

23-11-07

- 다차원 배열 (배열안에 배열이 들어가 있다)

<다차원배열>

배열안에 배열은 => 2차원 배열이다(표의 형태이다)

```
let a=[[1,2,3,4],[5,6,7,8],[9,10,11,12]];
```

배열안에 배열이 들어가 있는 구조 -> [[]]

[] -> 오브젝트 덩어리로 a에는 덩어리가 3개 있다는 것

```
console.log(a.length)
```

결과값 : 3 -> 이렇게 하면 결과는 3이 나온다.

```
for (let i=0; i<a.length; i++){  
  for(let j =0; j<a[i].length; j++){  
    console.log(a[i][j]); } }  
}
```

배열의 개수는 3개이고 배열안의 배열의 개수는 4개이다.

두번째 for문에 a[i].length라고 한 것은 배열의 개수가 아니라 배열 안에 배열의 개수 4개가 들어  
가야 한다.

Handwritten notes and code explaining 2D arrays. The code defines a 2D array 'a' with 3 rows and 4 columns. Handwritten notes explain that 'a' is an array of 3 elements, each being an array of 4 elements. It also shows how to access elements like a[3][4] and how to iterate through them with nested loops.

a	1	2	3	4
b	5	6	7	8
c	9	10	11	12

a[3][4] -> 3행4열인 2차원 배열이다.

A의 배열안에 a,b,c의 배열이 있다고 생각하면 a=[a,b,c]의 형태로 간단하게 도식화 할 수 있다.

```
let a=[[1, 2, 3, 4],[5, 6, 7, 8, 13, 14],[9, 10, 11, 12]];
```

원래 만들었던 것에서 두번째 배열에 13,14를 추가해도 for문을 돌리면 값이나옴을볼수있음

➔ 정형화된 배열 뿐만 아니라 비정형화 된 배열도 만들 수 있다.

```
let b = [[1,1,3,4,5,6],[2,3],[4,5,15],[5,7,21,27,35,43],[7,9,27,34]];
for (let i=0;i<b.length;i++){
  for (let j=0;j<b[i].length;j++){
    console.log(b[j]);
  }
} console.log("")
```

console.log(b[j]); 를 했을 때

j는 b의 배열의 열의 기준으로 돌아간다

j가 6까지 돌아가지만 b[j]로 설정해줬기 때문에

행으로 출력되고 행의 개수는 5개만있으니까

6번째 값은 undefined로 나오게 되는 것.

그 다음 두번째는 값이 2개니까 두번돌아가는 것

그렇게 코드가 돌아가는 것

그렇게 6개 > 2개 > 3개 > 6개 > 4개 순으로

돌아감

```
PS C:\work\javascript\11-07> node exm08.js
[ 1, 1, 3, 4, 5, 6 ]
[ 2, 3 ]
[ 4, 5, 15 ]
[ 5, 7, 21, 27, 35, 43 ]
[ 7, 9, 27, 34 ]
undefined

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

[ 1, 1, 3, 4, 5, 6 ]
[ 2, 3 ]
[ 4, 5, 15 ]

[ 1, 1, 3, 4, 5, 6 ]
[ 2, 3 ]
[ 4, 5, 15 ]
[ 5, 7, 21, 27, 35, 43 ]
[ 7, 9, 27, 34 ]
undefined

[ 1, 1, 3, 4, 5, 6 ]
[ 2, 3 ]
[ 4, 5, 15 ]
[ 5, 7, 21, 27, 35, 43 ]
```

컴퓨터의 모든 순서는 행 > 열 순서대로 읽는다.

(x,y)의 기준이다.

- 3차원 배열

DB 생성할 때 필요한 기본적 이론이다.

배열을 하나 더 씌운 것.

```
PS C:\work\javascript\11-07> node exm08.js
1
2
3
4
5
6      [
7        [ [ 1, 2 ], [ 3, 4 ] ],
8          [ [ 5, 6 ], [ 7, 8 ] ],
9          [ [ 9, 10 ], [ 11, 12 ] ]
10       ]
11
12
```

//3 차원 배열

```
let c = [[[1,2],[3,4]], [[5,6],[7,8]], [[9,10],[11,12]]];
for (let i=0;i<c.length;i++){
  for (let j=0;j<c[i].length;j++){
    for (let k=0;k<c[i][j].length;k++){
      console.log(c[i][j][k]);
    } } } console.log(c);
```

<문자열>

//문자열

```
let s = "hello world";
for(let i =0;i<s.length; i++){ // 길이 수는 빈칸도 포함한다
  console.log(s[i]); //s[i] 배열형태로 만든것
}
```

Length는 길이구나! 길이 값을 정할 때 빈칸도 포함한다

s[i] 출력하는 건 배열형태로 만든 것

```
PS C:\work\javascript\11-07> node exm10.js
h
e
l
l
o

w
o
r
l
d
```

```
console.log(s[0]); //위에 h 가 출력됨
```

```
s[0] = "H";
```

```
console.log(s[0]); // 값을 넣어준 H가 출력될 것 같은데, h 가 출력됨
```

여기서 s[0]를 출력하면 h가 출력한다

s[0]에 대문자"H"를 넣어줬는데 똑같이 s[0]출력했을 때 H가 나올 것 같은데 h가 출력됨

-> 대소문자 구분하지 않고 출력되기 때문이다.

배열의 주소address가 0부터 시작한다

문자열도 마찬가지이다.

```
let s="one two one two";
console.log(s.indexOf("two")); //two 시작하는 곳이 4 이다
console.log(s.indexOf("ne")); //ne 가 시작하는 곳 1 이다
console.log(s.indexOf("one")); //0 부터 시작함
console.log(s.indexOf("two",6)); //6 은 시작위치 -> 6 번 이후로 two 가 시작하는 값 찾아라
console.log(s.indexOf("Two")); //값 -1 나온다 값이 없다는 것. T 가 없으니까.
console.log(s.indexOf("Two",6));
//예를들어 4 번째 자리에 T 가 있다 그래도 이것은 6 부터 시작하니까 값은 -1 이 나온다
```

S 를 문자열리주고 s 의 시작하는 값의 자리수를 출력하는 문장을 설정해준다.

값이 있으니까 양의 정수로 결과 값이 나온다

그 문자가 없을 때 값이 정수로 나오긴 한다 그런데 음수로 나옴

출력하는 값에 대해 문장 나오는 것 설정하고 싶다. -> 조건문을 사용한다

```
let s="one two one two";
a = s.indexOf("two");
b = s.indexOf("Two",6);
console.log(a);
console.log(b);
if(a>=0){
  console.log("해당 문자열이 있습니다.");
}
if(b>=0){
  console.log("해당 문자열이 있습니다.");
}
```

조건값으로 0 보다 크면 해당문자열이 있다고 출력하는 것이다.

```
4
-1
해당 문자열이 있습니다.
```

## <연산자>

### - 삼항연산자

삼항연산자는 네/아니요로 처리할 때 사용한다(두개의 답으로 처리할 때 사용)

조건이 더 길어지고 나오는 개수의 개수가 더 많다고 하면 if 와 else if 와 else 를 사용한다.

```
(a>0) ? console.log("양수"):console.log("음수");
```

a 가 0 보다 큰 것이 맞으면 앞의 값인 "양수"가 나오고 아니면 뒤의 값인 "음수"가 나온다

```
let result = a>0 ? "양수":"음수";
```

### - 이항연산자

```
b=a+b
```

이항연산은 사칙연산에 부등식

```
let result = a>0
```

## <문제>

1. 아래 표와 같이 생긴 배열 만들어보기

1	1	3	4	5	6
2	3				
4	5	15			
5	7	21	27	35	43
7	9	27	34		

```
let b = [[1,1,3,4,5,6],[2,3],[4,5,15],[5,7,21,27,35,43],[7,9,27,34]];
```

b의 length의 값은 행의 개수이고, 그 안에 배열들은 열의 개수이다(각각 다르게 설정되어 있음)

- 2.

어제 배운 것 중 sort 정렬하는 함수가 있다.

```
sort
```

1. 빈 배열 작성
2. random 함수로 1 이상 100이하의 100개 작성
3. 배열 오름차순 정렬
4. 배열 출력

```
let d=[];
for (i=0;i<=100;i++){
  let ran = Math.floor(Math.random()*100+1);
  d.push(ran);
}
d.sort();
console.log(d);
```

d.push(ran) -> ran의 값을 d에 저장하는 것

d.sort() -> 정렬한 것

정렬은 컴퓨터기준으로 정렬한 것이다.

```
PS C:\work\javascript\11-07> node ex08.js
```

```
[
  1, 1, 1, 100, 12, 13, 13, 14, 14, 16, 16, 16,
  17, 18, 18, 18, 18, 19, 2, 2, 20, 21, 21, 22,
  27, 28, 28, 29, 29, 30, 34, 38, 38, 4, 40, 42,
  43, 48, 48, 48, 48, 49, 5, 5, 50, 50, 52, 53,
  56, 57, 57, 58, 58, 59, 6, 60, 63, 63, 63, 63,
  64, 64, 65, 66, 68, 68, 69, 7, 7, 70, 73, 73,
  74, 75, 77, 78, 78, 79, 8, 82, 82, 82, 82, 83,
  84, 84, 84, 85, 86, 9, 91, 92, 93, 93, 94, 94,
  95, 95, 95, 97,
  ... 1 more item
]
```

```
var d=[];
```

Let 대신 var라고 하면 숫자, 수치로 만들라는 것 (let : 지역변수, var : 전역변수)

이렇게 해도 위의 값과 동일하게 수치로 정렬되지 않고 문자열로 정렬된다.

```
hint (method) Array<number>.sort(compareFn?: ((a: number, b: number) =>
js number) | undefined): number[]
// Sorts an array in place. This method mutates the array and returns a reference to the
// same array.
@param compareFn
Function used to determine the order of the elements. It is expected to return a
negative value if the first argument is less than the second argument, zero if they're
va equal, and a positive value otherwise. If omitted, the elements are sorted in
fo ascending, ASCII character order.
[11,2,22,1].sort((a, b) => a - b)
d.sort();
```

- sort 사용하는 방법

함수는 미리정해져 있는 값 부르는 것이다.

```
d.sort(function(a,b){
    return a-b; //return 돌려주는 값이다.
});
```

```
PS C:\work\javascript\11-07> node exm08.js
[
  1, 3, 3, 5, 6, 7, 7, 8, 10, 11, 11, 16,
  16, 16, 16, 18, 18, 19, 20, 21, 22, 27, 28, 28,
  29, 29, 32, 32, 33, 35, 36, 36, 36, 40, 40, 41,
  43, 46, 49, 49, 49, 50, 51, 51, 56, 60, 61, 63,
  65, 65, 66, 68, 68, 69, 69, 71, 71, 72, 72, 72,
  73, 74, 75, 76, 77, 78, 78, 79, 79, 80, 80, 80,
  82, 82, 83, 83, 83, 84, 84, 85, 85, 87, 87, 87,
  87, 89, 90, 90, 91, 93, 94, 94, 94, 95, 95, 96,
  96, 97, 97, 100
]
```

그냥 sort는 let이 되었든 var이 되었든 문자 기준의 오름차순 정렬이 된다.

그런데 sort안에 function을 넣으면 함수기능을 사용해서 이렇게순서대로 수치로 가지고 올수있다

Function 함수 : `(local function)(a: number, b: number): number`

그러면 결과값은 이전과 다르게 수치로 오름차순 정렬이 된 것을 볼 수 있다.

\* function함수 사용하는 사람도 있고, 그거 사용하지 않고 하는 사람도 있고, 문자열로만 정렬하는 사람도 있다. 이것 비교할 것 (오후시간에)

## 다른방법 #1 팀

```
let a = [];
for(i=0; i<100; i++){
    let b = Math.floor(Math.random()*100)+1;
    a.push(b);
}
a.sort();
console.log(a);
```

## 다른방법 #2 팀

```
function one(){
    let arr = [];
    for(let k=0; k<100; k++){
        let makeNum = Math.floor((Math.random()*100)) + 1 ;
        arr.push(makeNum);
    }
    console.log(JSON.stringify(arr));
    for(let j=0; j<arr.length; j++){
        for(let i=0; i<arr.length; i++){
            if(arr[i]>arr[i+1]){
                let data = arr[i+1];
                arr[i+1] = arr[i];
                arr[i] = data;
            }
        }
    }
    console.log(JSON.stringify(arr));
}
```

```

function two() {
  let arr = [];
  for (let k = 0; k < 100; k++) {
    let makeNum = Math.floor(Math.random() * 100) + 1;
    if (JSON.stringify(makeNum).length === 2) {
      makeNum = "0" + JSON.stringify(makeNum);
    } else if (JSON.stringify(makeNum).length === 1) {
      makeNum = "00" + JSON.stringify(makeNum);
    }
    arr.push(makeNum);
  }
  console.log(JSON.stringify(arr));
  arr.sort();
  console.log(JSON.stringify(arr));
  let makeInt = [];
  for(let j=0; j<arr.length; j++){
    makeInt.push(parseInt(arr[j]))
  }
  console.log(JSON.stringify(makeInt));
}
one();
console.log("")
console.log("")
console.log("")
two();

```

### 다른방법 #3 팀

```

let a=[];
for(let i=0; i<100; i++){
  a.push(Math.floor(Math.random()*100)+1);
}
console.log(a);
a.sort((a,b)=>a-b);
console.log(a);

```

a.sort((a,b) -> **a-b**); //오름차순으로 정렬해라 라는 것

a.sort((a,b) -> **b-a**); //내림차순으로 정렬해라 라는 것, b 에서부터 a 로 출력해라

3. 아래  $b \geq 0$  일 때 해당 출력문이 왜 출력되지 않는지 이유찾기

```
a = s.indexOf("two");
b = s.indexOf("Two" , 6);
console.log( a );
console.log( b );

if(a>=0){
    console.log("해당 문자열이 있습니다.");
}
|
if(b>=0){
    console.log("해당 문자열이 있습니다.");
}
```

4. a 가 85 점일 때 조건문 사용하여 점수대에 맞는 등급을 나누는 코드 만들기

```
let a=85;

if(a>=90) {console.log("A");}
else if(a>=80) {console.log("B");}
else if(a>=70) {console.log("C");}
else if(a>=60) {console.log("D");}
else {console.log("F");} //그 외 나머지값은 F 이다. 라고 해주는 것
```



5.

5. 사용자 정의 함수 'fn가입상태'를 작성하여 [표1]의 가입상태[H4:H39]를 표시하시오. (6점)

- ▶ 'fn가입상태'는 가입기간, 미납기간을 인수로 받아 값을 되돌려줌
- ▶ 미납기간이 가입기간 이상이면 '해지예상', 미납기간이 가입기간 미만인 경우 중에서 미납기간이 0 이면 '정상', 미납기간이 2 초과이면 '휴면보험', 그 외는 미납기간과 '개월 미납'을 연결하여 표시 [ 표시 예: 1개월 미납 ]
- ▶ If 문, & 연산자 사용

```
Public Function fn가입상태(가입기간, 미납기간)
End Function
```

*console.log ?*

*if (c >= b) { a = "해지예상" } else if (c <= 0) { } else if (c > 2) { } else { }*

5점

미납기간이

1) 정상  
2) 휴면보험  
3) %d개월 미납

```
//a=가입상태
//b=가입기간
//c=미납기간
let c=0;
let b=4;
if (c>=b) { //미납기간이 가입기간 이상일 때
    console.log("해지예상");
}
else if (c<b) { //미납기간이 가입기간 미만일 때
    if(c==0) {console.log("정상");} //미납기간이 0 이면 정상 출력
    else if(c>2) {console.log("휴면보험");} //미납기간이 2 초과이면 휴면보험 출력
    else {console.log("%d 개월 미납" ,c);} //미납기간이 그 외의 값이면 %d 개월 미납 출력
}
```

```
PS C:\work\javascript\11-07> node exm11.js
휴면보험
PS C:\work\javascript\11-07> node exm11.js
%d 개월미납1
PS C:\work\javascript\11-07> node exm11.js
1 개월미납
PS C:\work\javascript\11-07> node exm11.js
1개월 미납
PS C:\work\javascript\11-07> node exm11.js
정상
```

6.

3.사용자정의 함수 'fn판매액'을 작성하여 판매액[H4:H17]을 계산하여 표시하시오.(6점 )

▶ 'fn판매액'은 판매량과 가격을 인수로 받아 판매액을 계산하는 함수이다.

▶ 판매액은 판매량×가격으로 계산하되, 판매량이 5개 이상이면 5%, 그렇지 않으면 3%를 할인한 금액으로 계산하시오.

```
//f=판매액
//g=판매량
//h=판매가격
g=4
h=1000
if (g>=5) {
  f=g*h-g*h*0.05; //전체금액에 *0.95 를 해도된다
  console.log(f);
}
else {
  f=g*h-g*h*0.03; //전체금액에 *0.97 를 해도된다
  console.log(f);
}
```

PS C:\work\javascript\11-07> node exm11.js  
3880

7.

3.사용자정의 함수 'fn판매액'을 작성하여 판매액[H4:H17]을 계산하여 표시하시오.(6점 )

▶ 'fn판매액'은 판매량과 가격을 인수로 받아 판매액을 계산하는 함수이다.

▶ 판매액은 판매량×가격으로 계산하되, 판매량이 5개 이상이면 5%, 그렇지 않으면 3%를 할인한 금액으로 계산하시오.

AND OR

AND	OR
T   T   F	F   F   F
이면서	이거나
이고	또는
그리고	

판매량 10개 이상 이고 AND  
판매금액이 100,000 이상이면 5%  
판매량 10개 이상 이거나 OR  
판매금액이 100,000 이상이면 3%  
나머지 할인 없음

논리연산자 AND 와 OR 이 있다.

AND 는 위에 '이고'처럼 두개의 조건이 참인경우 출력하는 경우이고, 연산자는 &&를 사용한다.

OR 은 위에 '이거나'처럼 두개의 조건중 하나만 참이어도 출력하는 것이고, 연산자는 || 이다.

```
// 논리 연산자 사용해서 나온 코드

// 7번문제
// f=판매액
// a=판매량
// b=판매가격
a=10
b=1000
f=a*b
if (a>=10 && f>100000) {
    f=f-f*0.05; //전체금액에 *0.95를 해도된다
    console.log(f);
}
else if (a>=10 || f>100000) {
    f=f-f*0.03; //전체금액에 *0.97를 해도된다
    console.log(f);
}
```

```
//if 문만 사용해서 나온 코드 (논리연산자 사용하지 않음)

a=10
b=1000
f=a*b
if (a>=10) {
    if (f>=100000){
        f=f-f*0.05;
        console.log(f);
    }
    else {
        f=f-f*0.03;
        console.log(f);
    }
}
else if (f>=100000) { //a인 판매량이 10개 미만만 오기 때문에 if(a>=10)할 필요없다
    f=f-f*0.03; //전체금액에 *0.97를 해도된다
    console.log(f);
}
else {
    console.log(f);
}
```

나는 f를 지정할 때 전체 '판매값 - 할인된 금액' 해서 구했지만  
f\*0.97 할인된 금액 빼 전체 퍼센트 곱해도 된다.

a>=10의 값 이후 else if에서는 a의 판매량이 이미 10개미만이 오는 것임을 기억하는 게 중요!

## 다른방법 #선생님방법

```
let a=0; //판매량
let b=36; //가격
let c=a*b; //판매금액
let d=0; //할인금액
if( a>=10){
    if(c>=100000){
        d = c*0.95
    }else{
        d = c*0.97
    }
}
```

판매량이 10개이상 이고  
판매금액이 100,000 이상일 때

```
let c=a*b; //판매금액
let d=0; //할인금액
if( a>=10){
    if(c>=100000){
        d = c*0.95;
        console.log(d);
    }else{
        d = c*0.97;
        console.log(d);
    }
}
else if(c>=100000){
    d=c*0.97;
    console.log(d);
}
else{
    d=c;
    console.log(d);
}
```

논리연산자 사용하지 않은 방법

// 판매금액 100000 미만 판매 수량이 10개 이상은 된다.

//판매량 10개 미만만 남아 있음.

논리연산자 사용한 방법

```
if( a>=10 && c>=100000 ){
    console.log(c*0.95);
}
else if(a>=10 || c>=100000){
    console.log(c*0.97);
}
else{
    console.log(c);
}
```

a가 10이상과 c가 100000이상인거 둘다 포함

논리연산자로 사용하면 코드의 개수가 확 줄어든다.

## 다른방법 #2 팀

- 삼항연산자 이용한 방법

```
// d : 최종금액  
// b : 수량  
// c : 가격  
d = b>=10 && b*c>=100000 ? (b*c)/100*95 : b>=10 || b*c>=100000 ? (b*c)/100*97 : b*c;
```

d 에 값을 넣을 건데 먼저 삼항연산자를 이용하여 참과 아닌 것 구분해서 넣는다  
아닌것에서 다시 참과 아닌것의 조건으로 구분하여 최종인 값이 들어가는 것.

```
b>=10 && b*c>=100000 T/F 의 조건  
?  
(b*c)/100*95 //T?  
:  
b>=10 || b*c>=100000 //F, T/F의 조건  
?  
(b*c)/100*97 //t?  
:  
b*c; //F
```