

행렬계산기

MATRIX CALCULATOR

라인컴퓨터아트학원 / JAVA기반 스마트 웹&앱 개발자 양성과정 17기 / 서동민

01

1. SCHEDULE

P. 03 ~ P. 04

02

2. SKETCH

P. 05 ~ P. 06

03

3. PROTOTYPING

P. 07 ~ P. 11

04

4. PROGRAMMING

P. 12 ~ P. 17

05

5. Result

P. 18 ~ P. 19

06

6. Review

P. 20 ~ P. 21

SCHEDULE



디자인, 코딩, 보고 순으로 분류하였다.

디자인은 스케치와 프로토타이핑을 했다.

기본적인 틀은 PUG로 구현, SCSS를 통해서 디자인을 구현, JavaScript를 이용해서 동작을 구현했다.

보고 과정에서는 계획서를 제출하고 디자인을 확인 받았다. 이후 디자인을 개선하며 피드백 받으며 기능을 구현했다.

SKETCH

Array 1

Number	"	"	"	"	"
"	"	"	"	"	"
"	"	"	"	"	"
"	"	"	"	"	"
"	"	"	"	"	"
"	"	"	"	"	"
"	"	"	"	"	"

리셋

Array 2

RANDOM + - x =

number	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"

리셋

Output

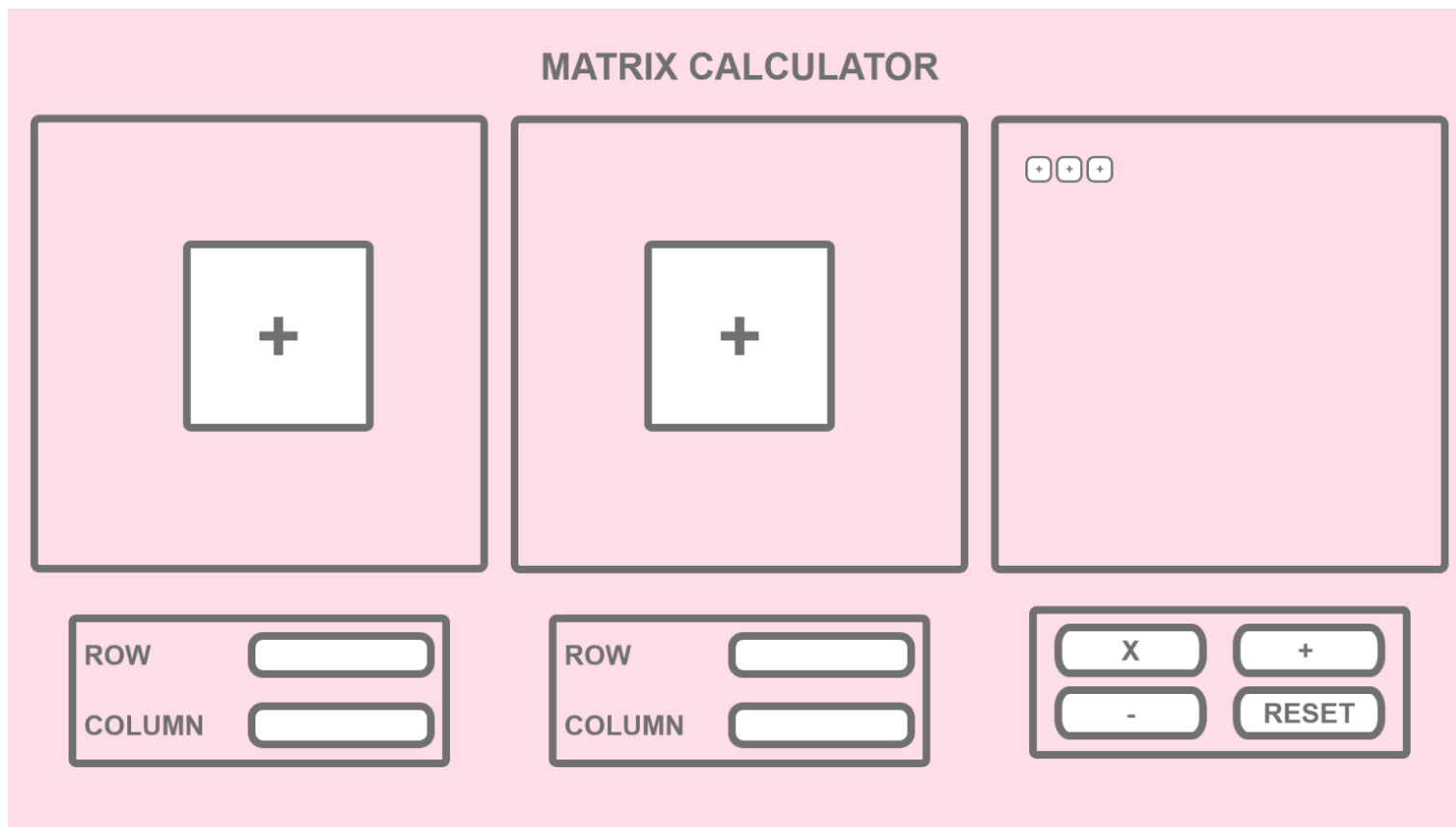
출력	"	"	"	"	"	"
"	"	"	"	"	"	"
"	"	"	"	"	"	"
"	"	"	"	"	"	"
"	"	"	"	"	"	"
"	"	"	"	"	"	"
"	"	"	"	"	"	"

리셋
랜덤 곱하기 빼기 더하기
Random x - +

스케치를 진행하면서 입력 받을 공간 2개와 출력 받을 공간 한 개가 필요하다고 생각했다. 그래서 3개의 공간을 배치했다.

숫자를 Random하게 입력해주는 기능과 더하기, 빼기, 곱하기를 선택할 수 있는 버튼과 출력 버튼이 필요하다고 생각했다. 그래서 동작을 하는 버튼을 만들었다.

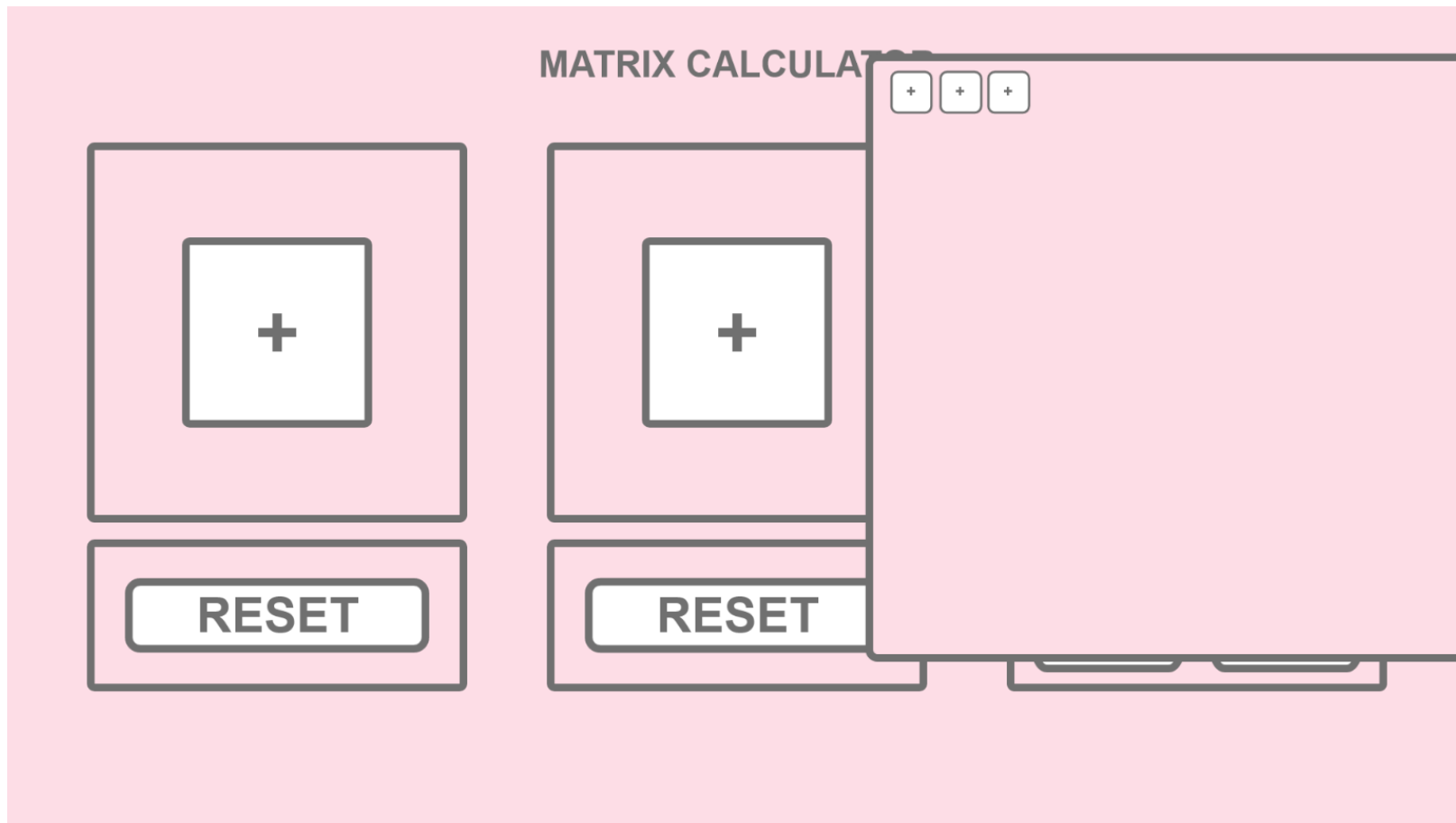
PROTOTYPING



페이지에 들어오면 처음으로 나오는 화면이다. 스케치에서 구상했던 곱하기, 더하기, 빼기를 위한 버튼이 있다.

1. ROW와 COLUMN에 숫자를 입력한다.
2. 큰 더하기 버튼을 누르면 행렬이 생성된다.
3. 생성된 행렬에 숫자를 입력한다.
4. 더하기, 곱하기, 빼기를 눌러 연산한다.

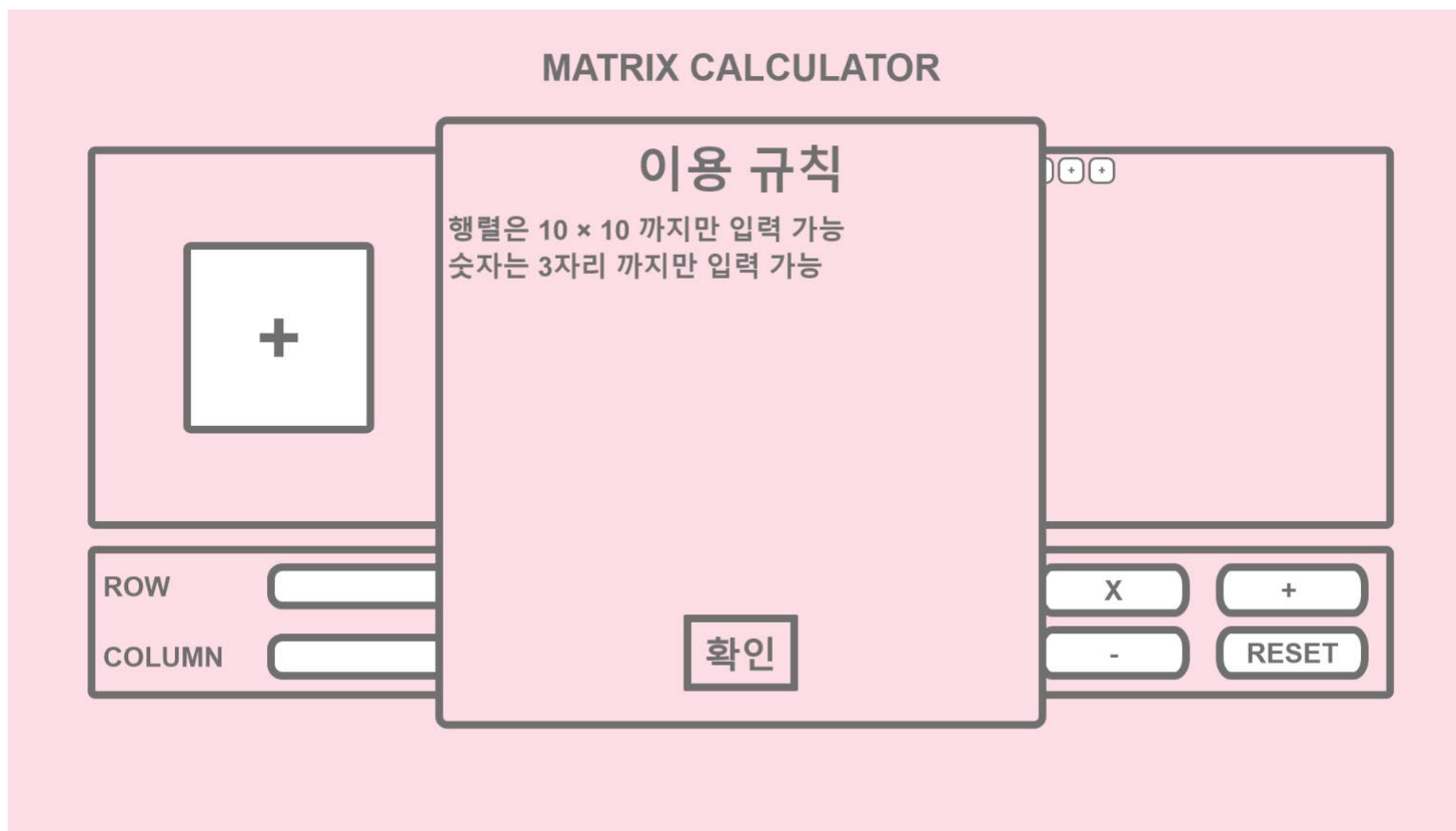
순으로 설계하였다.



결과를 보거나 입력할 때 잘 안보일 수 있어서 확대하는 기능을 추가했다.

이후 사람들의 불편하다는 의견을 수렴하여 제거된 기능이다.

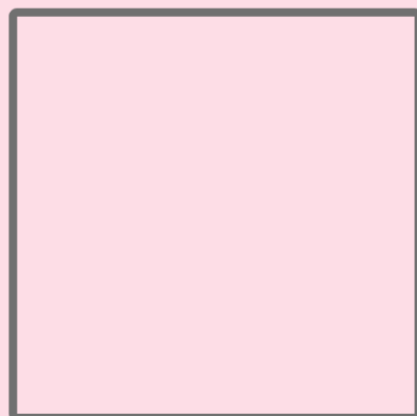
프로토타이핑 과정에서부터 생각했던 기능이였다.



현재 페이지는 처음 들어갔을 때 이용자에게 이용 방법을 안내하는 모달의 디자인을 만든 것이다.

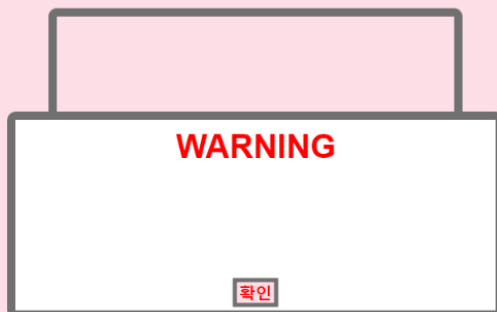
이용자는 이용 규칙에 적힌 내용을 보고 행렬 계산기를 이용하면 된다.

MATRIX CALCULATOR



RESET

RANDOM

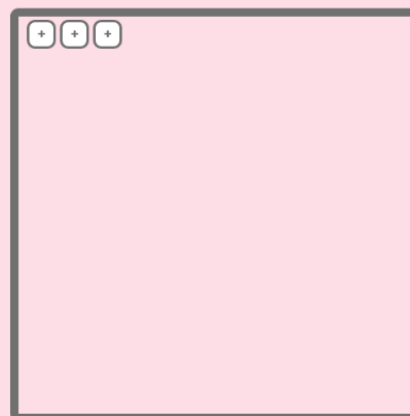


WARNING

확인

RESET

RANDOM



x

+

-

RESET

현재 페이지는 이용자가 잘못된 입력을 했을 경우에 출력되는 모달이다.

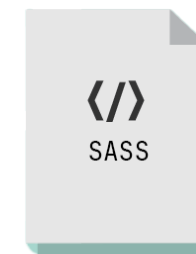
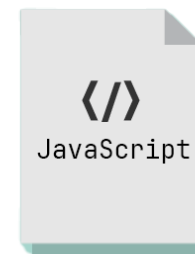
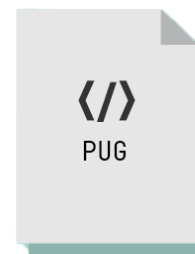
이용자가 발생시킨 문제에 따라서 다른 내용이 출력된다.

PROGRAMMING

개발언어

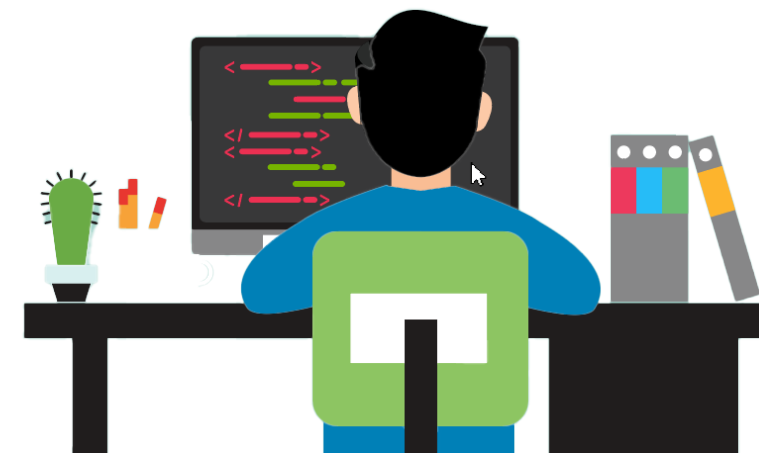
PUG, SASS, JavaScript

개발에는 효율적인 개발을 위하여 PUG라는 템플릿 언어를 사용하게 되었다.
반복적인 작업이나 효율적인 수정을 위하여 SASS를 도입하였다.



pug

Sass





```
1  doctype html
2  html(lang="ko-KR")
3    head
4      meta(charset="UTF-8")
5      meta(name="viewport", content="width=device-width, initial-scale=1.0")
6      title MATRIX CALCULATOR
7      link(rel="stylesheet", href="../scss/style.css", type="text/css")
8    body
9      header
10       h1 MATRIX CALCULATOR
11       section.calculator
12         .arrayBox
13           #inputArray1.box
14             button#arrayPlus1.arrayPlus +
15             #inputArrayNum1
16           #inputArray2.box
17             button#arrayPlus2.arrayPlus +
18             #inputArrayNum2
19           #outputArray.box
```

PUG를 이용해서 페이지 구성 내용들을 구현하였다.

PUG는 들여쓰기로 태그를 구별하기 때문에 적은 타이핑만으로 틀을 모두 구현할 수 있었다.

```
1  @mixin box() {  
2    width: 31.25vw;  
3    height: 31.25vw;  
4    float: left;  
5    margin-right: 1.56vw;  
6    border-radius: $b10;  
7  }  
8  @mixin borderSize($borderSize: $b10) {  
9    border: $borderSize solid #707070;  
10   box-sizing: border-box;  
11  }  
12  @mixin transitionHoverActive() {  
13    transition: all 0.3s;  
14  }  
15  $b10: 0.52vw; // 10px  
16  $f36: 1.87vw; // 36px  
17
```

SASS의 MIXIN기능과 변수 기능을 이용하여 반복적인 코드를 제거하는데 활용하였다.

기본적인 틀을 구성하는 박스를 구성할 때와 박스들의 태두리를 구성할 때, 변화에 대해서 애니메이션을 구현할 때 MIXIN을 사용하였다.

변수는 자주 사용하는 크기에 선언하였다.

변수를 활용하여 크기를 수정할 때 한번에 동일한 변수를 사용하는 값들이 수정되어서 빠른 수정을 진행할 수 있었다.



```
1  let gettingArray = (column, row, rememberArray, alpha) => {  
2    makingArrayBucket(row, rememberArray);  
3    let i = 0;  
4    while (i < row) {  
5      let j = 0;  
6      while (j < column) {  
7        arrayNum = document.querySelector(`#${alpha}${i}_${j}`);  
8        rememberArray[i].push(arrayNum.value);  
9        j++;  
10     }  
11     i++;  
12   }  
13 }; // 숫자 가져와서 임시 배열에 입력하는 장치
```

만들어진 행렬의 입력
된 값들을 가져와서 연
산을 하기 위한 배열로
변환해주는 함수이다.



```
1  const plusComma3Number = () => {  
2    outOfValue = document.querySelectorAll(".outOfValue");  
3    outOfValue.forEach((nopValue, index, arrow) => {  
4      outOfValueNumber = outOfValue[index].value;  
5      outOfValueNumberChangeValue = outOfValueNumber  
6        .toString()  
7        .replace(/\B(?=(\d{3})+(?!\d))/g, ",");  
8      outOfValue[index].value = outOfValueNumberChangeValue;  
9    });  
10  };
```

정규식을 이용하여 입력 받는 문자의 종류를 제한하였다.

정규식을 이용하여 출력된 값에 3번째 자릿수마다 쉼표를 찍는 기능을 구현하였다.

RESULT

MATRIX CALCULATOR

-30	56	-96	-98	62	-46	-61	74	78	-41
42	49	89	-15	14	-11	68	84	52	-38
82	73	-38	-44	28	93	23	41	0	61
-32	9	-3	-69	40	18	53	29	66	-57
-10	82	27	0	-44	-2	22	0	24	88
-65	56	36	-15	75	-92	-31	-38	58	85
53	17	53	-33	23	-37	-11	-74	-27	-89
-7	-64	9	22	-18	5	-28	-36	-13	-47
42	-5	85	0	-91	93	73	36	75	-16
-10	52	43	-43	0	45	-96	26	-65	90

RESET

RANDOM

54	71	44	9	68	87	48	53	79	36
42	77	48	37	41	53	93	27	78	70
31	94	49	92	10	75	60	7	17	31
88	24	16	77	43	54	5	12	12	34
47	72	43	35	69	90	62	46	98	30
9	56	46	47	87	1	85	6	65	87
78	5	87	11	69	45	10	74	66	59
72	14	71	72	12	56	4	69	19	48
33	64	29	82	22	26	53	25	69	50
6	92	22	81	47	27	7	78	4	22

RESET

RANDOM

-5,470	-5,355	-3,047	-6,836	-8,174	-4,280	985	-6	4,874	-2,659
19,164	16,501	20,969	17,179	10,177	21,455	15,277	13,734	18,650	16,462
9,709	20,340	16,282	10,052	21,266	13,368	18,687	16,658	22,328	18,986
2,585	22	7,651	168	2,300	1,546	6,531	2,773	10,760	6,370
4,691	14,604	7,381	13,132	5,924	5,529	7,976	8,929	5,083	9,157
-1,395	13,814	-1,498	10,509	-2,562	6,141	4,210	4,201	2,609	-2,451
-4,548	-2,161	-4,448	-12,365	-4,644	1,994	2,961	-9,458	1,516	-6,419
-7,139	-10,867	-9,534	-8,077	-7,730	-8,636	-7,771	-11,117	-10,184	-8,874
11,918	13,440	16,868	17,448	11,680	8,486	13,651	7,571	12,113	18,591
-7,623	12,828	-866	7,370	436	710	6,875	1,014	-3,561	1,380

+

-

×

RESET

랜덤으로 생성 받아서 곱하기 연산을 하여 결과를 출력한 것이다.

REVIEW

JavaScript

처음에 기능을 구현하기 위해서 만든 코드보다 버그를 막거나 이용자가 잘못된 입력을 했을 때 발생하는 문제를 막기 위한 코드를 짜는 것이 더 힘들었다고 느껴졌다.

숫자만 입력해야 하는 공간에 수학 기호만 입력하거나 앞에 0을 붙여서 입력하는 행위를 막아야 했다. 한정된 공간에서 최대 표시가능한 숫자의 길이를 한정해야 하는 문제도 있었다.

문제들을 해결하면서 이용자가 문제를 발생시켰을 경우 그 문제가 입력되거나 작동하지 않게 막는 것이 중요하다는 것을 깨달을 수 있었다.

발생 가능한 문제들을 방지하며 코드를 작성했다면 JavaScript 코드가 좀 더 깔끔할 수 있지 않았을까 생각된다.

SCSS

CSS 방법론을 알고 있었다면 더 효율적인 SCSS를 작성할 수 있었을 것이라고 생각한다.

다음에는 배운 내용을 추가로 적용해서 더 효율적인 SCSS를 작성 해야 겠다.

PUG

PUG를 도입함으로써 코드를 작성하는데 시간을 더 많이 아낄 수 있었다. PUG의 유용함을 체험하는 중요한 기회가 되었다.