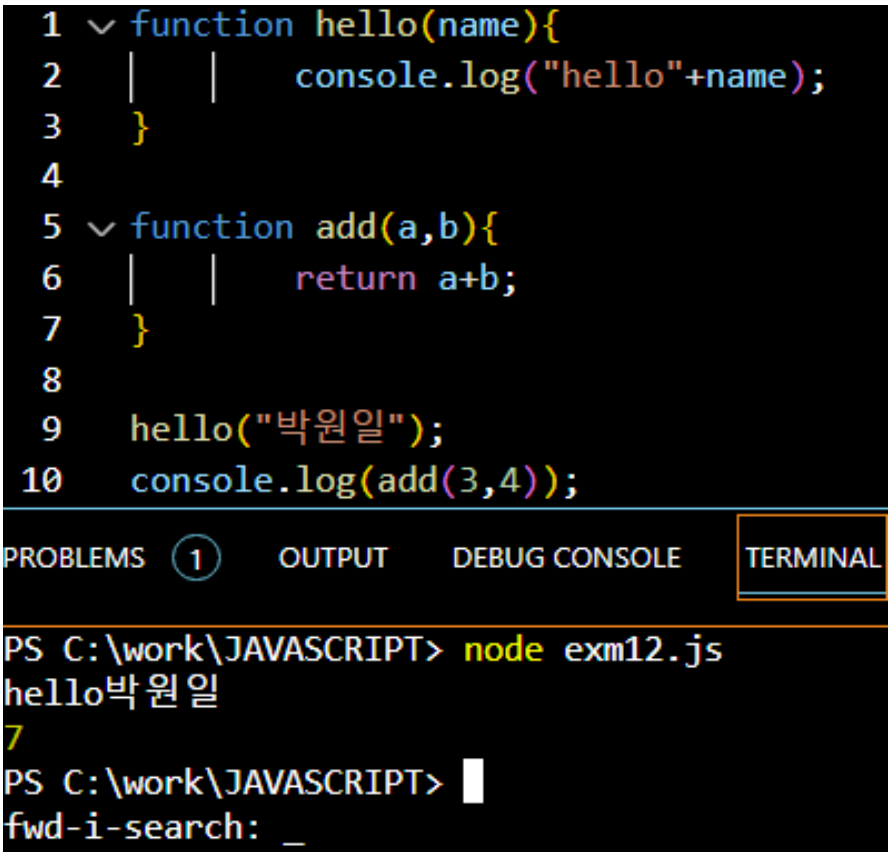
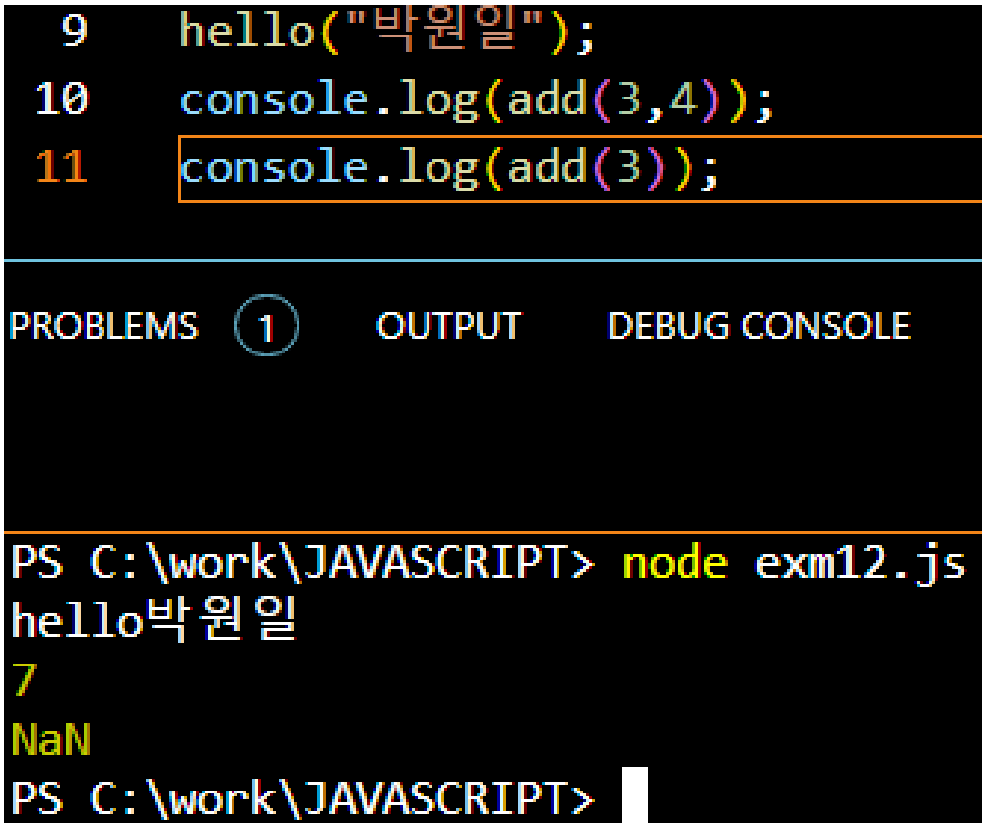
231108 26일차 수업 :: 츄르 없는 집

함수 => 미리 약속된 계산식
sqrt - 제곱근
sum - A+B
function
hello(name){
console.log("hello"+name);
}
위의 과정이 함수지정
function => 기능정의->함수지정
hello => 함수명
(name) =>파자미터(대입자)
function add(a,b){
return a+b;
}





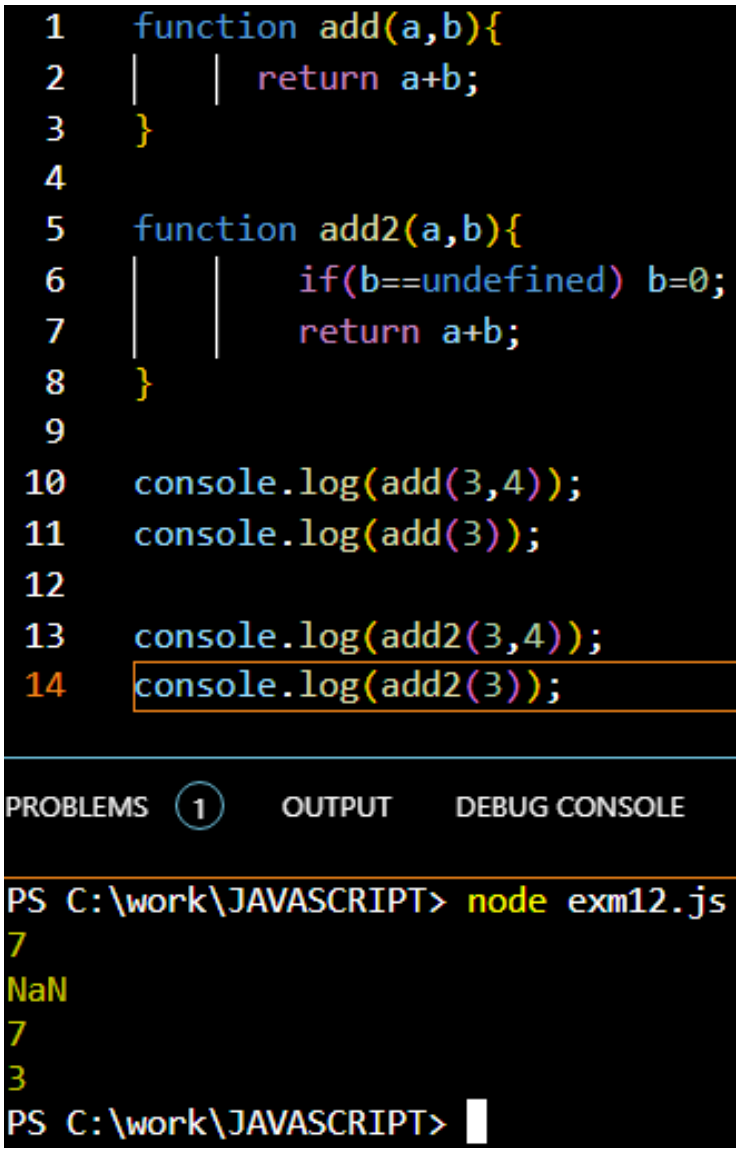
a,b가 주어져야 하는 상황에서 b의 값이 정의되지 않아서 NaN 으로 표기 됨

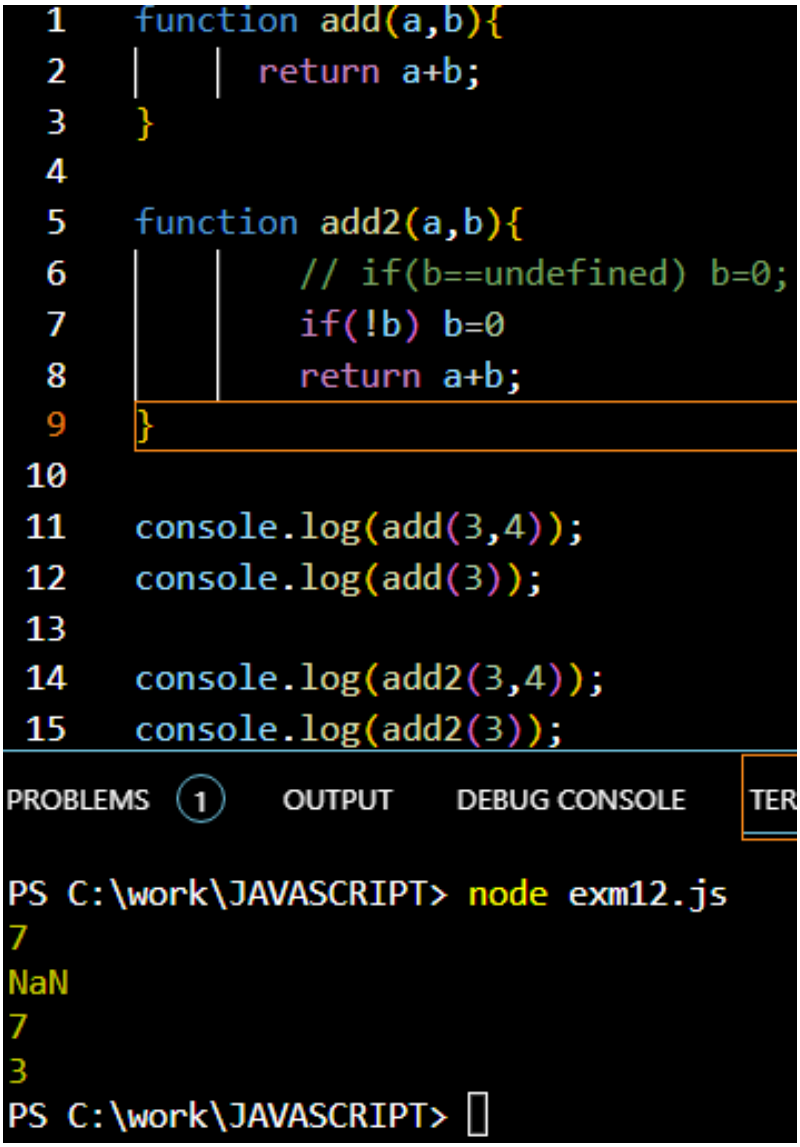
```
11    console.log(add("",4));

PROBLEMS (1)    OUTPUT    DEBUG CONSOLE

PS C:\work\JAVASCRIPT> node exm12.js
hello박원일
7
4
```

a를 Null 값을 주니 나옴

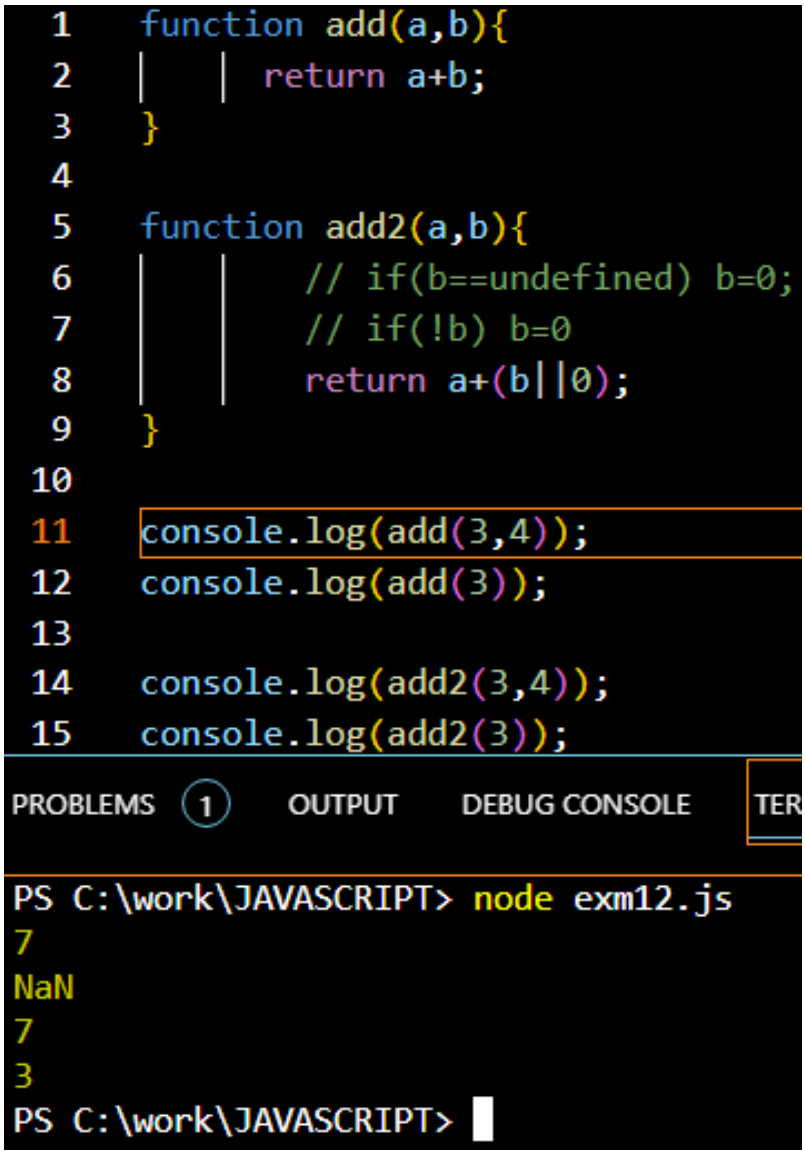
```
 9    hello("박원일");
10    console.log(add(3,4));
11    console.log(add("papa",4));

PROBLEMS (1)    OUTPUT    DEBUG CONSOLE

PS C:\work\JAVASCRIPT> node exm12.js
hello박원일
7
papa4
```

자바스크립트는 자료형 지정을 하지 않기 때문에 자료형 지정이 가능한 타입 스크립트가 추후에 나오게 된다.


1 function add(a,b){
2 return a+b;
3 }
4
5 function add2(a,b){
6 if(b==undefined) b=0;
7 return a+b;
8 }
9
10 console.log(add(3,4));
11 console.log(add(3));
12
13 console.log(add2(3,4));
14 console.log(add2(3));
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
PS C:\work\JAVASCRIPT> node exm12.js
7
NaN
7
3
PS C:\work\JAVASCRIPT>


1 function add(a,b){
2 return a+b;
3 }
4
5 function add2(a,b){
6 // if(b==undefined) b=0;
7 if(!b) b=0
8 return a+b;
9 }
10
11 console.log(add(3,4));
12 console.log(add(3));
13
14 console.log(add2(3,4));
15 console.log(add2(3));
PROBLEMS 1 OUTPUT DEBUG CONSOLE TER
PS C:\work\JAVASCRIPT> node exm12.js
7
NaN
7
3
PS C:\work\JAVASCRIPT>

!b => b의 값이 비어있으면



power 함수 작성

파라미터 a,b,c

return a*b*c

b또는c undrfined 이면 0으로 처리

```
result = result + a[i];
result =+ a[i];
```

동일함

```
29  function power(...a){
30      let result = 0;
31      for(let i=0; i<a.length; ++i)
32          result += a[i]
33      return result;
34  }
35
36  function power2(){
37      let result = 0;
38      for(let i=0; i<arguments.length; ++i)
39          result = result + arguments[i];
40      return result;
41  }
42
43  console.log(power(1, 2, 3, 4));
44  console.log(power(1, 2));
45  console.log(power());
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\work\JAVASCRIPT> node exm12.js
10
3
0
PS C:\work\JAVASCRIPT>
```

```
1   // callback
2
3   function add(a,b){
4       return a+b;
5   }
6
7   let a= add(3,4);
8   console.log(a);
9
10  let f = add;
11  console.log(typeof f);
12
13  let b = f(3,4);
14  console.log(b);
```

```
PROBLEMS 2   OUTPUT   DEBUG CONSOLE

PS C:\work\JAVASCRIPT> node exm13.js
7
function
7
```

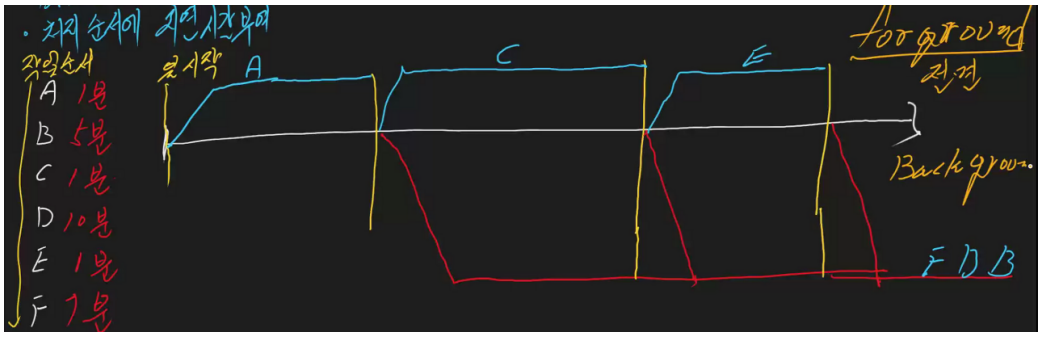
callback 함수 => 언제든 호출할 수 있다 -> 과거의 것을 호출할 수 있다

ㄴ SAS 호출에서 DAS 호출이 가능하다.

ㄴ 처리 순서에 지연시간 부여가 가능하다.

SASP: 순차접근 / 비디오 Tape

DASD : 직접접근 / CD

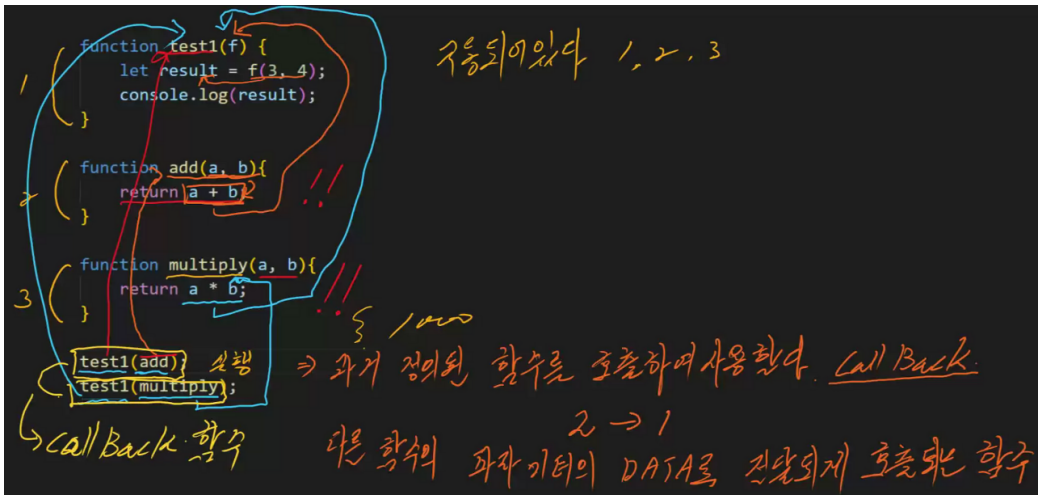


fooground 먼저 진행 시키고, background는 나중에 진행돼도 상관없을 때

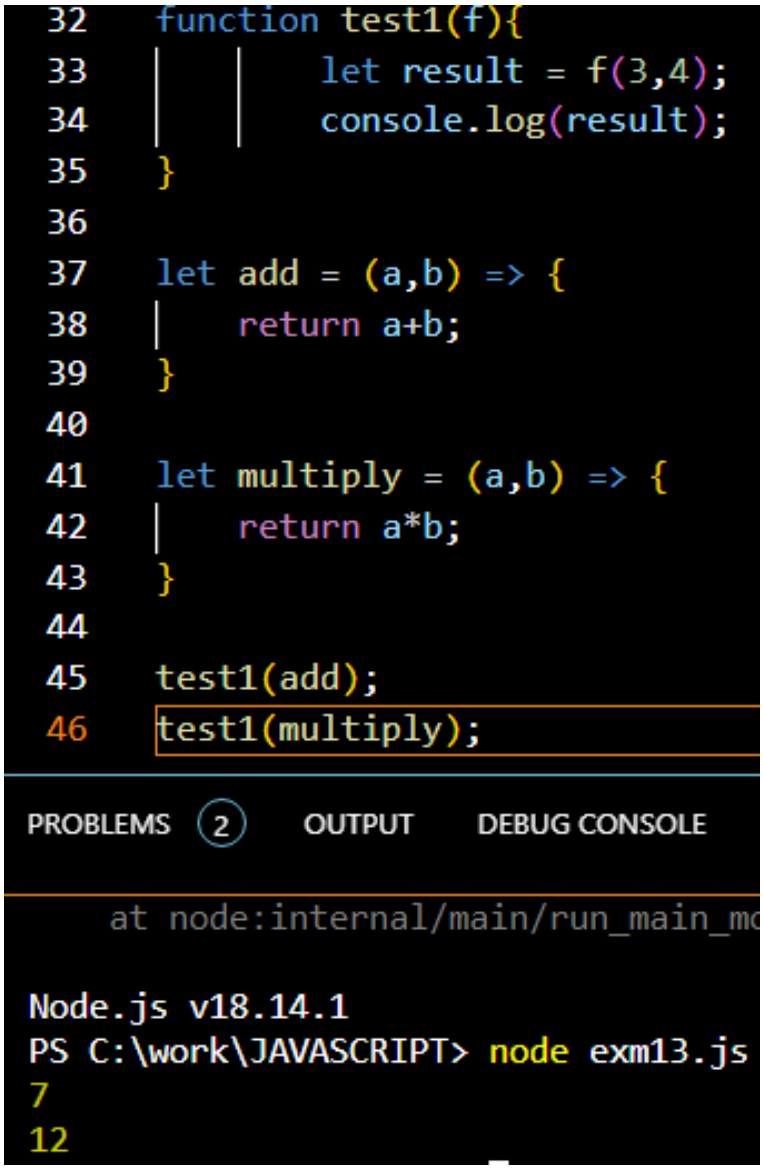
callback은 함수 자체가 자료형이기도 하고 자료이기도 하다

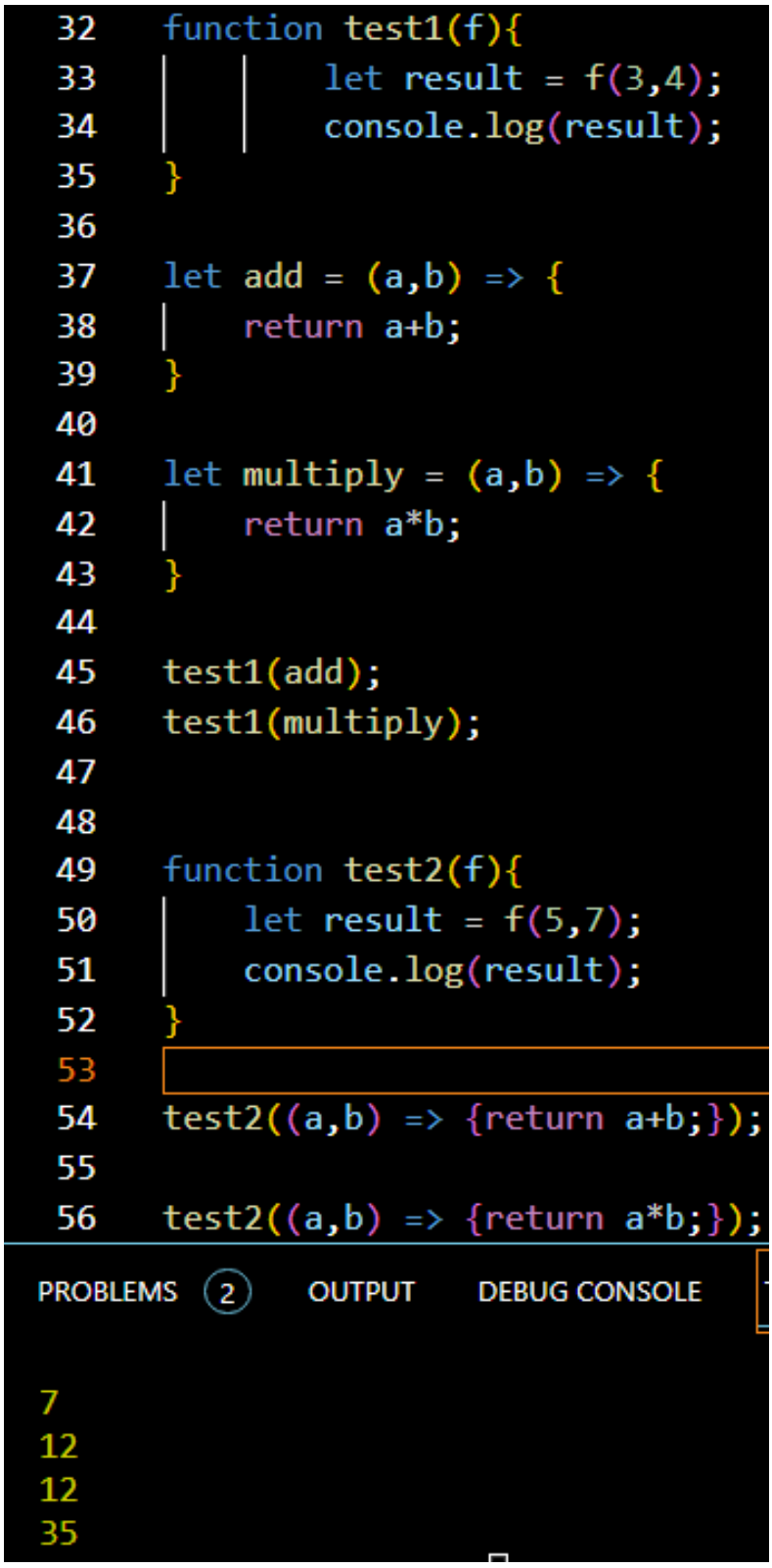
```
16  function test1(f) {
17      let result = f(3,4);
18      console.log(result);
19  }
20
21  function add(a,b){
22      return a+b ;
23  }
24
25  function multiply(a,b){
26      return a*b;
27  }
28
29  test1(add);
30  test1(multiply);
```

```
PROBLEMS 2   OUTPUT   DEBUG CONSOLE

PS C:\work\JAVASCRIPT> node exm13.js
7
12
PS C:\work\JAVASCRIPT>
```



다른 함수의 파라미터의 DATA로 전달되게 호출되는 함수


function test1(f){
let result = f(3,4);
console.log(result);
}
let add = (a,b) => {
return a+b;
}
let multiply = (a,b) => {
return a*b;
}
test1(add);
test1(multiply);
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
at node:internal/main/run_main_mo
Node.js v18.14.1
PS C:\work\JAVASCRIPT> node exm13.js
7
12


32 function test1(f){
33 let result = f(3,4);
34 console.log(result);
35 }
36
37 let add = (a,b) => {
38 return a+b;
39 }
40
41 let multiply = (a,b) => {
42 return a*b;
43 }
44
45 test1(add);
46 test1(multiply);
47
48
49 function test2(f){
50 let result = f(5,7);
51 console.log(result);
52 }
53
54 test2((a,b) => {return a+b;});
55
56 test2((a,b) => {return a*b;});
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
7
12
12
35

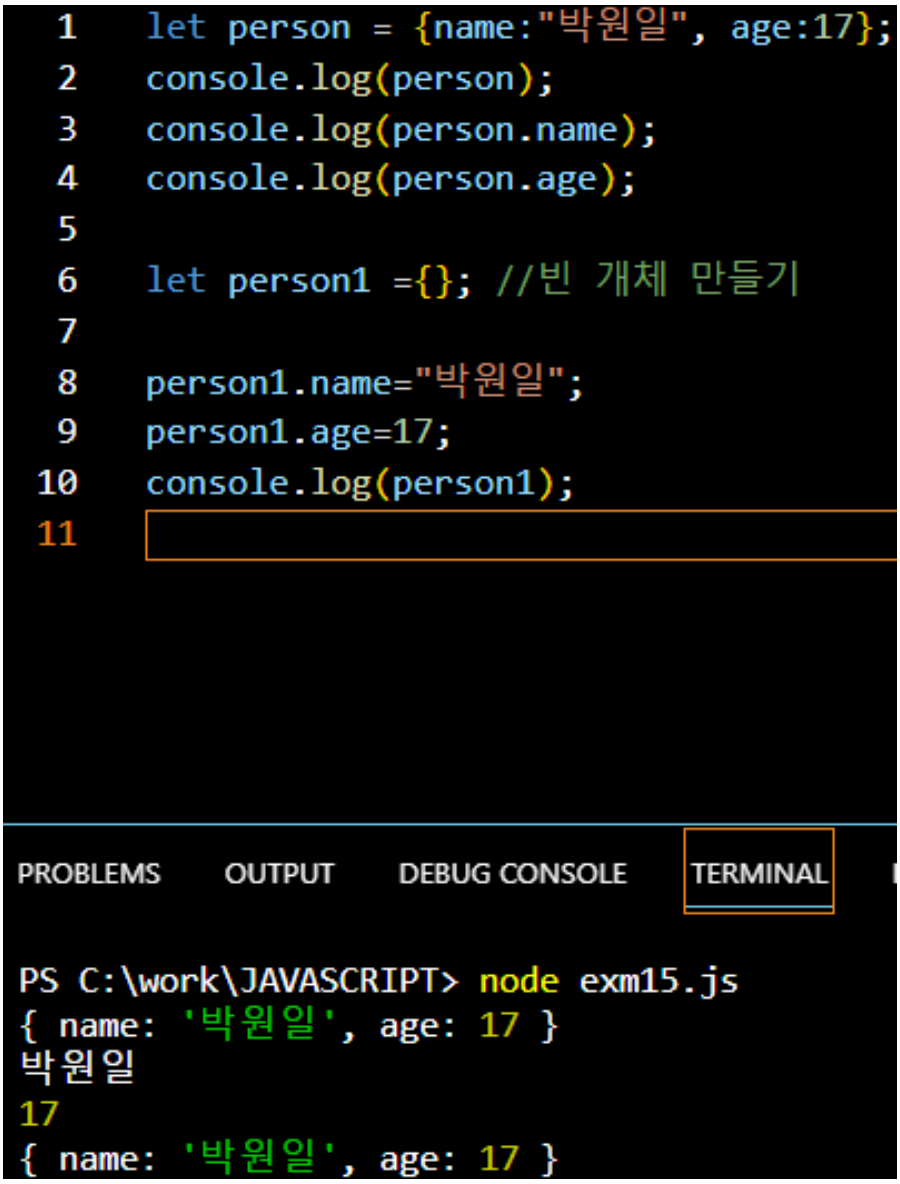
```
1   function printTime(msg){
2       console.log(msg, new Date());
3   }
4
5   setTimeout(printTime, 1000, "1초 후");
6   setTimeout(printTime, 1000, "2초 후");
7   setTimeout(printTime, 1000, "3초 후");
```

```
PROBLEMS (2)   OUTPUT   DEBUG CONSOLE   TERMINAL

PS C:\work\JAVASCRIPT> node exm14.js
1초 후 2023-11-08T05:17:06.170Z
2초 후 2023-11-08T05:17:06.186Z
3초 후 2023-11-08T05:17:06.187Z
```

```
1   let person = {name:"박원일", age:17};
2   console.log(person);
3   console.log(person.name);
4   console.log(person.age);
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

{ name: '박원일', age: 17 }
박원일
17
```

person이 오브젝트임


1 let person = {name:"박원일", age:17};
2 console.log(person);
3 console.log(person.name);
4 console.log(person.age);
5
6 let person1 ={}; //빈 개체 만들기
7
8 person1.name="박원일";
9 person1.age=17;
10 console.log(person1);
11
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PS C:\work\JAVASCRIPT> node exm15.js
{ name: '박원일', age: 17 }
박원일
17
{ name: '박원일', age: 17 }

```javascript
let person = {name:"박원일", age:17};
console.log(person);
console.log(person.name);
console.log(person.age);

let person1 ={}; //빈 개체 만들기

person1.name="박원일";
person1.age=17;
console.log(person1);

let person2 ={name:"박원일"};
person2.age=53;
console.log(person2);
```

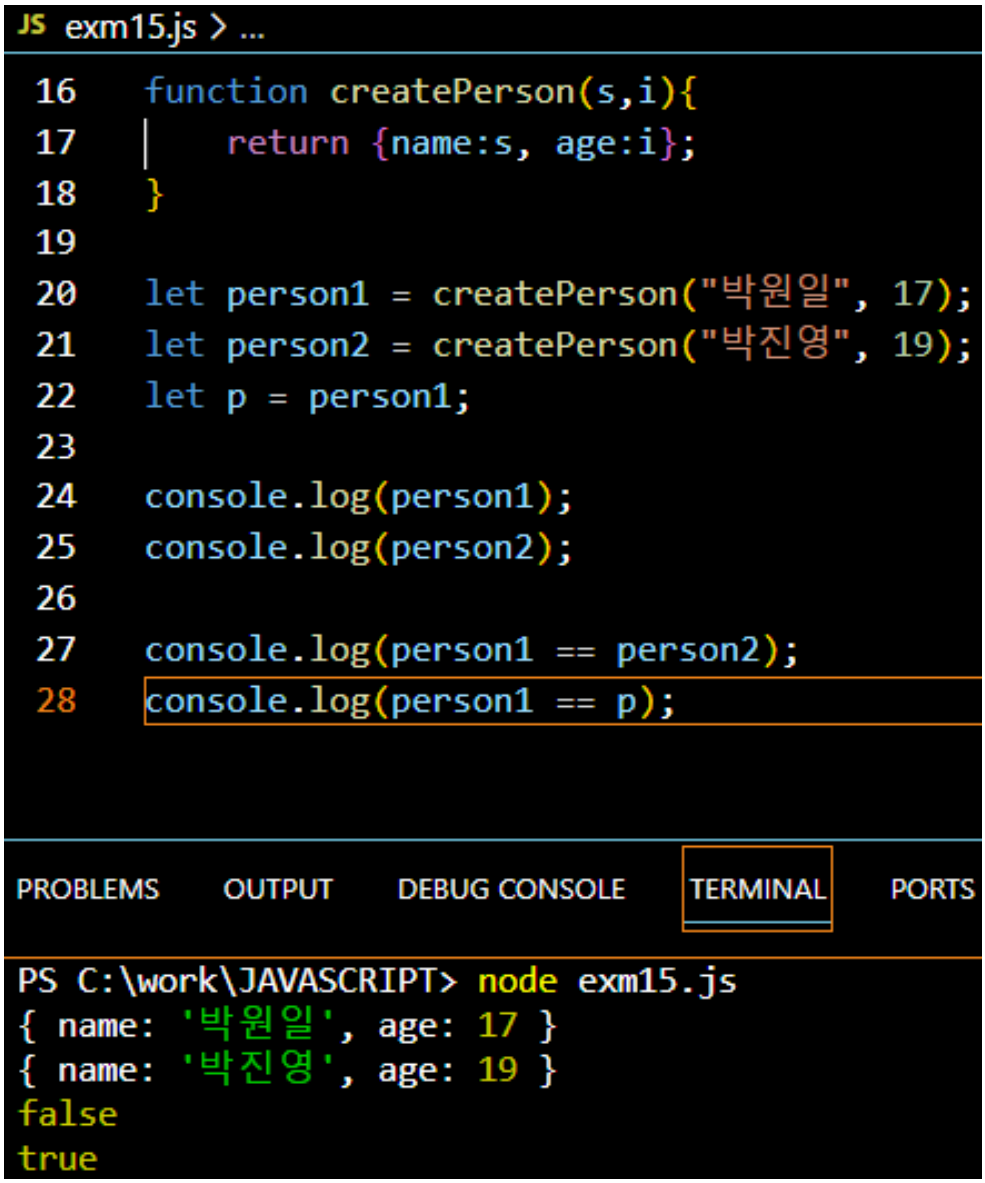
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
{ name: '박원일', age: 17 }
박원일
17
{ name: '박원일', age: 17 }
{ name: '박원일', age: 53 }
```

```javascript
function createPerson(s,i){
    return {name:s, age:i};
}

let person1 = createPerson("박원일", 17);
let person2 = createPerson("박진영", 19);

console.log(person1);
console.log(person2);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\work\JAVASCRIPT> node exm15.js
{ name: '박원일', age: 17 }
{ name: '박진영', age: 19 }
PS C:\work\JAVASCRIPT>
```

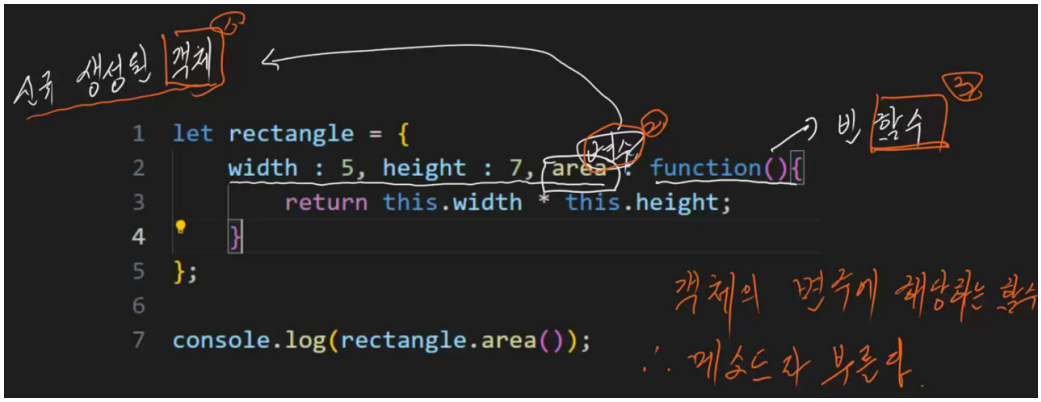


자바스크립트의 객체(오브젝트)명은 유일해야 한다. 겹치면 안 됨

```
36    let ps1 = {name:"박지영", age:48};
37    let ps2 = {name:"박지영", age:17};
38    let ps3 = {name:"박태민", age:20};
39    let ps4 = {name:"박태석", age:26};
40
41    let person =[ps1, ps2, ps3, ps4];
42    console.log(person);
43    for( let i=0; i<person.length; ++i)
44    console.log(person[i]);
45    console.log(ps1.name);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

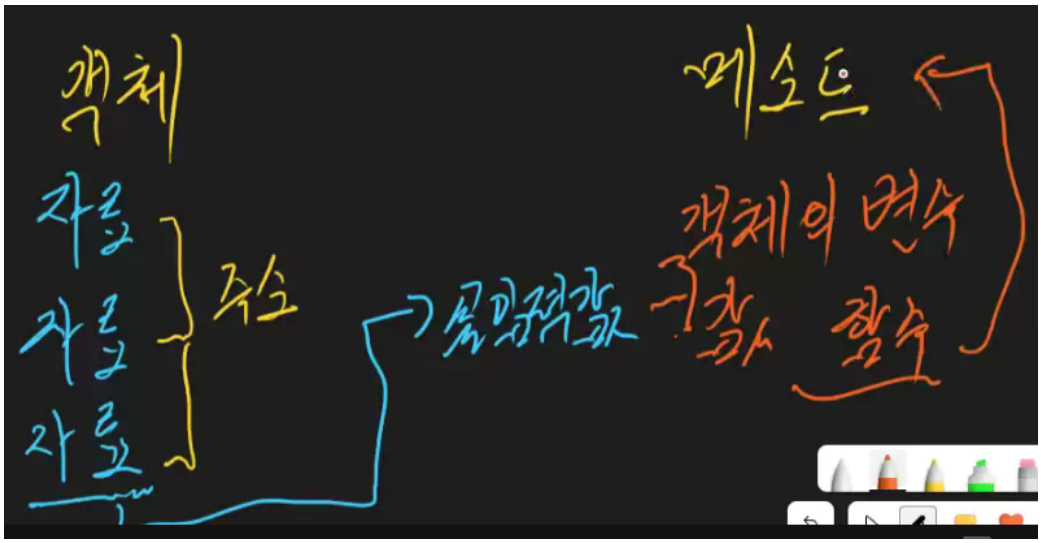
```
}
PS C:\work\JAVASCRIPT> node exm15.js
[
  { name: '박지영', age: 48 },
  { name: '박지영', age: 17 },
  { name: '박태민', age: 20 },
  { name: '박태석', age: 26 }
]
{ name: '박지영', age: 48 }
{ name: '박지영', age: 17 }
{ name: '박태민', age: 20 }
{ name: '박태석', age: 26 }
박지영
```



자바스크립트에서 메소드 생성시 변수명 앞에 this는 무조건 붙인다

객체의 변수에 해당하는 함수 -> 메소드라 부른다.

자료의 주소값을 기억하고 있는 자료의 집합 -> 객체라 부른다.



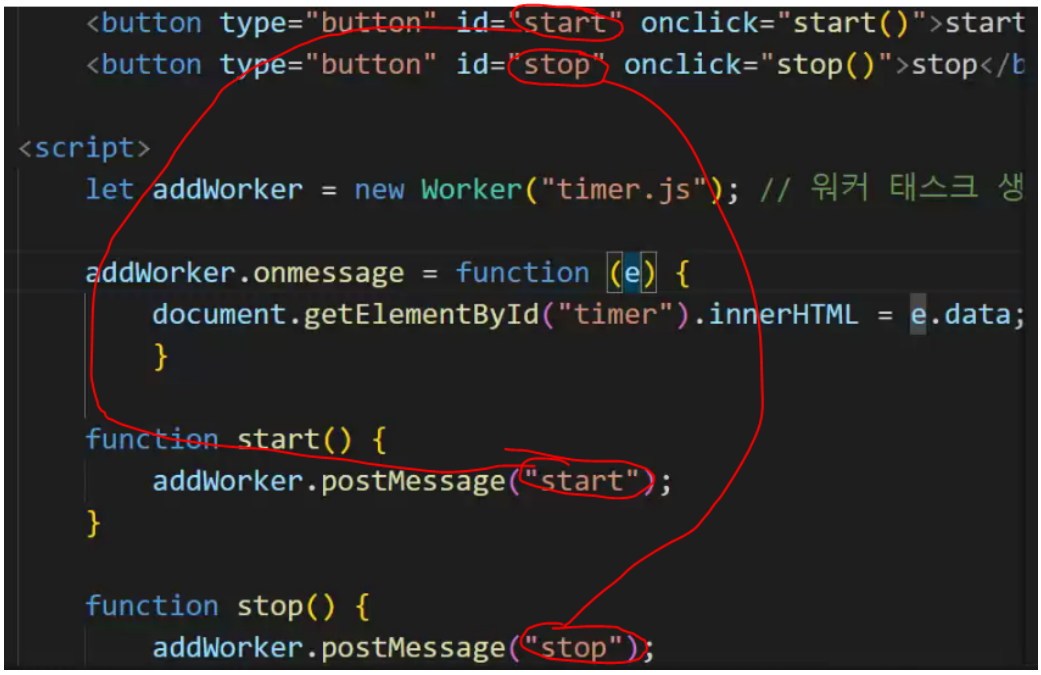
자료의 주소값을 기억하고 있는 자료의 집합을 객체라고 부르고, 자료의 실입력값을 객체의 변수값이라고 하며,

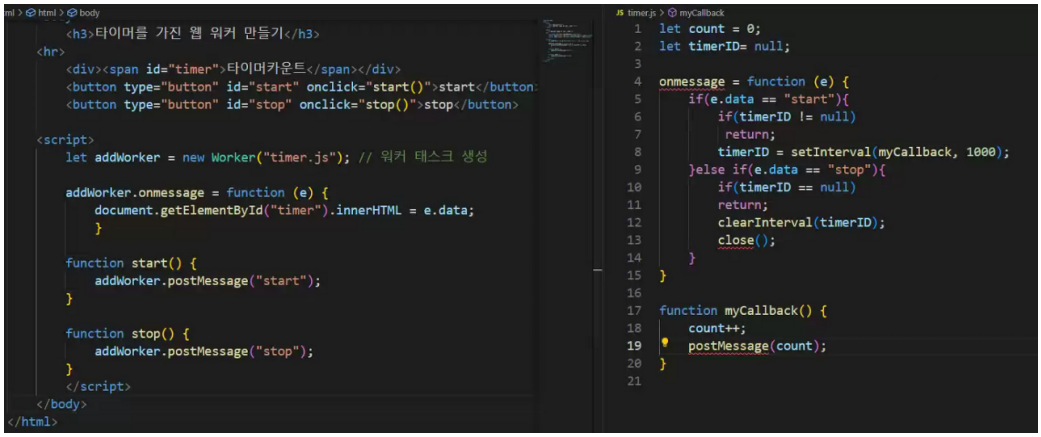
```
<body>
    <h3>타이머를 가진 웹 워커 만들기</h3>
<hr>
    <div><span id="timer">타이머카운트</span></div>
    <button type="button" id="start" onclick="start()">start</button>
    <button type="button" id="stop" onclick="stop()">stop</button>

<script>
    let addWorker = new Worker("timer.js"); // 워커 테스크 생성

    addWorker.onmessage = function (e) {
        document.getElementById("timer").innerHTML = e.data;
        }

    function start() {
        addWorker.postMessage("start");
    }

    function stop() {
        addWorker.postMessage("stop");
    }
    </script>
```

onmessage => 브라우저 상에서 무언가 문제가 생겼을 때

document => 문서

getElementById => 스타트와 스톱에 의해 html 상에서 데이터 만들어냄 / 행동 기억





좋아요공감

공유하기

통계

게시글 관리