배열

배열 만들기

```
let a = [1, 2, 3, 4];

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

```
1
2
3
4
```

```
let a = ["one", "two", "three", "four"];

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

```
"one"
"two"
"three"
"four"
```

length 속성

배열의 length 속성은 배열의 길이를 리턴한다.

배열 원소의 타입

javascript 언어는 약타입 언어이므로, 배열 원소의 타입에 제약이 없다.
즉 배열 원소가 모두 같은 타입이지 않아도 된다.

```
let a = [3, "hello", true];

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

```
"number"
"string"
"boolean"
```

빈 배열 만들기

길이 0 인 배열 만들기.

```
let a = [];
console.log(a.length);
```

```
0
```

`let a = [];`

지역 변수 a 에 빈 배열이 대입된다.

배열은 객체

```
let a1 = [1, 2, 3];
let a2 = ["a", "b", "c"];

console.log(typeof a1);
console.log(typeof a2);
```

```
"object"
"object"
```

javascript 언어에서 배열은 객체(object) 이다.

배열이 객체이므로, 메소드를 호출할 수 있다.

구조 분해 할당 (destructuring assignment)

```
let a = [3, 4, 5];

let [e1, e2, e3] = a;
console.log(e1, e2, e3);

[e1, e2] = a;
console.log(e1, e2);
```

```
3 4 5
3 4
```

destructuring assignment

배열의 원소들 각각을 동시에 변수들에 대입한다.

`let [e1, e2, e3] = a;`

e1, e2, e3 지역 변수가 생성된다.
a[0] 값이 e1 변수에 대입된다.
a[1] 값이 e2 변수에 대입된다.
a[2] 값이 e3 변수에 대입된다.

`[e1, e2] = a;`

이미 존재하는 e1, e2 변수를 사용한다.
a[0] 값이 e1 변수에 대입된다.
a[1] 값이 e2 변수에 대입된다.

```
let a = [3, 4, 5, 6, 7, 8];

let [e1, e2, ...e3] = a;
console.log(e1, e2);
console.log(e3);
```

```
3 4

// [object Array] (4)
[5,6,7,8]
```

`let [e1, e2, ...e3] = a;`

e1, e2, e3 지역 변수가 생성된다.
a[0] 값이 e1 변수에 대입된다.
a[1] 값이 e2 변수에 대입된다.
a[2] 부터 배열 끝까지의 값들이 들어있는 배열이 생성되고, 이 배열이 e3 변수에 대입된다.

swap

```
let a, b;
a = 5;
b = 6;
console.log(a, b);

//a 변수의 값과 b 변수의 값을 서로 교환(swap)하는 코드
let temp = a;
a = b;
b = temp;
console.log(a, b);
```

```
5 6
6 5
```

```
let a, b;
[a, b] = [5, 6];
console.log(a, b);

[a, b] = [b, a];
console.log(a, b);
```

[a, b] = [5, 6];

a 변수에 5 값이 대입되고,
b 변수에 6 값이 대입된다.
위 코드에 사용된 문법이 구조분해 할당(destructuring assignment) 이다.

[a, b] = [b, a];

a 변수에 b 값이 대입되고,
b 변수에 a 값이 대입된다.
a 변수의 값과 b 변수의 값을 서로 교환(swap)하는 코드이다.

```
5 6
6 5
```

```
let a = [];
a[0] = 3;  //a[0]에 3을 대입
a[1] = 4;  //a[1]에 4를 대입
console.log(a);

//a[0] 값과 a[1] 값을 서로 교환(swap)하는 코드
let temp = a[0];
a[0] = a[1];
```

```
a[1] = temp;
console.log(a);
```

```
// [object Array] (2)
[3,4]
```

```
// [object Array] (2)
[4,3]
```

```
let a = [];
[a[0], a[1]] = [3, 4]; //a[0]에 3을 대입하고, a[1]에 4를 대입한다.
console.log(a);

[a[0], a[1]] = [a[1], a[0]]; //a[0] 값과 a[1] 값을 서로 교환(swap)하는 코드
console.log(a);
```

```
// [object Array] (2)
[3,4]
```

```
// [object Array] (2)
[4,3]
```

```
let i = 3, j = 4;
let a = [5, 6, 7]

let a1 = [i, j, a];
console.log(a1);

let a2 = [i, j, ...a];
console.log(a2);
```

`let a1 = [i, j, a];`

a1 배열의 크기는 3 이다.
a[0] 값은 숫자 3, a[1] 값은 숫자 4, a[2] 값은 [5, 6, 7] 배열이다.

`let a2 = [i, j, ...a];`

a2 배열의 크기는 5 이다.
a2 배열의 값은 [3, 4, 5, 6, 7] 이다.

... 연산자 = spreading operator

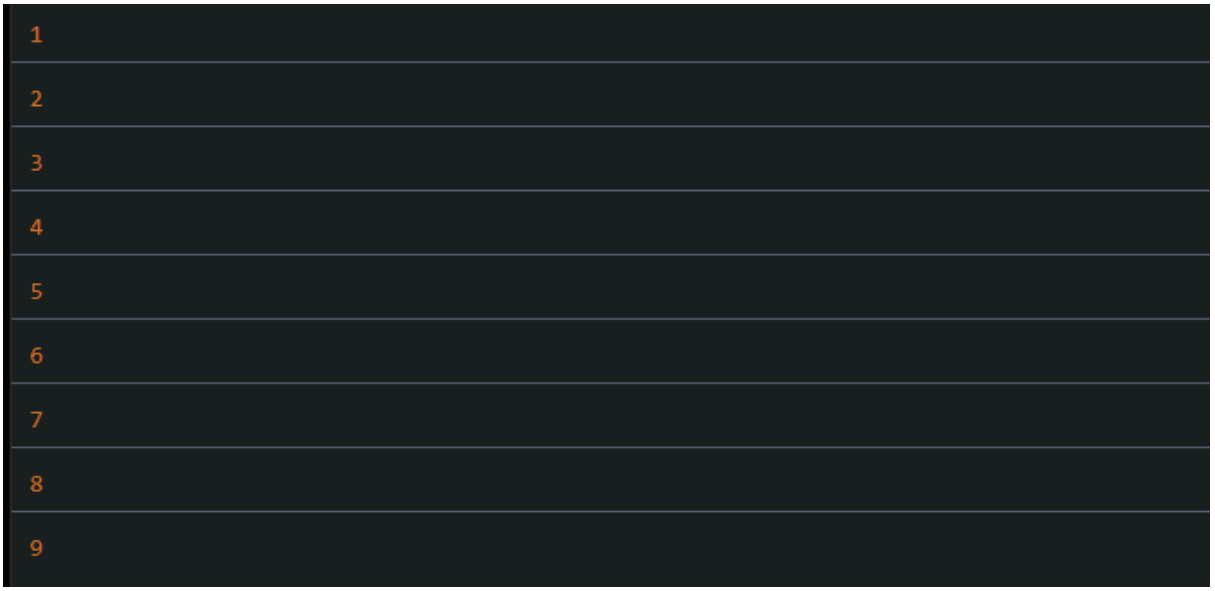
다차원 배열

중첩 for문

```
for (var i = 1; i <= 6; i++) { //i값이 1부터 6까지 1씩 증가하며 실행
  for (var j = 1; j <= 6; j++) { //j가 1부터 6까지 1씩 증가하며 실행
    //i+j값이 6일 경우에만 i와 j값을 출력
      if (i + j === 6) console.log(`[${i}, ${j}]`);
      //결과 값 [1,5] [2,4] [3,3] [4,2] [5,1]
  }
}
```

2차원 배열 #1

```
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[i].length; ++j)
    console.log(a[i][j]);
```



a[0] 값은 [1, 2, 3] 일차원 배열이다.
a[1] 값은 [4, 5, 6] 일차원 배열이다.
a[2] 값은 [7, 8, 9] 일차원 배열이다.

a 값은 3행 3열 2차원 배열이다.

```
let a = [];
a[0] = [1, 2, 3];
a[1] = [4, 5, 6];
a[2] = [7, 8, 9];

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

위 두 코드의 실행 결과와 메모리 구조는 동일하다.

`let a = [];`

최초 a 배열의 크기는 0 이다.

`a[0] = [1, 2, 3];`

이 줄을 실행하는 순간, a 배열의 크기는 1로 자동 확대된다.
a[0] 값은 [1, 2, 3] 일차원 배열이다.

a 배열은 1행 3열 2차원 배열이다.

`a[1] = [4, 5, 6];`

이 줄을 실행하는 순간, a 배열의 크기는 2로 자동 확대된다.
a[1] 값은 [4, 5, 6] 일차원 배열이다.
a 배열은 2행 3열 2차원 배열이다.

`a[2] = [7, 8, 9];`

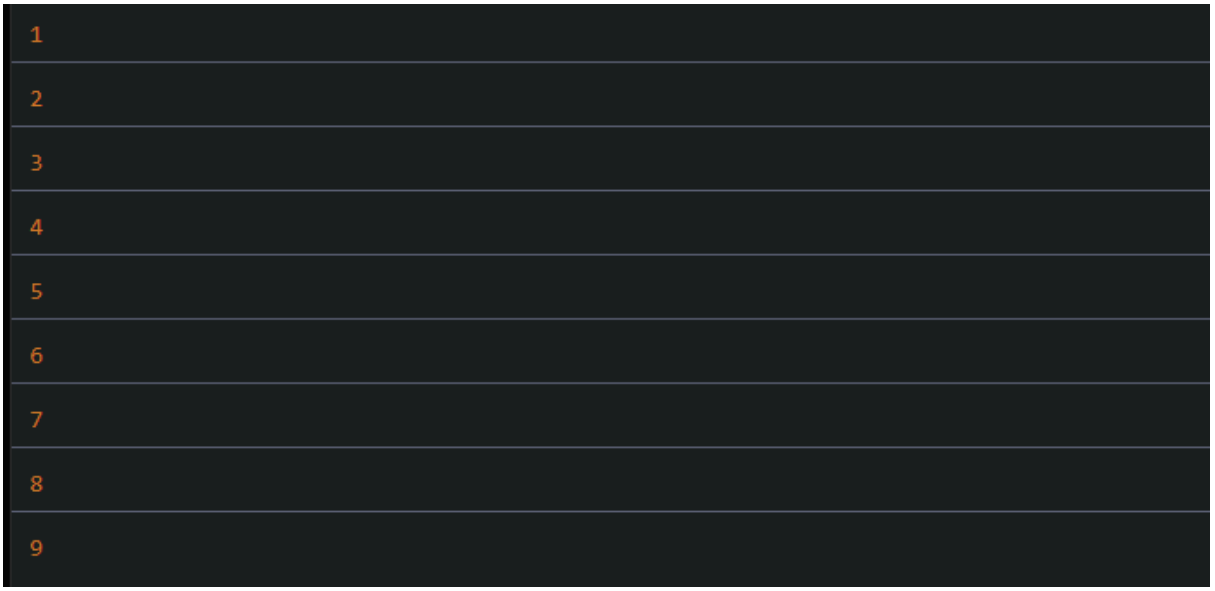
이 줄을 실행하는 순간, a 배열의 크기는 3으로 자동 확대된다.
a[2] 값은 [7, 8, 9] 일차원 배열이다.
a 배열은 3행 3열 2차원 배열이다.

```
let a1 = [1, 2, 3];
let a2 = [4, 5, 7];
let a3 = [7, 8, 9];
let a = [a1, a2, a3];

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

a 배열은 3행 3열 2차원 배열이다.

2차원 배열 #2

```
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[i].length; ++j)
    console.log(a[i][j]);
```



2차원 배열에서 열의 크기가 모두 같지 않아도 된다.

a 배열은 3행 ?열 2차원 배열이다.
a[1] 값은 [1, 2] 일차원 배열이다.
a[2] 값은 [3, 4, 5] 일차원 배열이다.
a[3] 값은 [6, 7, 8, 9] 일차원 배열이다.

3차원 배열

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

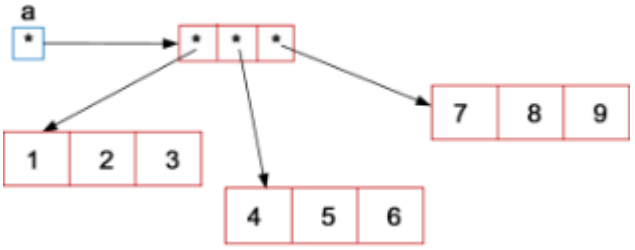
메모리 구조

1차원 배열

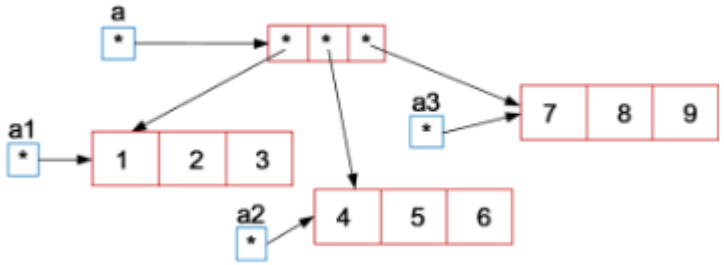
```
let a = [1, 2, 3, 4];
```

2차원 배열 #2

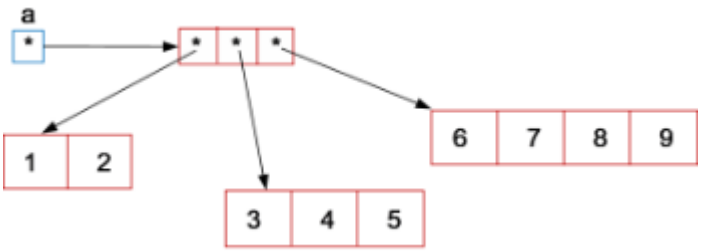
```
let a = [[1, 2, 3], [4, 5, 6], [7, 8, 9]];
```




```
let a1 = [1, 2, 3];
let a2 = [4, 5, 7];
let a3 = [7, 8, 9];
let a = [a1, a2, a3];
```




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


a
1
2
3
4
5
6
7
8
9