6강

2차원

배열(Array)

2차원 배열(Array) 선언

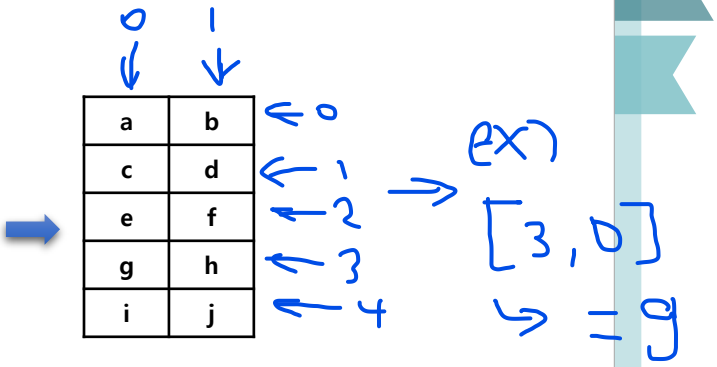
JavaScript

2차원 배열(Array) 선언

- 자바스크립트는 진정한 2차원 배열은 없다
- var arr = [][]; 이와 같은 한 번에 2차원 배열 선언이 불가능하다
- 약간의 트릭을 통하여 2차원 배열과 비슷한 배열을 만들 수 있다

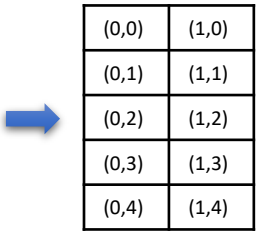
초기값을 할당하여 배열(Array) 생성

```
// arr[5][2]
var arr = [ ['a','b'], ['c', 'd'], ['e', 'f'], ['g', 'h'], ['i', 'j'] ];
```



반복문을 사용하여 빈 배열(Array) 생성

```
// arr[5][2]
var arr = new Array(5);
for (var i = 0; i < arr.length; i++) {
        arr[i] = new Array(2);
}
```



2차원 배열 생성 함수를 만들어 사용

```
function create2DArray(rows, columns) {
  var arr = new Array(rows);
  for (var i = 0; i < rows; i++) {
   arr[i] = new Array(columns);
  }
   return arr;
}  // arr[5][2]
var arr = create2DArray(5, 2);
```

☞ 여기서 잠깐

- 다음과 같은 배열을 선언하고, 각 행을 콘솔에 출력하세요.

```
[['a','b'], ['c','d'], ['e','f']]
```

- 출력 예시

```
a b
c d
e f
```

	Index : 0	Index : 1
Index : 0	a	b
Index : 1	c	d
Index : 2	e	f

```
let arr = [
        ['a', 'b'],
        ['c', 'd'],
        ['e', 'f'],
      ];
      for (let i = 0; i < arr.length; i++) {
        //행
        for (let j = 0; j < arr[i].length; j++) {
          //열
          console.log(arr[i][j]);
        }
      }
```

2차원 배열 메모리 구조

☞ **arr → 3칸짜리 1차원 배열**

arr[0] → ['a', 'b'] 배열 객체를 가리킴

arr[1] → ['c', 'd'] 배열 객체를 가리킴

arr[2] → ['e', 'f'] 배열 객체를 가리킴

☞ **각 내부 배열**

arr[0][0] = 'a'

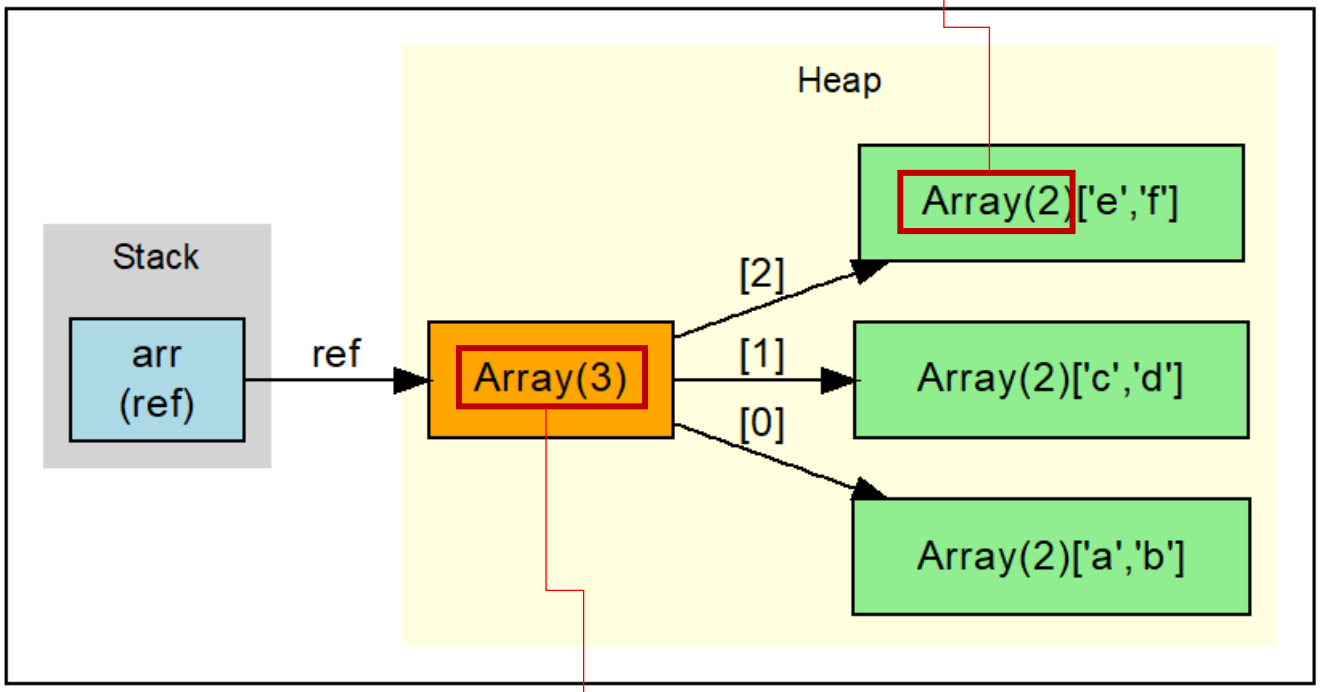
arr[0][1] = 'b'

arr[1][0] = 'c'

arr[1][1] = 'd'

arr[2][0] = 'e'

arr[2][1] = 'f'



☞ 여기서 잠깐

- 2행 3열 배열을 만들고, 각 요소에 (행,열) 좌표를 넣어 출력하세요.
- 출력 예시

```
(0,0) (0,1) (0,2)
(1,0) (1,1) (1,2)
```

```
// 2행 3열
    let arr2 = new Array(2);
    for (let i = 0; i < arr2.length; i++) {
      arr2[i] = new Array(3);
    }
    // 출력
    for (let i = 0; i < arr2.length; i++) {
      for (let j = 0; j < arr2[i].length; j++) {
        console.log(`${[i]},${[j]}`);
      }
    }
```

☞ 여기서 잠깐

- 2행 3열 배열을 만들고, 각 요소에 0 값을 넣기
- 출력 예시

```
0,0 : 0
0,1 : 0
0,2 : 0
1,0 : 0
1,1 : 0
1,2 : 0
```

```
// 2행 3열
    let arr2 = new Array(2);
    for (let i = 0; i < arr2.length; i++) {
      arr2[i] = new Array(3);
      for (let j = 0; j < arr2[i].length; j++) {
        arr2[i][j] = 0;
      }
    }
    // 출력
    for (let i = 0; i < arr2.length; i++) {
      for (let j = 0; j < arr2[i].length; j++) {
        console.log(`${[i]},${[j]} : ${arr2[i][j]}`);
      }
    }
```

☞ 여기서 잠깐

- 4행 2열 배열을 만들고, 좌표 출력하기

- 출력 예시

```
0,0
0,1
1,0
1,1
2,0
2,1
3,0
3,1
```

```
// 2차원 배열 함수 이용 생성하기
    function array2D(row, col) {
      let arr3 = new Array(row);
      for (let i = 0; i < arr3.length; i++) {
        arr3[i] = new Array(col);
      }
      return arr3;
    }
    let arr3 = array2D(4, 2);


    // 출력하기
    for (let i = 0; i < arr3.length; i++) {
      for (let j = 0; j < arr3[i].length; j++) {
        console.log(`${[i]},${[j]}`);
      }
    }
```

JavaScript

➢ ES6를 지원하는 최신 브라우저라면 사용 가능한 방법

```
// arr[6][7] (빈 배열 생성)
const arr1 = Array.from(Array(6), () => new Array(7));
```

- 자바스크립트의 2차원 배열은 1차원 배열에 또 다른 배열 객체를 추가하여 2차원 배열 만드는 방법을 사용한다. 자바스크립트 배열의 특성을 잘 모른다면 조금 이상한 방법으로 보일 수도 있다.

- 자바스크립트의 배열은 동적으로 배열의 크기를 조절할 수 있으며, 배열에는 모든 유형의 변수 그리고 함수, 객체도 담을 수 있어서 유연하게 사용할 수 있지만 그만큼 충분히 이해를 하고 사용해야 한다.

☞ 여기서 잠깐

- 6행 7열 배열을 만들고, 각 요소에 i * j 값을 넣기
- 출력 예시

```
▸ (7) ['0,0', '0,1', '0,2', '0,3', '0,4', '0,5', '0,6']
▸ (7) ['1,0', '1,1', '1,2', '1,3', '1,4', '1,5', '1,6']
▸ (7) ['2,0', '2,1', '2,2', '2,3', '2,4', '2,5', '2,6']
▸ (7) ['3,0', '3,1', '3,2', '3,3', '3,4', '3,5', '3,6']
▸ (7) ['4,0', '4,1', '4,2', '4,3', '4,4', '4,5', '4,6']
▸ (7) ['5,0', '5,1', '5,2', '5,3', '5,4', '5,5', '5,6']
```

```
// 6행 7열 2차원 배열 생성 (빈 배열)
// ES6를 지원하는 최신 브라우저
const arr1 = Array.from(Array(6), () => new Array(7));


// 값 채우기 예시: 행 번호 + 열 번호
for (let i = 0; i < arr1.length; i++) {
  for (let j = 0; j < arr1[i].length; j++) {
    arr1[i][j] = `${i},${j}`; // 각 요소에 "i,j" 문자열 저장
  }
}


// 콘솔에 출력
for (let i = 0; i < arr1.length; i++) {
  console.log(arr1[i]);
}
```

☞ 여기서 잠깐

- Fill( ) 메소드 이용하여 빈 배열에 값 채우기

Fill(value, start, end)

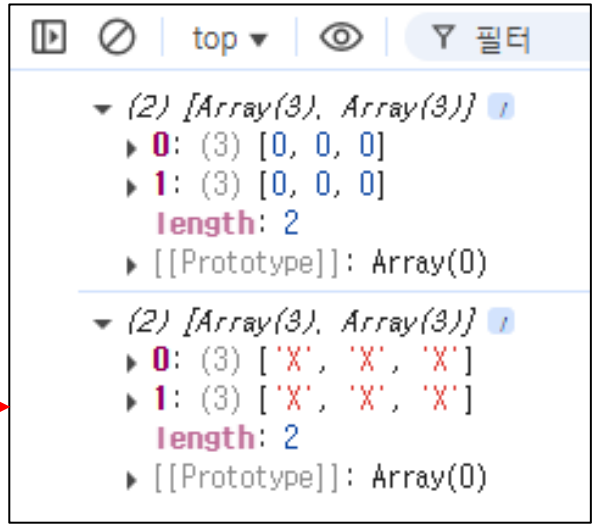
value : 배열을 채울 값

start : 채우기 시작할 인덱스, 기본값 :0

end : 채우기를 끝낼 인덱스 , 기본값 : 배열길이

```
const arr01 = Array.from(Array(2), () => new Array(3).fill(0));
console.log(arr01);

const arr02 = Array.from(Array(2), () => new Array(3).fill('X'));
console.log(arr02);
```



배열(Array) 에서 사용하는 반복문

JavaScript

➢ forEach() : 배열 순회 전용 메서드

용도: 배열의 각 요소를 한 번씩 순회하면서 함수를 실행.

배열 전용: 배열이 아니면 사용할 수 없음.

형식 • 배열이름.forEach((value, index, array){...반복 수행 코드...})

콜백함수의 매개변수로 value에 요소값, index에 인덱스, array에 원본 배열이 들어온다.

순서가 중요: 첫 번째 매개변수 = 요소, 두 번째 = 인덱스, 세 번째 = 원본 배열.

```
const arr02 =[10,20,30];
arr02.forEach((value, index, array)=>{
    console.log(`${index} : ${value}`); // 0 : 10, 1 : 20, 2: 30 출력
    console.log(`${array}`);
  })
```

```
0 : 10
10,20,30
1 : 20
10,20,30
2 : 30
10,20,30
```

JavaScript

➢ forEach() 간단한 예제

```
let arr09 = ['apple', 'banana', 'cherry'];


    arr09.forEach((value, index, array) => {
      console.log('value:', value); // 현재 요소
      console.log('index:', index); // 현재 인덱스
      console.log('array:', array); // 원본 배열
      console.log('---');
    });
```

```
value: apple
index: 0
array: ▸ (3) ['apple', 'banana', 'cherry']
---
value: banana
index: 1
array: ▸ (3) ['apple', 'banana', 'cherry']
---
value: cherry
index: 2
array: ▸ (3) ['apple', 'banana', 'cherry']
---
```

JavaScript

➢ 배열.map()

용도: 배열의 각 요소를 변형하고 새로운 배열을 반환.

배열 전용: 배열이 아니면 사용할 수 없음.

형식: • 배열.map(**(value, index, array))=>{...반복 수행 코드...}**

forEach와 동일(순회 방식, 콜백함수 매개변수 등)

다른점 => 각 콜백함수에서 return 하는 값들로 새로운 배열을 만들어 반환한다.

순서가 중요: 첫 번째 매개변수 = 요소, 두 번째 = 인덱스, 세 번째 = 원본 배열.

```
let arr = [1, 2, 3];
let doubled = arr.map((value, index, array) => {
  console.log(value);
  return value * 10;
});


console.log(doubled);
```

```
1
2
3
▸ (3) [10, 20, 30]
```

JavaScript

➤ 배열.map() 간단한 예제

순서가 중요: 첫 번째 매개변수 = 요소, 두 번째 = 인덱스, 세 번째 = 원본 배열.
Value는 출력하지 않고 index만 추출하려면 _, 또는 아무 이름이나 입력한다.

```
let arr10 = ['apple', 'banana', 'cherry'];

    let indices = arr10.map((_, index) => {
      console.log('index:', index); // index 출력
      return index; // map은 새 배열 반환
    });



    console.log('indices array:', indices);
```

```
index: 0
index: 1
index: 2
indices array: ▸ (3) [0, 1, 2]
```

2차원 배열

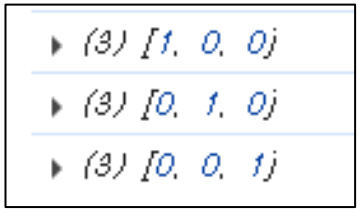
기초문제

연습문제

예제1] 3행 3열짜리 배열을 만들고, 대각선(왼쪽 위 → 오른쪽 아래) 요소만 1로 채운 후 출력하세요.

(단, Ex_arr01.html 파일로 작성)

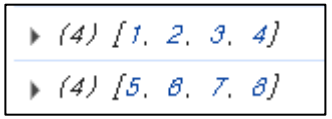
<출력결과>



예제2] 2행 4열짜리 배열을 만들고, 1부터 8까지 숫자를 순서대로 채운 후 출력하세요.

(단, Ex_arr02.html 파일로 작성)

<출력결과>

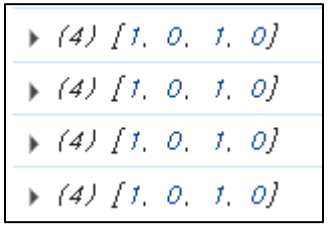


연습문제

예제3] 4행 4열 배열을 만들고, 짝수열은 0, 홀수열은 1로 채운 후 출력하세요.

(단, Ex_arr03.html 파일로 작성)

<출력결과>

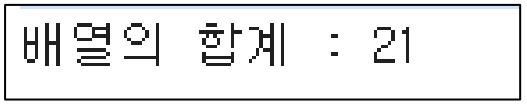


예제4] 2차원 배열의 합 구하기, 아래와 같은 배열이 있을 때, 전체 합을 구하세요.

(단, Ex_arr04.html 파일로 작성)

```
let arr06 = [
  [1, 2, 3],
  [4, 5, 6]
];
```

<출력결과>



연습문제

예제5] 2차원 배열이 아래와 같은 배열이 있을 때, 행별의 합을 구하세요.

(단, Ex_arr05.html 파일로 작성)

```
let arr06 = [
  [1, 2, 3],
  [4, 5, 6]
];
```

<출력결과>

```
배열의 행의합계 : 6
배열의 행의합계 : 15
```

예제6] 2차원 배열을 선언하고, 가장 큰 값을 찾아 출력하세요.

(단, Ex_arr06.html 파일로 작성)

```
let arr07 =[
  [3, 9, 2],
  [8, 7, 6],
  [1, 4, 5]
]
```

<출력결과>

```
최대값 : 9
```

배열 연습문제

실습문제1

예제] 아래 <조건>에 만족하도록 , HTML과 JavaScript를 작성하시오.

(단, Real_arr01.html 파일로 작성)

조건

좌석 배치표 만들기

설명: 5x5 2차원 배열을 만들어서 각 칸에 좌석 번호(1~25)를 넣고 출력하기.

요구사항: for문을 이용하여 배열 생성 및 출력.

```
let result = '<table border="1">';
```

<출력결과>

1. 좌석 배치표 (5x5)

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

실습문제2

예제] 아래 <조건>에 만족하도록 , HTML과 JavaScript를 작성하시오.

(단, Real_arr02.html 파일로 작성)

조건

성적표 만들기

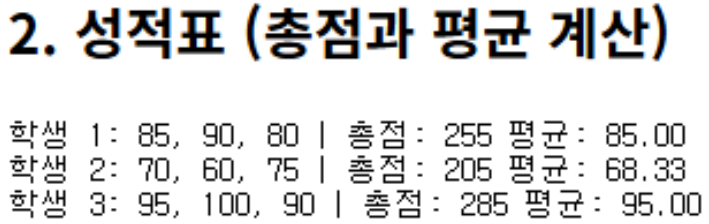
설명: 학생 3명의 국, 영, 수 점수를 2차원 배열에 저장하고, 각 학생의 총점과 평균 계산하기.

요구사항: 총점과 평균은 배열을 순회하여 계산. ForEach()문 이용

```
let scores = [
        [85, 90, 80],
        [70, 60, 75],
        [95, 100, 90],
      ];
```

toFixed(2) => 평균을 소수2자리까지 표시 하는 메서드

<출력결과>



실습문제3

예제] 아래 <조건>에 만족하도록 , HTML과 JavaScript를 작성하시오.

(단, Real_arr03.html 파일로 작성)

조건

달력 만들기

설명: 6x7 2차원 배열을 이용해 1~31일까지 달력 형태로 채우기.

요구사항:

let result = '<table border="1">';

const weekdays = ['일', '월', '화', '수', '목', '금', '토'];

<출력결과>

3. 달력 만들기 (6x7)

일	월	화	수	목	금	토
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

실습문제4

예제] 아래 <조건>에 만족하도록 , HTML과 JavaScript를 작성하시오.

(단, Real_arr04.html 파일로 작성)

조건

상품 재고 현황

설명: 3개의 카테고리(과일, 음료, 간식), 각 카테고리별 4개의 상품 수량을 2차원 배열에 저장하고 출력.

요구사항: 카테고리별 총 재고 계산. ForEach( ) 이용

```
Let stock = [
   [10, 5, 8, 12], // 과일
   [20, 15, 18, 25], // 음료
   [30, 25, 28, 15]  // 간식
 ];
```

<출력결과>

상품 재고 현황

카테고리1: 10,5,8,12 | 총재고: 35
카테고리2: 20,15,18,25 | 총재고: 113
카테고리3: 30,25,28,15 | 총재고: 211

실습문제5

예제] 아래 <조건>에 만족하도록 , HTML과 JavaScript를 작성하시오.

(단, Real_arr05.html 파일로 작성)

조건

성적 분석 프로그램 만들기

요구사항: 아래 주어진 배열 데이터를 이용하여 작성

```
let names = ['홍길동', '개나리', '진달래'];
let subjects = ['국어', '영어', '수학'];
let scores = [
  [85, 90, 78], // 홍길동
  [92, 88, 95], // 개나리
  [76, 85, 83], // 진달래
];
```

다음을 구하여 출력하시오.

1. 학생별 총점과 평균
2. 과목별 평균 (국어, 영어, 수학)
3. 최고 점수를 받은 학생의 이름과 점수

<출력결과>

성적 분석 프로그램

[학생별 총점과 평균]
홍길동 총점: 253, 평균: 84.3
개나리 총점: 275, 평균: 91.7
진달래 총점: 244, 평균: 81.3

[과목별 평균]
국어 평균: 84.3
영어 평균: 87.7
수학 평균: 85.3

[최고 점수 학생]
개나리 (95점)