



라인컴퓨터아트학원  
JAVA기반 스마트웹 디지털컨버전스  
김하늘



# 행렬 계산기

행렬 계산기 바로가기



# INDEX

---

## 1. 행렬이란? ★ 1p

- 행렬의 모양 / 덧셈과 뺄셈 / 곱셈

## 2. 행렬 계산기 소개 ★ 4p

- 전체 화면 / 이용방법

## 3. 코딩 ★ 8p

- HTML / JAVASCRIPT / SASS

## 4. 후기 ★ 18p



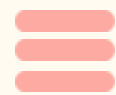


## Chapter 01

행렬이란?

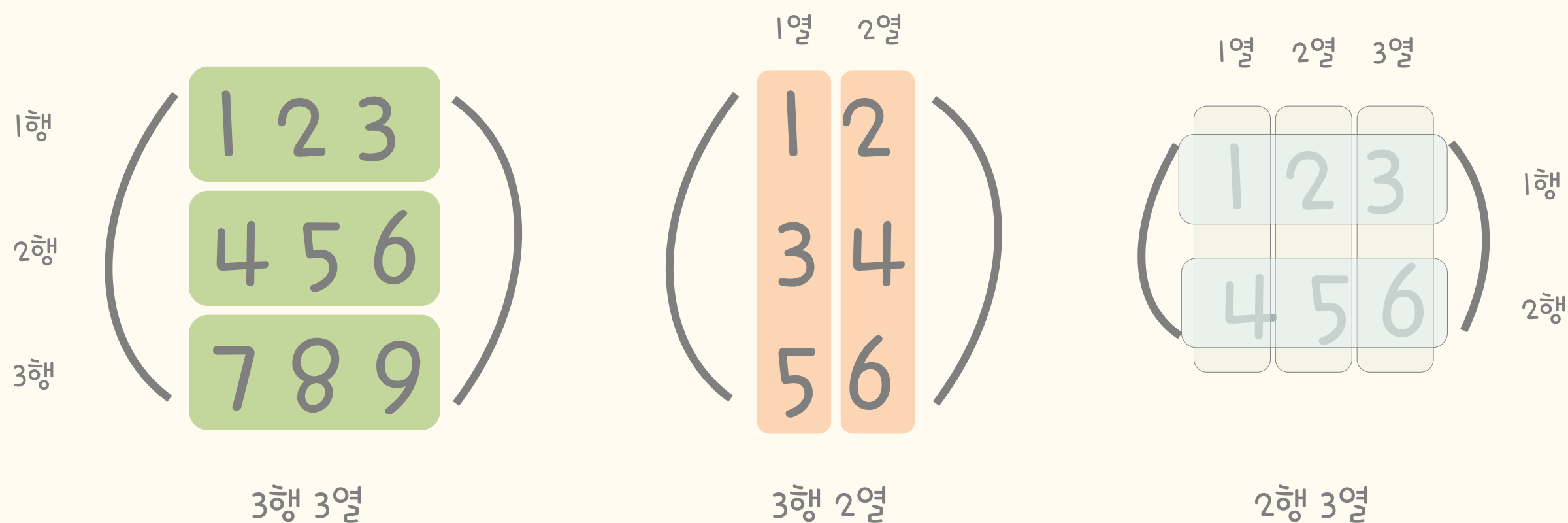
- 행렬의 모양 / 더하기&빼기 / 곱하기





# 행렬이란?

- 행렬의 모양



행렬은 몇 개의 숫자들을 사각형 모양으로 배치하고 괄호로 양옆을 닫아 놓은 모습을  
가로줄을 '행(Row)'이라고 하고 세로줄을 '열(Column)'이라고 한다.

# 행렬이란?

- 덧셈과 뺄셈

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

$$A \pm B = \begin{pmatrix} a_{11} \pm b_{11} & a_{12} \pm b_{12} \\ a_{21} \pm b_{21} & a_{22} \pm b_{22} \end{pmatrix}$$

A와 B의 행과 열의 개수가 같아야 하며, 동일 위치의 값끼리 더하거나 뺄 수 있다.

# 행렬이란?

- 곱셈

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

$$A \times B = \begin{pmatrix} a_{11} \times b_{11} + a_{12} \times b_{21} & a_{11} \times b_{12} + a_{12} \times b_{22} \\ a_{21} \times b_{11} + a_{22} \times b_{21} & a_{21} \times b_{12} + a_{22} \times b_{22} \end{pmatrix}$$

A의 열과 B의 행의 수가 같아야 하고, A의 1행의 첫 번째 값과 B의 1열의 첫 번째 값과 곱하고  
A의 2행의 두 번째 값과 B의 2열의 두 번째 값과 곱한 값을 더하여 한 개의 값을 구할 수 있으며  
마지막에 도달할 때까지 반복한다.

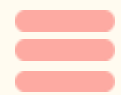


## Chapter 02

행렬 계산기 소개

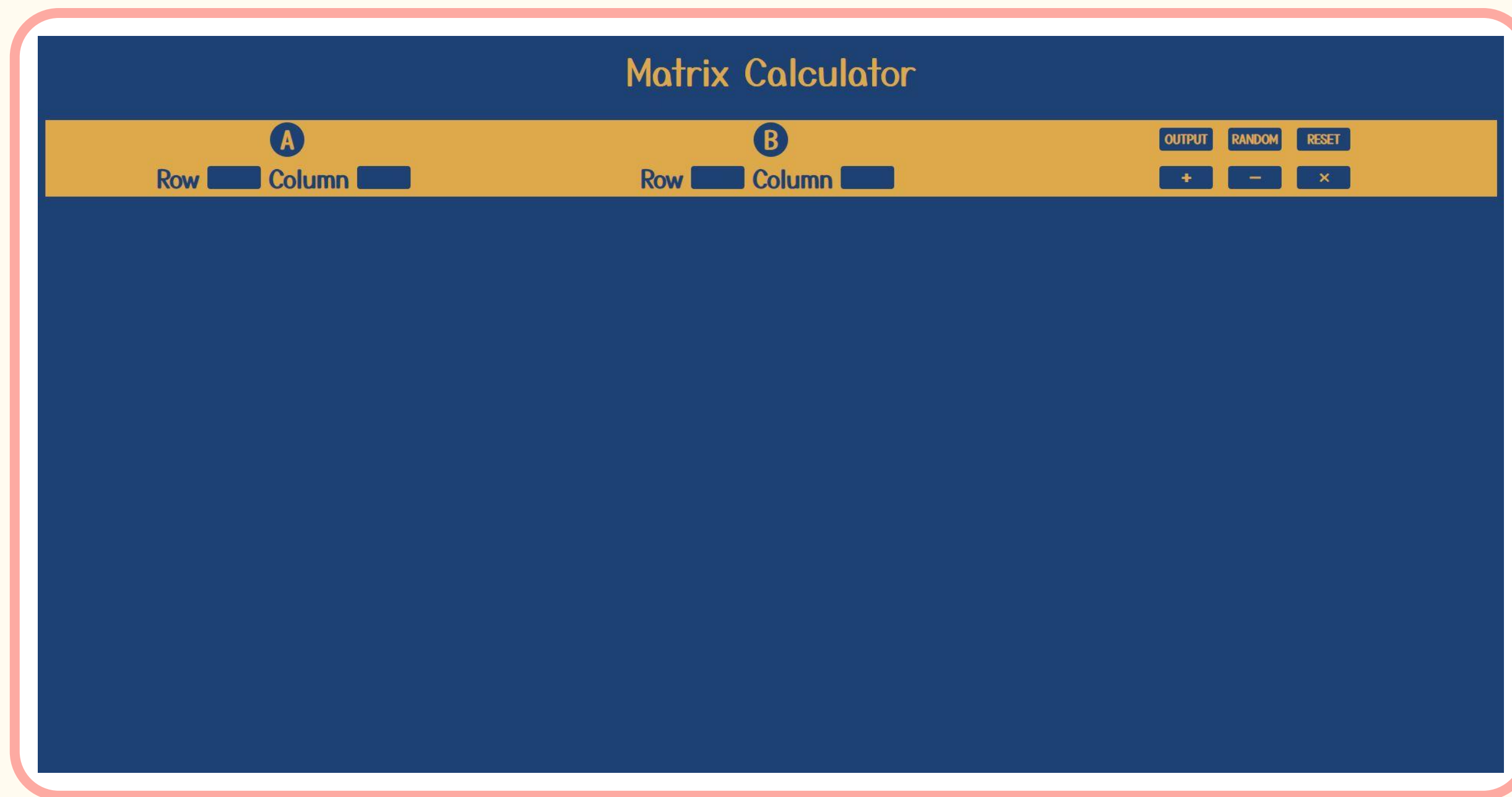
- 전체 화면 / 이용방법





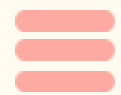
# 행렬 계산기 소개

- 전체 화면



[행렬 계산기 바로가기](#)





# 행렬 계산기 소개

- 이용방법

Matrix Calculator

1 **A** Row  Column  1 **B** Row  Column  2 **OUTPUT** **RANDOM** **RESET** **+** **-** **x**

- 1 그리려고 하는 행과 열의 개수를 입력한다.
- 2 OUTPUT 버튼을 클릭한다.

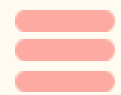


**A** Row **3** Column **2** **B** Row **2** Column **3**

0	0
0	0
0	0

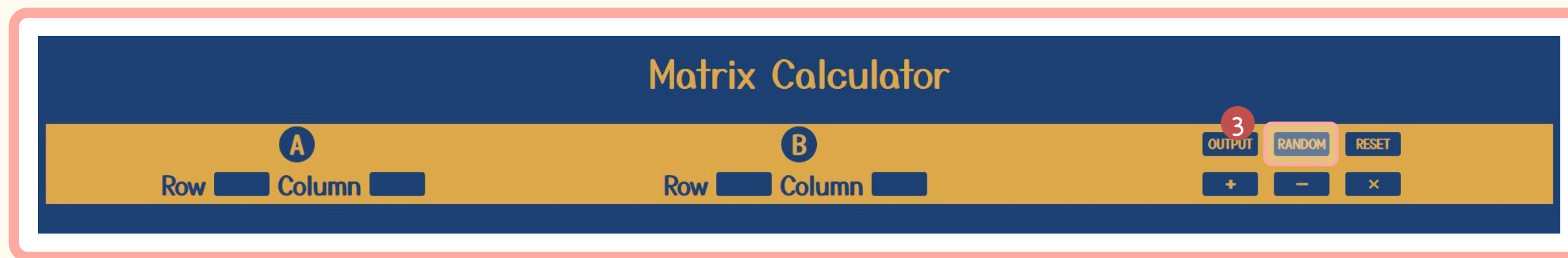
0	0	0
0	0	0

[행렬 계산기 바로가기](#)

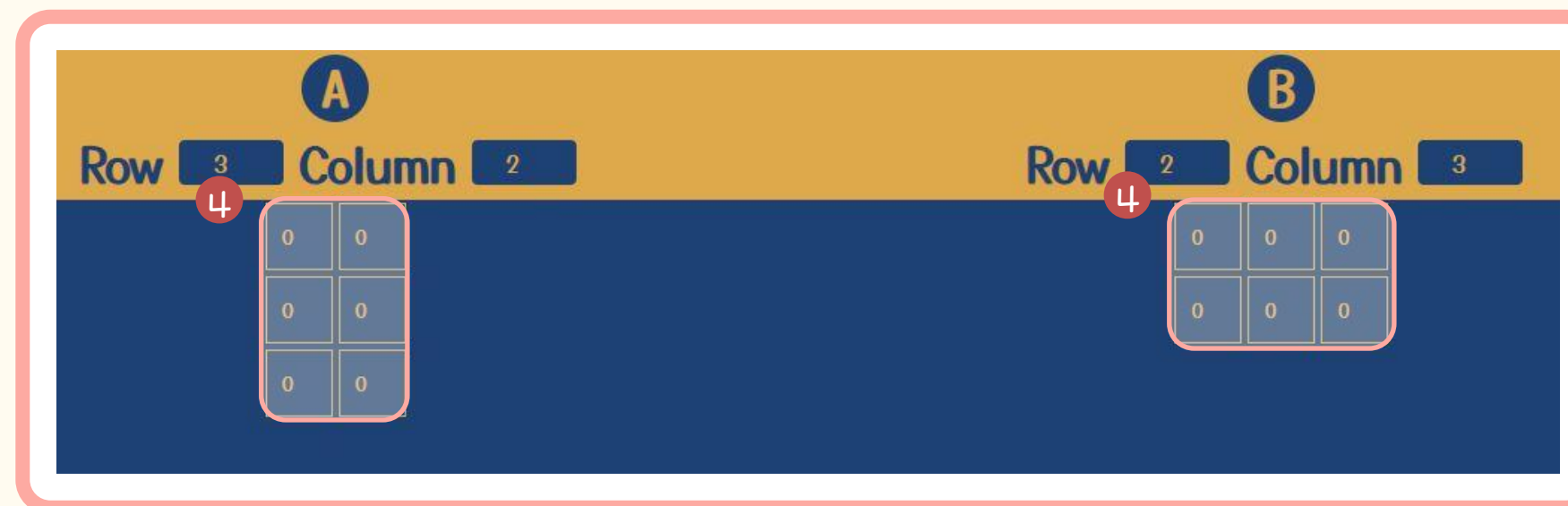


# 행렬 계산기 소개

- 이용방법

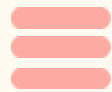


3 RANDOM 버튼 클릭 시 그려진 행렬의 값이 1~99까지의 랜덤 값으로 변경된다.



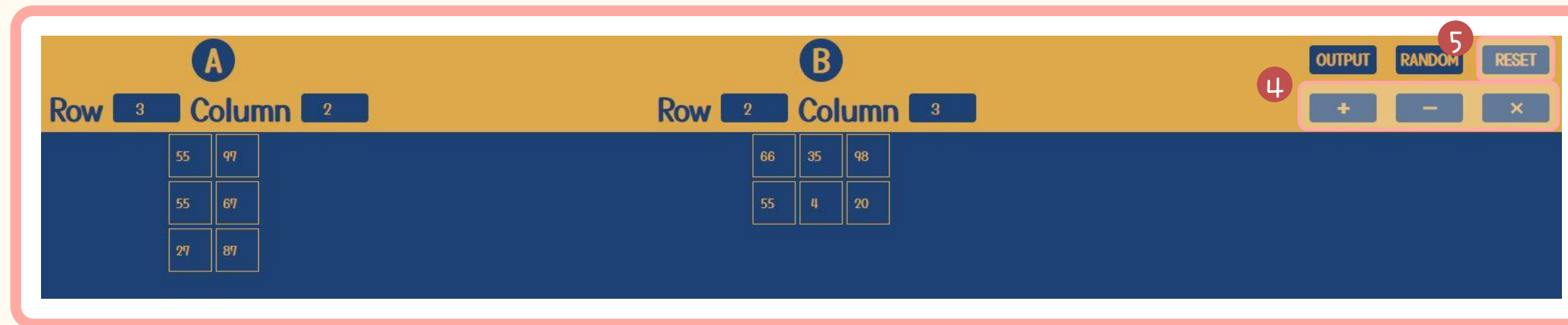
4 그려진 행렬에 직접 값을 기입할 수 있다.

[행렬 계산기 바로가기](#)



# 행렬 계산기 소개

- 이용방법



④ 연산하려는 값을 기입 후 연산 버튼을 클릭한다.

⑤ RESET 버튼은 클릭 시 값을 모두 지울 수 있다.

**주의!**

덧셈과 뺄셈은 A와 B의 행과 열의 수가 동일해야 한다.

곱셈은 A의 열과 B의 행의 수가 동일해야 한다.

[행렬 계산기 바로가기](#)

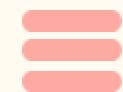


## Chapter 03

코딩

- HTML / JAVASCRIPT / SASS





# 코딩

## • HTML



```
1 <!DOCTYPE html>
2 <html lang="ko">
3   <head>
4     <meta charset="utf-8">
5     <link rel="stylesheet" href="./dist/matrixcalculator.css">
6     <link rel="preconnect" href="https://fonts.gstatic.com">
7     <link href="https://fonts.googleapis.com/css2?family=Stylish&family=Yusei+Magic&display=swap" rel="stylesheet">
8     <title>행렬 계산기</title>
9     <link rel="shortcut icon" type="image" href="./image/mc.jpg">
10    <script src="./jquery/jq.js"></script>
11  </head>
12  <body>
13    <div class="main_box">
14      <!-- 상단 타이틀 -->
15      <header class="title_box">Matrix Calculator</header>
```

- 외부 CSS 링크 및 글꼴 링크
- 문서의 타이틀 설정
- 파비콘 설정
- 제이쿼리 로드

- 상단 대표 제목 작성

## Matrix Calculator

행렬 계산기 바로가기

# 코딩

## • HTML

```
1 <section class="md_box">
2 <!-- 상단 A와 B의 영역 구분 -->
3   <div class="small_box"><p class="mid_title">A</p></div>
4   <div class="small_box"><p class="mid_title">B</p></div>
5 <!-- 버튼 모음1(그리고 지우는 버튼) -->
6   <div class="small_box">
7     <div id="print_bt" class="button_box">OUTPUT</div>
8     <div id="random_bt" class="button_box">RANDOM</div>
9     <div id="reset_bt" class="button_box">RESET</div>
10  </div>
11 <!-- 그리려는 행렬의 값을 받는 INPUT 박스 -->
12   <div class="small_box">
13     Row <input id="x1_num" class="in_num" type="number" min="1" max="12" value="">
14     Column <input id="y1_num" class="in_num" type="number" min="1" max="12" value="">
15   </div>
16   <div class="small_box">
17     Row <input id="x2_num" class="in_num" type="number" min="1" max="12" value="">
18     Column <input id="y2_num" class="in_num" type="number" min="1" max="12" value="">
19   </div>
20 <!-- 버튼 모음2(연산 버튼) -->
21   <div class="small_box">
22     <div id="plus_bt" class="button_box">&#10010;</div>
23     <div id="minus_bt" class="button_box">&#10134;</div>
24     <div id="multi_bt" class="button_box">&#10060;</div>
25   </div>
26 </section>
```

• section 영역의 시작 및 A/B 영역 구분을 위한 중간 제목

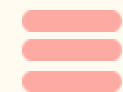
• 행렬을 그릴 수 있는 버튼과 지울 수 있는 RESET 버튼

• 그리려는 행렬의 개수를 입력받는 INPUT 영역

• A/B 행렬의 연산 버튼(더하기, 빼기, 곱하기)

A		B		OUTPUT	RANDOM	RESET
Row	<input type="text"/>	Row	<input type="text"/>	+	-	×
Column	<input type="text"/>	Column	<input type="text"/>			

행렬 계산기 바로가기



# 코딩

## • HTML

```
1 <section class="big_box">
2     <!-- 팝업 박스 -->
3     <div class="pop_up">숫자만 입력하세요.</div><p class="x_box">X</p>
4     <!-- 행렬이 그려지는 영역 -->
5     <div class="print_array">
6         <div id="print_a" class="print_a"></div>
7     </div>
8     <div class="print_array">
9         <div id="print_b" class="print_a"></div>
10    </div>
11    <!-- 연산된 값이 그려지는 영역 -->
12    <div class="print_array">
13        <div id="print_o" class="print_a"></div>
14    </div>
15 </section>
16 </div>
17 <script src="./matrixcalculator.js"></script>
18 </body>
19 </html>
```

• 입력 오류 시 알려주는 대표 팝업

• 행렬이 그려지는 영역

• 메인 자바스크립트 로드

A				B				OUTPUT RANDOM RESET			
Row	3	Column	3	Row	3	Column	3	+	-	×	
	28	96	40		6	98	94		9,096	15,008	15,296
	60	39	28		78	99	99		4,410	11,673	11,713
	27	1	99		36	69	79		3,804	9,576	10,458

행렬 계산기 바로가기

## • JAVASCRIPT

```
1 // 행렬 객체의 시작
2 class xy_cal{
3     constructor( x_num, y_num ){
4         this.x_num = x_num;
5         this.y_num = y_num;
6         this.val_num = [];
7         this.print_a = "";
8         this.id_m = "";
9         this.cal_num = "";
10    }
11    //OUTPUT버튼 클릭 시 그려지는 메소드
12    zero_array(id_name){
13        let i = 0;
14        while( i < this.x_num ){
15            this.val_num.push([]);
16            let j = 0;
17            while( j < this.y_num ){
18                this.val_num[i][j] = 0;
19                j++;
20            }
21            i++;
22        }
23        i = 0;
24        while( i < this.x_num ){
25            let j = 0;
26            while( j < this.y_num ){
27                this.print_a += '<input id="'+id_name+i+"_"+j+"class="min_box" type="number" value="'+this.val_num[i][j]+'>';
28                j++;
29            }
30            i++;
31        }
32        return this.print_a;
33    }
```

• class 객체의 constructor 선언

• OUTPUT 버튼 클릭 시 그려지는 행렬의 메소드  
(값은 0으로 표시됨)



## • JAVASCRIPT

```

1  //RANDOM버튼 클릭 시 그려지는 메소드
2  random_array(id_name){
3      let i = 0;
4      while( i < this.x_num ){
5          this.val_num.push([]);
6          let j = 0;
7          while( j < this.y_num ){
8              this.val_num[i][j] = Math.floor(Math.random()*100);
9              j++;
10         }
11         i++;
12     }
13     i = 0;
14     while( i < this.x_num ){
15         let j = 0;
16         while( j < this.y_num ){
17             this.print_a += '<input id="'+id_name+i+'_'+j+'"class="min_box" type="number" value="'+this.val_num[i][j]+'">';
18             j++;
19         }
20         i++;
21     }
22     return this.print_a;
23 }
24 //행렬을 그릴 때 영역을 구분
25 print_array(pt_id){
26     $("#"+pt_id).css("width", (49*this.y_num)+"px");
27 }

```

- RANDOM 버튼 클릭 시 그려지는 행렬의 메소드  
(값은 0~99까지 랜덤으로 표시됨)

- 열의 구분을 하기 위해 width를 열의 개수만큼 제한

## • JAVASCRIPT

```
1 minus_array(){
2     let i = 0;
3     while( i < this.x_num ){
4         let j = 0;
5         while( j < this.y_num ){
6             this.cal_num += '<div class="min_box">'+(parseInt($("#in_a"+i+"_"+j).val())-parseInt($("#in_b"+i+"_"+j).val()))+'</div>';
7             j++;
8         }
9         i++;
10    }
11 }
12 //PLUS 연산
13 plus_array(){
14     let i = 0;
15     while( i < this.x_num ){
16         let j = 0;
17         while( j < this.y_num ){
18             this.cal_num += '<div class="min_box">'+(parseInt($("#in_a"+i+"_"+j).val())+parseInt($("#in_b"+i+"_"+j).val()))+'</div>';
19             j++;
20         }
21         i++;
22     }
23 }
24 }
```

• 그려진 행렬의 ID의 값을 가져와서 MINUS 연산

• 그려진 행렬의 ID의 값을 가져와서 PLUS 연산

## • JAVASCRIPT

```

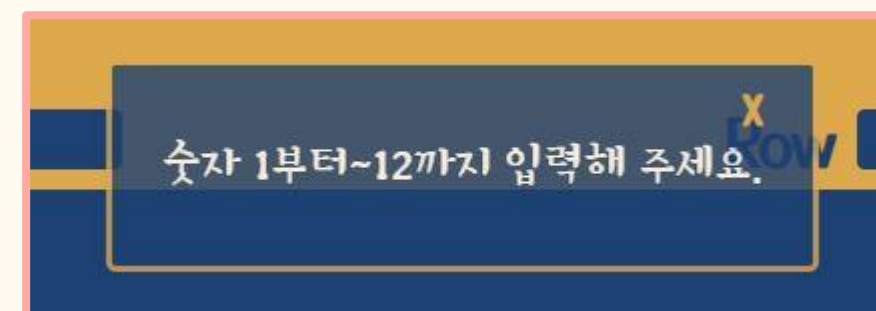
1  //전체 팝업 실행 함수
2  pop_event=( id, body )=>{
3      $(".pop_up, .x_box").css("display", "block");
4      $(".pop_up").html(body);
5      $("#"+id).val("");
6      $(".pop_up, .x_box").click(function(){
7          $(".pop_up, .x_box").css("display", "none");
8      });
9  }
10 // 그리려는 행렬 값을 입력 시 검사 진행하여 팝업 노출
11 $("input").on("keyup", function(){
12     const check_in = /^[0-9]![\.]/g;
13     if(check_in.test($("#x1_num").val())){
14         pop_event("x1_num", "숫자 1부터~12까지 입력해 주세요.");
15         $("#print_a, #print_b, #print_o").empty();
16     }

```

중간생략

- 팝업이 필요한 곳에 ID와 오류 문구를 넣어 필요할 때 사용 가능

- 그리려는 행렬의 값 입력 시 keyup 이벤트를 사용하여 팝업 노출



- JAVASCRIPT

```

1  //이벤트 영역
2  $(document).on("click", function(e){
3      //A와 B의 입력 값
4      var xa_v = Number($("#x1_num").val());
5      var ya_v = Number($("#y1_num").val());
6      var xb_v = Number($("#x2_num").val());
7      var yb_v = Number($("#y2_num").val());
8      //영역별 객체 복제
9      var a_val = new xy_cal(xa_v, ya_v);
10     var b_val = new xy_cal(xb_v, yb_v);
11     //곱하기 함수
12     multi_array =()=> {
13         let m_st = "";
14         let new_array = "";
15         let total_num = 0;
16         for( let i = 0; i < a_val.x_num ; i++ ){
17             m_st = [];
18             for( let j = 0; j < b_val.y_num; j++ ){
19                 for( let k = 0; k < a_val.y_num; k++ ){
20                     total_num += parseInt($("#in_a"+i+"_"+k).val())*parseInt($("#in_b"+k+"_"+j).val());
21                 }
22                 m_st += '<div class="min_box">'+total_num.toString().replace(/\B(?=(\d{3})+(?! \d))/g, ',')+</div>';
23                 total_num = 0;
24             }
25             new_array += m_st;
26         }
27         return new_array;
28     }

```

- 행렬의 개수 입력 값을 변수에 담고, A/B 영역별 객체 복제

- 곱하기 연산은 함수로 만들어 이벤트 영역 내 배치
- 곱한 수가 3자리가 넘을 경우 정규식을 사용하여 ,로 구분

# 코딩

## • JAVASCRIPT

```
1 //행렬이 그려진 후 값이 변경되었을 경우 클릭 이벤트 발생 시 검사 요소
2 const check_in = /^[0-9]|\./g;
3 const html_testa = $("#print_a").html();
4 const html_testb = $("#print_b").html();
5 // 각 버튼별 검사 후 이벤트
6 switch(e.target.id){
7     case 'print_bt' :
8         if( check_in.test(xa_v) || check_in.test(xb_v) || check_in.test(ya_v) || check_in.test(yb_v) || xa_v > 12 || xb_v > 12 || ya_v > 12 || yb_v > 12 ){
9             pop_event("x1_num, #x2_num, #y1_num, #y2_num ", "숫자 1부터~12까지 입력해 주세요.");
10            $("#print_a, #print_b, #print_o").empty();
11        } else {
12            a_val.zero_array("in_a");
13            a_val.print_array("print_a");
14            $("#print_a").html(a_val.print_a);
15            b_val.zero_array("in_b");
16            b_val.print_array("print_b");
17            $("#print_b").html(b_val.print_a);
18        }
19        break;
```

중간생략

- RESET 버튼 클릭 시 모든 값에 대한 초기화
- 연산 버튼 클릭 시 A/B 영역이 그려져 있는지 확인 후 연산 진행 (그려져 있지 않을 경우 팝업 노출)
- 연산 불가할 경우 팝업 노출

- 행렬의 개수 제한 및 복제한 객체의 메소드 사용

```
1 case 'reset_bt' :
2     $("#print_a, #print_b, #print_o").empty();
3     $("#x1_num, #x2_num, #y1_num, #y2_num").val("");
4     break;
5 case 'plus_bt' :
6     if( html_testa == "" && html_testb == ""){
7         pop_event("plus_bt", "OUTPUT을 먼저 눌러주세요.");
8     } else if( xa_v == xb_v && ya_v == yb_v ){
9         a_val.plus_array();
10        a_val.print_array("print_o");
11        $("#print_o").html(a_val.cal_num);
12    } else {
13        pop_event("plus_bt", "A와 B의 행렬이 같아야 합니다.");
14        $("#print_a, #print_b, #print_o").empty();
15    }
16    break;
```

중간생략

행렬 계산기 바로가기

# 코딩

## • SASS

```
1 // 대표 색
2 $back_g : #DDA94B;
3 $scan_c1 : #1E4174;
4 //리셋과 글꼴 설정
5 * { margin : 0px; padding : 0px; font-family: 'Yusei Magic', sans-serif;}
6 //전체화면 설정
7 html, body{
8     width : 100%;
9     height : 100%;
10    background-color: $scan_c1;
11 }
```

- 대표 색상 설정 및 리셋과 전체 글꼴 설정
- 전체 화면 사용으로 html, body를 100% 설정 후 background-color 지정

```
1 //버튼 시작
2 .button_box{
3     width : 70px;
4     height : 30px;
5     background-color: $scan_c1;
6     color : $back_g;
7     border-radius: 4px;
8     margin : 10px;
9     line-height: 28px;
10    font-size: 1rem;
11    float: left;
12    cursor: pointer;
13 }
14 .button_box:hover{
15     background-color: white;
16     color : $scan_c1;
17 }
18 #multi_bt{
19     font-size: 0.8rem;
20     line-height : 30px;
21 }
22 #plus_bt, #minus_bt{
23     font-size: 1rem;
24     line-height : 28px;
25 }
26 .in_num{
27     @extend .button_box;
28     text-align: center;
29     outline: none;
30     border : none;
31     font-size: 1rem;
32 }
33 }
34 }
```

- 전체 버튼에 CSS 설정
- 연산 버튼은 글꼴에 따라 크기가 상이하여 ID를 사용하여 글꼴 크기 및 줄 간격 설정
- 행렬 개수를 받는 INPUT들에 대해서 @extend를 사용하여 동일하게 주고 추가 필요한 속성에 대해 작성

행렬 계산기 바로가기



# Chapter 04

후기



## 후기

처음 과제가 주어졌을 때 어렵다고 느껴져 많이 더딘 프로젝트였다.

어떤 식으로 로직을 구성해야 하는지 처음에는 막막하기만 했었지만, 많은 고민과 수정을 거쳐 완성이 되었고, 로직을 구현하고 객체 사용에 대해 이해하고 경험한 프로젝트여서 나에게 뜻깊은 과제였다고 생각한다.

생각대로 되지 않을 때 많이 힘들었지만 어려웠던 것들이 순차적으로 해결이 되면서 그만큼 성장한 계기가 되었다.

어떤 프로젝트가 주어진다고 해도 겁먹지 말고 차근차근 해나갈 수 있다고 생각한다. 다음에는 좀 더 깔끔하고 순서 있는 로직을 구현해야겠다.





THANK 😊 YOU!

