

이지원

행력 계상기!

목차

01

행렬 계산기 소개

행렬 소개
덧셈, 뺄셈, 곱셈

3p

02

기획

스케줄
스케치 & 디자인
컨셉트

7p

03

작업 설명

계산기 설명
코드 설명

11p

04

후기

작업 후기

18p

헤헤



01

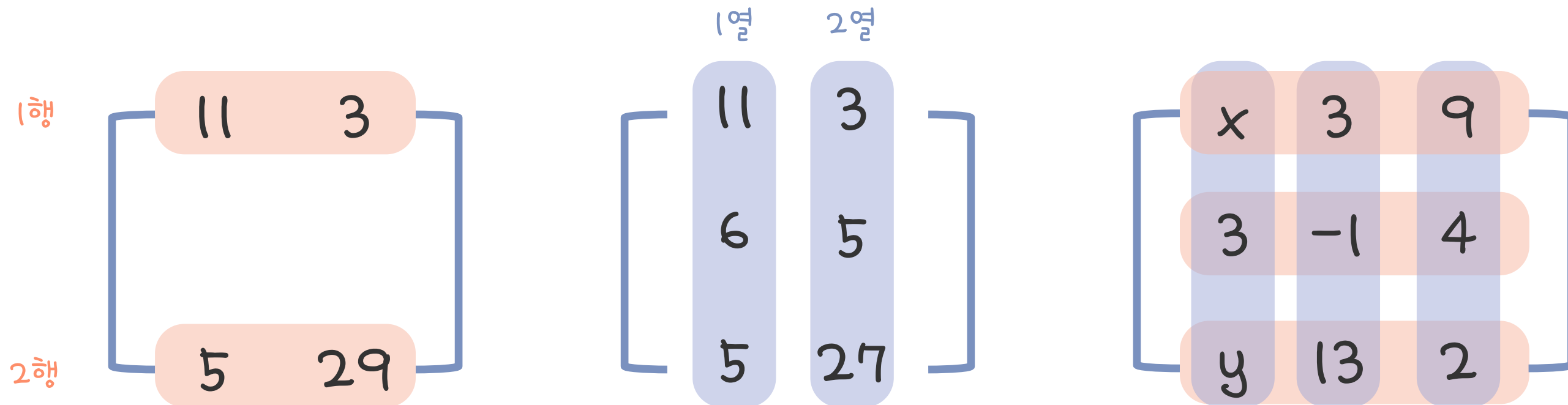
행렬 계산기 소개

행렬 소개
덧셈, 뺄셈, 곱셈



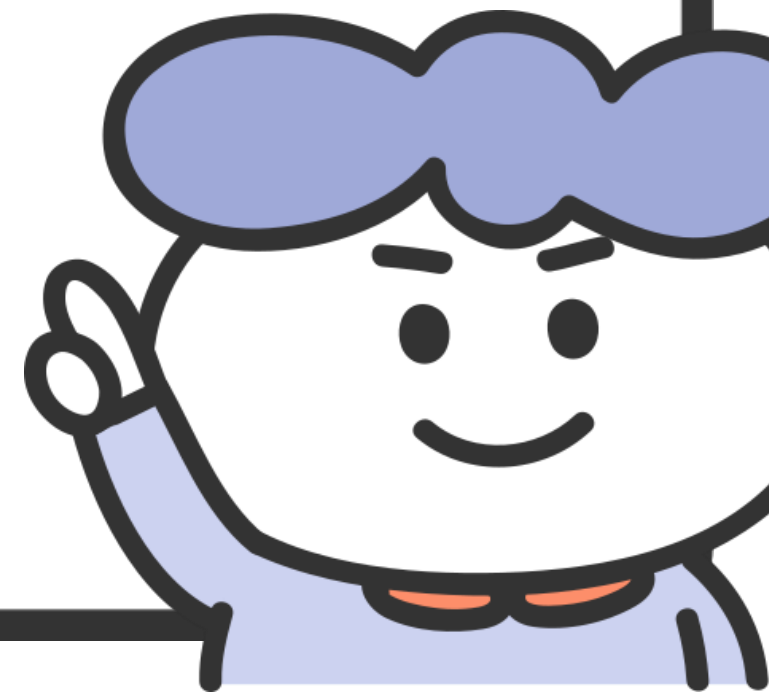
행렬 소개

+ 행렬의 생김새



행렬(Matrix)은 1개 이상의 수나 식을 사각형 모양으로 나열한 것을 의미

가로줄을 **행(row)**이라고 부르고, 세로줄을 **열(column)**이라고 부릅니다.



덧셈 뺄셈

A행렬

$$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}$$

A의 1,1

B행렬

$$\begin{bmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{bmatrix}$$

B의 1,1

결과 행렬

$$\begin{bmatrix} 10 & 12 & 14 \\ 16 & 18 & 20 \\ 22 & 24 & 26 \end{bmatrix}$$

결과 1,1

+

=

+ 덧셈, 뺄셈 계산 조건

행렬의 행과 행, 열과 열의 개수가 같아야 하고,
같은 동일한 위치 값을 가진 값을 더하거나 뺄 수 있음



곱셈

A행렬

$$\begin{bmatrix} 0 & 2 \\ 3 & 5 \\ 6 & 8 \end{bmatrix}$$

$1,1$ $1,2$

B행렬

$$\begin{bmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \end{bmatrix}$$

$1,1$ $2,1$

×

=

결과 행렬

$$\begin{bmatrix} 26 & 28 & 30 \\ 82 & 89 & 96 \\ 164 & 178 & 192 \end{bmatrix}$$

$1,1 = A_{1,1} \times B_{1,1} + A_{1,2} \times B_{2,1}$

+ 곱셈 계산 조건

A의 **열**과 B의 **행**이 같아야 함

결과 행렬의 크기는 A의 **행** x B의 **열**이 됨



기획

02

기획

스케줄
스케치 & 디자인
콘셉트



스케줄

[콘셉트 선정] × [틀 작업
HTML] × [javascript
구현]

8월 12일

콘셉트 선정 및 스케치 작업

8월 13일

HTML, CSS 작업

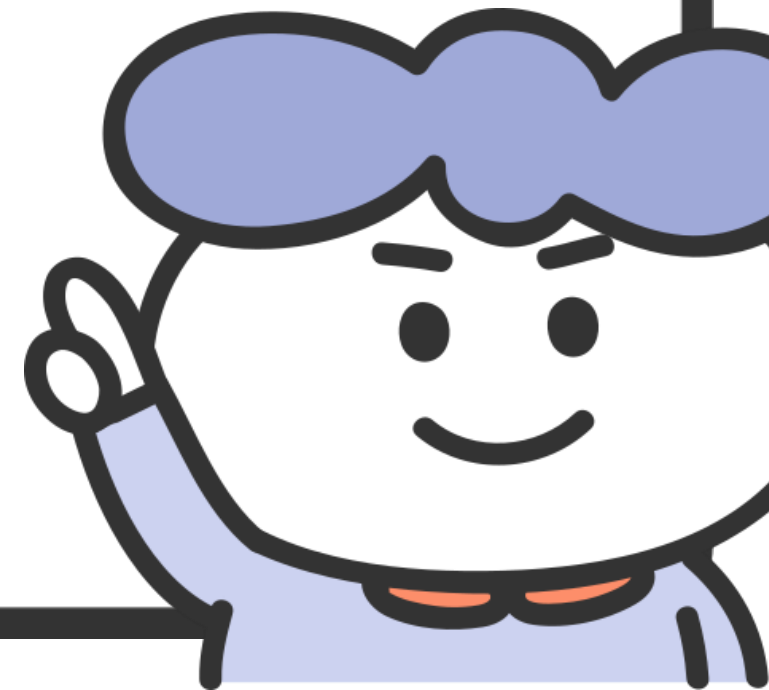
8월 14일 ~ 17일

jQuery, javascript로 기능 구현

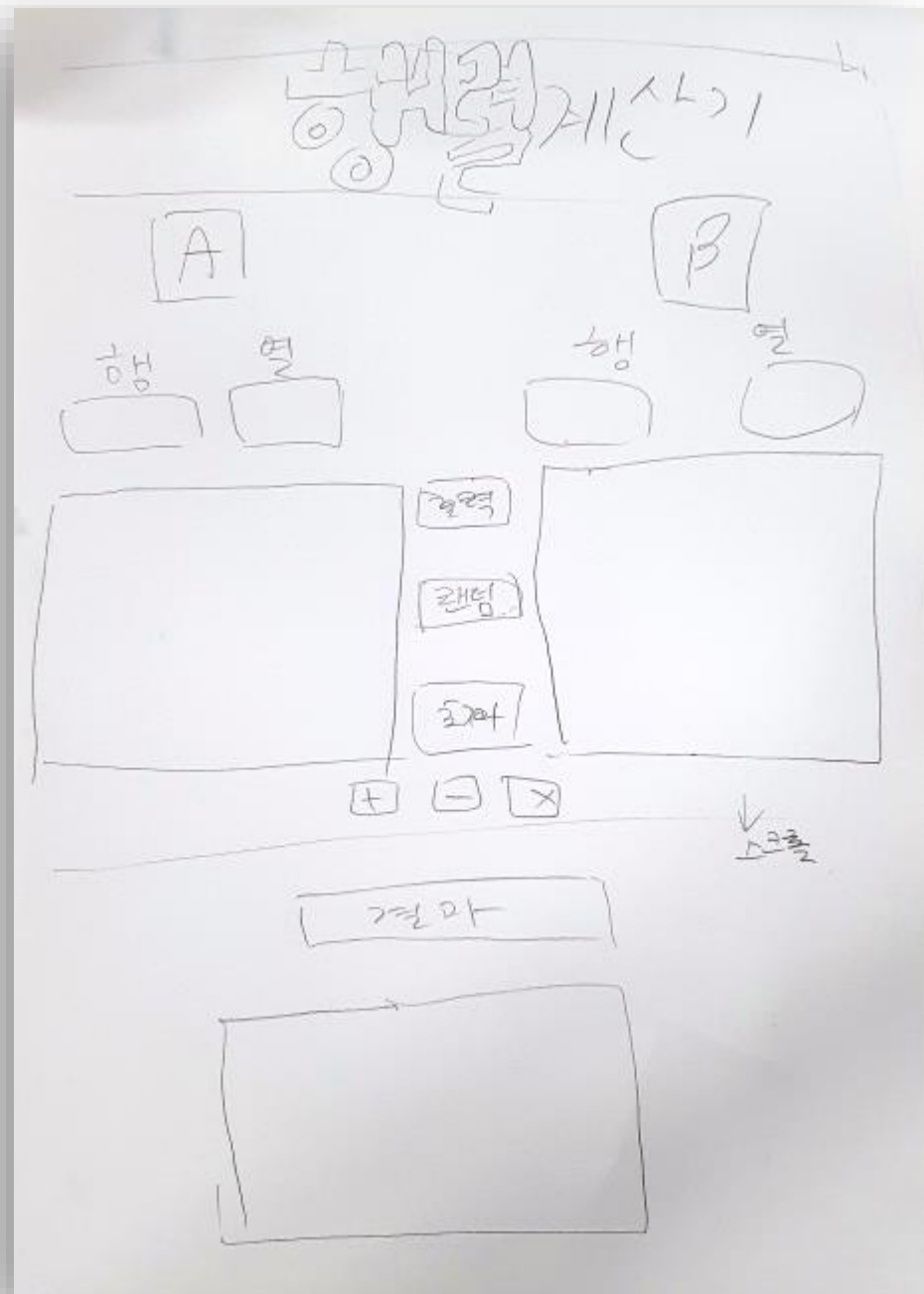
[PPT]

8월 18일

PPT 작업



스케치 & 디자인



콘셉트

01

02

03

장난감 같은 폰트

리치 와글와글

210 리치 망고

+ 특 특 튀는 색상

주황색: #ec573b

보라색: #7A91BF

아기자기한 모형

폰트와 색상 콘셉트에 맞춰

아기자기한 느낌을 주기 위해

전체적으로 동글동글한 느낌을 줌



03

작업 설명

계산기 설명
코드 설명



계산기 설명

행렬 계산기 I

A

행: 10 열: 10

4	-77	-90	77	-26	-51	-19	-53	-64	31
36	-52	22	53	-87	-34	-4	-20	-2	-16
64	-14	-45	-13	25	96	85	80	95	67
76	66	-30	-42	44	64	-16	-77	9	-42
21	5	-84	18	60	52	92	46	-54	12
94	39	56	-78	-70	-64	70	81	-95	28
65	26	-83	66	-57	28	49	2	23	11
-74	-100	99	46	-61	-86	-99	-1	-84	-21
-23	-64	-97	5	4	15	-49	-19	-94	62
93	8	-78	99	-80	-9	-90	-98	69	-10

B

행: 10 열: 10

-38	96	65	49	-88	89	-48	-68	25	76
76	37	-100	36	41	23	67	-78	-24	70
42	91	46	-58	99	81	64	5	14	-42
-30	-12	-5	6	46	98	-18	-99	-89	-22
90	15	78	-88	2	56	-32	-37	73	-35
-32	-39	63	81	20	54	92	94	54	-30
11	-80	-66	-17	93	-41	46	96	-84	7
25	48	-3	-70	-28	-33	-12	100	2	57
54	59	14	77	100	97	-85	10	51	35
-100	86	42	-90	54	5	46	70	-4	62

출력

랜덤

초기화

+ - ×

+ 출력

- + A의 행과 열, B의 행과 열 입력 후 출력을 누르면 행렬 박스가 생성
- + 1부터 10까지의 정수만 입력 가능

랜덤 +

행렬에 값이 랜덤으로 들어 감 +

값은 -100부터 100까지 들어 감 +

행렬 계산기 I

A

행: 10 열: 10

4	-77	-90	77	-26	-51	-19	-53	-64	31
36	-52	22	53	-87	-34	-4	-20	-2	-16
64	-14	-45	-13	25	96	85	80	95	67
76	66	-30	-42	44	64	-16	-77	9	-42
21	5	-84	18	60	52	92	46	-54	12
94	39	56	-78	-70	-64	70	81	-95	28
65	26	-83	66	-57	28	49	2	23	11
-74	-100	99	46	-61	-86	-99	-1	-84	-21
-23	-64	-97	5	4	15	-49	-19	-94	62
93	8	-78	99	-80	-9	-90	-98	69	-10

B

행: 10 열: 10

-38	96	65	49	-88	89	-48	-68	25	76
76	37	-100	36	41	23	67	-78	-24	70
42	91	46	-58	99	81	64	5	14	-42
-30	-12	-5	6	46	98	-18	-99	-89	-22
90	15	78	-88	2	56	-32	-37	73	-35
-32	-39	63	81	20	54	92	94	54	-30
11	-80	-66	-17	93	-41	46	96	-84	7
25	48	-3	-70	-28	-33	-12	100	2	57
54	59	14	77	100	97	-85	10	51	35
-100	86	42	-90	54	5	46	70	-4	62

출력

랜덤

초기화

+ - ×



계산기 설명

행렬 계산기

A

행: 10 열: 10

4	-77	-90	77	-26	-51	-19	-53	-64	31
36	-52	22	53	-87	-34	-4	-20	-2	-16
64	-14	-45	-13	25	96	85	80	95	67
76	66	-30	-42	44	64	-16	-77	9	-42
21	5	-84	18	60	52	92	46	-54	12
94	39	56	-78	-70	-64	70	81	-95	28
65	26	-83	66	-57	28	49	2	23	11
-74	-100	99	46	-61	-86	-99	-1	-84	-21
-23	-64	-97	5	4	15	-49	-19	-94	62
93	8	-78	99	-80	-9	-90	-98	69	-10

B

행: 10 열: 10

-38	96	65	49	-88	89	-48	-68	25	76
76	37	-100	36	41	23	67	-78	-24	70
42	91	46	-58	99	81	64	5	14	-42
-30	-12	-5	6	46	98	-18	-99	-89	-22
90	15	78	-88	2	56	-32	-37	73	-35
-32	-39	63	81	20	54	92	94	54	-30
11	-80	-66	-17	93	-41	46	96	-84	7
25	48	-3	-70	-28	-33	-12	100	2	57
54	59	14	77	100	97	-85	10	51	35
-100	86	42	-90	54	5	46	70	-4	62

출력

랜덤

초기화

+ - ×

+ 덧셈 뺄셈 버튼

- + A, B 행렬의 같은 위치의 값을 더하거나 뺌
- + A행, B행 or A열, B열의 값이 같아야 함

곱셈 버튼 +

A행의 값 x B의 열의 값을 곱함 +

A의 열 and B의 행의 값이 같아야 함 +

행렬 계산기

A

행: 10 열: 10

4	-77	-90	77	-26	-51	-19	-53	-64	31
36	-52	22	53	-87	-34	-4	-20	-2	-16
64	-14	-45	-13	25	96	85	80	95	67
76	66	-30	-42	44	64	-16	-77	9	-42
21	5	-84	18	60	52	92	46	-54	12
94	39	56	-78	-70	-64	70	81	-95	28
65	26	-83	66	-57	28	49	2	23	11
-74	-100	99	46	-61	-86	-99	-1	-84	-21
-23	-64	-97	5	4	15	-49	-19	-94	62
93	8	-78	99	-80	-9	-90	-98	69	-10

B

행: 10 열: 10

-38	96	65	49	-88	89	-48	-68	25	76
76	37	-100	36	41	23	67	-78	-24	70
42	91	46	-58	99	81	64	5	14	-42
-30	-12	-5	6	46	98	-18	-99	-89	-22
90	15	78	-88	2	56	-32	-37	73	-35
-32	-39	63	81	20	54	92	94	54	-30
11	-80	-66	-17	93	-41	46	96	-84	7
25	48	-3	-70	-28	-33	-12	100	2	57
54	59	14	77	100	97	-85	10	51	35
-100	86	42	-90	54	5	46	70	-4	62

출력

랜덤

초기화

+ - ×



계산기 유효성



출력 버튼

1부터 10까지
숫자만 입력 가능



연산 버튼

행렬 안의 값은
-100부터 100까지
정수만 입력 가능



+ - 버튼

행과 행, 열과 B열이
같아야 함



x 버튼

A의 열과 B의 행이
같아야 함

코드 설명 - 출력 버튼

```

1  $(function(){
2      $(':button').on('click',function(e){
3          switch(e.target.id){
4              case 'outputBtn':
5              case 'randomBtn':
6                  let btnNameFucOR = new btnNameFuc(e.target.id);
7                  btnNameFucOR.outBtnFuc();
8                  break;
9              case 'clearBtn':
10                 let btnNameFucCL = new btnNameFuc(e.target.id);
11                 btnNameFucCL.clBtnFuc();
12                 break;
13                 case 'plusBtn':
14                 case 'minusBtn':
15                     let btnNameFucPM = new btnNameFuc(e.target.id);
16                     btnNameFucPM.pmBtnFuc();
17                     break;
18                     case 'multiplyBtn' :
19                         let btnNameFucMU = new btnNameFuc(e.target.id);
20                         btnNameFucMU.muBtnFuc();
21                         break;
22                 }
23             })
24         });

```

```

1  //출력, 랜덤
2  this.outBtnFuc = function(){
3      let aRowNum = $('#aRowNum').val();
4      let aColumnNum = $('#aColumnNum').val();
5      let bRowNum = $('#bRowNum').val();
6      let bColumnNum = $('#bColumnNum').val();
7      if( !validationClick('aRowNum') || !validationClick('aColumnNum') ||
8          !validationClick('bRowNum') || !validationClick('bColumnNum') ){
9          return;
10     }
11     //a 행렬 2차원 배열 생성
12     aRowColArray = [];
13     for(let i = 0; i<aRowNum; i++){
14         let aMatrix = [];
15         for(let j = 0; j<aColumnNum; j++){
16             aMatrix.push(0);
17         }
18         aRowColArray.push(aMatrix);
19     }
20     //랜덤 버튼일 경우 2차원 배열에 랜덤 값 뿌려줌
21     if(anybtn == "randomBtn"){
22         for(let i=0; i < aRowColArray.length; i++){
23             for(let j=0; j<aRowColArray[i].length; j++){
24                 let randomNum = Math.floor(Math.random() * 201) + -100;
25                 do{
26                     randomNum = Math.floor(Math.random() * 201) + -100;
27                 }
28                 while(randomNum==0 );
29                 aRowColArray[i][j] = randomNum;
30             }
31         }
32         1 drawFirst();
33         2 drawSecond();

```

+ 버튼에 따른 함수 호출

- + A, B의 각 행, 열 값을 받아온 후 2차원 배열로 생성
- + drawFirst(), drawSecond()를 호출하여, 행렬 값을 입력할 input 박스를 띄움

코드 설명 - 출력 버튼

```
1 //A 행 열 그리는 함수
2 function drawFirst() {
3     let tempStringA = "";
4     for(let i =0; i < aRowColArray.length; i++){
5         for(let j =0; j<aRowColArray[i].length; j++){
6             tempStringA += '<input type="text" ';
7             tempStringA += ' class="arrayInNum" ';
8             tempStringA += ' value="" + aRowColArray[i][j] + ''';
9             tempStringA += ` oninput="this.value = this.value.replace(/^[^0-9]/g,'')";`
10            tempStringA += ` onKeyUp="changeFirst(this, ${i}, ${j});"`; //this는 value
11            tempStringA += ' id="';
12            tempStringA += 'a_' + i + '_' + j;
13            tempStringA += '>';
14        }
15        tempStringA += '<br>';
16    }
17    $("#a_matrix_input_box").html(tempStringA);
18 }
```

```
1 // 숫자가 바뀔 때 마다 값을 A행렬 배열에 값을 넣는 함수
2 function changeFirst(param, i, j) {
3     aRowColArray[i][j] = parseInt($(param).val());
4 }
```

+ 값 대입, 출력

+ 입력한 행, 열의 값으로 박스를 그리는 함수

+ onKeyUp를 사용하여 값이 바뀔 때 마다 changeFirst를 호출

+ changeFirst()

매개변수로 입력한 값과 i, j를 받아와서 생성한
2차원 배열에 [i][j]에 매개변수로 받은 param을 넣어줌

코드 설명 - 연산 버튼

```
1 //플러스, 마이너스 버튼
2 this.pmBtnFuc = function(){
3     if( !validationClick('aRowNum') || !validationClick('aColumnNum') || !validationClick('bRowNum') || !validationClick('bColumnNum')){
4         return;
5     }
6     if(aRowColArray.length != bRowColArray.length || aRowColArray[0].length != bRowColArray[0].length){
7         popUpEvent('A와 B의', ' 행렬이 같지 않음');
8         return;
9     }
10    if(!validation(aRowColArray, 'A') || !validation(bRowColArray, 'B')) {
11        return;
12    }
13    resultRowColArray = [];
14    for(let i=0; i < aRowColArray.length; i++){
15        let resultMatrix = [];
16        let resultN = 0;
17        for(let j =0; j < aRowColArray[i].length; j++){
18            if(anybtn == "plusBtn"){
19                resultN = (aRowColArray[i][j]) + (bRowColArray[i][j]);
20            }else{
21                resultN = (aRowColArray[i][j]) - (bRowColArray[i][j]);
22            }
23            resultMatrix.push(resultN);
24        }
25        resultRowColArray.push(resultMatrix);
26    }
27    drawResult();
28    scrollDisplayfuc();
29 }
```

+ 덧셈, 뺄셈

+ 연산한 값을 결과배열에 넣고 drawResult 호출

+ If로 눌린 버튼에 따라 연산을 다르게 함

+ drawResult()
결과 배열에 담긴 값을 그리는 함수

코드 설명 - 유효성 검사

+ 유효성 검사

- + 유효성 검사를 한 후 popupEvent를 호출
- + 매개변수 rowColVal, flag_Text를 받아서 팝업 창에 매개변수 값을 띄움

```

14 // A행렬, B행렬 값 유효성
15 function validation(arrParam, nameText){
16     let temp_AFlag = true;
17     let regExp100 = /^(-100)$|^(-0)$|^[-]?[1-9][0-9]?$|^100$/; //-100~100 유효성
18
19     //행렬의 유효성
20     $.each (arrParam, function (index, item) {
21         $.each(arrParam[index],function(i,it){
22             if(!regExp100.test(it)){
23                 temp_AFlag = false;
24             }
25         });
26     });
27
28     if(temp_AFlag == false){
29         let flag_Text = ' 행렬 숫자 -100~100까지 입력해 주세요.';
30         popUpEvent(nameText, flag_Text)
31     }
32     return temp_AFlag;
33 }

```

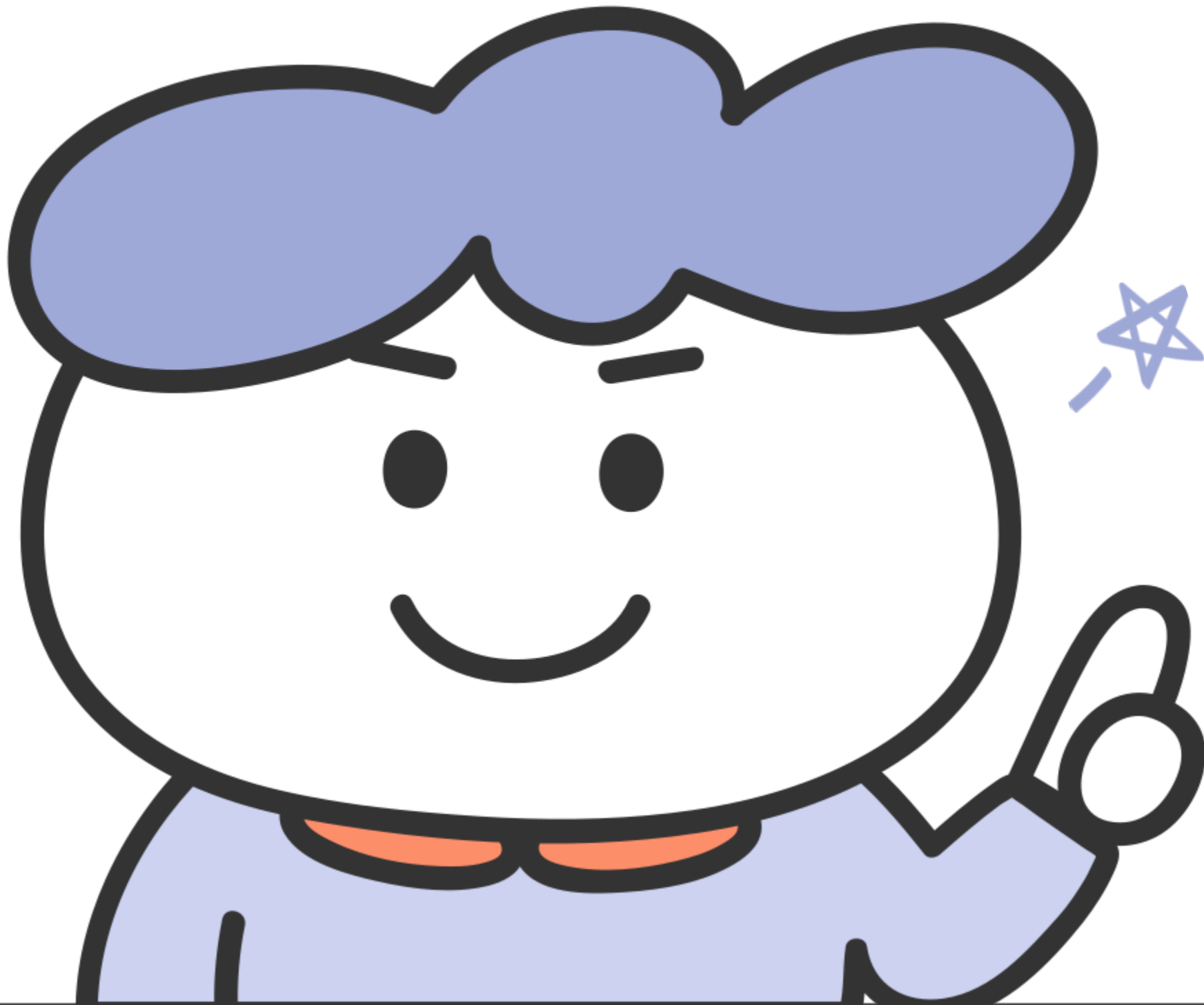
```

13
14
15
16
17
18
19
20
21
22
function popUpEvent(rowColVal, flag_Text){
    if(rowColVal == 'aRowNum' || rowColVal == 'aColumnNum' || rowColVal == 'bRowNum' || rowColVal == 'bColumnNum'){
        switch(rowColVal){
            case 'aRowNum' :
                rowColVal = 'A의 행';
                break;
            case 'aColumnNum' :
                rowColVal = 'A의 열';
                break;
            case 'bRowNum' :
                rowColVal = 'B의 행';
                break;
            case 'bColumnNum' :
                rowColVal = 'B의 열';
                break;
        }
        $('.popup_string').html(rowColVal + flag_Text);
    }else{
        $('.popup_string').html(rowColVal + flag_Text);
    }
    $('.popup_wrap').css("display","block");
}

```

04

후기





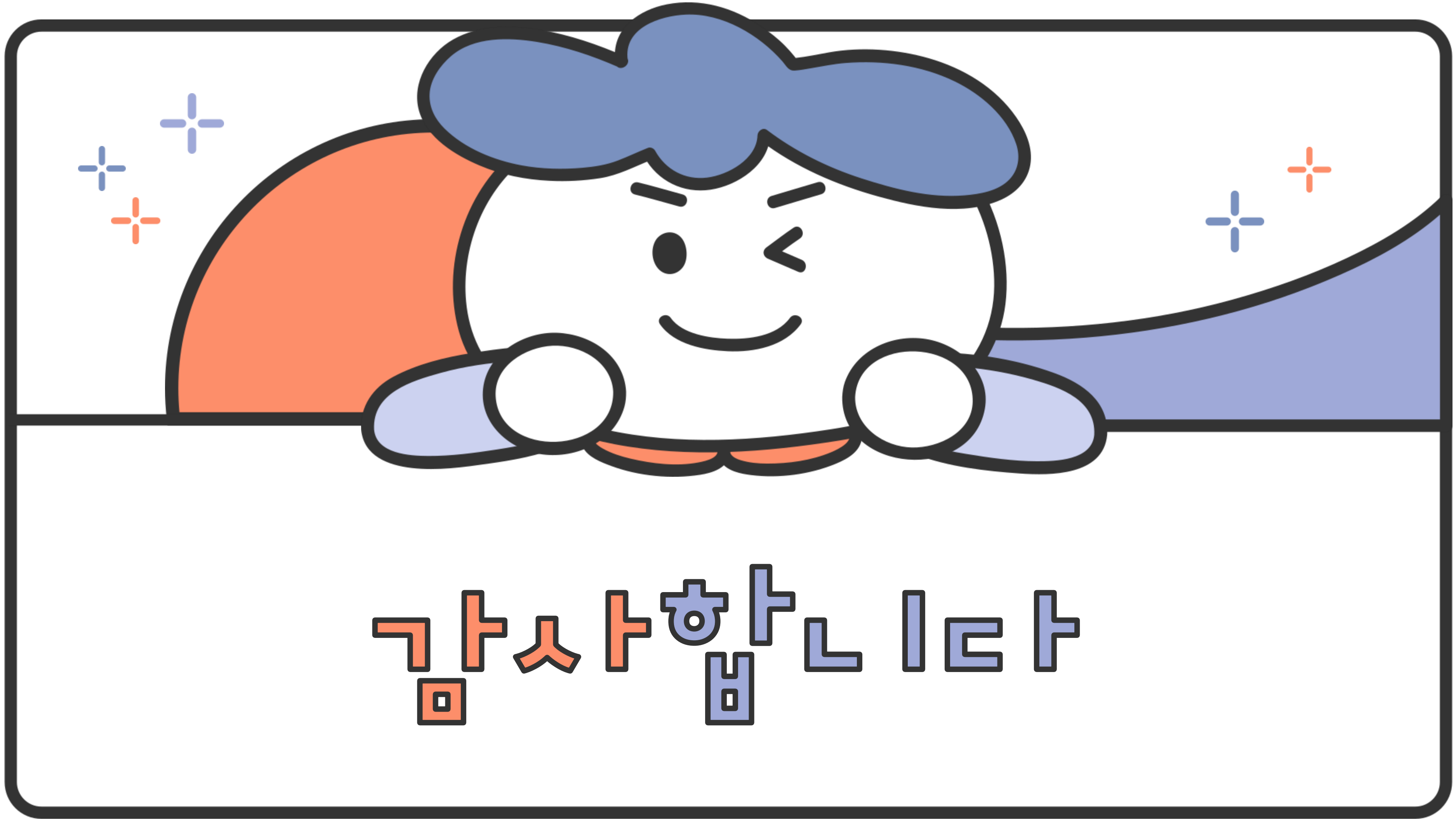
후기



저는 자바스크립트의 반복문과 함수에 대한 이해가 부족했습니다.
하지만 이 프로젝트를 통해 코드를 여러 번 수정하고 다양한 종류의 반복문과 함수를 사용하면서,
로직에 대해 많이 고민하고 학습하게 되었습니다.

이 프로젝트는 제 실력을 성장시키는 큰 도약의 발판이 되었으며,
앞으로 다른 일에도 겁먹지 않고 도전할 수 있는 용기를 얻었습니다.

감사합니다.



가시하리다