[11/07] Javascript Language - 다중 배열, 오름*내림차순, if & else if & else, and & or, 3항연산

- 다차원 배열 : 배열내의 배열, 2차원 배열 (행*열 순서로)
  - 배열내의 배열은 1개로 인식, 3개의 객체(요소의 집합)를 하나로 인식

```
let a=[[1,2,3],[4,5,6],[7,8,9]];
console.log(a.length);
```
```
3
```

  - 정형화 배열(N*N) : j < a . length #이중 반복문과 유사

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

    - a.length는 3개
    - a[0] 에 해당하는 [1],[2],[3] 중 3개만큼의 [1],[2],[3] 출력
    - a[1] 에 해당하는 [4],[5],[6] 중 3개만큼의 [4],[5],[6] 출력
    - a[2] 에 해당하는 [7],[8],[9] 중 3개만큼의 [7],[8],[9] 출력

  - 비정형화 배열(N*M) : j < a [i] . length

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

    - a[0].length 는 4개 (a[1],a[2],a[3] 도 동일한 4개)
    - a[0] 에 해당하는 [1,2,3,4] 중 4개만큼의 [1],[2],[3],[4] 를 출력

  - 타 비정형화 배열도 적용되는 것을 확인

```
let a=[[1,2,3,4],[5,6,7,8,13,14],[9,10,11,12]];
for(let i=0; i<a.length; i++){
    for (let j=0; j<a[i].length; j++){
        console.log(a[i][j]);
    }
}
```
```
1
2
3
4
5
6
7
8
13
14
9
10
11
12
```

  - 아래 구조와 동일

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

1	2	3	4		
5	6	7	8	13	14
9	10	11	12		

- a [j] 만 남길 경우

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

```
[ 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 ]
```

  - 행 : for(let i=0; i<a.length; i++)
  - 열 : for(let j=0; j<a[i].length; j++)
  - j < a [0] 는 6개 이므로, a ([j]) 는 a ([0]) ~ a([5]) 를 호출하므로
    - a ([0]) : [1, 1, 3, 4, 5, 6]
    - a ([1]) : [2, 3]
    - a ([2]) : [4, 5, 15]
    - a ([3]) : [5, 7, 21, 27, 35, 43]
    - a ([4]) : [7, 9, 27, 34]
    - a ([5]) : undefined (a의 6번째는 없으므로)

- 3차원 배열(행*열*높이)
  - DB 생성시 필요한 기본적 이론

```
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]);
    }
}
```

```
1
2
3
4
5
6
7
8
9
10
11
12
```

- 대소문자 미구분

```
let s="hello World";
for (let i=0; i<s.length; i++){
    console.log(s[i]);
}

s[0] = "H"
console.log(s[0]);
```

```
h
e
l
l
o
W
o
r
l
d
h
```

  - 대문자 H 를 선언했으나 반영되지 않음

- 오름차순과 내림차순

  Q) 빈 배열, random 함수로 1이상 101이하의 100개 작성, 배열 오름/내림차순 정렬, 출력

  - 일반적인 오름차순 정렬

```
let a = [];
    for(i=0; i<100; i++){
        let b = Math.floor(Math.random()*101+1)
        a.push(b)
    }
    a.sort();
    console.log(a);
```

```
[
  1,  1, 10, 100, 101, 101, 12, 13, 14, 15, 15, 16,
 17, 18, 19,  19,   2,  22, 23, 23, 23, 24, 25, 26,
 26, 30, 30,  30,  31,  33, 34, 34, 37, 38,  4, 40,
 41, 41, 42,  42,  42,  44, 44, 45, 46, 47, 48, 49,
 50, 52, 55,  56,  59,  59, 61, 61, 62, 62, 66, 66,
 69, 71, 71,  71,  73,  73, 74, 74, 77, 78, 78, 78,
 78, 78, 78,   8,  81,  81, 81, 82, 82, 82, 83, 85,
 86, 88, 89,  89,   9,   9,  9, 90, 91, 93, 95, 95,
 97, 97, 97,  98
]
```

  - function을 통한 오름차순 정렬 (1), (2)

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

```
[
  2,  4,  5,   7,  7,  7,  7,  8,  8, 10, 10, 10,
 11, 11, 12,  12, 15, 15, 16, 18, 19, 19, 19, 20,
 21, 22, 23,  25, 26, 27, 28, 32, 32, 34, 35, 35,
 39, 42, 44,  44, 46, 49, 50, 51, 56, 57, 58, 58,
 60, 61, 61,  61, 62, 62, 63, 64, 65, 65, 66, 66,
 66, 67, 67,  68, 69, 69, 70, 71, 71, 72, 72, 73,
 74, 74, 74,  75, 75, 75, 76, 80, 81, 82, 82, 83,
 83, 84, 87,  87, 88, 88, 91, 92, 93, 93, 94, 94,
 96, 98, 99, 100
]
```

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

    - var : 수치형, 배열을 숫자로 지정, 전역변수 (vs let 은 지역변수)
    - .sort(function (a, b) { return a - b })
    - .sort((a,b)=>a-b)

- 내림차순 정렬시

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

```
[
 98, 97, 96, 96, 96, 94, 93, 93, 93, 92, 90, 89,
 88, 87, 85, 84, 84, 83, 83, 83, 82, 79, 79, 78,
 78, 76, 76, 74, 73, 72, 70, 70, 70, 69, 69, 68,
 65, 65, 65, 62, 62, 58, 58, 57, 56, 53, 50, 50,
 47, 44, 42, 42, 41, 40, 40, 38, 38, 38, 38, 37,
 36, 36, 36, 35, 34, 32, 31, 30, 29, 29, 27, 27,
 26, 26, 26, 25, 24, 24, 24, 24, 23, 21, 20, 20,
 16, 15, 15, 14, 12, 12, 12, 11, 11,  8,  8,  7,
  5,  5,  4,  2
]
```

  - .sort((a,b)=>b-a)
  - 오름차순은 a-b, 내림차순은 b-a

- function을 통한 오름차순 정렬 (3)

```
// 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(arr);
}
```

```
[
  1,  2,  3,  3,  4,  5,  5,  5,  6,  7,  7,  8,
  8, 10, 10, 11, 11, 13, 14, 14, 15, 15, 17, 19,
 21, 21, 21, 23, 24, 24, 25, 29, 29, 30, 31, 32,
 32, 35, 38, 39, 40, 40, 43, 44, 47, 47, 47, 48,
 49, 50, 53, 53, 53, 54, 55, 55, 56, 57, 57, 57,
 58, 58, 58, 60, 61, 62, 63, 66, 66, 66, 67, 68,
 69, 71, 73, 74, 76, 78, 79, 79, 82, 83, 84, 84,
 84, 84, 84, 84, 87, 88, 88, 90, 91, 92, 93, 94,
 94, 95, 96, 97
]
```

  - arr[0] = 100 이고, arr[1] = 99 이어서 뒷자리가 작을 경우
  - data = arr[i+1] : data 에 뒷자리의 값 99를 임시값으로 지정
  - arr[1] = arr[0] : arr[1] 과 arr[0] 로 지정, 이때 arr[1] = 100 이 됨
  - arr[0] = data : 앞선 임시값인 99로 지정, 이때 arr[0] = 99 가 됨
  - 해당 과정의 반복(if)을 통해 오름차순으로 정렬

- indexof : text 의 띄어쓰기 미포함하며 순서대로 인식

```
let s = "one two one two";
console.log(s.indexOf("one"));
console.log(s.indexOf("two"));
console.log(s.indexOf("two", 6));
console.log(s.indexOf("Two"));
```

```
0
4
12
-1
```

  - 출력값 0 : text의 띄어쓰기 미포함, 0번째 text에 있음을 의미
  - 출력값 4 : text의 띄어쓰기 미포함, 4번째 text에 있음을 의미
  - indexof(6) : 6번째부터 찾기
    - 출력값 12 : 띄어쓰기를 포함한 12번째 text에 있음을 의미
  - 출력값 -1 : 없음, 대문자 T 가 없음을 의미

  - 조건부 출력

```
let s = "one two one two";

a = s.indexOf("two");
b = s.indexOf("Two");

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

```
해당 문자열이 있습니다
해당 문자열이 없습니다
```

```
if(a>=0){
    console.log("해당 문자열이 있습니다");
}else if(a<0) {
    console.log("해당 문자열이 없습니다");
}

if(b>=0){
    console.log("해당 문자열이 없습니다");
}else{
    console.log("해당 문자열이 없습니다");
}
```

  - 출력값이 양수면 '있습니다', 음수면 '없습니다'.

- if & else if & else

```
let a = 85;
                    B
if(a>=90){
    console.log("A");
}else if(a>=80){
    console.log("B");
}else if(a>=70){
    console.log("C");
}else{
    console.log("F");
}
```

  - else if : 조건식 (a>=70) 이 붙을 수 있음, 조건식에 유용
  - else : 뒤에는 조건식이 붙지 못함, 나머지 처리에 유용

```
let a = 50;
// fn 가입상태
let b = 20;
// 가입기간
let c = 10;
// 미납기간
```

```
if(c>=b){
    console.log("해지예상");
    // 미납기간이 가입기간 보다 이상인 경우
}else if(c<b){
    // 미납기간이 가입기간 미만인 경우 중에서
    if(b==0){
    console.log("정상");
    //가입기간이 0 인 경우
    }else if(b>2){
    console.log("휴면보험");
    //가입기간이 0 인 경우
    }else{
    console.log("%d개월미납", c);
    }
    //그 외의 처리
}
```

컴활 기출문제중 발췌

- and & or

Q) 판매량이 10개 이상 이고 판매금액이 100,000 이상이면 5%
판매량이 10개 이상 이거나 판매금액이 100,000 이상이면 3%
나머지 할인없음시

#and : &&
#or : ||

```
let a = 10;
//판매량
let b = 1000;
//가격
let c = a*b;
//판매액
```

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

  - and : &&
  - or : ||

- 3항 연산자

```
let a=85;
(a>0) ? console.log("양수"):console.log("음수");
```
양수

```
let a=85;
let result = a>0 ? "양수" : "음수";
console.log(result);
```
양수

  - 2항 연산 : 사칙 연산 or 부등식에 활용 ex); b=a+c
  - 3항 연산 : True or False 의 간단한 처리에 활용
    - a>0 ? "양수" : "음수"
    - 조건 ? True시 출력 : False시 출력

Q) 판매량이 10개 이상 이고 판매금액이 100,000 이상이면 5%
판매량이 10개 이상 이거나 판매금액이 100,000 이상이면 3%
나머지 할인없음시 - 3항 연산자로 표현

```
let a = 10;
//판매량
let b = 1000;
//가격
let c = a*b;
//판매액
```

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

if 문의 경우

```
d = a>=10 && c>=100000 ? c*0.95 : a>=10 || c>=100000 ? c*0.97 : "할인없음";
console.log(d);
```

다중 연산자의 경우

  - 다중 연산자는 좌->우측 순서로 조건이 맞으면 해당 위치에서 중단,
    조건이 틀리면 우측으로 계속 진행하는 형식
  - 조건 ? T : F // 이때 F 는 F 의 역할과 다음 조건의 시작
    조건 ? T : F
  - a>=10 && c>=100000 ? c*0.95 : a>=10 || c>=100000
    - a>=10 || c>=100000 : 첫 조건문의 T/F 중 F의 역할이며,
      a>=10 || c>=100000 : 두번째 조건문의 시작이기도 함
    - a>=10 || c>=100000 ? c*0.97 : c