

행렬계산기 구현

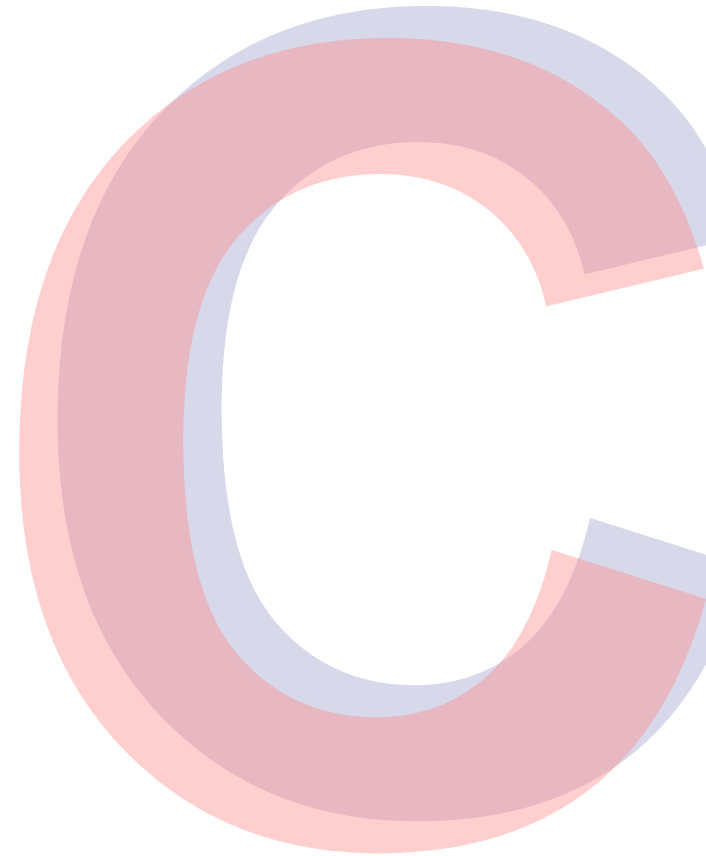
URL : <http://pager.kr/~c10st06/Portfolio/matrix/calculator.html>

C10st06 김은숙

INDEX

01 제작 일정 표

02 구현



제작일정표

행렬계산기 제작

총 제작기간: 2020.01.21-2020.02.03 (2주)

2020.01.20 (월)	2020.01.21 (화)	2020.01.22 (수)	2020.01.23 (목)	2020.01.24 (금)	2020.01.25 (토)	2020.01.26 (일)
	행렬 알고리즘 복습	코딩(30%)	코딩(50%)	코딩(70%)	코딩(80%)	코딩(90%)
2020.01.27 (월)	2020.01.28 (화)	2020.01.29 (수)	2020.01.30 (목)	2020.01.31 (금)	2020.02.01 (토)	2020.02.02 (일)
디버깅 테스트(2)	디버깅 테스트(3)	디버깅 테스트(4)	디버깅 테스트(5)	디버깅 테스트(6)	디버깅 테스트(7)	디버깅 테스트(8)
2020.02.03 (월)						
제출						

구현 - 제작과정(코드)

행렬계산기 제작



```

let makeMatrixBtn = document.getElementById('makeMatrixBtn'); //생성하기 버튼
let madeMatrix = document.getElementById('madeMatrix'); //생성된 행렬 나오는 박스
let Bank=[]; //공배열
//행렬 개체
function makeMatrixBox(matrixX, matrixY, matrixNum) {
  this.matrixX = matrixX; //행의 값
  this.matrixY = matrixY; //열의 값
  this.matrixNum = matrixNum; //행렬 번호A=0, B=1
  this.temporaryArray=[];
  this.matrixArray=[];
  //행렬 생성
  this.makeMatrixV = function() {
    madeMatrix.innerHTML += '<div class="inMatrixBox"></div>';
    let inMatrixBox = document.getElementsByClassName('inMatrixBox')[this.matrixNum];
    i=0;
    while(i<this.matrixX) {
      j=0;
      while(j<this.matrixY) {
        document.getElementsByClassName('inMatrixBox')[this.matrixNum].innerHTML += '<input class="inMatrixValue" type="number" value="0">';
        j++;
      }
      inMatrixBox.innerHTML += "<br>";
      i++;
    }
  }
}
  
```

URL : <http://pager.kr/~c10st06/Portfolio/matrix/calculator.html>

구현 - 제작과정(코드)

행렬계산기 제작



```

// 만든 A, B 행렬의 값 배열에 담기
this.tempArray = function() {
  let inMatrixValue = document.getElementsByClassName('inMatrixValue');
  i=0;
  while(i<inMatrixValue.length) {
    Bank[i] = inMatrixValue[i].value;
    i++;
  }
}

this.makeArray = function() {
  // 임시 배열에 행렬값 담기
  i=0;
  while(i<(this.matrixX*this.matrixY)) {
    this.tempArray.push(Bank.shift());
    i++;
  }
  //matrixArray[i]에 값 담기
  i=0;
  while(i<this.matrixX) {
    this.matrixArray[i]=[];
    j=0;
    while(j<this.matrixY) {
      this.matrixArray[i][j] = this.tempArray[0];
      this.tempArray.shift();
      j++;
    }
    i++;
  }
  return(this.matrixArray);
}

```

행렬생성 개체 완성

```

//행과 열 입력받을 변수 선언
let matrixAX = document.makeMatrix.matrixAX;
let matrixAY = document.makeMatrix.matrixAY;
let matrixBX = document.makeMatrix.matrixBX;
let matrixBY = document.makeMatrix.matrixBY;
//행렬 생성 버튼 실행
makeMatrixBtn.addEventListener('click',function() {
  madeMatrix.innerHTML = "";
  let matrixA = new makeMatrixBox(matrixAX.value, matrixAY.value, 0);
  matrixA.makeMatrixV();
  let matrixB = new makeMatrixBox(matrixBX.value, matrixBY.value, 1);
  matrixB.makeMatrixV();
  let buttonBox = document.getElementById('buttonBox');
  buttonBox.style.display="block";
});

```

행렬 생성 완료

URL : <http://pager.kr/~c10st06/Portfolio/matrix/calculator.html>

구현 - 제작과정(코드)

행렬계산기 제작



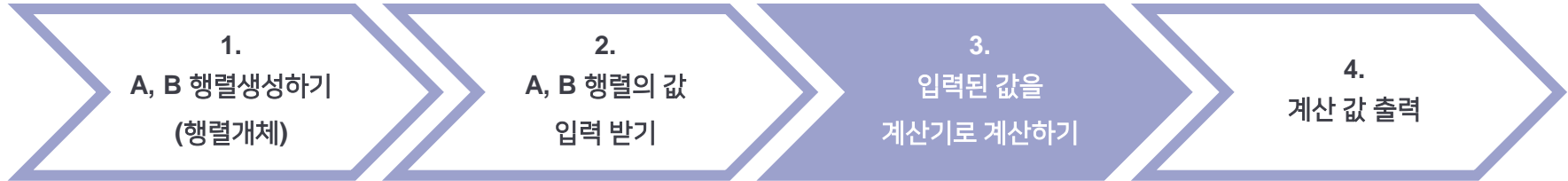
계산기객체에 값을 랜덤으로 넣어주는 기능 구현

```
//랜덤 버튼 실행
let inMatrixValue = document.getElementsByClassName('inMatrixValue');
case "randomAB":
  i=0;
  while(i<(matrixAX.value*matrixAY.value)+(matrixBX.value*matrixBY.value)){
    document.getElementsByClassName('inMatrixValue')[i].value = parseInt((Math.random()*50)-(Math.random()*50));
    i++;
  }
  break;
```

URL : <http://pager.kr/~c10st06/Portfolio/matrix/calculator.html>

구현 - 제작과정(코드)

행렬계산기 제작



계산기 객체 만들기

```

// 계산기 객체
function matrixCalculator(mA, mB) {
  this.mA = mA;
  this.mB = mB;
  this.plus=[];
  this.minus=[];
  this.multi=[];
  // 덧셈 계산기
  this.plusmAB = function() {
    i=0;
    while(i<this.mA.length){
      j=0;
      this.plus[i]=[];
      while(j<this.mA[0].length) {
        this.plus[i][j] =parseInt(this.mA[i][j])+parseInt(this.mB[i][j]);
        j++;
      }
      i++;
    }
    return(this.plus);
  }
  // 뺄셈 계산기
  this.minusmAB = function() {
    i=0;
    while(i<this.mA.length){
      j=0;
      this.minus[i]=[];
      while(j<this.mA[0].length) {
        this.minus[i][j] = parseInt(this.mA[i][j])-parseInt(this.mB[i][j]);
        j++;
      }
      i++;
    }
    return(this.minus);
  }
}
  
```

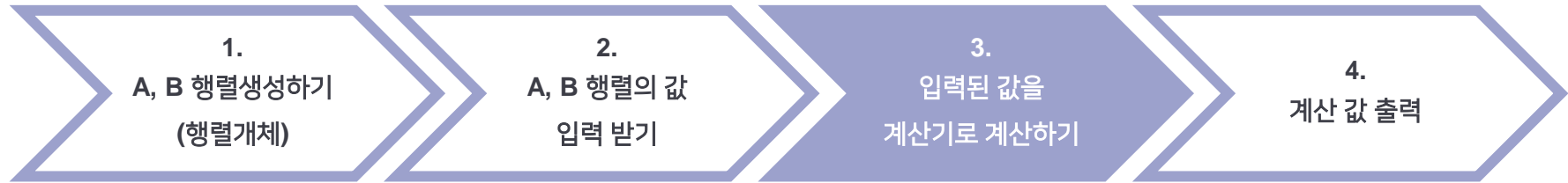
```

//AB 곱셈 계산기
let myA = [], myV=0;
this.multimAB = function() {
  i=0;
  while(i<this.mA.length) {
    j=0;
    while(j<this.mB[0].length){
      k=0;
      while(k<this.mA[0].length) {
        myV += parseInt(this.mA[i][k]*this.mB[k][j]);
        k++;
      }
      myA.push(myV);
      myV=0;
      j++;
    }
    i++;
  }
  i=0;
  while(i<this.mA.length) {
    j=0;
    this.multi[i]=[];
    while(j<this.mB[0].length) {
      this.multi[i][j]=myA[0];
      myA.shift();
      j++;
    }
    i++;
  }
  return(this.multi);
}
  
```

URL : <http://pager.kr/~c10st06/Portfolio/matrix/calculator.html>

구현 - 제작과정(코드)

행렬계산기 제작



```

//resultBox에 값 입력하기
let resultBox = document.getElementById('resultBox'); //결과값 입력할 박스
//덧셈, 뺄셈 계산과 입력
this.resultPMAB = function(res) {
  resultBox.innerHTML = "";
  resultBox.innerHTML += '<div class="inResultBox" id="inResultBox"></div>';
  let inResultBox = document.getElementById('inResultBox');
  i=0;
  while(i<this.mA.length) {
    j=0;
    while(j<this.mA[0].length) {
      inResultBox.innerHTML += '<div class="inResultM" type="number">'+res[i][j]+'</div>';
      j++;
    }
    i++;
    inResultBox.innerHTML += "<br>";
  }
}
//곱셈 계산과 입력
this.resultAB = function(res) {
  resultBox.innerHTML = "";
  resultBox.innerHTML += '<div class="inResultBox" id="inResultBox"></div>';
  let inResultBox = document.getElementById('inResultBox');
  i=0;
  while(i<this.mA.length) {
    j=0;
    while(j<this.mB[0].length) {
      inResultBox.innerHTML += '<div class="inResultM">'+res[i][j]+'</div>';
      j++;
    }
    i++;
    inResultBox.innerHTML += "<br>";
  }
}

```

계산기 객체 완성

```

//계산 실행
this.addEventListener('click',function(e) {
  switch(e.target.id){
    //덧셈 버튼 실행
    case "plusAB":
      if (matrixAX.value==matrixBX.value && matrixAY.value==matrixBY.value) {
        let matrixA = new makeMatrixBox(matrixAX.value, matrixAY.value, 0);
        let matrixB = new makeMatrixBox(matrixBX.value, matrixBY.value, 1);
        matrixA.tempArray();
        let resultmA = matrixA.makeArray();
        let resultmB = matrixB.makeArray();
        let plusmAxB = new matrixCalculator(resultmA, resultmB);
        let resultPlus = plusmAxB.plusAB();
        plusmAxB.resultPMAB(resultPlus);
      } else{
        displayAlert('계산할 수 없습니다. <br>행렬A와 행렬B의 행의 값과 열의 값이 각각 동일해야 합니다. <br>예시: 행렬A(3X2), 행렬B(3X2)');
      }
      break;
    //뺄셈 버튼 실행
    case "minusAB":
      if (matrixAX.value==matrixBX.value && matrixAY.value==matrixBY.value) {
        let matrixA = new makeMatrixBox(matrixAX.value, matrixAY.value, 0);
        let matrixB = new makeMatrixBox(matrixBX.value, matrixBY.value, 1);
        matrixA.tempArray();
        let resultmA = matrixA.makeArray();
        let resultmB = matrixB.makeArray();
        let minusmAxB = new matrixCalculator(resultmA, resultmB);
        let resultMinus = minusmAxB.minusAB();
        minusmAxB.resultPMAB(resultMinus);
      } else{
        displayAlert('계산할 수 없습니다. <br>행렬A와 행렬B의 행의 값과 열의 값이 각각 동일해야 합니다. <br>예시: 행렬A(3X2), 행렬B(3X2)');
      }
      break;
    //곱셈 버튼 실행
    case "multiAB":
      if (matrixBX.value == matrixAY.value) {
        let matrixA = new makeMatrixBox(matrixAX.value, matrixAY.value, 0);
        let matrixB = new makeMatrixBox(matrixBX.value, matrixBY.value, 1);
        matrixA.tempArray();
        let resultmA = matrixA.makeArray();
        let resultmB = matrixB.makeArray();
        let multimAxB = new matrixCalculator(resultmA, resultmB);
        let resultMultiAB = multimAxB.multiAB();
        multimAxB.resultAB(resultMultiAB);
      } else{
        displayAlert('계산할 수 없습니다. <br>행렬A의 열의 값과 행렬B의 행의 값이 동일해야 합니다. <br>예시: 행렬A(3X4), 행렬B(4X2)');
      }
      break;
  }
});

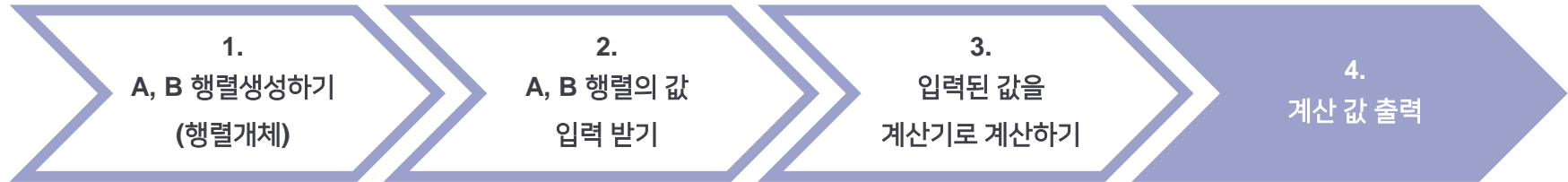
```

계산기 실행

URL : <http://pacer.kr/~c10st06/Portfolio/matrix/calculator.html>

구현 - 제작과정(코드)

행렬계산기 제작



계산기 실행 시
계산 불가일 때
덧셈, 뺄셈 알림창

계산할 수 없습니다.
행렬A와 행렬B의 행의 값과 열의 값이 각각 동일해야 합니다.
예시: 행렬A(3X2), 행렬B(3X2)

닫기

계산기 실행 시
계산 불가일 때
곱셈 알림창

계산할 수 없습니다.
행렬A의 열의 값과 행렬B의 행의 값이 동일해야 합니다.
예시: 행렬A(3X4), 행렬B(4X2)

닫기

URL : <http://pager.kr/~c10st06/Portfolio/matrix/calculator.html>

구현 - 사용방법

행렬계산기 제작

행렬계산기

행렬 A의 행: 열: 행렬 B의 행: 열: 행렬 생성

-10	36
11	-23

13	-23
3	-8

A+BA-BABRandom

-22	-58
74	-69

URL : <http://pager.kr/~c10st06/Portfolio/matrix/calculator.html>

구현 - 사용방법

행렬계산기 제작

- 행렬A, B 각각의 행과 열의 값을 입력 받아 행렬을 생성합니다. 최대 10X10, 기본값은 2X2로 지정하였습니다.
- 각 행렬에 값을 입력 한 후 버튼을 눌러주면 계산결과가 아래의 창에 나타납니다.

행렬계산기

행렬 A의 행: 열: 행렬 B의 행: 열: 행렬 생성

-10	36
11	-23

13	-23
3	-8

A+BA-BAB

Random

-22	-58
74	-69

계산결과창

- 왼쪽이 A행렬, 오른쪽이 B행렬이며 생성된 행렬에 값을 직접 입력하거나 random버튼을 이용하여 값을 임의로 넣을 수 있습니다. (음수값 포함).

URL : <http://pager.kr/~c10st06/Portfolio/matrix/calculator.html>

감사합니다