

A+

please

Made By 서영윤

행렬 계산기

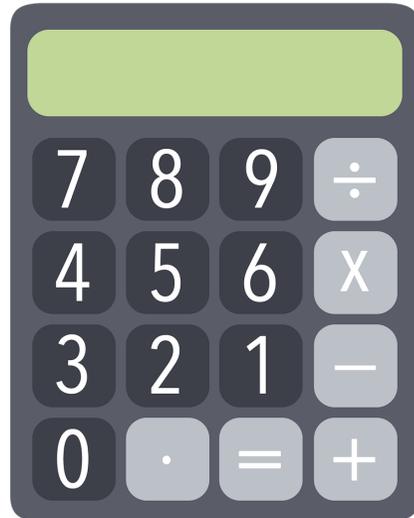
JAVA기반 스마트웹 디지털컨버전스
라인컴퓨터아트학원

[보러가기](#)

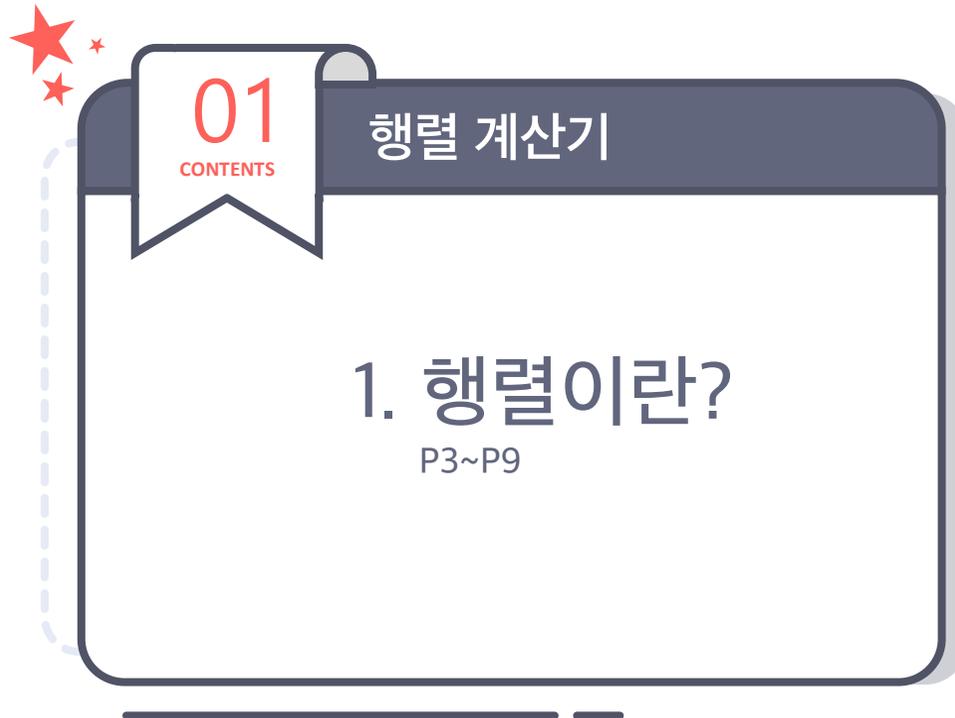
00

CONTENTS

행렬 계산기



- 행렬이란
(P3~P9)
- 스케치
(P10~P13)
- HTML
(P14~P16)
- JavaScript
(P17~P25)
- Sass
(P26~P28)
- 후기
(P29)



행렬이란

행렬의 모양

$$\begin{pmatrix} 17 & -29 \\ 3 & 65 \end{pmatrix}, \quad \begin{pmatrix} 22 & 37 & 87 \\ 1 & 55 & 66 \end{pmatrix}, \quad \begin{pmatrix} 66 & 23 \\ 9 & 61 \\ 0 & 28 \end{pmatrix}, \quad (7 \quad 11 \quad 66)$$

$$\begin{array}{l} \text{1행} \\ \text{2행} \end{array} \begin{pmatrix} 22 & 37 & 87 \\ 1 & 55 & 66 \end{pmatrix}$$

$$\begin{array}{l} \text{1열} \\ \text{2열} \end{array} \begin{pmatrix} 66 & 23 \\ 9 & 61 \\ 0 & 28 \end{pmatrix}$$

행렬의 모양

$$\begin{array}{cccc} a_{00} & a_{01} & a_{02} & a_{0j} \\ a_{10} & a_{11} & a_{12} \cdots & a_{1j} \\ a_{20} & a_{21} & a_{22} & a_{2j} \\ \cdots & \cdots & \cdots & \cdots \\ a_{i0} & a_{i1} & a_{i2} \cdots & a_{ij} \end{array} \left. \vphantom{\begin{array}{cccc} a_{00} & a_{01} & a_{02} & a_{0j} \\ a_{10} & a_{11} & a_{12} \cdots & a_{1j} \\ a_{20} & a_{21} & a_{22} & a_{2j} \\ \cdots & \cdots & \cdots & \cdots \\ a_{i0} & a_{i1} & a_{i2} \cdots & a_{ij} \end{array}} \right\} \begin{array}{l} (i, j)\text{형} \\ \text{행렬} \end{array}$$

행렬의 모양

$$A = \begin{pmatrix} 22 & 37 & 87 \\ 1 & 55 & 66 \end{pmatrix}$$

$$B = \begin{pmatrix} 66 & 23 \\ 9 & 61 \\ 0 & 28 \end{pmatrix}$$

$$A + B = X$$

※ 계산이 불가능합니다.

A와 B의 행과 열의 개수가 같아야 합니다.

행렬의 덧셈/뺄셈

$$\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \pm \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix} = \begin{pmatrix} a_{00} \pm a_{01} & b_{00} \pm b_{01} \\ a_{10} \pm a_{11} & b_{10} \pm b_{11} \end{pmatrix}$$

$$(a_{ij}) \pm (b_{ij}) = (a_{ij} \pm b_{ij})$$

행렬의 곱셈

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{pmatrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{pmatrix}$$

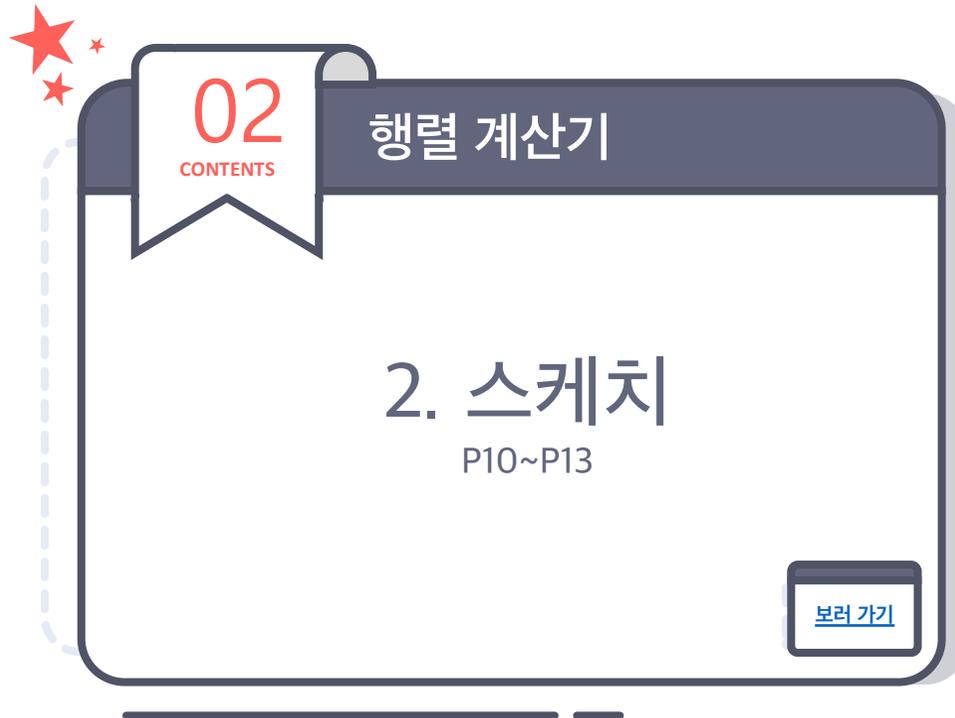
행렬의 곱셈

$$\begin{array}{c} \text{A 배열} \\ \left(\begin{array}{cc} a & b \\ c & d \end{array} \right) \end{array} \times \begin{array}{c} \text{B 배열} \\ \left(\begin{array}{cc} e & f \\ g & h \end{array} \right) \end{array} = \begin{array}{cc} ae + bg & af + bh \\ ce + dg & cf + dh \end{array}$$

A 배열 1 행 × B 배열 1 열을 각각 곱하고
나오는 값을 더하여 계산합니다.

$$\begin{array}{c} \text{A 배열} \\ \left(\begin{array}{cc} a & b \\ c & d \end{array} \right) \end{array} \times \begin{array}{c} \text{B 배열} \\ \left(\begin{array}{cc} e & f \\ g & h \end{array} \right) \end{array} = \begin{array}{cc} ae + bg & af + bh \\ ce + dg & cf + dh \end{array}$$

A 배열 1 행 × B 배열 2 열을 각각 곱하고
나오는 값을 더하여 계산합니다.



스케치

스케치

[보러가기](#)

행	열	자동완성	직접입력	더하기	빼기	곱하기	Reset	행	열
A 배열		결과배열						B 배열	

A배열의 행과 열을
개수를 지정합니다.

자동완성: 입력된 행과 열의 개수에 임의의 값을 넣습니다.
 직접입력: 입력된 행과 열의 개수에 0을 집어 넣습니다.
 더하기: 생성된 A 와 B 배열의 값을 더합니다.
 빼기: 생성된 A 와 B 배열의 값을 뺍니다.
 곱하기: 생성된 A 와 B 배열의 값을 곱합니다.
 Reset: A / B 배열과 출력된 값을 초기화 합니다.

B배열의 행과 열을
개수를 지정합니다.

스케치

[보러가기](#)

A배열

54	51	92	4	54	17	22	55	17	83	73	60
42	40	61	66	30	0	27	38	83	48	29	23
31	84	95	35	50	27	54	11	71	21	96	47
72	71	74	91	89	54	26	74	15	57	81	3
49	39	82	4	49	55	11	9	8	54	15	85
11	4	20	28	94	56	20	90	66	43	68	71
26	64	86	67	63	61	7	91	14	83	3	47
11	22	78	91	64	70	12	27	99	24	70	52
10	56	76	77	61	67	48	34	1	58	63	9
8	61	62	86	64	25	61	38	44	0	19	5
99	5	65	83	57	93	77	38	98	78	68	43
28	56	87	22	50	20	41	15	37	42	10	66

↑

A배열의 행과 열을 생성합니다.

결과배열

122	120	102	67	128	36	109	110	36	177	78	95
65	75	109	119	120	54	114	105	138	59	72	44
114	93	186	94	82	48	107	39	90	74	112	91
160	100	133	124	171	77	44	104	110	142	171	95
127	69	172	92	75	110	70	67	26	145	59	92
31	55	77	44	110	143	61	150	149	49	139	102
61	141	95	157	75	127	53	177	101	163	96	49
27	37	112	101	97	153	95	54	106	89	71	114
30	153	170	114	122	94	122	67	90	110	154	46
65	62	161	101	123	112	62	88	58	32	75	88
194	43	77	138	86	140	90	114	108	141	101	123
61	87	180	60	115	31	130	22	72	99	25	162

↑

버튼에 따라 더하기, 빼기, 곱하기, 초기화가 진행 되는 결과를 표시합니다.

B배열

68	69	10	63	74	19	87	55	19	94	5	35
23	35	48	53	90	54	87	67	55	11	43	21
83	9	91	59	32	21	53	28	19	53	16	44
88	29	59	33	82	23	18	30	95	85	90	92
78	30	90	88	26	55	59	58	18	91	44	7
20	51	57	16	16	87	41	60	83	6	71	31
35	77	9	90	12	66	46	86	87	80	93	2
16	15	34	10	33	83	83	27	7	65	1	62
20	97	94	37	61	27	74	33	89	52	91	37
57	1	99	15	59	87	1	50	14	32	56	83
95	38	12	55	29	47	13	76	10	63	33	80
33	31	93	38	65	11	89	7	35	57	15	96

↑

B배열의 행과 열을 생성합니다.

스케치

스케치

[보러가기](#)

4 2 자동완성 직접입력 더하기 빼기 곱하기 Reset 2 4

A 배열 결과배열 B 배열

49	66	18	77	55	38
65	14	53	62	82	54
93	11				
81	52				

좌측 행과 우측 열
좌측 열과 우측 행이 같아야합니다.
돌아가기

덧셈 / 뺄셈 시 숫자가 잘못 입력 되면 경고문을 표시합니다.

1 2 자동완성 직접입력 더하기 빼기 곱하기 Reset 3 4

A 배열 결과배열 B 배열

34	7	65	43	80	71
		19	10	42	60
		20	71	46	51

좌우 행렬의
행과열이 서로 같아야 합니다.
돌아가기

곱셈 시 숫자가 잘못 입력 되면 경고문을 표시합니다.



HTML

[보러 가기](#)

```

1 <div class="left_input_area">
2   <input id="x_array_l" class="input_design" type="number" min="1" max="12" placeholder="행">
3   <input id="y_array_l" class="input_design" type="number" min="1" max="12" placeholder="열">
4 </div>
5 <div class="button_area">
6   <div id="button_random" class="button_design">자동완성</div>
7   <div id="button_empty" class="button_design">직접입력</div>
8   <div id="button_plus" class="button_design">더하기</div>
9   <div id="button_minus" class="button_design">빼기</div>
10  <div id="button_multi" class="button_design">곱하기</div>
11  <div id="button_reset" class="button_design">Reset</div>
12 </div>
13 <div class="right_input_area">
14  <input id="x_array_r" class="input_design" type="number" min="1" max="12" placeholder="행">
15  <input id="y_array_r" class="input_design" type="number" min="1" max="12" placeholder="열">
16 </div>

```

→ 좌측 행과 열의 값을 입력합니다.

→ 자동완성, 직접입력, 더하기, 빼기, 곱하기, 초기화 버튼입니다.

→ 우측 행과 열의 값을 입력합니다.

HTML

[보러 가기](#)

```
1 <p class="explain_ment">A 배열</p>
2 <p class="explain_ment">결과배열</p>
3 <p class="explain_ment">B 배열</p>
4 <div class="print_div">
5     <div id="print_area_l" class="print_area_l"></div>
6 </div>
7 <div class="print_div">
8     <div id="print_result" class="print_result"></div>
9 </div>
10 <div class="print_div">
11     <div id="print_area_r" class="print_area_r"></div>
12 </div>
13 </div>
14 <div id="wrong_box_t" class="alert_div">좌측 행과 우측 열<br>좌측 열과 우측 행이 같아야합니다.
15     <div id="back_plus_button" class="back_button">돌아가기
16 </div>
17 </div>
18 <div id="wrong_box_m" class="alert_div">좌우 행렬의<br>행과열이 서로 같아야 합니다.
19     <div id="back_multi_button" class="back_button">돌아가기</div>
20 </div>
```

결과 값을 출력하는 곳입니다.

경고문구입니다.



JavaScript

[보러가기](#)

```

1 class generate_matrix{
2   constructor(id_num, x_array, y_array){
3     this.x_array = x_array;
4     this.y_array = y_array;
5     this.id_num = id_num;
6     this.return_left = this.array_left_bank();
7     this.return_right = this.array_right_bank();
8     this.check_comma = this.add_comma();
9   };

```

Class로 객체를 만들고 부여할 ID값, 행의 개수, 열의 개수를 매개 변수로 가져 옵니다. 추후 값을 저장할 함수, 숫자에 콤마(,)를 찍을 함수를 지정합니다.

다중 배열을 만들고 랜덤 값으로 A/B행렬을 생성합니다.

```

1 make_random_number(){
2   let empty_array = [];
3   let empty_string = "";
4   let i=0;
5   let j=0;
6   while(i<this.x_array) {
7     empty_array.push([]);
8     while(j<this.y_array){
9       empty_array[i][j] = Math.floor(Math.random()*(99-0+1)+0);
10      empty_string += '<input id="'+this.id_num+i+'_'+j+'" class="each_cell" type="number" value="'+Number(empty_array[i][j])+'>';
11      j++;
12    };
13    j=0;
14    i++;
15  };
16  i=0;
17  return empty_string;

```

JavaScript

[보러 가기](#)

```

1  make_empty_number(){
2    let empty_array = [];
3    let empty_string = "";
4    let i=0;
5    let j=0;
6    while(i<this.x_array) {
7      empty_array.push([]);
8      while(j<this.y_array){
9        empty_array[i][j] = 0;
10       [ empty_string += <input id="`+this.id_num+i+`_`+j+`" class="each_cell" type="number" value="+Number(empty_array[i][j])+`">;];
11       j++;
12     };
13     j=0;
14     i++;
15   };
16   i=0;
17   return empty_string;

```

다중 배열을 만들고 직접 입력에 사용할 A/B행렬을 생성합니다.

생성된 값을 지정된 함수에 담습니다.

```

1  array_left_bank = () => {
2    let empty_left_bank = [];
3    let div_string = "";
4    let i=0;
5    let j=0;
6    while(i<this.x_array){
7      empty_left_bank.push([]);
8      while(j<this.y_array){
9        empty_left_bank[i][j]=`${this.id_num+i+`_`+j}.val();
10       j++;
11     };
12     j=0;
13     i++;
14   };
15   return empty_left_bank;
16 };

```

JavaScript

[보러 가기](#)

```

1 plus_two = (arg1, arg2) => {
2   let i=0;
3   let j=0;
4   let part_sum = "";
5   let array_string = "";
6   if( arg1.length == arg2.length && arg1[0].length == arg2[0].length){
7     while(i<this.x_array){
8       while(j<this.x_array){
9         part_sum = parseInt(arg1[i][j]) + parseInt(arg2[i][j]);
10        array_string += '<div class="result_cell">'+part_sum+ '</div>';
11        j++;
12      };
13      j=0;
14      i++;
15    };
16    return array_string;
17  } else {
18    $("#wrong_box_t").css("display","block");
19  };
20 };

```

지정된 함수에서 값을 가져와 빼기를 합니다.
조건문으로 행/열의 개수가 공식에 부합되지 않을 경우
경고문을 띄웁니다.

지정된 함수에서 값을 가져와 더하기를 합니다.
조건문으로 받은 행/열의 개수가 공식에 부합되지
않을 경우 경고문을 띄웁니다.

```

1 minus_two = (arg1, arg2) => {
2   let i=0;
3   let j=0;
4   let part_sum = "";
5   let array_string = "";
6   if( arg1.length == arg2.length && arg1[0].length == arg2[0].length){
7     while(i<this.x_array){
8       while(j<this.x_array){
9         part_sum = parseInt(arg1[i][j]) - parseInt(arg2[i][j]);
10        array_string += '<div class="result_cell">'+part_sum+ '</div>';
11        j++;
12      };
13      j=0;
14      i++;
15    };
16    return array_string;
17  } else {
18    $("#wrong_box_t").css("display","block");
19  };
20 };

```

JavaScript

[보러가기](#)

```
1 multi_two = (arg1, arg2) => {
2     let i=0;
3     let j=0;
4     let k=0;
5     let part_sum = 0;
6     let array_string = "";
7     if( arg1.length == arg2[0].length && arg1[0].length == arg2.length){
8         while(i<arg1.length){
9             while(j<arg1.length){
10                while(k<arg1[0].length){
11                    part_sum += parseInt(arg1[i][k]) * parseInt(arg2[k][j]);
12                    k++;
13                };
14                array_string += '<div class="result_cell">' + this.add_comma(part_sum) + '</div>';
15                part_sum=0;
16                k=0;
17                j++;
18            };
19            j=0;
20            i++;
21        };
22        return array_string;
23    } else {
24        $("#wrong_box_m").css("display", "block");
25    };
26 };
```

3중 반복문을 활용하여 저장된 값을 더합니다.
조건문으로 받은 행/열의 개수가 공식에 부합
되지 않을 경우 경고문을 띄웁니다.

JavaScript

[보러가기](#)

```
1 add_comma(arg){
2   let comma_length= 1;
3   let comma_array="";
4   let without_comma="";
5   let result_number=String(arg);
6   let string_revers="";
7   let i=0;
8   while(i<result_number.length){
9     if(result_number != ","){
10      without_comma += result_number[i];
11    };
12    i++;
13  };
```

```
1   i=without_comma.length-1;
2   while(i>=0){
3     comma_array+=without_comma[i];
4     if(comma_length%3 == 0 && i !=0){
5       comma_array+=",";
6     };
7     comma_length++;
8     i--;
9   };
10  i=comma_array.length-1;
11  while(i>=0){
12    string_revers+=(comma_array[i]);
13    i--;
14  };
15  return string_revers;
16  };
```

계산된 값이 천 단위가 넘어가면 콤마(,)를 찍습니다.

JavaScript

[보러 가기](#)

```

1 case 'button_random':
2   $("#wrong_box_t").css("display", "none");
3   $("#wrong_box_m").css("display", "none");
4   make_left_array = new generate_matrix('each_idl', $("#x_array_l").val(), $("#y_array_l").val());
5   make_right_array = new generate_matrix('each_idr', $("#x_array_r").val(), $("#y_array_r").val());
6   $(".print_area_l").css("width", 50*$("#y_array_l").val()+ 'px').css("height", 50*$("#x_array_l").val()+ 'px');
7   $(".print_area_r").css("width", 50*$("#y_array_r").val()+ 'px').css("height", 50*$("#x_array_r").val()+ 'px');
8   $("#print_area_l").html(make_left_array.make_random_number());
9   $("#print_area_r").html(make_right_array.make_random_number());
10  break;

```

버튼을 누르면 A / B 배열 생성을 위해 객체를 복사하고 ID / 행의 개수 / 열의 개수를 인자로 하여 값이 랜덤(위) / 직접입력(아래)인 행렬을 만듭니다.

결과가 출력되는 곳의 크기를 조절하기 위해 행 / 열의 개수로 높이와 너비를 결정합니다.

```

1 case 'button_empty':
2   $("#wrong_box_t").css("display", "none");
3   $("#wrong_box_m").css("display", "none");
4   make_left_array = new generate_matrix('each_idl', $("#x_array_l").val(), $("#y_array_l").val());
5   make_right_array = new generate_matrix('each_idr', $("#x_array_r").val(), $("#y_array_r").val());
6   $(".print_area_l").css("width", 50*$("#y_array_l").val()+ 'px').css("height", 50*$("#x_array_l").val()+ 'px');
7   $(".print_area_r").css("width", 50*$("#y_array_r").val()+ 'px').css("height", 50*$("#x_array_r").val()+ 'px');
8   $("#print_area_l").html(make_left_array.make_empty_number());
9   $("#print_area_r").html(make_right_array.make_empty_number());
10  break;

```

JavaScript

[보러 가기](#)

```

1 case 'button_plus':
2   $("#wrong_box_t").css("display", "none");
3   $("#wrong_box_m").css("display", "none");
4   $("#print_result").html("");
5   bank_plus_left = new generate_matrix('each_idl', $("#x_array_l").val(), $("#y_array_l").val());
6   bank_plus_right = new generate_matrix('each_idr', $("#x_array_r").val(), $("#y_array_r").val());
7   plus_div = new generate_matrix('each_id', $("#x_array_r").val(), $("#y_array_r").val());
8   $(".print_result").css("width", 50*($("#x_array_l").val())+'px').css("height", 50*($("#x_array_l").val())+'px');
9   $("#print_result").html(plus_div.plus_two(bank_plus_left.array_left_bank(), bank_plus_right.array_right_bank()));
10  break;

```

```

1 case 'button_minus':
2   $("#wrong_box_t").css("display", "none");
3   $("#wrong_box_m").css("display", "none");
4   $("#print_result").html("");
5   bank_minus_left = new generate_matrix('each_idl', $("#x_array_l").val(), $("#y_array_l").val());
6   bank_minus_right = new generate_matrix('each_idr', $("#x_array_r").val(), $("#y_array_r").val());
7   minus_div = new generate_matrix('each_id', $("#x_array_r").val(), $("#y_array_r").val());
8   $(".print_result").css("width", 50*($("#x_array_l").val())+'px').css("height", 50*($("#x_array_l").val())+'px');
9   $("#print_result").html(minus_div.minus_two(bank_minus_left.array_left_bank(), bank_minus_right.array_right_bank()));
10  break;

```

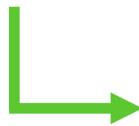
버튼을 누르면 생성된 값을 인자로 하여 객체를 복사 한 후 덧셈(위) / 뺄셈(아래)인 결과값을 만들고 출력합니다.

결과가 출력되는 곳의 크기를 조절하기 위해 행 / 열의 개수로 높이와 너비를 결정합니다.

JavaScript

[보러가기](#)

```
1 case 'button_multi':
2     $("#wrong_box_t").css("display", "none");
3     $("#wrong_box_m").css("display", "none");
4     $("#print_result").html("");
5     bank_multi_left = new generate_matrix('each_idl', $("#x_array_l").val(), $("#y_array_l").val());
6     bank_multi_right = new generate_matrix('each_idr', $("#x_array_r").val(), $("#y_array_r").val());
7     multi_div = new generate_matrix('each_id', $("#x_array_r").val(), $("#y_array_r").val());
8     $(".print_result").css("width", 50*($("#x_array_l").val()+ 'px').css("height", 50*($("#x_array_l").val()+ 'px');
9     $("#print_result").html(multi_div.multi_two(bank_multi_left.array_left_bank(), bank_multi_right.array_right_bank()));
10    break;
```



버튼을 누르면 생성된 값을 인자로 하여 객체를 복사 한 후 두 배열을 곱하여 결과값을 만들고 출력합니다.

결과가 출력되는 곳의 크기를 조절하기 위해 행 / 열의 개수로 높이와 너비를 결정합니다.



Sass

[보러가기](#)

```
1  .button_area{
2    width: $width;
3    height: $main_input_height;
4    display: $display_in_bl;
5    margin: auto;
6    margin-top: $margin-top;
7    float: left;
8    margin-bottom: 2px;
9    .button_design{
10     display: inline-block;
11     width: 80px;
12     height: 30px;
13     font-size: 18px;
14     line-height: 30px;
15     border: 1px solid #d8d8d8;
16     border-radius: $border_radius;
17     cursor: pointer;
18     margin: 0px 2px 1px 2px;
19   }
20 }
```



```
1  $width: 33%;
2  $main_input_height: 35px;
3  $display_in_bl: inline-block;
4  $margin-top: 5%;
```



width, height, display, margin-top 등 중복되는 값을
Sass의 변수 선언으로 미리 설정하여 작성하였습니다.

Sass

[보러 가기](#)

```

1  .each_cell{
2    width: 40px;
3    height: 40px;
4    float: left;
5    border: 1px solid #d8d8d8;
6    line-height: 45px;
7    margin-left: $cell_margin_left;
8    margin-top: $cell_margin_top;
9    font-size: $font_size;
10   border-radius: $border_radius;
11   color: $cell_text_color;
12   background-color: $cell_background_color;
13   outline: none;
14 }

```

```

1  .result_cell{
2    width: 40px;
3    height: 40px;
4    float: left;
5    border: 1px solid #d8d8d8;
6    line-height: 45px;
7    font-size: 12px;
8    margin-left: $cell_margin_left;
9    margin-top: $cell_margin_top;
10   border-radius: $border_radius;
11   color: $cell_text_color;
12   background-color: $cell_background_color;
13 }

```

```

1  $cell_margin_top: 7px;
2  $cell_margin_left: 7px;
3  $result_font_size: 12px;
4  $font_height: -7px;
5  $border_radius: 12.5px 14px 12.5px 14px;

```

margin, font-size, border 등 Sass의 변수 선언으로 미리 설정하여 작성하였습니다.

코드 작성

처음 객체를 사용함에 많은 어려움을 느껴 객체를 사용하지 않는 형식으로 먼저 작업을 하였습니다. 작업시간이 그만큼 증가 하였으나, 객체로 전환하며 꼼꼼히 코드를 볼 수 있었으며 객체 사용에 한단계 발전 할 수 있었습니다.

공식 분석

수학적 접근이 필요한 반복문을 위하여 타인의 도움 없이 공식을 만들었으며 코드를 더욱 이론적으로 접근 할 수 있었습니다.

개선 요구사항

많은 시간을 들여 객체 접근법을 공부하였으나 아직 부족함을 느끼며 더욱 노력해야 함을 느꼈습니다.



06

CONTENTS

후기

감사합니다