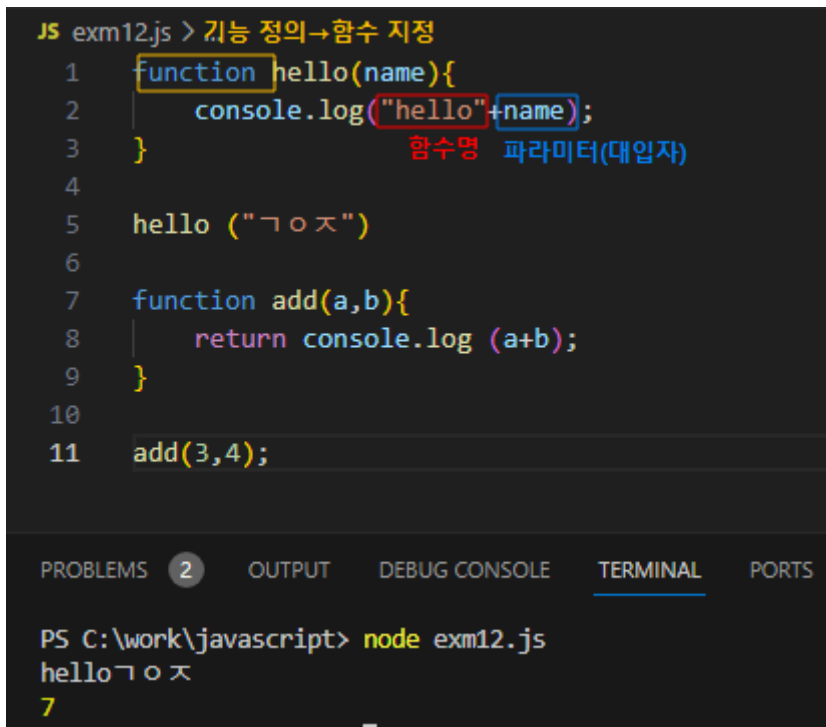


함수 지정하기



```
JS exm12.js > 기능 정의→함수 지정
1  function hello(name){
2      console.log("hello"+name);
3  }
4
5  hello("ㄱㅇㅈ")
6
7  function add(a,b){
8      return console.log(a+b);
9  }
10
11 add(3,4);
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\work\javascript> node exm12.js
helloㄱㅇㅈ
7
```

```
function hello(name){
    return console.log("hello"+name);
}
```

function 으로 함수 지정하기

function으로 add함수 지정하기

```
5 function add(a,b){
6     return a+b; // return 보관하고 있어
7 }
8 console.log(add(3,4)); // 터미널이 NaN 이라고 자료형이 없어서 계산할 수 없
9 console.log(add(3)); // 자바스크립트는 자료유형을 지정하지 않으므로,
10 //자료형 지정 가능한 '타입스크립트'가 나온다
11
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm12.js

7

NaN

```
function add(a,b){
    return a+b;
}
```

add 함수 a+b 형태로 지정하기

```
13 function add2(a,b){
14     if(b==undefined) b=0; // b가 undefined (자료값 못찾으면?) (true값)b=0으로 지정
15     if(!b) b=0; // (!b) = (b==undefined) 와 같은 뜻
16     return a+b;
17 }
18 console.log(add2(3,4));
19 console.log(add2(3));
20
21 function add3(a,b){
22     return a+(b|0);
23 }
24 console.log(add3(3,4));
25 console.log(add3(3));
26
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm12.js

7

3

7

3

if

```
33 function power(a,b,c){
34   if(b==undefined) b=0;{
35   if(!c) c=0;}
36   return a*b*c;
37 }
38 console.log(power(3,3,3))
39
40
41 function power1(a,b,c){
42   return a*(b||0)*(c||0);
43 }
44 console.log(power1(3,3,0))
45
46
47
48
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm12.js

27

0

가변 파라미터

```
49 function power(...a){ // ...a(가변 파라미터): 파라미터의 개수를 정해두지 않는다
50   let result =0;
51   for(let i=0; i<a.length; ++i) //
52     result=result+a[i]; // 같은것 result += a[i]
53   return result;
54 }
55
56 console.log (power(1,2,3,4));
57 console.log (power(1,2));
58 console.log (power());
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm12.js

10

3

0

```

59 //function power2 함수 + arguments.length 사용하여 배열로 받기
60 function power2(){ // 파라미터 괄호를 비우고 arguments.length 쓸 수 있다.
61     let result =0;
62     for(let i=0; i<arguments.length; ++i) //a.length 대신 arguments.length 가능.(배열로 받겠다
63     result=result+arguments[i]; // 같은것 result += a[i]
64     return result;
65 }
66 console.log (power2(1,2,3,4));
67 console.log (power2(1,2));
68 console.log (power2());
69 |

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm12.js

10

3

0

call back 함수

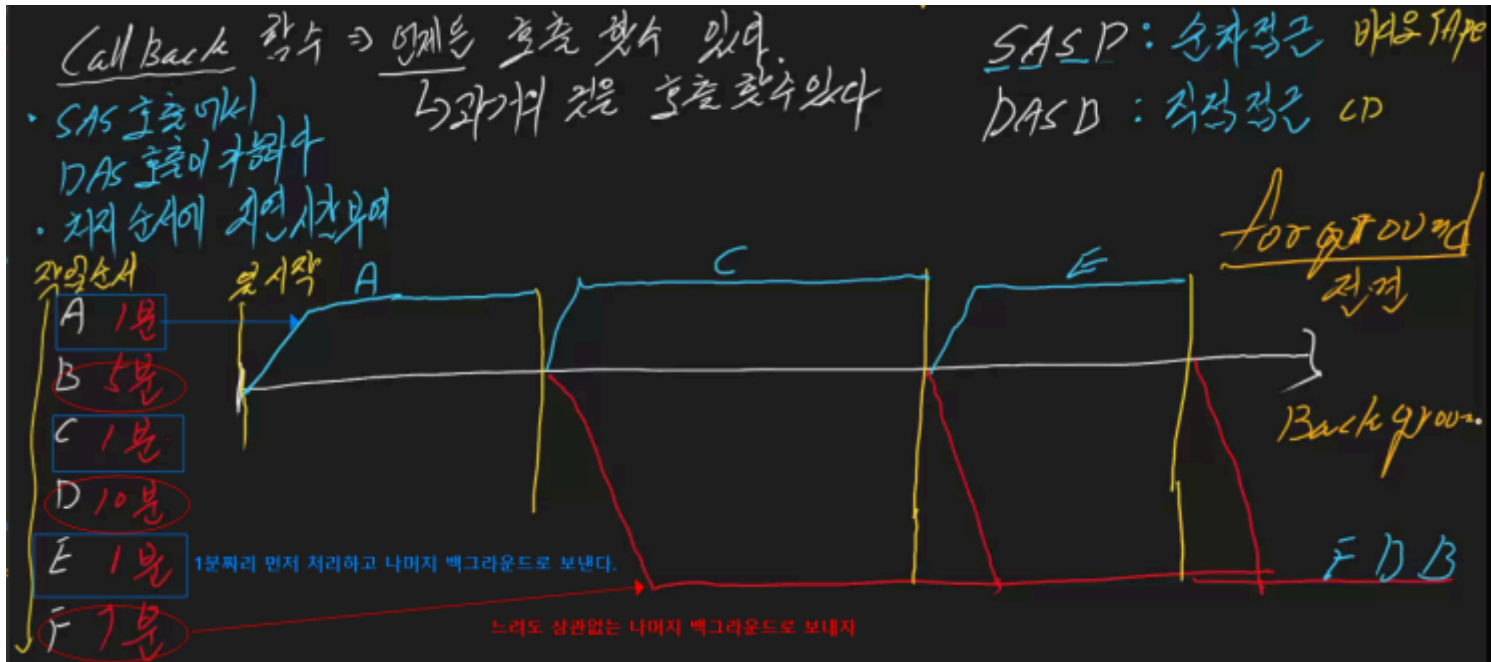
callback 함수 : 언제든지 과거의 것을 호출할 수 있다. 처리 속도가 빠른 애들 먼저 처리하는것.

SAS에서 DAS호출이 가능하다

처리 순서에 지연시간 부과

SASD: Sequential access storage device 순차 접근 저장 장치 SAS ex)비디오테이프

DASD: Direct access storaged device 직접 접근 DAS, ex) CD



멀티프로세싱

멀티프로세서 (ex. 계산기가 2개이면)

스레드 (ex. 스레드는 4개)

```

23 //예제시작
24 function add(a,b){
25   return a+b;
26 }
27 let f=add;
28
29 //위와 아래는 둘다 동일한 코드
30 let f= function(a,b){
31   return a+b;
32 }
33

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

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

```

35 //예시2.
36 function test1(f) {
37     let result = f(3,4);
38     console.log(result);
39 }
40
41
42 function add(a,b){
43     return a+b;
44 }
45
46 function multiply(a,b) {
47     return a*b;
48 }
49
50 test1(add);
51 test1(multiply);

```

어떻게 연결?
test1 (result = f (3,4))

어떻게 연결?
test1 (add (a,b)) { return a+b }

test1 (multiply (a,b)){ return a*b }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm13.js

7

12

```

55 function test1(f){
56     let result=f(3,4);
57     console.log(result);
58 }
59
60 let add=(a,b) =>{
61     return a + b;
62 }
63 let multiply=(a,b) =>{
64     return a * b;
65 }
66
67 test1 (add);
68 test1 (multiply);
69
70 //
71
72 function test2(f){
73     let result=f(5,7);
74     console.log(result);
75 }
76 test2((a,b) =>{return a + b;});
77 test2((a,b) =>{ return a * b;});
78
79
80
81

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\work\javascript> node exm13.js
7
12
12
35

```

객체 만들기

JS exm15.js > ...

```
1 // object 객체 지정, 필드는 name, 결과값은 ㄱㅇㅈ
2 //객체 만드는법 1.
3 let person ={name:"ㄱㅇㅈ", age:17}; // person 이 object 객체이다
4 console.log(person);
5 console.log(person.name);
6 console.log(person.age);
7
8 //객체 만드는법 2.
9 let person1={}; // 빈 object 만들어 두고, 결과값 지정해도 된다.
10 person1.name="김은진";
11 person1.age=19;
12 console.log(person1);
13
14 //객체 만드는법 3.
15 let person2={name:"김ㅇㅈ"};
16 person2.age=33;
17 console.log(person2);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm15.js

```
{ name: 'ㄱㅇㅈ', age: 17 }
```

```
ㄱㅇㅈ
```

```
17
```

```
{ name: '김은진', age: 19 }
```

```
{ name: '김ㅇㅈ', age: 33 }
```

```
19 // 객체 만드는 법 4. (콜백함수)
20 function createPerson(s,i){
21     return{name:s, age:i};
22 }
23 let person1 = createPerson("ㄱㅇㅈ",17);
24 let person2 = createPerson("ㄱㅂㅈ",19);
25
26 console.log(person1)
27 console.log(person2)
28
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm15.js

```
{ name: 'ㄱㅇㅈ', age: 17 }
```

```
{ name: 'ㄱㅂㅈ', age: 19 }
```


JS exm15.js > ...

```
19 // 객체 만드는 법 4. (콜백함수)
20 function createPerson(s,i){
21     return{name:s, age:i};
22 }
23 let person1 = createPerson("ㄱㅇㅈ",17);
24 let person2 = createPerson("ㄱㅂㅈ",17);
25 let p=person1; // p 하나 추가해보자
26
27 console.log(person1);
28 console.log(person2);
29
30 console.log(person1 == person2); // person1==person2로 비교해 보자. 달라서 false
31 console.log(person1 == p); // p==person1 비교해보니 같아서 true
32
33 function equals(person1, person2){
34     return person1.name == person2.name || person1.age == person2.age ;
35 }
36
37 console.log(equals); //자료값 없어서 function equals라고 나옴
38 console.log(equals(person1, person2)); // 비교하니 다른사람이라서 false나옴다
39 // 자바스크립트 object 객체 이름은 무조건 유일해야 함
40
41
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm15.js

```
{ name: 'ㄱㅇㅈ', age: 17 }
```

```
{ name: 'ㄱㅂㅈ', age: 17 }
```

```
false
```

```
true
```

```
[Function: equals]
```

```
true
```

```
43 let ps1 = {name:"박지영", age:48};
44 let ps2 = {name:"박지영", age:17};
45 let ps3 = {name:"박태민", age:20};
46 let ps4 = {name:"박태민", age:26};
47
48 let person=[ps1, ps2, ps3, ps4];
49 console.log(person);
50
51 for(let i=0; i<person.length; ++i)
52 console.log(person[i]);
53
```

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: '박태민', age: 26 }
]
{ name: '박지영', age: 48 }
{ name: '박지영', age: 17 }
{ name: '박태민', age: 20 }
{ name: '박태민', age: 26 }
```

배열

하나하나의 결과값

```
43 let ps1 = {name:"박지영", age:48};
44 let ps2 = {name:"박지영", age:17};
45 let ps3 = {name:"박태민", age:20};
46 let ps4 = {name:"박태민", age:26};
47
48 let person=[ps1, ps2, ps3, ps4];
49 console.log(person);
50
51 for(let i=0; i<person.length; ++i)
52 console.log(person[i]);
53 console.log(ps1.name);
54
55
56
```

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: '박태민', age: 26 }
]
{ name: '박지영', age: 48 }
{ name: '박지영', age: 17 }
{ name: '박태민', age: 20 }
{ name: '박태민', age: 26 }
박지영
```

주소

	name	age
0	박지영	48
1	박지영	17
2	박태민	20
3	박태민	26

JS exm16.js > [?] rectangle > [?] area

```
1 let rectangle = { // 객체 만들겠다 / let rectangle = { }
2   width: 5, height: 7, area: function() { //객체의 변수는 width 와 height으로 area에 넣을것임.
3   // (함수명 area도 괄호 바깥. 변수도 바깥)
4   return this.width * this.height; // 객체의 변수에 해당하는 함수는 '메소드'라고 함
5   }
6 }
7 console.log(rectangle.area());
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm16.js
35

JS exm16.js > ... 객체 만들기

```
1 let rectangle = { // 객체 만들겠다 / let rectangle = { }
2   width: 5, height: 7, area: function() { //객체의 변수는 width 와 height으로 area에 넣을것임.
3   // (함수명 area도 괄호 바깥. 변수도 바깥)
4   return this.width * this.height; // 객체의 변수에 해당하는 함수는 '메소드'라고 함
5   // 자바스크립트에서 메소드 생성시, 변수이름 앞에 this 는 무조건 붙
6   }
7 }
8 console.log(rectangle.area());
9 // 객체란? 자료의 주소값을 기억하고있는 자료의 집합이다. 주소로 기억
10 // 메소드 자료의 실입력값을 객체의 변수값이라고 하는데 이것 취급하는 것.
11
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\work\javascript> node exm16.js
35

