

231106 24일차 수업 :: 츠르 없는 집



```
20 let sum=0;
21 for( let i=0; i<=100; ++i){
22     if (i%3==0){
23         sum=sum+i;}
24 }
25 console.log(sum);
26
```

PROBLEMS **TERMINAL** ... powershell

PS C:\work\JAVASCRIPT> node exm04.js
1683

1~100 사이의 정수 중, 3의 배수의 합

"a*b" => string 문자

a*b => NUM 숫자

+ => 문자와 문자 결합, 문자와 숫자의 결합

```
33 let sum=0;
34 for(let a=1; a<=5; a++){
35     for(let b=1; b<4; b++){
36         sum=sum+ a*b}
37 }
38 console.log(sum);
```

PROBLEMS **TERMINAL** ... powershell

PS C:\work\JAVASCRIPT> node exm04.js
90

a*b의 총합을 구하기.

for => 이중 반복문

1. for(1~5) => Row 행

2. for(1~5) => Col 열

1	2	3	4	5
2				
3				
4				
5				

array = 객체(object)

결합체

[]의 주소는 0부터 시작

address => [0,1,2,3] / 갯수 4

let a = [3]; <- 배열 정의 / a 배열에 3이라는 자료값 넣음. 그래서 1개임

console.log(a.length);

= 결과값 1

a[5] = 456; => a배열 5번째 address에 456 집어넣기

└ (a 배열의 다섯번째 값)

= 결과값 6

for(i=0; i<a.length; i++){

 console.log("a[%d] : %s", i, a[i]);

}

a배열에 %d의 자료값을 i의 값으로 출력해라

a[3]=12;

for(i=0; i<a.length; i++){

 console.log("a[%d] : %s", i, a[i]);

}

=

a[0] : 3

a[1] : undefined

a[2] : undefined

a[3] : 12

a[4] : undefined

a[5] : 456

let a=[3, "", "", 12, "", 456]

을 주면 null 값 갖게 됨.

```
let a=[]
```

console.log(a.length); => address 지정한 게 없다 / 빈배열 만들어둔 것
결과값 = 0

```
let a = [3,4,5]; / a에 3,4,5를 넣기
```

```
let [a1, a2, a3]=a; / a1, a2, a3에 a를 넣기
```

```
let a1 = 3
```

```
let a2 = 4
```

```
let a3 = 5
```

```
console.log("%d %d %d", a1, a2, a3);
```

```
let a,b ;
```

```
a= 5;
```

```
b= 6;
```

```
console.log(a,b);
```

└ a는 5, b는 6이라고 선언 됨

```
let temp = a;
```

└ temp는 a라고 선언 됐으니 temp는 5

```
a= b;
```

└ a는 b라고 다시 선언 됐으니 a는 6

```
b= temp;
```

└ b는 temp라고 다시 선언 됐으니 b는 5

```
console.log(a,b);
```

└ a는 6, b는 5라고 선언 됨

```
let a=[0, 1, 2, 3];  
a.splice(1,0,"a");  
console.log(a);  
console.log(a.toString());
```

PS C:\work\javascript> node exm07

[0, 'a', 1, 2, 3]

0,a,1,2,3

```
let a=[0,1,2,3];  
a.splice(1,0,"a")  
// a의 1번 주소값을 0개 지우고 a를 삽입하라.  
console.log(a);  
// [0,a,1,2,3]  
console.log(a.toString());  
// 0,a,1,2,3
```

```
let a=[0,1,2,3];
a.splice(1,1,"a")
console.log(a);
// [0,a,2,3]
console.log(a.toString());
// 0,a,2,3
```

a의 1번 주소값을 1개 지우고 a 삽입

```
86 let a=[0,1,2,3];
87 a.splice(1,0,"a")
88 console.log(a);
89
90 let b=[0,1,2,3];
91 console.log(b.slice(0,1));
92 console.log(b.slice(0,2));
93 console.log(b.slice(1,2));
94 console.log(b.slice(1,3));
```

PROBLEMS **TERMINAL** ... powershell

```
[ 0, 'a', 1, 2, 3 ]
[ 0 ]
[ 0, 1 ]
[ 1 ]
[ 1, 2 ]
```

slice -> 반복

slice(x,y)

x -> 시작값 / 배열 add주소값

y -> 종료값 / 단, 위치를 셀 때는 처음 값부터 시작함.

let b=a.slice(0);

a를 b에 복제하라

인덱스 0번부터 종료점을 지정하지 않았으므로 전체에 해당한다.

```
96 let a=[];
97 for(let i=0; i<10; i++){
98     a.push(i);
99 }
100 }
101 
102 console.log(a);
103 console.log(a.toString())
```

PROBLEMS **TERMINAL** ... powershell

```
[
  0, 1, 2, 3, 4,
  5, 6, 7, 8, 9
]
0,1,2,3,4,5,6,7,8,9
```

push => 빈 배열에 항목 추가

```
96 let a=[10];
97 for(let i=0; i<10; i++){
98     a.push(i);
99 }
100 }
101 
102 console.log(a);
103 console.log(a.toString())
```

PROBLEMS **TERMINAL** ... power

```
10, 0, 1, 2, 3,
  4, 5, 6, 7, 8,
  9
]
10,0,1,2,3,4,5,6,7,8,9
```

배열의 맨 뒤에서부터 추가

```

96   let a=[10];
97   for(let i=0; i<10; i++)
98     // 증가식
99     { a.push(i); }
100
101   console.log(a);
102   console.log(a.toString());
103
104   for (let j=a.length; j>0; j--)
105     // 감소식
106     { a.pop(j); }
107   console.log("length %d array %s", i, a)}

```

PROBLEMS

TERMINAL

...



powershell



```

]
length 9 array [
  10, 0, 1, 2,
  3, 4, 5, 6
]
length 8 array [
  10, 0, 1, 2,
  3, 4, 5
]
length 7 array [ 10, 0, 1, 2, 3, 4 ]
length 6 array [ 10, 0, 1, 2, 3 ]
length 5 array [ 10, 0, 1, 2 ]
length 4 array [ 10, 0, 1 ]
length 3 array [ 10, 0 ]
length 2 array [ 10 ]
length 1 array []

```

%d => 해당하는 수치의 값

```

109 let a=[1,2,3];
110 let b=[4,5,6];
111 let c= a+b;
112 let d=a.concat(b);
113 // 기존의 배열은 건드리지 않고 하나의 배열로 만들
114 console.log(a);
115 console.log(b);
116 console.log(c);
117 console.log(d);

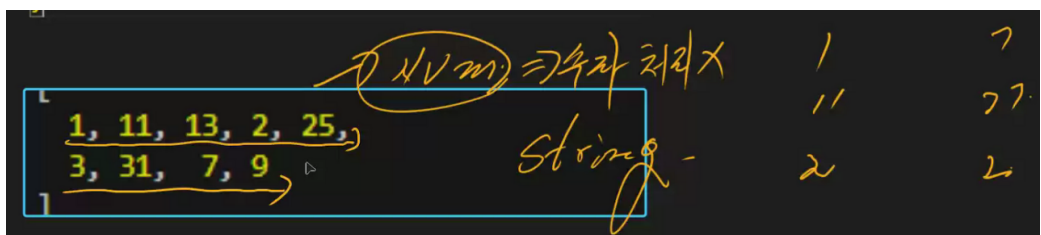
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** ... po

```

[ 1, 2, 3 ]
[ 4, 5, 6 ]
1,2,34,5,6
[ 1, 2, 3, 4, 5, 6 ]

```



자릿수가 다르기 때문에 숫자처리가 안 되어 오름차순 정렬이 되지 않은 모습

순서		순서
1		001
11		003
111		011
3		031
31		044
44		066
451		111
66		451+
779		779

문자랑 숫자가 결합되어 있는 구조