

HELLO ALPHA

프로젝트목표



Review



By my
self



Develop

INDEX

1. 프로젝트 주제 및 선정이유
2. 프로젝트 개발환경 구성도
3. 개발환경, 언어
4. 프로젝트 웹 화면 구성
5. 기능 및 핵심 코드 소개



1.프로젝트 주제

HELLOV T 100



VT100?

DEC (Digital Equipment Corporation)에서 개발 한 컴퓨터 비디오 터미널로 1978 년 8 월에 출시.

커서를 제어하고 다른 작업을 수행하기 위해 ANSI 이스케이프 코드를 완벽하게 지원하는 최초의 터미널 중 하나



왜 VT100인가?

주제선택이유

VT100 (CLASS)

동작(METHOD)

커서 제어
램프 제어

속성(FIELD)

램프
(터미널에 나타나는 색)

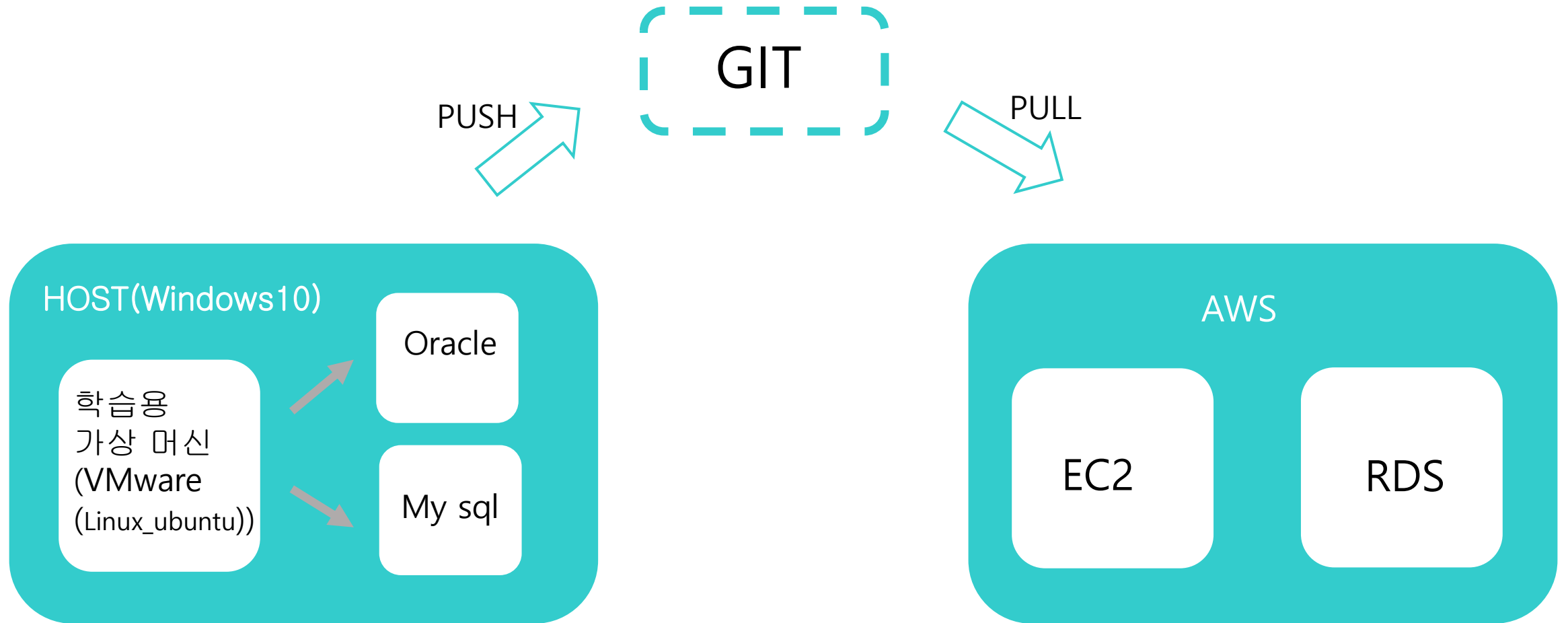


클래스 객체에 대한
기초적 개념 학습에
효과적!!



- 프로그래밍
역량 강화!
- 기초가 튼튼한
개발자로 성장!

2. 프로젝트 개발환경 구성도

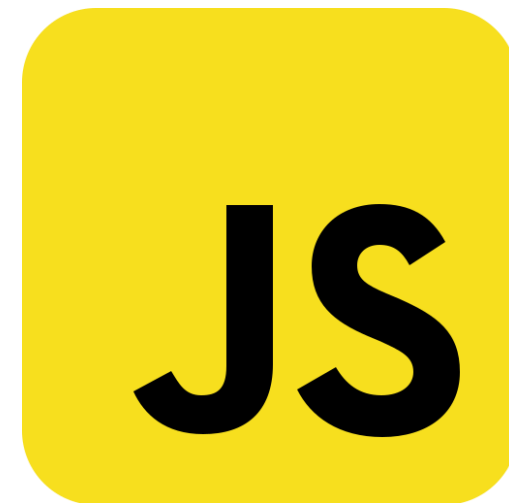




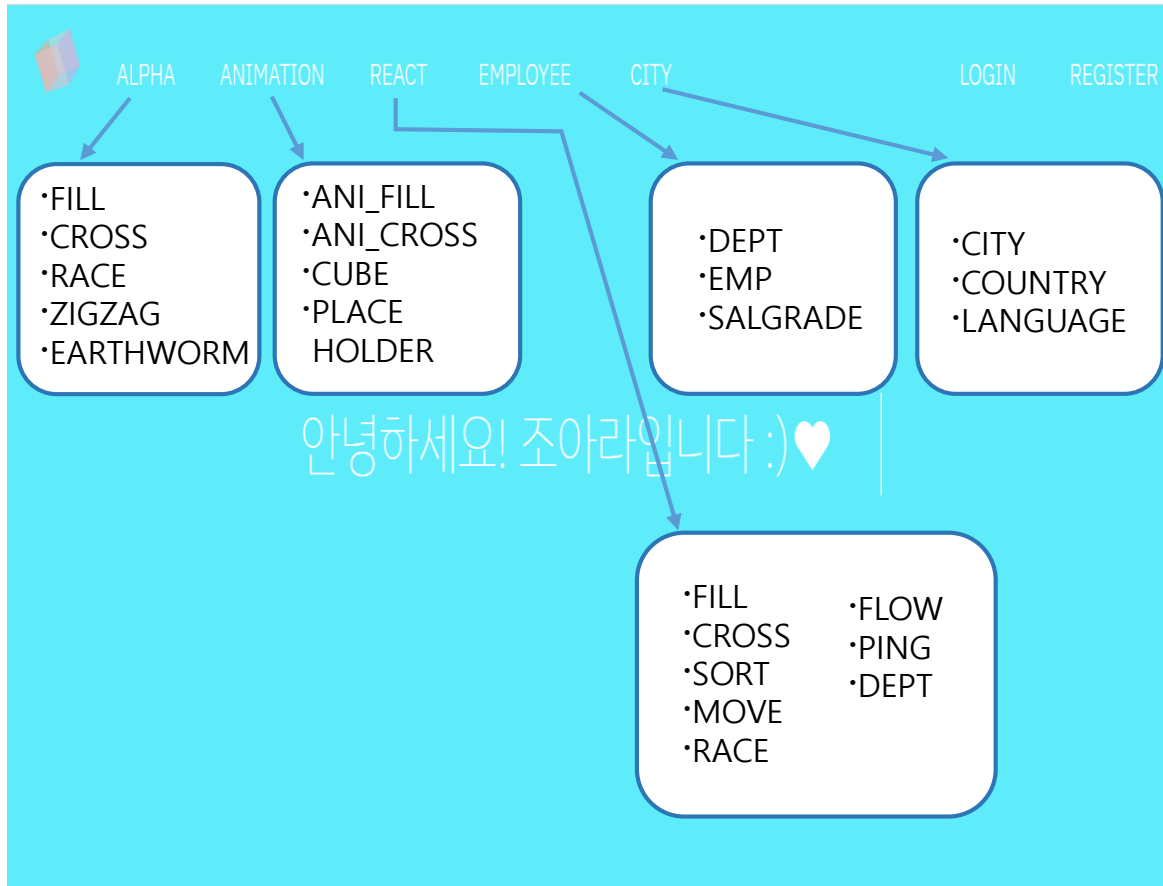
3.개발환경



Spring Boot



4. 프로젝트 웹 화면 구성



<nav>을 사용하여 다른 메뉴 접근을
쉽고 빠르게!



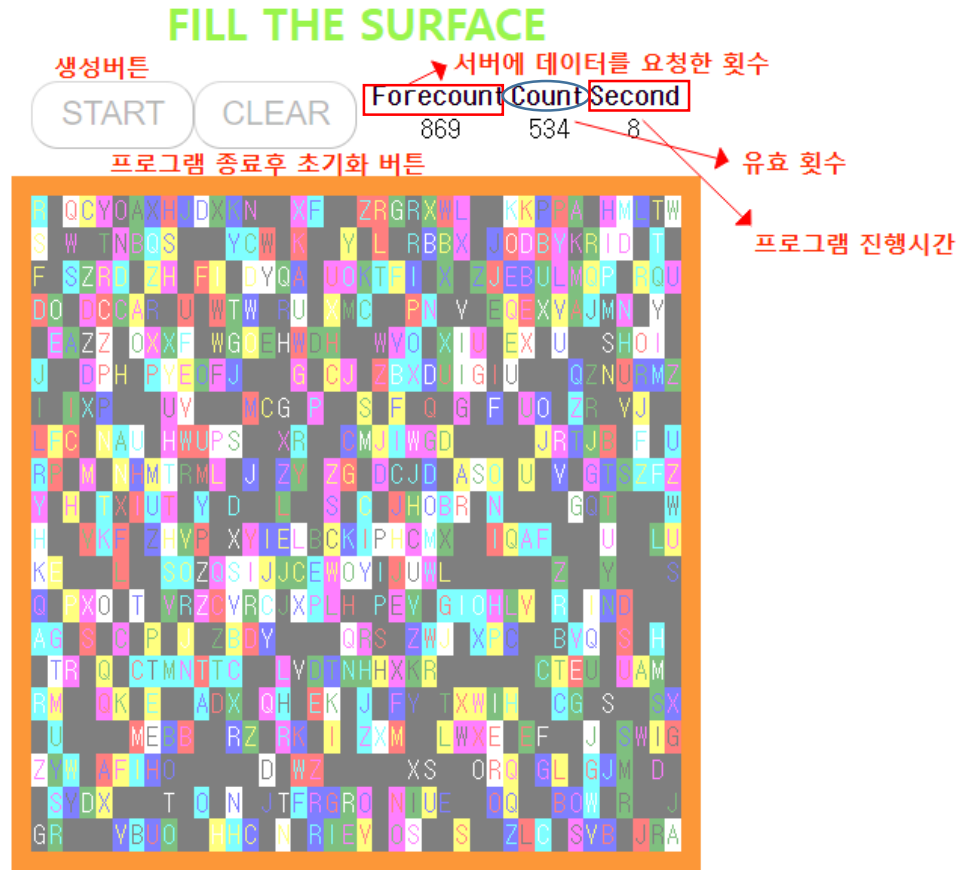
<iframe>을 사용하여 코드 중복 해소!!!

5. 기능 및 핵심 코드

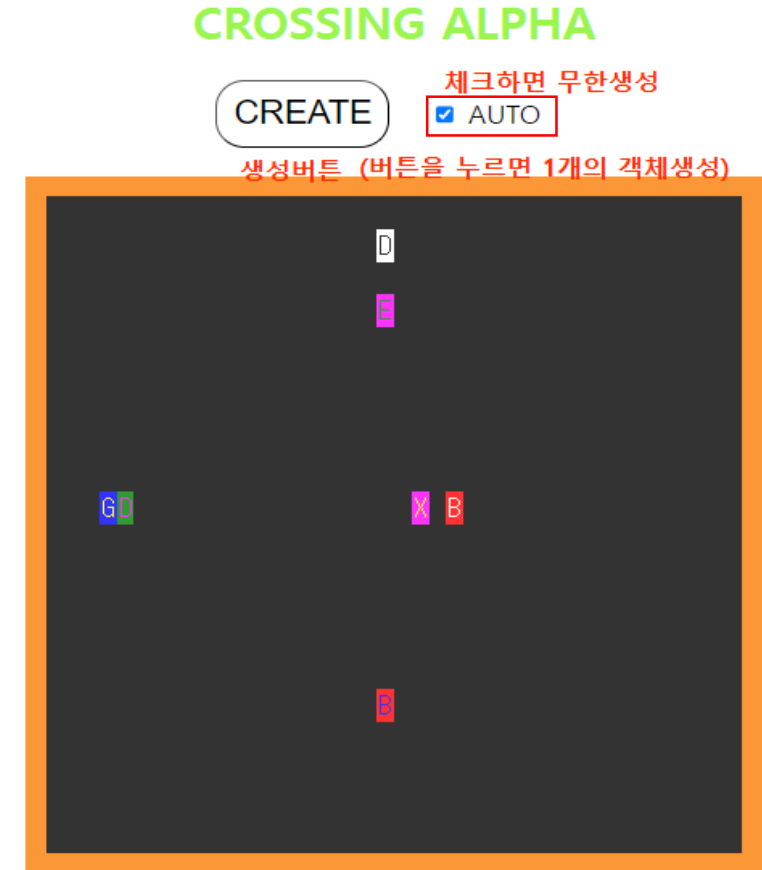
FIRST!!

- 프로그램명은 작동방식을 설명.
(예: **FILL**, ANI_**FILL**, REACT_**FILL**의 알고리즘은 동일함.)
- 버튼을 누르면 데이터(알파 객체) 생성.
 - 알파 객체 : 서버에서 가공한 데이터. 무작위로 생성됨.
(속성: LINE, COLUMN(웹 화면에 좌표로 기능함.),
FG(글자 색), BG(배경색), CH(알파벳))
 - 기본 화면 : 20 *40의 테이블

5. 프로젝트 기능 (FILL, CROSS)



- 테이블을 800개의 알파 객체로 채운다.
- 좌표가 중복된 데이터는 유효 횟수로 카운팅 되지 않는다.



- 객체는 테이블 가운데서 생성되어 가장자리를 향해 나아간다.
- 방향은 객체 생성 시점에 무작위로 설정된다.
- 테이블을 벗어나면 객체는 사라진다

5. 프로젝트 기능 (RACE)

RACING ALPHA

START

객체들이 테이블을 순회한 횟수의 총합

round count second

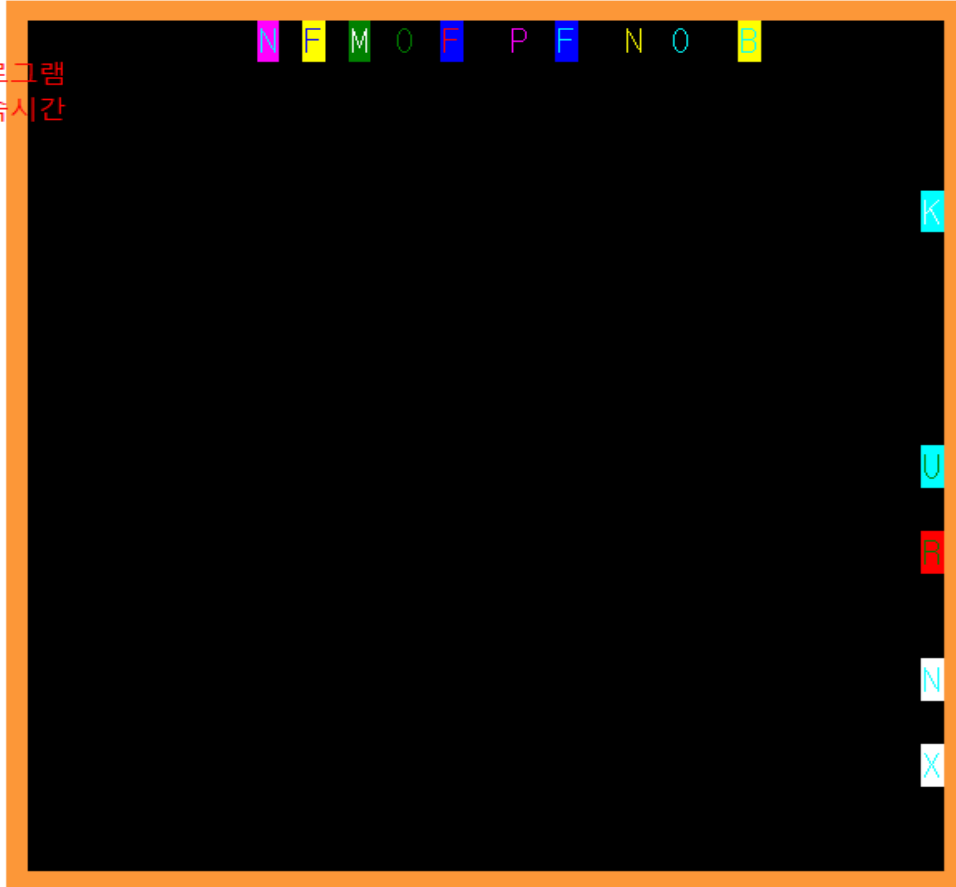
101

15

92

프로그램
지속시간

서버에 데이터를
요청한 횟수
(생성된 객체의 갯수)



각 객체가 테이블을
순회한 횟수

객체에 대한 정보

Count	roundingAlpha
7	K
7	B
7	O
7	N
7	F
7	P
7	F
7	O
7	M
7	F
7	N
6	X
6	N
6	R
6	U

객체 정보 테이블

- 객체가 테이블의 가장자리를 무한히 순회한다.

- 정보테이블을 클릭하면 해당하는 객체가 테이블에서 사라진다.

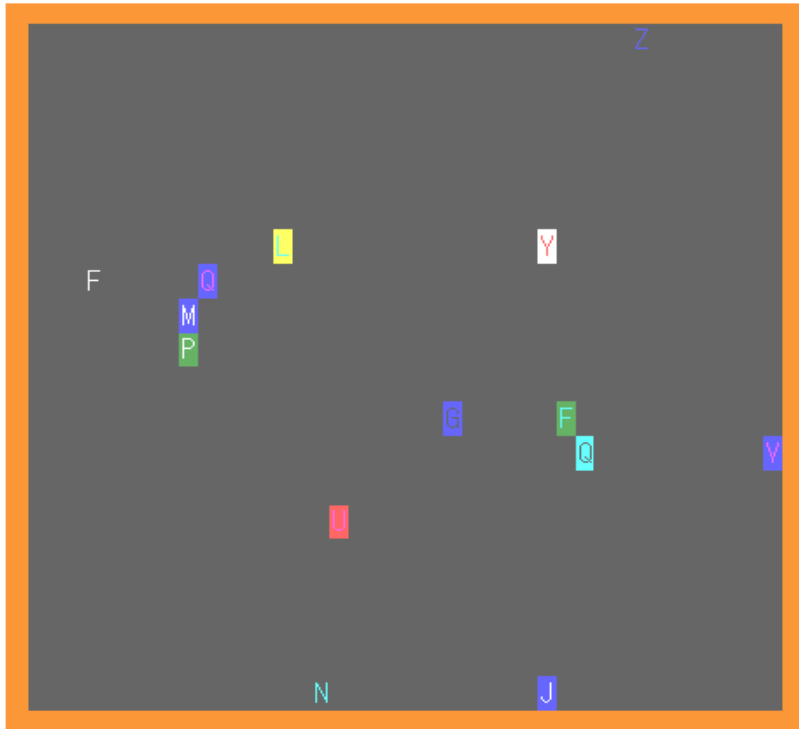
```
function list_click(e) {  
  let td = e.target;  
  console.log("targetsibling : ", e.target.nextSibling.innerText);  
  num = parseInt(e.target.nextSibling.innerText);  
  console.log(num);  
  index=td.indexOf(num);  
  console.log(a[index]);  
  let d=$('#table1')[0].rows[a[index].line-1].cells[a[index].column-1];  
  d.style.visibility='hidden';  
  td.remove();  
}
```

5. 프로젝트 기능 (ZIGZAG,EARTHWORM)

ZIGZAG ALPHA

CREATE

생성버튼

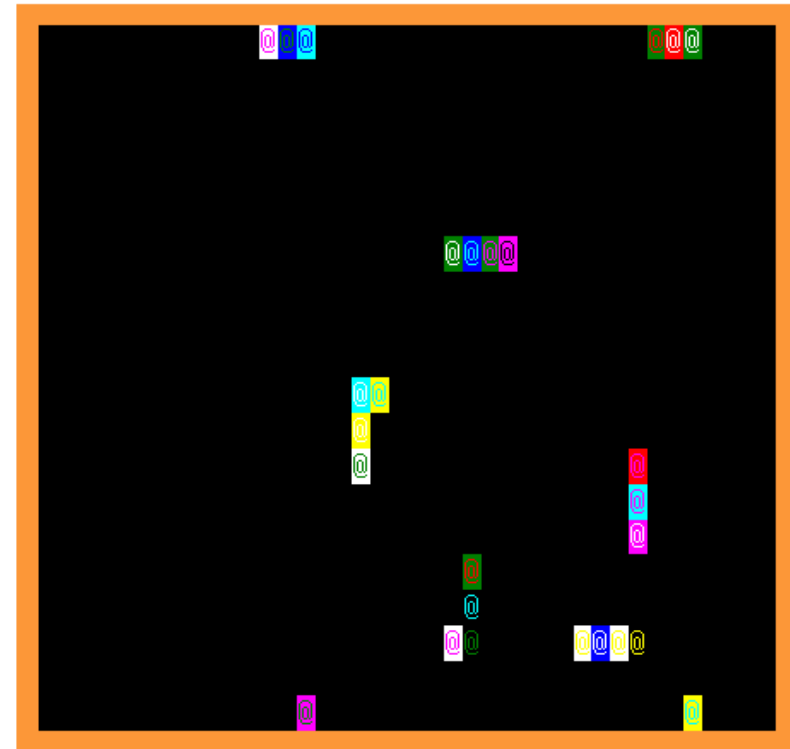


- 알파 객체가 자유자재로 움직임.
- 조건 처리로 인하여 테이블을 벗어나지 못함.

EARTHWORM

CREATE

생성버튼



- 4개의 객체가 한 몸이 되어 지렁이처럼 자유자재로 움직임.
- 데이터가 서버가 아닌, 웹에서 생성됨.

5. 프로젝트 핵심 코드(ALPHA)

```
let xhr = new XMLHttpRequest();
xhr.open('GET', '/alpha/data')
xhr.responseType='json';
```

**AJAX(XMLHttpRequest)방식으로
데이터 요청.**

```
xhr.onload = e =>{
  let alpha = xhr.response;
  console.log(alpha);

  let td = surface.rows[alpha.line-1].cells[alpha.column-1];

  if(td.style.color == 'black' && td.style.background == 'black') {
    count.innerText = ++count.innerText;
  }

  td.style.background = alpha.bg;
  td.style.color = alpha.fg;
  td.innerText = alpha.ch;
}
xhr.send();
}, 10);
```

```
$(function () {
  a=[];

  startbtn.onclick = e => {
    fetch('/alpha/data')
      .then(r => r.json())
      .then(alpha => {
        a.push(alpha);
        console.log("buttonEvent"+a);
        race(alpha);
      });
  }
});
```

**Jquery, AJAX(fetch,then)을
활용하여 데이터요청**

```
tb.onclick = e=>{ list_click(e); clearInterval(tid[num]);};
```

```
async run(){
  let response = await fetch('/alpha/data');
  this.alpha = await response.json();
  this.alpha.line=10;
  this.alpha.column=20;
```

```
this.show();
for(;;){
  await sleep(this.speed);

  if(!this.move()){
    break;
  }
}
```

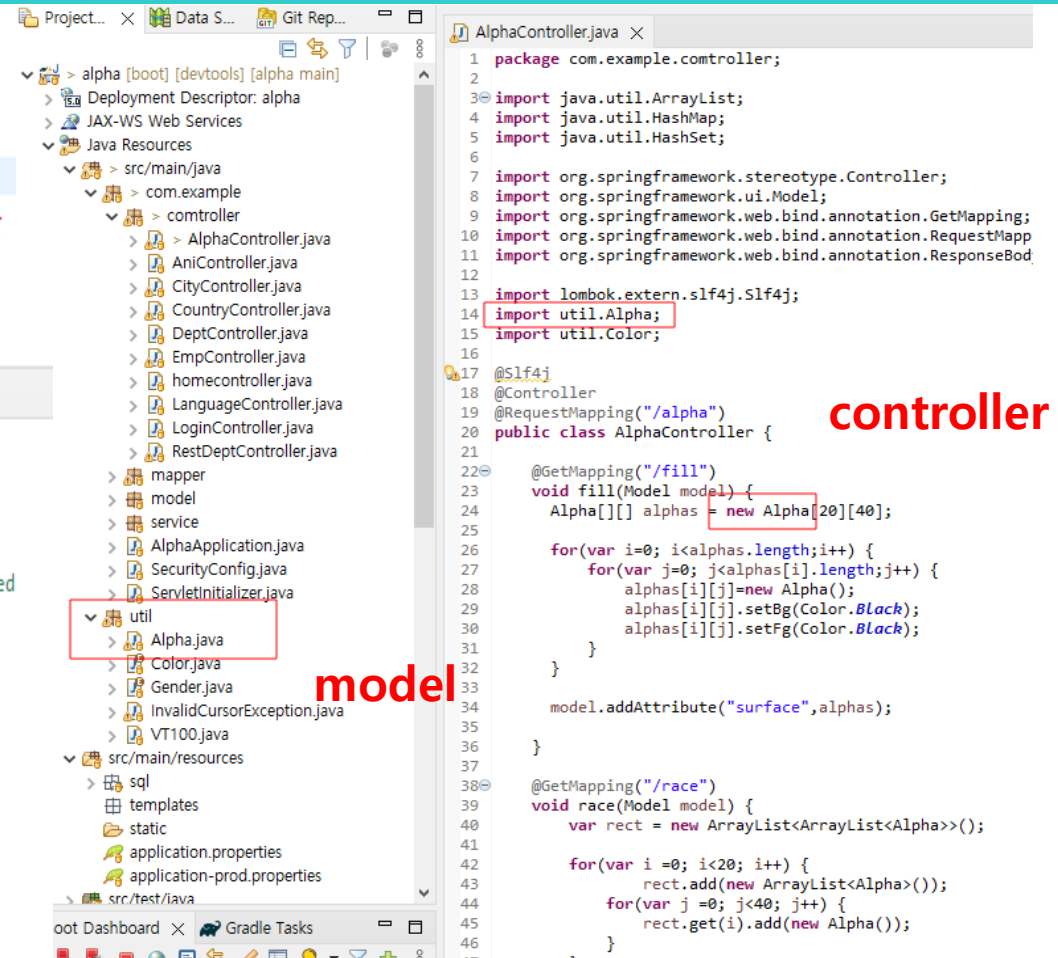
**AJAX(async - await) 를
활용하여 데이터요청**

```
}
@ResponseBody
@GetMapping("/data")
Alpha date() {
  return new Alpha();
}
```

**AJAX로 통신 시 json데이터를
반환하는 메소드**

```
AlphaController.java Alpha.java X
14 @AllArgsConstructor
15
16 public class Alpha {
17
18   static int count;
19   protected int line; //작식 객체는 접근할 수 있다. protected
20   protected int column;
21   protected Color fg;
22   protected Color bg;
23   protected char ch;
24
25   public Alpha() {
26     line = (int)(Math.random()*20 + 1);
27     column = (int)(Math.random()*40 + 1);
28
29     do {
30       fg = Color.values()[((int)(Math.random()*8))];
31       bg = Color.values()[((int)(Math.random()*8))];
32     }while(fg==bg);
33
34
35     ch=(char)(Math.random()*26+'A');
36
```

**Model이 되는
Alpha class**



controller

```
AlphaController.java X
1 package com.example.controller;
2
3 import java.util.ArrayList;
4 import java.util.HashMap;
5 import java.util.HashSet;
6
7 import org.springframework.stereotype.Controller;
8 import org.springframework.ui.Model;
9 import org.springframework.web.bind.annotation.GetMapping;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.ResponseBody;
12
13 import lombok.extern.slf4j.Slf4j;
14 import util.Alpha;
15 import util.Color;
16
17 @Slf4j
18 @Controller
19 @RequestMapping("/alpha")
20 public class AlphaController {
21
22   @GetMapping("/fill")
23   void fill(Model model) {
24     Alpha[] alphas = new Alpha[20][40];
25
26     for(var i=0; i<alphas.length;i++) {
27       for(var j=0; j<alphas[i].length;j++) {
28         alphas[i][j]=new Alpha();
29         alphas[i][j].setBg(Color.Black);
30         alphas[i][j].setFg(Color.Black);
31       }
32     }
33
34     model.addAttribute("surface", alphas);
35   }
36
37   @GetMapping("/race")
38   void race(Model model) {
39     var rect = new ArrayList<ArrayList<Alpha>>();
40
41     for(var i =0; i<20; i++) {
42       rect.add(new ArrayList<Alpha>());
43       for(var j =0; j<40; j++) {
44         rect.get(i).add(new Alpha());
45       }
46     }
47
```

MVC + AJAX

5. 프로젝트 기능 및 핵심 코드 (ANI_FILL)

AnimationFill-ALPHA

START

CLEAR

Forecount


Count

Second

454

353

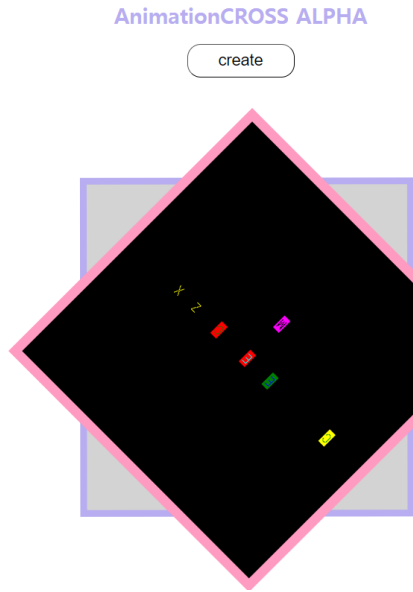
5



MVC + AJAX + CSS(POSITION, TRANSFORM, TRANSITION)

```
0
1 #surface td {
2   position: relative;
3   transition: left 3s, transform 4s;
4   transition-delay: 500ms;
5
6
7   top: 0px;
8   left: 0px;
9 }
10
11 if(td.style.color == 'black' && td.style.background == 'black') {
12   count.innerText = ++count.innerText;
13
14   td.style.left = "700px";
15   td.style.opacity = "0.5";
16   td.style.transform = `rotate(\${360*10}deg)`;
17 }
18
19 td.style.background = alpha.bg;
20 td.style.color = alpha.fg;
21 td.innerText = alpha.ch;
```


5. 프로젝트 기능, 핵심 코드 (ANI_CROSS, CUBE)

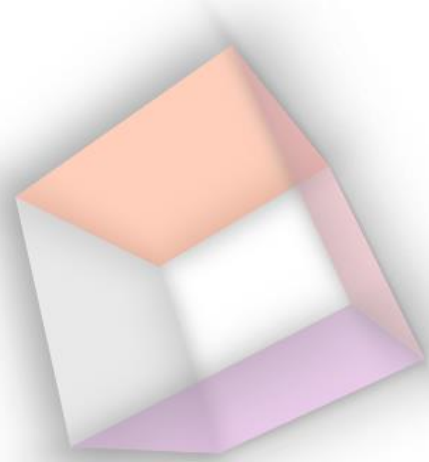


```
.configure-border-1 {  
  width: 478px;  
  height: 485px;  
  position: absolute;  
  background: lightgrey;  
  animation: configure-clockwise 3s ease-in-out 0s infinite alternate;  
  border: 10px solid #b7adf0;  
}  
  
@keyframes configure-clockwise {  
  0% {  
    transform: rotate(0);  
  }  
  25% {  
    transform: rotate(90deg);  
  }  
  50% {  
    transform: rotate(180deg);  
  }  
  75% {  
    transform: rotate(270deg);  
  }  
  100% {  
    transform: rotate(360deg);  
  }  
}
```

MVC + AJAX + CSS

(POSITION, TRANSFORM, ANIMATION)

CSS 3D 응용 큐브 만들기

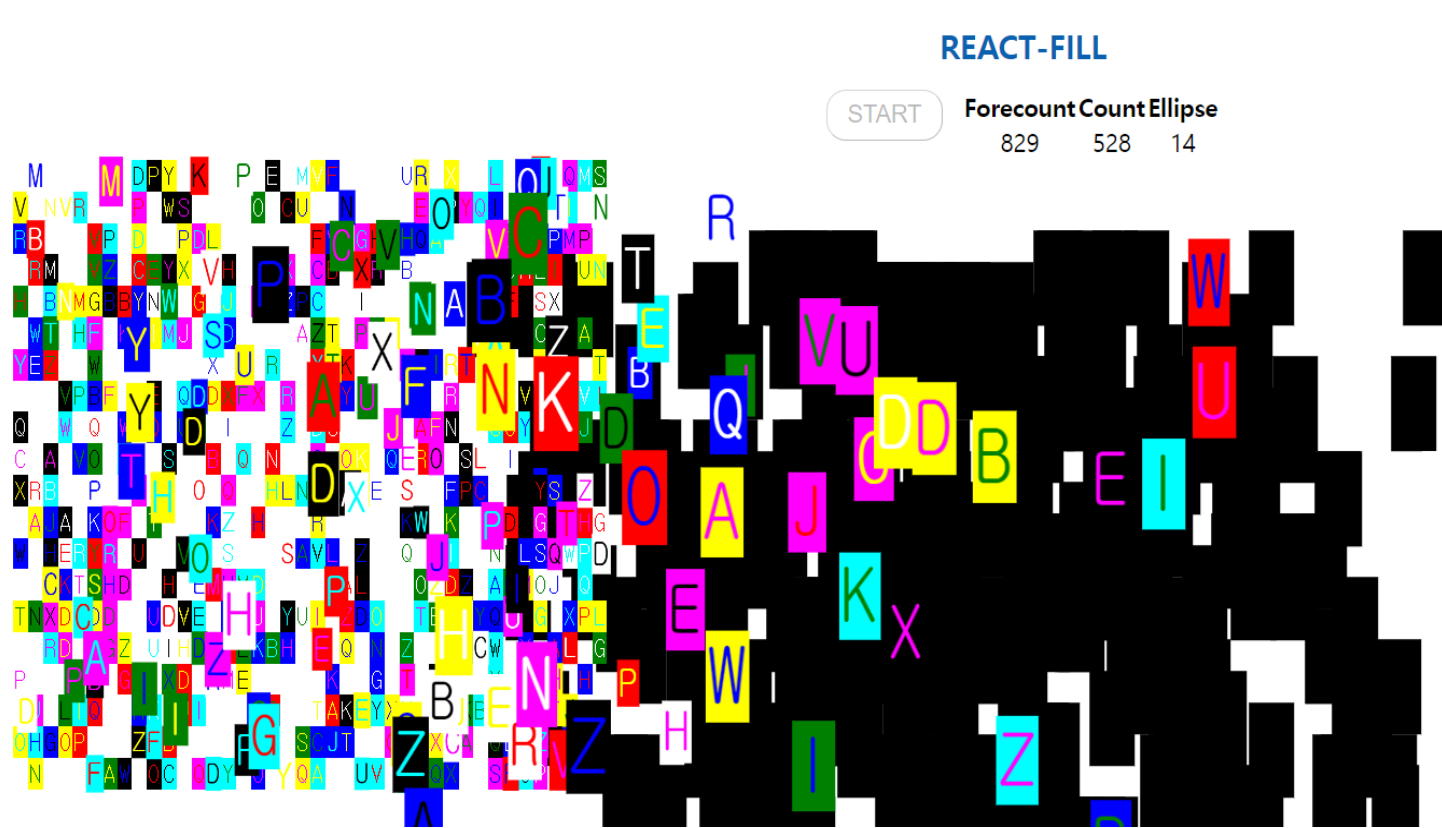


```
.front, .back, .left, .right, .top, .bottom {  
  position: absolute;  
  top: 0px;  
  left: 0px;  
  opacity: 0.5;  
  box-shadow: 0px 0px 50px 10px gray;  
}  
  
.front {  
  transform: translateZ(100px);  
  background: lightpink;  
}  
  
.back {  
  transform: rotateY(180deg) translateZ(100px);  
  background: lightgray;  
}  
  
.left {  
  transform: rotateY(-90deg) translateZ(100px);  
  background: lightsalmon;  
}
```

```
.cube {  
  display: inline-block;  
  transform-style: preserve-3d;  
  animation: rotate 10s infinite linear;  
}
```

CSS(POSITION, TRANSFORM, ANIMATION)

5. 프로젝트 기능, 핵심 코드 (REACT_FILL)



```
#surface td {  
  position: relative;  
  transition: left 3s, top 3s, transform 5s;  
}
```

```
class Fill extends Alpha {  
  left = 0;  
  top = 0;  
  zIndex = 0;  
  scale = 1;  
  
  constructor () {  
    super();  
    this.init();  
  }  
  
  init() {  
    this.left = parseInt(Math.random()*500+400);  
    this.top = 100;  
    this.zIndex = 0;  
    this.scale = 3;  
  }  
  
  clear() {  
    this.left = 0;  
    this.top = 0;  
    this.zIndex = 0;  
    this.scale = 1;  
  }  
}  
  
if(a.fg=='black' && a.bg == 'black') {  
  this.state.count++;  
  alpha.clear();  
  alpha.zIndex = zIndex++;  
  this.state.surface[alpha.line-1][alpha.column-1] = alpha;  
}
```

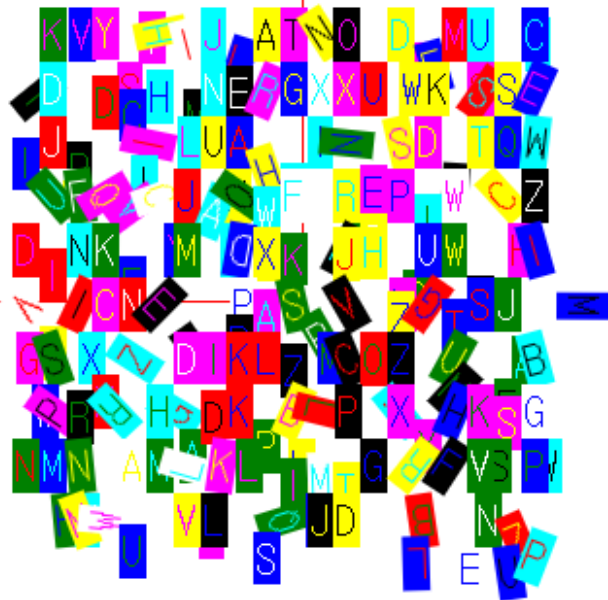
REACT + CSS(Position, Transition, Transform)

5. 프로젝트 기능, 핵심 코드 (REACT_MOVE)

REACT_MOVE

start

stop



```
class Move extends Alpha {
  /*핵심은 left, top, 좌표와 position : relative속성;
  left =0;
  top=0;
  speedLeft = 0;
  speedRotate = 0;
  speedTop = 0;
  degree = 0;
  constructor () {
    super();
    this.init();
  }

  init(){
    this.speedLeft = Math.random()*3;
    this.speedRotate = Math.random()*3;
    this.speedTop = Math.random()*3;
  }
}
```

```
1<#surface td {
2  position: relative;
3}
```

```
async move(){
  for(;;){
    let a = Math.random()*700;

    for(let i=0; i<10; i++){
      for(let j=0; j<20; j++){
        this.state.surface[i][j].left = a;
        this.state.surface[i][j].top = a;
        this.state.surface[i][j].degree = 360*parseFloat(Math.random()*10);
      }
    }
    this.forceUpdate();
    await sleep(3000);
  }
}
```

```
row.map((v,k) =>
  <td key={k} style={{
    color:v.fg,
    background:v.bg,
    left:v.left,
    top:v.top,
    transition:`left ${v.speedLeft}s,top ${v.speedTop}s, transform ${v.speedRotate}s`,
    transform:`rotate(${v.degree}deg)`,
  }}>{v.ch}</td>
)
```

REACT + CSS(position,TRANSITION,TRANSFORM)

5. 프로젝트 기능, 핵심 코드 (REACT_SORT)

REACT_SORT



```
class Sort extends Alpha {  
  left = 0;  
  top = 0;  
  scale = 1;  
  zIndex = 0;  
  
  constructor () {  
    super();  
    this.init();  
  }  
  
  init() {  
    this.left = 800;  
    this.top = 100;  
    this.scale = 10;  
  }  
}  
  
async sort1() {  
  let w = $("#surface td:first-child").outerWidth();  
  let h = $("#surface td:first-child").outerHeight();  
  console.log(`${w},${h}`)  
  
  for(let i=0; i<20; i++){  
    for(let j=0; j<40; j++){  
      this.state.surface[i][j].left=w*j;  
      this.state.surface[i][j].top=h*i;  
      this.state.surface[i][j].scale=1;  
      this.forceUpdate();  
      await sleep(100);  
    }  
  }  
  await sleep(2000);  
  await this.restore();  
}
```

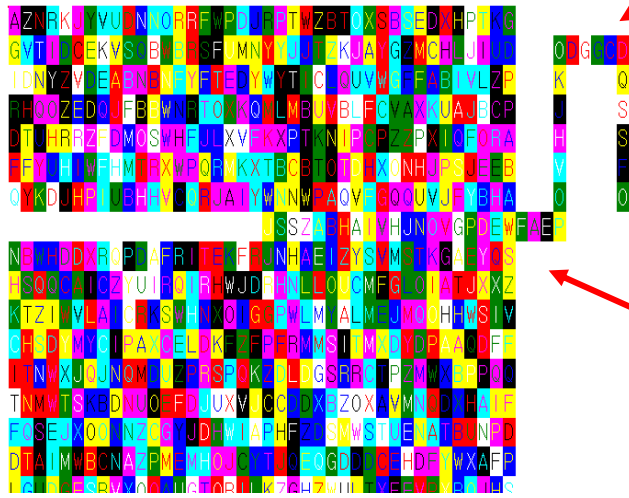
```
1< @surface td {  
2   position: absolute;  
3   transition: left 2s, top 2s, transform 2s;  
4 }
```

REACT + CSS(POSITION,TRANSITION,TRANSFORM)

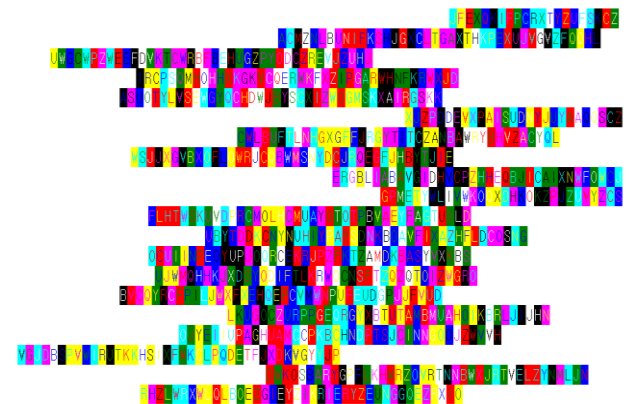
5. 프로젝트 기능, 핵심 코드 (REACT_FLOW,PING)

REACT_FLOW

LEFT RIGHT UP DOWN 7



REACT_PING



버튼이벤트로 제어

```
btnLeft_click(e){
  this.move(this.state.surface[this.state.rowNum], 4);
}

btnRight_click(e){
  this.move(this.state.surface[this.state.rowNum], 2);
}

btnUp_click(e){
  this.move(this.state.surface[this.state.rowNum], 1);
}

btnDown_click(e){
  this.move(this.state.surface[this.state.rowNum], 3);
}
```

몸통부분

```
for(let i=0; i<body.length-1; i++){
  body[i].setColumn(body[i+1].column);
  body[i].setLine(body[i+1].line);
}
```

방향을 움직이는 머리 부분

```
let head = body[body.length-1];

switch(direction){
  case 1: //top
    head.setLine(--head.line);
    break;
  case 2: //right
    head.setColumn(++head.column);
    break;
  case 3: //down
    head.setLine(++head.line);
    break;
  case 4: //left
    head.setColumn(--head.column);
    break;
}
```

```
async pingpong(body) {
  let isRight = true;
  let head = body[body.length-1];

  for(;;){
    if(isRight)
      this.move(body,2);
    else
      this.move(body,4);
    await sleep(head.speed);

    if(head.column == 100)
      isRight =false;
    else if(head.column == 0)
      isRight =true;
  }
}
```

```
<td style="color: red; background: blue; opacity:0.0;" >A</td>
```

```
#surface {
  position: relative;
  border: 1px solid red
}
```

테이블을 구성하는 코드

```
constructor (w,h) {
  super();
  this.w =w;
  this.h =h;

  setLine(line) {
    this.line = line;
    this.top = this.line * this.h;
  }

  setColumn(column) {
    this.column = column;
    this.left = this.column * this.w;
  }
}
```

```
class App extends React.Component {

  constructor() {
    super();
    let w = $('.collapse td').outerWidth();
    let h = $('.collapse td').outerHeight();

    for(let i=0; i<20; i++){
      this.state.surface[i]=[];
      for(let j=0; j<40;j++){
        let flow = new Flow(w,h);
        flow.position='absolute';
        flow.setLine(i);
        flow.setColumn(j);
        this.state.surface[i][j]=flow;
      }
    }
  }
}
```

REACT+CSS(POSITION)

5. 프로젝트 기능, 핵심 코드 (REACT_DEPT)

부서관리

부서목록

부서추가

deptno	dname	loc	수정	삭제
10	경리부	인천	수정	삭제
20	인사부	인천	수정	삭제
30	영업부	용인	수정	삭제
40	디자인부	대전	수정	삭제
50	개발부	준천	수정	삭제
60	식품사업부	부천	수정	삭제

부서관리

부서추가

목록으로이동

부서번호

부서이름

위치

부서추가

부서관리

부서수정

목록으로이동

부서번호

부서이름

위치

부서수정

부서관리

부서삭제

목록으로이동

부서번호

부서이름

지역

부서삭제

```
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
class App extends React.Component {
  render(){
    return(
      <Routes>
        <Route path="/rest/dept" element={<List/>}/>
        <Route path="/rest/dept/insert" element={<Insert/>}/>
        <Route path="/rest/dept/update" element={<Update/>}/>
        <Route path="/rest/dept/delete" element={<Delete/>}/>
      </Routes>
    )
  }
}
```

```
24
25 @Controller
26 @RequestMapping("/rest/dept")
27 public class RestDeptController {
28
29   DeptMapper mapper;
30
31   @GetMapping(headers = "Accept=text/html")
32   void view() {
33   }
34
35   ResponseEntity<Object> get(){
36
37   }
38
39   ResponseEntity<Object> post(@RequestBody Dept dept)
40
41   ResponseEntity<Object> put(@RequestBody Dept dept){
42
43   }
44
45   ResponseEntity<Object> delete(@RequestBody Dept dept)
46
47 }
```

```
class Insert extends React.Component{
  async onSubmit(e) {
    console.log("submit..");
    e.preventDefault();

    console.log(this.state.dept)

    let response = await fetch('/rest/dept',{
      method : "POST",
      headers : {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(this.state.dept)
    });

    if(response.ok){
      alert("부서추가성공! 부서목록으로 이동합니다.");
      this.props.navigate("/rest/dept");
    }else{
      let error = await response.json();
      alert("부서추가실패" + error.message);
    }

    let result = await response.json();

    console.log(result);

    state = {
      dept : {}
    }
  }
}
```

SPA, RESTFUL_API

5. 프로젝트 기능, 핵심 코드 (EMPLOYEE_DEPT,EMP)

부서목록

부서명으로 검색

부서번호	부서명	위치(지역)
10	경리부	인천
20	인사부	인천
30	영업부	용인
40	디자인부	대전
50	개발부	춘천
60	식품사업부	부천

```
<form action="/dept/list" class="text_center">
<input type="text" name="search" style="width: 150px; height: 20px;" />
<input id="input" type="submit" value="검색" />
</form>
```

```
@GetMapping("/dept/list")
String list(HttpServletRequest request) {
    var search = request.getParameter("search");
    var list = mapper.selectByName(search);
    request.setAttribute("list", list);
}
```

```
@Select("""
select *
from dept
where dname like '${search}%'
""")
List<Dept> selectByName(String search);
```

직원정보

사 번	직원명	성 별	직 급	직속상사사번	입사일	연 봉	커미션	부서번호
1001	김사항	F	사원	1013	2007-03-01	300.0		20
1002	한예슬	F	대리	1005	2007-04-02	250.0	80.0	30
1003	오지호	M	과장	1005	2005-02-10	500.0	100.0	30
1004	이병현	M	부장	1008	2003-09-02	600.0		20
1005	신동철	M	과장	1005	2005-04-07	450.0	200.0	30
1006	장동건	M	부장	1008	2003-10-09	480.0		30
1007	이문세	F	부장	1008	2004-01-08	520.0		10
1008	강우성	M	차장	1003	2004-03-08	500.0	0.0	30
1009	안성기	M	사장		1996-10-04	1000.0		20
1010	이병현	M	과장	1003	2005-04-07	500.0		10

```
@SelectProvider(type = SqlProvider.class, method = "selectBySearch")
List<Emp> selectBySearch(Integer deptno, String search);
```

```
public static class SqlProvider {
    public static String selectBySearch(Integer deptno, String search) {
```

```
var s = new SQL();
s.SELECT("*")
  .FROM("emp");

if(deptno != null) {
    s.WHERE(String.format("deptno=%d", deptno));
}
if(search != null && !search.trim().equals("")) {
    s.AND();
    s.WHERE(String.format("ename like '%s'" or job like '%s'", search, search));
}
```

sqlprovider를 사용하여 효율적으로 query문 처리

```
<form action="/emp/list" class="text_center">
<select name="deptno" style="width: 150px; height: 20px;">
<:choose>
<:forEach var="dept" items="${depts}">
</select>
<input type="text" name="search" style="width: 150px; height: 20px;" />
<input id="input" type="submit" value="검색" />
</form>
</div>
```

```
@GetMapping("/list")
void selectAll(Integer deptno, String search, HttpServletResponse response) {
    var list = empmapper.selectBySearch(deptno, search);
    request.setAttribute("list", list);
}
```

MVC PATTERN

5. 프로젝트 기능, 핵심 코드 (EMPLOYEE_SALGRADE)

직원별 급여정보

사 번	성 명	부서번호	직 급	급 여	급여등급
1001	김사랑	20	사원	300.0	2
1002	한예슬	30	대리	250.0	2
1003	오지호	30	과장	500.0	3
1004	이병현	20	부장	600.0	4
1005	신동협	30	과장	450.0	3
1006	장동건	30	부장	480.0	3
1007	이문세	10	부장	520.0	3
1008	감우성	30	차장	500.0	3
1009	안성기	20	사장	1000.0	5
1010	이병현	10	과장	500.0	3
1011	조항기	30	사원	280.0	2
1012	강혜정	20	사원	300.0	2
1013	박중훈	20	부장	560.0	3

급여등급안내

grade	losal	hisal
1	100	199
2	200	399
3	400	599
4	600	799
5	800	9999

Result Grid

empno	ename	gender	job	mgr	hiredate	sal	comm	deptno
1001	김사랑	F	사원	1013	2007-03-01	300.00	NULL	20
1002	한예슬	F	대리	1005	2007-04-02	250.00	80.00	30
1003	오지호	M	과장	1005	2005-02-10	500.00	100.00	30
1004	이병현	M	부장	1008	2003-09-02	600.00	NULL	20
1005	신동협	M	과장	1005	2005-04-07	450.00	200.00	30
1006	장동건	M	부장	1008	2003-10-09	480.00	NULL	30
1007	이문세	F	부장	1008	2004-01-08	520.00	NULL	10
1008	감우성	M	차장	1003	2004-03-08	500.00	0.00	30
1009	안성기	M	사장	NULL	1996-10-04	1000.00	NULL	20
1010	이병현	M	과장	1003	2005-04-07	500.00	NULL	10
1011	조항기	F	사원	1007	2007-03-01	280.00	NULL	30
1012	강혜정	F	사원	1006	2007-08-09	300.00	NULL	20
1013	박중훈	M	부장	1003	2002-10-09	560.00	NULL	20
1014	조인성	M	사원	1006	2007-11-09	250.00	NULL	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Sal이 grade의 대상.
조건은 losal, hisal

grade	losal	hisal
1	100	199
2	200	399
3	400	599
4	600	799
5	800	9999

```

@Select("""
    select empno, ename, deptno, job, sal, grade
    from emp, salgrade
    where sal between losal and hisal
""")

```

List<Emp> selectGrade();

```

@Select("select * from salgrade")
List<Salgrade> selectGradelist();

```

NONEQUI JOIN

MVC PATTERN

```

Emp.java
1 package com.example.model;
2
3 import java.time.LocalDate;
4
5
6 @Data
7 @Builder
8 @NoArgsConstructor
9 @AllArgsConstructor
10 public class Emp {
11
12     Integer empno;
13     String ename;
14     Gender gender; //Enum
15     String job;
16     Integer mgr;
17     LocalDate hiredate; //localdate
18     Double sal; //double
19     Double comm; //double
20     Integer deptno;
21     Integer grade;
22 }

```

MY BATIS로
MAPPING

5. 프로젝트 기능, 핵심 코드 (CITY_CITY)

CITYLIST

6,400,000이상

ID/NO	NAME	COUNTRYCODE	
1	Kabul	AFG	Kabol
2	Qandahar	AFG	Qandahar
3	Herat	AFG	Herat
4	Mazar-e-Sharif	AFG	Balkh
5	Amsterdam	NLD	Noord-Holland
6	Rotterdam	NLD	Zuid-Holland
7	Haag	NLD	Zuid-Holland
8	Utrecht	NLD	Utrecht
9	Eindhoven	NLD	Noord-Brabant
10	Tilburg	NLD	Noord-Brabant

ABW
 [ABW] Aruba
 AFG
 [AFG] Afghanistan
 AGO
 [AGO] Angola
 AIA
 [AIA] Anguilla
 ALB
 [ALB] Albania
 AND
 [AND] Andorra
 ANT
 [ANT] Netherlands Antilles
 ARE
 [ARE] United Arab Emirates
 ARG
 [ARG] Argentina
 ARM

```

<form action="Xcit
<input type="range
<label><span id="
<input type="search
<datalist id="code
<:forEach var="c
<option value="
</forEach>
        
```

```
@GetMapping("/list")
String list(Long population, String code, Model model) {
    log.info("list...");
```

controller

```
List<City> list = null;  
list = citymapper.searchAll(code, population);
```

```
model.addAttribute("list", list);  
  
var codes = citymapper.selectCountryCodes();  
model.addAttribute("codes", codes);
```

mapper

```
@ServiceProvider(type = SqlProvider.class, method = "searchAll")
List<City> searchAll(String code, Long population);
public static class SqlProvider {
```

```
public static String searchAll(String code, Long population) {
```

```
var s = new SQL();
s.SELECT("*");
s.FROM("city");

if(code != null && !code.trim().equals("")) {
    s.WHERE(String.format("countryCode='%s'", code));
}
if(population != null) {
    s.WHERE(String.format("population >= %d", population))
}
```

```
return s.toString();
```

3

MVC PATTERN

5. 프로젝트 기능, 핵심 코드 (CITY_COUNTRY)

나라정보

North America | 전체 | 검색 | 처음으로

전체
North America
Asia
Africa
Europe
South America
Oceania
Antarctica

NAME	GOVERNMENT	FORM	GNP_OLD	GNP	LIFE_EXPECTANCY	POPULATION
1	Mohammad Omar	Islamic Emirate		5976.0	45.9	22720000
56	Jos	Republic	7984.0	6648.0	38.3	12878000
62	Elisabeth II	Dependent Territory of the UK		63.2	76.1	8000
34	Rexhep Mejdani	Republic	2500.0	3205.0	71.6	3401200
55		Parliamentary Coprincipality		1630.0	83.5	78000
33	Beatrix	Nonmetropolitan Territory of The Netherlands		1941.0	74.7	217000

나라정보

North America | 전체 | 검색 | 처음으로

전체
Caribbean
Central America
North America

COUNTRYNAME	CAPITAL	HEADOFSTATE	GOVERNMENT	FORM	GNP	LIFE_EXPECTANCY	POPULATION
129	Beatrix	Nonmetropolitan Territory of The Netherlands		793.0	828.0	78.4	103000
1	Mohammad Omar	Islamic Emirate		5976.0	45.9	22720000	
56	Jos	Republic		7984.0	6648.0	38.3	12878000
62	Elisabeth II	Dependent Territory of the UK		63.2	76.1	8000	
34	Rexhep Mejdani	Republic		2500.0	3205.0	71.6	3401200
55		Parliamentary Coprincipality		1630.0	83.5	78000	
33	Beatrix	Nonmetropolitan Territory of The Netherlands		1941.0	74.7	217000	
65	Zayid bin Sultan al-Nahayan	Emirate Federation		36846.0	37966.0	74.1	2441000

```
let continent = document.forms[0].continent;
continent.onChange = function(){
    console.log(this.value);
    let xhr = new XMLHttpRequest();
    xhr.open('GET', `/country/region?continent=${this.value}`);
    xhr.send();
    xhr.onload = e => {
        setRegions(JSON.parse(xhr.responseText));
    }
}
```

```
function setRegions(regions){
    console.log(regions);
    let region = document.forms[0].region;
    region.innerText="";
    let regionParam = new URL(location.href).searchParams.get("region");
    let option = document.createElement('option');
    option.value= '';
    option.innerText= '전체';
    if(regionParam == null || regionParam == ""){
        option.selected = true;
    }
    region.append(option);
    for(let re of regions){
        let option = document.createElement('option');
        option.value= re;
        option.innerText= re;
        if(re == regionParam){
            option.selected= true;
        }
        region.append(option);
    }
}
```

```
@Mapper
public interface CountryMapper {

    @Select("select * from country")
    List<Country> selectAll();

    @Select("select distinct continent from country")
    List<String> selectContinent();

    @Select("select distinct region from country where continent =#{continent} ,jdbcType = VARCHAR")
    List<String> selectRegions(String continent);

    @Select("select * from country where continent = #{continent}")
    List<Country> selectByContinent(String continent);

    @Select("select * from country where continent = #{continent} and region = #{region}")
    List<Country> selectByContinentAndRegion(@Param("continent") String continent, @Param("region") String region);
}
```

mapper

```
@Autowired
CountryMapper mapper;

@GetMapping("/list")
void list(String continent, String region, Model model) {
    List<Country> list = null;
    if(continent == null || continent.trim().equals("")) {
        list = mapper.selectAll();
    }
    else {
        if (region == null || region.trim().equals("")){
            list = mapper.selectByContinent(continent);
        }
        else {
            list = mapper.selectByContinentAndRegion(continent, region);
        }
    }
    model.addAttribute("list", list);
}
```

controller

MVC PATTERN

5. 프로젝트 기능, 핵심 코드 (CITY_LANGUAGE)

LANGUAGE

NO	국가코드	국가명	언어	공식언어확인	사용비율
1	ABW	Aruba	Dutch	T	5.3
2	ABW	Aruba	English	F	9.5
3	ABW	Aruba	Papiamentu	F	76.7
4	ABW	Aruba	Spanish	F	7.4
5	AFG	Afghanistan	Balochi	F	0.9
6	AFG	Afghanistan	Dari	T	32.1
7	AFG	Afghanistan	Pashto	T	52.4
8	AFG	Afghanistan	Turkmenian	F	1.9
9	AFG	Afghanistan	Uzbek	F	8.8
10	AGO	Angola	Ambo	F	2.4
11	AGO	Angola	Chokwe	F	4.2
12	AGO	Angola	Kongo	F	13.2
13	AGO	Angola	Luchazi	F	2.4

Code	Name	Continent	Region	SurfaceArea
ABW	Aruba	North America	Caribbean	193.00
AFG	Afghanistan	Asia	Southern and Central Asia	652090.00
AGO	Angola	Africa	Central Africa	1246700.00
AIA	Anguilla	North America	Caribbean	96.00
ALB	Albania	Europe	Southern Europe	28748.00
AND	Andorra	Europe	Southern Europe	468.00
ANT	Antigua and Barbuda	North America	Caribbean	800.00
ARE	United Arab Emirates	Asia	Middle East	83600.00
ARG	Argentina	South America	South America	2780400.00
ARM	Armenia	Asia	Southern and Central Asia	29800.00
ASM	American Samoa	Oceania	Western Pacific	199.00
ATA	Austria	Europe	Central Europe	13120000.00
ATF	French Polynesia	Oceania	Western Pacific	7780.00
ATG	Antigua and Barbuda	North America	Caribbean	442.00
AUS	Australia	Oceania	Southeastern Asia	7748200.00
AUT	Austria	Europe	Central Europe	83858.00
AZE	Azerbaijan	Asia	Southern and Central Asia	86600.00
BDI	Burundi	Africa	Central Africa	27834.00

countryCode	Language	IsOfficial	Percentage
ABW	Dutch	T	5.3
ABW	English	F	9.5
ABW	Papiamentu	F	76.7
ABW	Spanish	F	7.4
AFG	Balochi	F	0.9
AFG	Dari	T	32.1
AFG	Pashto	T	52.4
AFG	Turkmenian	F	1.9
AFG	Uzbek	F	8.8
AGO	Ambo	F	2.4
AGO	Chokwe	F	4.2
AGO	Kongo	F	13.2
AGO	Luchazi	F	2.4
AGO	Luimbe-ng...	F	5.5
AGO	Luvale	F	3.6
AGO	Mbundu	F	21.7
AGO	Nyaneka-n...	F	5.8

```
10 @NoArgsConstructor
11 @AllArgsConstructor
12 public class Language {
13     String countryCode;
14     String language;
15     String isOfficial;
16     Double percentage;
17     String name;
18 }
19
```

MY BATIS로
MAPPING

```
@Mapper
public interface LanguageMapper {

    @Select("""
        select name, countrycode, language, IsOfficial, percentage
        from countrylanguage, country
        where code=countrycode
        """)
    List<Language> selectAll();
}
```

공통 요소로 테이블을
조인하여 화면에 출력함.

MVC PATTERN

5. 프로젝트 기능, 핵심 코드 (LOGIN,SIGN IN)

LOGIN

아이디

비밀번호

LOGIN

```
@Override
public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
    var member = mapper.selectById(username);

    if(member == null)
        throw new UsernameNotFoundException(username + "Not found");

    var auth = AuthorityUtils.createAuthorityList(member.getRoles());
    var user = new User(username, member.getPassword(), auth);

    return user;
}
```

spring security를 사용하여 login 처리

```
@Bean
SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
    http.csrf(csrf -> csrf.disable())
        .cors(cors -> cors.disable())
        .headers(headers -> headers.disable())
        .authorizeHttpRequests(authorizeHttpRequests -> authorizeHttpRequests
            .anyRequest().permitAll()
        )
        .formLogin(formLogin -> formLogin
            .loginPage("/login")
        )
        .logout(logout -> logout
            .logoutUrl("/logout")
            .logoutSuccessUrl("/")
        )
        .userDetailsService(memberService)
    ;
}
```

SIGN IN

아이디

비밀번호

SIGNIN

```
@PostMapping("/signup")
String register(String id, String password) {
    var list = mapper.insert(id, password, "USER");
    return "redirect:/login";
}
```

MVC PATTERN



프로젝트를 마치며..

개발자에게 필요한 진심.

어떤 마음으로 도전하는 것인가

- 개발하는 언어, 자료구조, DB에 대한 지식의 유무가 만드는 큰 차이.
- 내가 작성한 코드, 보고 또 보고 ..
- 완성도 높은 프로그램을 위해서는 진심을 다해 공부하고, 리뷰해야 ..

뿌리 깊은 나무가 되자.

- 기본 문법에 충실하자. JAVA, SCRIPT, HTML, CSS...
- 꼼꼼히 기록하자. Git의 기능 공부하고 적극활용, log 성실히 기록
- 꾸준히 공부하자. 프레임워크, 라이브러리를 쓸 때는 알고 쓰자!