



## 포트폴리오 설명:

6개월의 딥러닝 데이터 분석 커리큘럼 과정 중, 2개월 차에 진행한 중간 웹개발 프로젝트입니다.

저는 전체 서비스 기획을 담당했고, 팀장으로서 프로젝트 매니징을 담당했습니다.

기술적으로는 조 내의 finance팀에서, 데이터 db저장, 분석 함수 개발, 데이터 시각화, 페이지 구현을 맡았습니다.

레퍼지토리: <https://github.com/kang-narae/covidEmperor.git>



# 코로나의 황제 ...를 꿈꾸던 개미들 2조

최종발표

발표자: 강나래(팀장)



# 프로젝트의 분석 목표

## “코로나”

**분석 가설 1. 코로나 확진자율과 백신 접종율은 반비례하지 않을 것이다.**

분석 데이터: 서울시 지역별 확진자율, 백신접종율, 지역별 상권, 유동인구, 종교시설 등

**분석 가설 2: 코로나 진행 추이에 따른 주식 섹터별 등락 영향이 다를 것이다.**

ex) 미디어 섹터는 확진자 발발하며 약간 하락 후 백신 보급 이전 빠르게 반등했을 것이다, 여행 섹터는 확진자 발발 이후 크게 하락 후 백신 보급 이후 완만히 반등했을 것이다.

분석 데이터: 시기별 국내 확진자율 추이, 사망자율 추이, 백신접종률 추이, 기준금리, 국내 증시 상장 etf 등



# 과정상의 목표

api 활용 db 저장, 데이터 분석 및 시각화, 웹 페이지 구현

+깃 숙지

+팀원 간의 원만한 협력 (의 상하지 않기)



## 업무 분담

Web팀 (박혜정, 김재욱): 메인페이지, **css** 전반, 공지게시판, 뉴스 스크랩 게시판

Seoul팀 (최하진, 이승채): 데이터 **db**저장, 데이터 시각화, 페이지 구현

Finance팀 (강나래, 김수찬): 데이터 **db**저장, 분석 함수 구현, 데이터 시각화, 페이지 구현

팀장 : “어디..하고 있어요?” “덜어내자..” “우선순위.. 체크..” “일정..맞춰주세요..”

# 사전 수립 일정

20	21	22	23	24	25	26
					금융 : db저장, 시각화test	
				금융: 주제, 분석방향 설립		
				서울 : 오픈API신청		
				업무분담 완료		
				전체 : 팀별 사전조사, git 공부		
27	28	3월 1일	2	3	4	5
전체 : 팀별 사전조사,...	금융: DB 정리	금융 : 시각화, 분석	금융 : 페이지 구현 (차트, 분석 출력)	금융 : 마무리		
	서울 : db저장	서울 : 시각화	서울: 분석결과 정리/페이지구현	웹 : 최종 점검		
	웹 : 메인페이지, header		웹 : 게시판, 댓글	서울: 페이지 구현		
				웹 : 모든 앱 url 연동, css 적용		
6	7	8	9	10	11	12



main page (박혜정)

Covidiamond

Seoul

Finance

Boards

## Covidiamond

### 찬란히 빛나는 삶을 위하여

(주)왕제는 2022년 서울디지털인재개발원에서 시작된 그룹으로 서울시 행정 자치구 별 코로나 백신 접종과 확진 비율을 살펴보고 이상 결과를 보이는 구의 원인에 대해 분석하고자 하였으며, 시민들의 재테크 관심이 크게 높아진 요즘 트렌드에 맞춰 코로나 현황과 주식의 상관 관계 데이터를 이용자의 편의에 맞게 제공합니다.



# 메인 페이지, base.html

# main.css • base.html

static > css > # main.css > ...

```
1  * {
2    margin: 0;
3    padding: 0px;
4    box-sizing: border-box;
5    font-family: "IBM Plex Sans KR", sans-serif;
6  }
7
8  a {
9    text-decoration: none;
10   color: black;
11 }
12
13 ul li {
14   list-style: none;
15   /* border: 1px solid red; */
16 }
17
18 header {
19   position: fixed;
20   top: 0;
21   left: 0;
22   right: 0;
23   z-index: 100;
24 }
25
26 #topLogo {
27   margin: 0 auto 0 10%;
```

# main.css • base.html ✕

home > templates > base.html

```
20   {% block head-block %}
21   {% endblock head-block %}
22 </head>
23
24 <body>
25   {% comment %} <div> {% endcomment %}
26   <header>
27     {% comment %} <div> {% endcomment %}
28     <nav class="topMenu">
29       <ul>
30         <li id="topLogo"><a href="{% url 'main' %}">Covidiamond</a></li>
31         <li class="colorMenu"><a href="{% url 'seoul:view' %}">Seoul</a></li>
32         <li class="colorMenu"><a href="{% url 'fin:plot' %}">Finance</a></li>
33         <li class="colorMenu"><a href="#">Boards</a>
34           <ul>
35             <li><a href="{% url 'news:nlist' %}">n e w s</a></li>
36             <li><a href="{% url 'board:blist' %}">n o t i c e</a></li>
37             <li><a href="#">sc Board</a></li>
38           </ul>
39         </li>
40       </ul>
41     </nav>
42     {% comment %} </div> {% endcomment %}
43   </header>
44
45   <div id="wrapper">
46     <div class="main">
47       <!-- 메인페이지 필드-->
48       {% block main-block %}
49       {% endblock main-block %}
50     </div>
```

# css -header, footer



Covidiamond

Seoul

Finance

Boards

(주)황제

Web | 김재욱 박혜정 유승찬  
Seoul | 이승채 최하진  
Finance | 강나래 김수찬

08390 서울 구로구 디지털로26길 111  
서울디지털인재개발원  
02) 581 - 2293

Copyright (c) 2022.  
Covid Emperor & Covidiamond Inc.  
All Rights Reserved.

# css -header, footer

# main.css ● <> plot.html ✕

fin > templates > <> plot.html

```
1  {% extends 'base.html' %}
2  {% load static %}
3
4
5  {% block head-block %}
6  <link rel="stylesheet" type="text/css" href="{% static 'css/fin.css' %}">
7  <script type="text/javascript" src="{% static 'js/fin.js' %}"></script>
8  {% endblock head-block %}
9
10 <!--요기 아래 블럭 속에 쓰면 메인 홈페이지에 바로 나옴-->
11 {% block fin-block %}
12 <div id='fbody'>
13     <div id = 'fhead'>
14         <div id = 'ftitle'>
15             <h1>코로나 진행 추이에 따른 주가 변동 </h1>
16         </div>
17         <div id = 'finfo'>
18             <h3>좌측의 y축은 코로나 발생 직전인 2020년 1월 20일의 주가를 100% 기
19             우측의 y축은 코로나 발생추이에 따른 인구수입니다.<br>
```



**Seoul (최하진, 이승채)**

# api 통해 데이터 db에 저장

```
def patientsum(request):

    serial_key= "6453454b53746d6438337244774474"
    url= "http://openapi.seoul.go.kr:8088/{}/json/TbCorona19CountStatusJCG/1/800/".format(serial_key) # API 호출
    response = requests.get(url) # URL 불러오기
    contents = response.text # JSON TEXT 변환
    json_ob = json.loads(contents) # JSON 로딩
    publicData = json_ob["TbCorona19CountStatusJCG"]["row"] # ROW 컬럼 출력
    df = pd.json_normalize(publicData) # JSON을 PANDAS 변환

    df['DATETIME'] = pd.to_datetime(df['JCG_DT']) # 날짜 형식을 PANDAS DATETIME 형식으로 변환
    df['YEAR'] =df['DATETIME'].dt.year # 날짜 형식을 PANDAS YEAR 형식으로 변환
    df['DATE'] =df['DATETIME'].dt.date # 날짜 형식을 PANDAS DATE 형식으로 변환
    df['DATE'] = df['DATE'].apply(lambda x: x.strftime('%Y-%m-%d')) # 날짜 형식 변환
    mask = df["YEAR"].isin([2022,2021,2020]) # 2020 ~2021 날짜만 포함
    df = df[mask]
    df = df[:-3] # -3 제외하고 출력 (오류값 처리)

    df = df[["DATETIME", "GANGNAM", "GANGDONG", "GANGBUK", "GANGSEO",
            "GWANAK", "GWANGJIN", "GURO", "GEUMCHEON", "NOWON",
            "DOBONG", "DDM", "DONGJAK", "MAPO", "SDM", "SEOCHO",
            "SEONGDONG", "SEONGBUK", "SONGPA", "YANGCHEON",
            "YDP", "YONGSAN", "EP", "JONGNO", "JUNGGU", "JUNGNANG"]]

    rows=[] #레코드 변환
    for x in df.to_records(index=False):
        row=[ x[0],int(x[1]),int(x[2]),int(x[3]),int(x[4]),int(x[5]),int(x[6]),int(x[7]),int(x[8]),int(x[9]),
            int(x[10]),int(x[11]),int(x[12]),int(x[13]),int(x[14]),int(x[15]),int(x[16]),int(x[17]),int(x[18]),int(x[19]),
            int(x[20]),int(x[21]),int(x[22]),int(x[23]),int(x[24]),int(x[25])]
        # 전체 list에 담음.
        rows.append(row)
        #변수에 지정
    date = row[0]
    gangnam = row[1]
    gangdong = row[2]
    gangbuk = row[3]
    gangseo = row[4]
    gwanak = row[5]
    gwangjin = row[6]
    guro = row[7]
    geumcheon = row[8]
    nowon = row[9]
    dobong = row[10]
```

# 서울시 지역구를 컬럼으로 갖는 table

```
class Patientadd(models.Model):

    date = models.DateTimeField(null=False, default=0 ,max_length=100,primary_key=True)
    gangnam = models.IntegerField(null=False, default=0)
    gangdong = models.IntegerField(null=False, default=0)
    gangbuk = models.IntegerField(null=False, default=0)
    gangseo = models.IntegerField(null=False, default=0)
    gwanak = models.IntegerField(null=False, default=0)
    gwangjin = models.IntegerField(null=False, default=0)
    guro = models.IntegerField(null=False, default=0)
    geumcheon = models.IntegerField(null=False, default=0)
    nowon = models.IntegerField(null=False, default=0)
    dobong = models.IntegerField(null=False, default=0)
    ddm = models.IntegerField(null=False, default=0)
    dongjak = models.IntegerField(null=False, default=0)
    mapo = models.IntegerField(null=False, default=0)
    sdm = models.IntegerField(null=False, default=0)
    seocho = models.IntegerField(null=False, default=0)
    seongdong = models.IntegerField(null=False, default=0)
    seongbuk = models.IntegerField(null=False, default=0)
    songpa = models.IntegerField(null=False, default=0)
    yangcheon = models.IntegerField(null=False, default=0)
    ydp = models.IntegerField(null=False, default=0)
    yongsan = models.IntegerField(null=False, default=0)
    ep = models.IntegerField(null=False, default=0)
    jongno = models.IntegerField(null=False, default=0)
    junggu = models.IntegerField(null=False, default=0)
    jungnang = models.IntegerField(null=False, default=0)
```

# 서울시/전국 코로나 확진자 표

127.0.0.1:8000/seoul/sview/

Covidiamond

Seoul

Finance

Boards

## 발생동향

### 서울시

(2022.03.08.00시 기준)

사망

신규확진자

누적확진자

2,607

37,484

1,133,589

### 대한민국

(2022.03.08.00시 기준)

사망

신규확진자

누적확진자

9,282

202,721

4,869,691

확진자 현황 ●

확진자 추이 ●

## 자치구별 확진자 현황

강남구	강동구	강북구	강서구	관악구	광진구	구로구	금천구	노원구	도봉구	동대문구	동작구	마포구
58,017	50,688	30,007	58,440	52,735	39,499	44,162	25,261	52,590	33,122	39,443	42,733	40,350
+1,845	+1,727	+973	+2,400	+1,686	+1,582	+1,598	+998	+1,548	+1,361	+1,359	+1,618	+1,500
서대문구	서초구	성동구	성북구	송파구	양천구	영등포구	용산구	은평구	종로구	중구	종랑구	



```

def sview(req):
    conn=ora.connect('ora_user/1234@localhost:1521/orcl')

    #구별누적확진자데이터
    qs = Patientsum.objects.all()
    object_df3 = pd.DataFrame.from_records(qs.values())
    #구별 일일 추가 확진자
    qs2 = Patientadd.objects.all()
    daily = pd.DataFrame.from_records(qs2.values())

    #전체/서울 신규,누적,사망자 데이터
    qs3 = Death.objects.all()
    totaldata = pd.DataFrame.from_records(qs3.values())

    #최신날짜로 정렬
    df = object_df3.sort_values(by=['date'], ascending=False)
    df2 = daily.sort_values(by=['date'], ascending=False)

    sumdata=[] #현재까지 누적 구별 확진자 리스트로(마포구까지)
    sumdata2=[] #현재까지 누적 구별 확진자 리스트로(서대문구~종로구)

    dailydata=[] #구별 일일 추가 확진자 리스트로(마포구까지)
    dailydata2=[] #구별 일일 추가 확진자 리스트로(서대문구~종로구)

    allcovid=[] #전국 신규,누적,사망자
    seoulcovid=[] #서울 신규,누적,사망자

    #날짜순으로 정렬했을 때 0행이 최신, 포문으로 컬럼값을 빈 리스트에 넣어서
    #html로 보내줌
    for i in range(13): ...

    for i in range(12): ...

    for i in range(3):
        alldata = format(totaldata.iloc[0][i+3],',')
        allcovid.append(alldata)
        alldata2 = format(totaldata.iloc[1][i+3],',')
        seoulcovid.append(alldata2)

    context = {'gudata':sumdata,'gudata2':sumdata2, 'dailydata':dailydata,
    print(allcovid)
    return render(req, 'sview.html', context)

```

## db의 데이터 불러와서 html에 전달

```

<div id="move-cont1" class="move-cont" style="display: block;">
    <h4>
        <span>자치구별 확진자 현황</span>
    </h4>
    <div></div>
    <div class="status-confirm" style="padding:0 0 20px;">
        <table class="">
            <caption>서울시 구별 확진자 현황</caption>
            <tbody>
                <tr>...
            </tr>
            <tr class="odd">
                {% for x in gudata %}
                <td>{{x}}</td>
                {% endfor %}
                {% comment %} <td>{{x[1]}}</td> {% endcomment %}
            </tr>
            <tr class="odd">
                {% for daily in dailydata %}
                <td class="today">+{{daily}}</td>
                {% endfor %}
            </tr>
            <tr>...
            </tr>
            <tr class="odd">
                {% for x in gudata2 %}
                <td>{{x}}</td>
                {% endfor %}
            </tr>
            <tr class="odd">
                {% for daily2 in dailydata2 %}
                <td class="today">+{{daily2}}</td>
                {% endfor %}
            </tr>
            </tbody>
        </table>
    </div>

```



# 확진자 현황과 확진자 추이 화면 전환

확진자 현황 ●

확진자 추이 ●

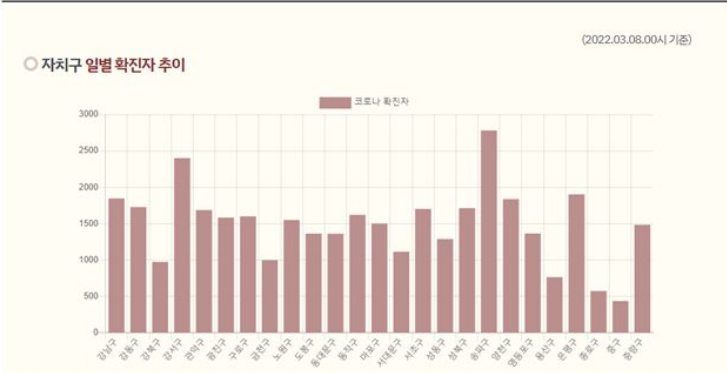
자치구별 확진자 현황

강남구	강동구	강북구	강서구	관악구	광진구	구로구	금천구	노원구	도봉구	동대문구	동작구	마포구
58,017	50,688	30,007	58,440	52,735	39,499	44,162	25,261	52,590	33,122	39,443	42,733	40,350
+1,845	+1,727	+973	+2,400	+1,686	+1,582	+1,598	+998	+1,548	+1,361	+1,359	+1,618	+1,500
서대문구	서초구	성동구	성북구	송파구	양천구	영등포구	용산구	은평구	종로구	중구	중랑구	
30,886	42,939	32,991	46,650	73,337	45,951	43,575	22,810	46,798	15,152	13,759	41,378	
+1,112	+1,701	+1,287	+1,712	+2,781	+1,836	+1,365	+763	+1,903	+572	+437	+1,485	

확진자 현황 ●

확진자 추이 ●

확진자 추이



## 확진자 현황과 확진자 추이 화면전환

```
<!--확진자 현황, 확진자 추이 div-->
<div class="move-tab move-tab-type2 move-tab-occur">
    <ul>
        {% comment %} 확진자현황 버튼을 클릭하게 되면 showCount1()명선 실행하면서
        "#move-cont1" div의 style.display="block"
        "#move-cont3" div의 style.display="none" 으로 바꿔서 하나의 div만 보여지게 해주고
        확진자추이 버튼을 클릭하면 showCount2()명선 실행되면서 block은none으로 none은 block으로
        <li>
            <button type="button" onclick="showCont1()" class="btn-tab2 on">확진자
                | <span>현황</span>
            </button>
        </li>
        <li>
            <button type="button" onclick="showCont3()" class="btn-tab3">확진자
                | <span>추이</span>
            </button>
        </li>
    </ul>
</div>
<div id="move-cont1" class="move-cont" style="display: block;">
    <h4>
        <span>자치구별 확진자 현황</span>
    </h4>
    <div></div>
    <div class="status-confirm" style="padding:0 0 20px;">
        <table class="">
            <caption>서울시 구별 확진자 현황</caption>
            <tbody>
                <tr>
                    <th scope="col">강남구</th>
                    <th scope="col">강동구</th>
```

```
function showCont1(){
    document.querySelector("#move-cont1").style.display="block"
    document.querySelector("#move-cont3").style.display="none"
}
function showCont3(){
    document.querySelector("#move-cont1").style.display="none"
    document.querySelector("#move-cont3").style.display="block"
    $.ajax({
        url: "/seoul/chartshow",
        type: "get",
        data: {},
        dataType: "json",
        success: function(data){
            // barChart();

            shapeChartData = {
                labels: ['강남구', '강동구', '강북구', '강서구', '관악구'],
                // labels: ['강남구', '강동구', '강북구'],
                datasets: [{
                    label: '코로나 확진자',
                    data: data.dailydata,
                    // data: [500, 300, 200],
                    borderColor: 'skyblue',
                    backgroundColor: 'skyblue',
                    borderWidth: 4
                }]
            }
            barChart()
        },
        error: function(){
            alert(data);
        })
    })
}
```

# 차트 그리기

```
function showCont3(){
    document.querySelector("#move-cont1").style.display="none"
    document.querySelector("#move-cont3").style.display="block"
    $.ajax({
        url: "/seoul/chartshow",
        type: "get",
        data: {},
        dataType: "json",
        success: function(data){
            // barChart();

            shapeChartData = {
                labels: ['강남구', '강동구', '강북구', '강서구', '관악구'],
                // labels: ['강남구', '강동구', '강북구'],
                datasets: [{
                    label: '코로나 확진자',
                    data: data.dailydata,
                    // data: [500, 300, 200],
                    borderColor: 'skyblue',
                    backgroundColor: 'skyblue',
                    borderWidth: 4
                }]
            }
            barChart()
        },
        error: function(){
            alert(data);
        }
    })
}
```

#차치구별 확진자 추이div에 차트 그리기

```
def chartshow(request):
    qs2 = Patientadd.objects.all()
    daily = pd.DataFrame.from_records(qs2.values())
    df2 = daily.sort_values(by=['date'], ascending=False)

    dailyData=[]
    #포문으로 전체 컬럼을 list로 담아서 html에 보내기
    for i in range(25):
        dailydatas = format(df2.iloc[0][i+1]) #format을 안써줘주면 TypeError가 뜬
        dailyData.append(dailydatas)
    print(dailydatas)
    context = {'dailydata':dailyData}
    return JsonResponse(context)
```

```
function barChart() {
    // id myChart의 html 삭제
    // $('#canvas2').remove();
    // id fchart의 자식으로 <canvas id="myChart"><canvas> 생성
    // $('#canvas2').append('<canvas id="canvas2"></canvas>');
    // var ctx = document.getElementById('canvas2').getContext('2d');
    var ctx = document.getElementById('canvas3');
    var myChart = new Chart(ctx, {
        type: 'bar',
        data: shapeChartData,
        options: {
            scales: {
                yAxes: [{
                    ticks: {
                        beginAtZero: true
                    }
                }]
            }
        }
    });
}
//createChart
```



**Finance(강나래, 김수찬)**

# 초기화면 차트

127.0.0.1:8000/fin/plot/

☆

Covidiamond

Seoul

Finance

Boards

## 코로나 진행 추이에 따른 주가 변동

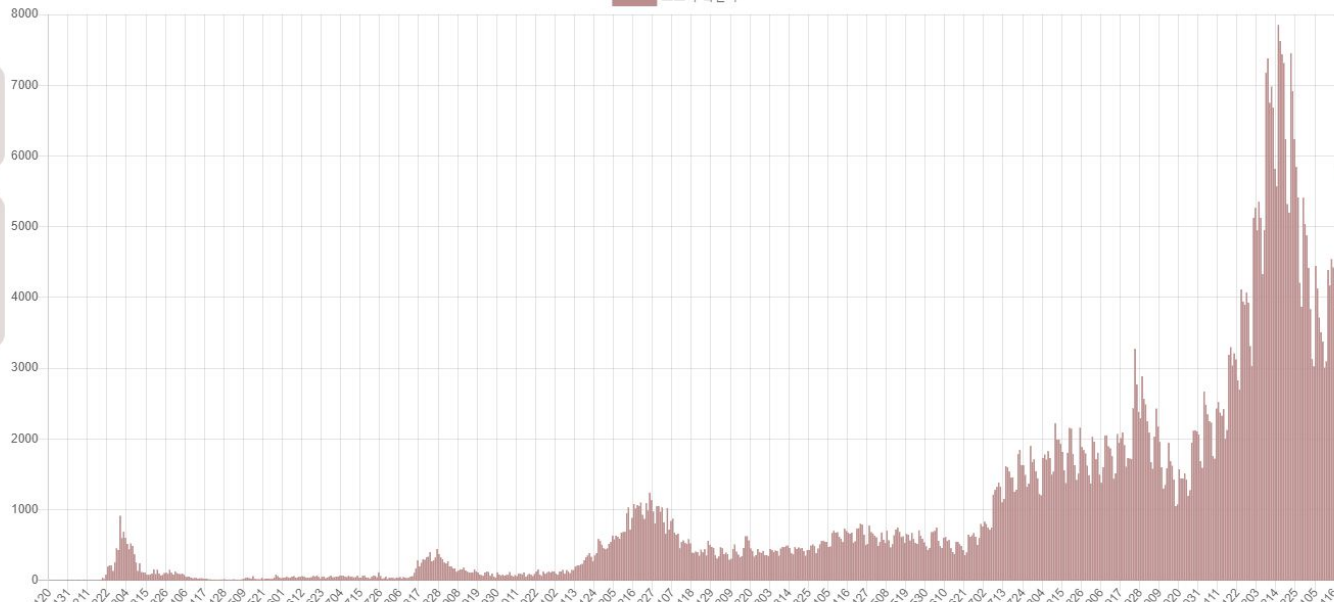
좌측의 y축은 코로나 발생 직전인 2020년 1월 20일의 주가를 100% 기준으로 합니다.

우측의 y축은 코로나 발생추이에 따른 인구수입니다.

x축은 국내 코로나 환자 첫 발생일인 2020년 1월 20일부터 2022년 1월 19일까지의 시기를 나타냅니다.

RESET

코로나 확진자



\*코로나 진행 동안의 주가 데이터

종목 코드(ticker)

GO

\*특정 일자의 주가 및 코로나 데이터

종목 코드(ticker)

20200120~20220119 사이의 날짜

GO

\*코로나 데이터

☒ 일별 신규 확진자

☐ 누적 확진자

☐ 일별 백신 접종자

☐ 누적 백신 접종자

Submit

\*확진자 추세

20200120~20220119 사이의 날짜

Submit

# 초기화면 차트

```
$(function(){
  // 자동으로 ajax 실행
  $.ajax({
    url: "/fin/DailyCovid",
    type: "get",
    data: {},
    dataType: "json",
    success: function(data){
      shapeChartData = {
        labels: data.intdate,
        datasets: [{
          label: '코로나 확진자',
          data: data.dailydecide,
          borderColor: 'skyblue',
          backgroundColor: 'skyblue',
          borderWidth: 4
        }]
      }
      barChart();
    }, error: function(){
      alert("실패");
    }
  });
});
```

```
def DailyCovid(request):
    covid = dailycovid.objects.all().values().order_by('intdate') #covid db 다 불러와서
    covid = pd.DataFrame(covid)
    covid.columns = ['strdate', 'intdate', 'deathCnt', 'decideCnt']
    # 앞의 것과 차이 앞에 없을 경우 0
    covid['dailydecide'] = covid['decideCnt'].diff().fillna(0) #covid 일일 신규 확진자 만들기
    intdate = covid['intdate'].values.tolist()
    dailydecide = covid['dailydecide'].values.tolist()
    context = {'intdate': intdate, 'dailydecide': dailydecide} #context에 태워보낼
    return JsonResponse(context)
```

```
function barChart() {
  // id myChart의 html 삭제
  $('#myChart').remove();
  // id fchart의 자식으로 <canvas id="myChart"></canvas> 생성
  $('#fchart').append('<canvas id="myChart"></canvas>');
  var ctx = document.getElementById('myChart').getContext('2d');
  var myChart = new Chart(ctx, {
    type: 'bar',
    data: shapeChartData,
    options: {
      scales: {
        yAxes: [{
          ticks: {
            beginAtZero: true
          }
        }]
      }
    }
  });
}
} //createChart
```



# db 모델링

covidPjt > settings.py > ...

```
88 # }
89
90 # sqlite3을 끄고 db를 oracle로 설정
91 DATABASES = {
92     'default': {
93         'ENGINE': 'django.db.backends.oracle',
94         'NAME': 'orcl',
95         'USER': 'covid_user',
96         'PASSWORD': '1234',
97         'HOST': 'localhost',
98         'PORT': '1521',
99     }
100 }
101
102
```

fin > models.py > ...

```
1 from django.db import models
2 from django.db import models
3 from django.forms import CharField
4
5
6 # Create your models here.
7
8 class dailycovid(models.Model):
9     strdate= models.CharField(max_length=30, primary_key=True)
10    intdate= models.IntegerField(default=0)
11    deathCnt = models.IntegerField( default=0)
12    decideCnt= models.IntegerField( default=0)
13
14    def __str__(self):
15        return self.strdate
16
17 class dailyvaccine(models.Model):
18     strdate= models.CharField(max_length=30, primary_key=True) # str 형식의 date
19     intdate= models.IntegerField(default=0) # int 형식의 date
20     firstCnt= models.IntegerField(default=0) # 해당 날짜 1차 접종자 수
21     secondCnt = models.IntegerField( default=0) # 해당 날짜 2차 접종자 수
22     thirdCnt= models.IntegerField( default=0) # 해당 날짜 3차 접종자 수
23     totalFirstCnt= models.IntegerField( default=0) # 누적 1차 접종자 수
24     totalSecondCnt= models.IntegerField( default=0) # 누적 2차 접종자 수
25     totalThirdCnt= models.IntegerField( default=0) # 누적 3차 접종자 수
26
27     def __str__(self):
28         return self.strdate
29
```

# 공공데이터 api로부터 db 저장

```
def vaccineData():
    service_key = 'JvU2PSq91h1mHsr1M5p7uq9GeNuR4KvrHcZ00jIb7unq51ANtM0HkaDA35GqYh3vhuWTXx1WrXqE8AZiqVSA%'
    vaccineData = [] # 전체 data 리스트(2중)
    for i in range(356): #페이지 별로
        url = f'http://api.odcloud.kr/api/15077756/v1/vaccine-stat?page={i}&perPage=18&serviceKey={service_key}'
        res = requests.get(url)
        contents = res.json()
        data = [] # 날짜별로 data를 넣을 리스트
        for i in range(18): #날짜 row 별로
            if contents['data'][i]['_sid0'] == '전국':
                data.append(contents['data'][i]['baseDate'][0:4]+contents['data'][i]['baseDate'][5:7]+contents['data'][i]['baseDate'][8:10])
                data.append(int(contents['data'][i]['baseDate'][0:4]+contents['data'][i]['baseDate'][5:7]+contents['data'][i]['baseDate'][8:10]))
                data.append(contents['data'][i]['firstCnt'])
                data.append(contents['data'][i]['secondCnt'])
                data.append(contents['data'][i]['thirdCnt'])
                data.append(contents['data'][i]['totalFirstCnt'])
                data.append(contents['data'][i]['totalSecondCnt'])
                data.append(contents['data'][i]['totalThirdCnt'])
        # writer.writerow(data)
        vaccineData.append(data)

    for i in vaccineData: #날짜 row 별로
        date = i[0]
        intdate = i[1]
        firstCnt = i[2]
        secondCnt = i[3]
        thirdCnt = i[4]
        totalFirstCnt = i[5]
        totalSecondCnt = i[6]
        totalThirdCnt = i[7]
        qs = dailyvaccine(strdate=date, intdate=intdate, firstCnt=firstCnt, secondCnt=secondCnt, thirdCnt=thirdCnt, totalFirstCnt=totalFirstCnt, totalSecondCnt=totalSecondCnt, totalThirdCnt=totalThirdCnt)
        qs.save()
```

주식데이터는 파이썬 라이브러리 사용

```
fin > views.py > covidData
412
413 def covidData(): #api 통해 데이터 읽어와서 dailycovid 모델에 저장
414
415     #인증키 넣은 url
416     url='http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19InfStateJson?serviceKey=...'
417
418     response= requests.get(url)
419     contents=response.text #받은 텍스트는 xml 형식.
420     dictionary = xmltodict.parse(contents) # xml을 dic형식으로 만들고
421     json_str= json.dumps(dictionary) #dic을 json 형식으로 만들었는데 이러면 json str타입이 나옴.
422     json_ob= json.loads(json_str) # json str을 다시 json dict 객체로 만듦.
423     covidData= json_ob['response']['body']['items']['item'] #해당 키 내의 벨류값 찾아옴
424
425     filename='확진 및 사망자.csv'
426     f= open(filename, 'w', encoding='utf-8-sig', newline='')
427     writer= csv.writer(f)
428     title= "date, intdate, deathCnt, decideCnt".split(',')
429     writer.writerow(title)
430
431     for i in covidData:
432         # rowdata=[]
433         date= i['stateDt']
434         deathCnt= i['deathCnt']
435         decideCnt= i['decideCnt']
436         qs = dailycovid(strdate=date,intdate=int(date), deathCnt=int(deathCnt), decideCnt=int(decideCnt))
437         qs.save()
438
```

```
def plot(request):
    # covidData()
    # vaccineData()
    qs = dailycovid.objects.all().values().order_by('intdate')
    # print(qs)
    return render(request, 'plot.html')
```



# 분석 함수

주식 ticker와 날짜 입력 시

\*특정 일자의 주가 및 코로나 데이터

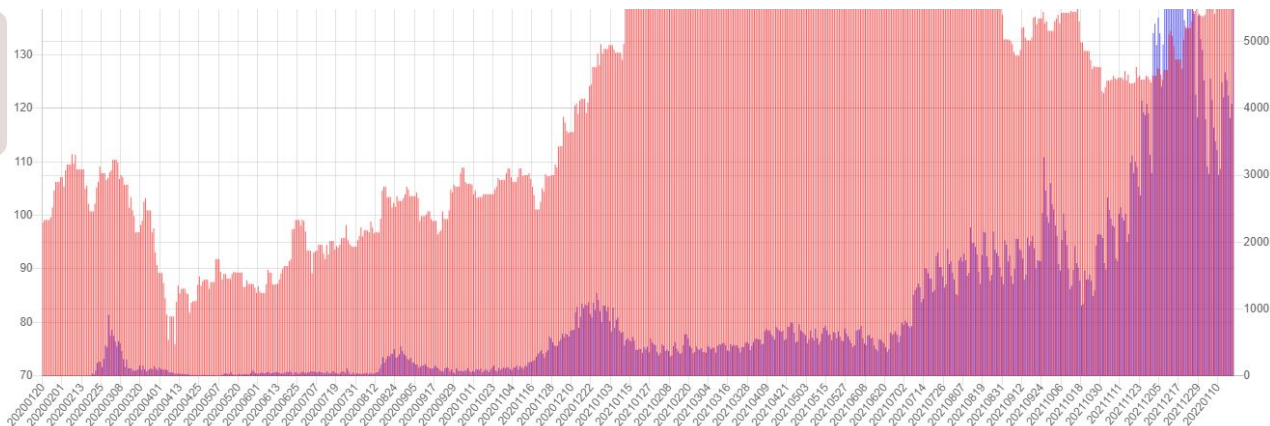
종목 코드(ticker)

005930

20200120~20220119 사이의 날짜

20200303

GO



주식 확인 일출시

Submit

\*확진자 추세

20200120~20220119 사이의 날짜

20200308

Submit

해당 날짜의 코로나 신규 확진자는 전일 대비 -13% 증감했습니다.

코로나 신규 확진자의 증감율이 -20%와-10% 사이 구간인 경우 해당 종목은 46.27%의 확률로 상승했습니다.

위의 그래프는, 코로나의 신규확진자 추세가 -20%와-10% 사이인 경우의 주가 증감율을 나타냅니다.

005930 종목이 코로나 진행 기간 동안 최저점을 찍은 날짜는 2020-03-23입니다.

코로나 발생 이전인 2020년 1월 1일을 기준으로 하였을 때, 해당 날의 하락율은 -23.42% 입니다.

해당 종목이 코로나 진행 기간 동안 최고점을 찍은 날짜는 2021-01-11입니다.

기준일과 비교했을 때 최고점 날의 상승률은 + 63.96% 입니다.

전국 코로나 확진율 : 0.01 지난 7일 평균 코로나 확진자 대비 확진자 증감률: 36.09%

# 분석 함수

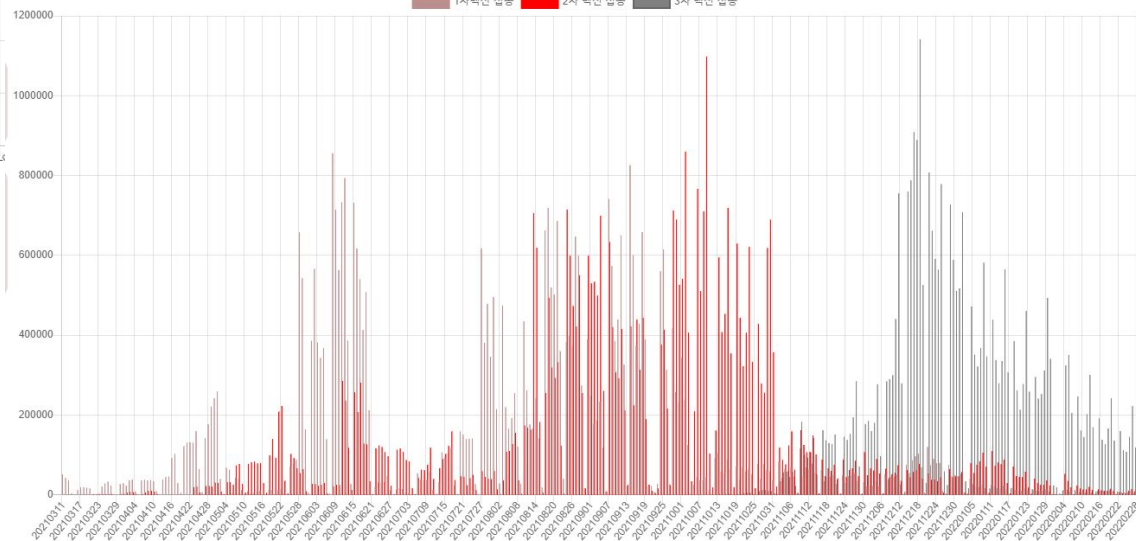
```
#종목 기준으로  
#언제 최저점? = 언제 반등?  
def lowpointday(ticker):  
    df = fdr.DataReader(ticker, '2020-01-01', '2022-01-19')  
    df.reset_index(inplace=True)  
    #close 값이 제일 낮은 row의 date 구하기  
    lowdayindex= df['Close'].argmin()  
    lowday = df.loc[lowdayindex]['Date']  
  
    c= ticker  
    a= " 종목이 코로나 진행 기간 동안 최저점을 찍은 날짜는 "  
    b= "입니다. <br> "  
  
    result=""  
    result= c+ a+ str(lowday)[0:10]+ b  
    return result
```

```
# 언제 oo과 크로스?  
def cross(ticker1, ticker2, startday, endday):  
    df1 = fdr.DataReader(ticker1, startday, endday)  
    df2 = fdr.DataReader(ticker2, startday, endday)  
  
    df1['percent']=df1['Close']/df1.iloc[0,1]*100  
    df2['percent']=df2['Close']/df2.iloc[0,1]*100  
#df1의 percent컬럼과 df2의 percent컬럼 비교.  
    df1['compare'] = df1['percent'] >= df2['percent']  
    df1higher= df1[df1['compare']]['Close'].count()  
    allday=df1['Close'].count()  
    df1higherpercent= df1higher/allday*100  
  
    return df1higherpercent.round(2)
```

```
def covidgoingon(date):    #코로나 누적 확진자율, #코로나 증가 세기, #백신접종율  
    #인자로날짜만 들어가고.  
    covid = dailycovid.objects.all().values().order_by('intdate')  
    covid = pd.DataFrame(covid)  
    covid.columns = ['strdate', 'intdate', 'deathCnt', 'decideCnt']  
    # 오늘 누적 확진자 - 어제 누적 확진자(diff()-> 하루 확진자, 값이 없을 경우 0)  
    covid['dailydecide']=covid['decideCnt'].diff().fillna(0)  
  
    # 입력 받은 날 구하기  
    todayCovid = covid['intdate'] == int(date)  
    # 7일의 확진자 합을 넣을 변수 설정  
    sevenCnt = 0  
    # 7일 간의 확진자 숫자를 구하기  
    for i in range(1,8):  
        # 자동 변수 생성  
        globals()['covid_'+str(i)] = covid['intdate'] == int(date) - i  
        globals()['covidCnt_'+str(i)] = int(covid[globals()[f'covid_{i}']][f'dailydecide'])  
        sevenCnt += globals()['covidCnt_'+str(i)]  
    sevenMean = sevenCnt/7  
    # 조건을 충족하는 데이터를 필터링하여 새로운 변수에 저장  
    toCovidCnt = int(covid[todayCovid]['decideCnt'])  
    # 전국의 누적 확진자 %  
    toCovidPer = round(int(covid[todayCovid]['decideCnt']/50000000*100,2)  
    covidCompare = round((toCovidCnt - sevenMean)/sevenMean*100,2)  
    context = {'toCovidPer':toCovidPer,'covidCompare':covidCompare}  
    return context
```

# 코로나 차트 변형

2020-01-01 증가 대비



\*코로나 데이터

- ☐ 일별 신규 확진자
- ☐ 누적 확진자
- ☒ 일별 백신 접종자
- ☐ 누적 백신 접종자

Submit

\*확진자 추세

20200120~20220119 사이의 날짜

20200308

Submit



# 코로나 데이터 라디오 버튼으로 불러오기

```


```
function barChart() {
  // id myChart의 html 삭제
  $('#myChart').remove();
  // id fchart의 자식으로 <canvas id="myChart"><canvas> 생성
  $('#fchart').append('<canvas id="myChart"></canvas>');
  var ctx = document.getElementById('myChart').getContext('2d');
  var myChart = new Chart(ctx, {
    type: 'bar',
    data: shapeChartData,
    options: {
      scales: {
        yAxes: [{
```



```
def TotalCovid(request):
    covid = dailycovid.objects.all().values().order_
    covid = pd.DataFrame(covid)
    covid.columns = ['strdate', 'intdate', 'deathCnt',
    intdate = covid['intdate'].values.tolist()
    decideCnt = covid['decideCnt'].values.tolist()
    context = {'intdate':intdate, 'decideCnt':decideC
    return JsonResponse(context)
```



```
$('#radioBtn').click(function () {
  // checked 상태의 radio의 값을 변수에 저장
  var radiovalue = $('input:radio[name=covid]:checked').val();
  // 해당 변수의 값이 무엇인가에 따라 다른 ajax 실행
  if(radiovalue == 'Total-Covid'){
    $.ajax({
      url:"/fin/TotalCovid",
      type:"get",
      data:{},
      dataType:"json",
      success:function(data){
        shapeChartData = {
          labels: data.intdate,
          datasets: [{
            label: '코로나 확진자',
            data: data.decideCnt,
            borderColor: 'skyblue',
            backgroundColor: 'skyblue',
            borderWidth: 4
          }]
        }
        barChart();
      },error:function(){
        alert("실패");
      }
    });
  }
```


```

# 티커와 일자(날짜)로 차트와 분석 불러오기

\*코로나 진행 동안의 주가 데이터

종목 코드(ticker)

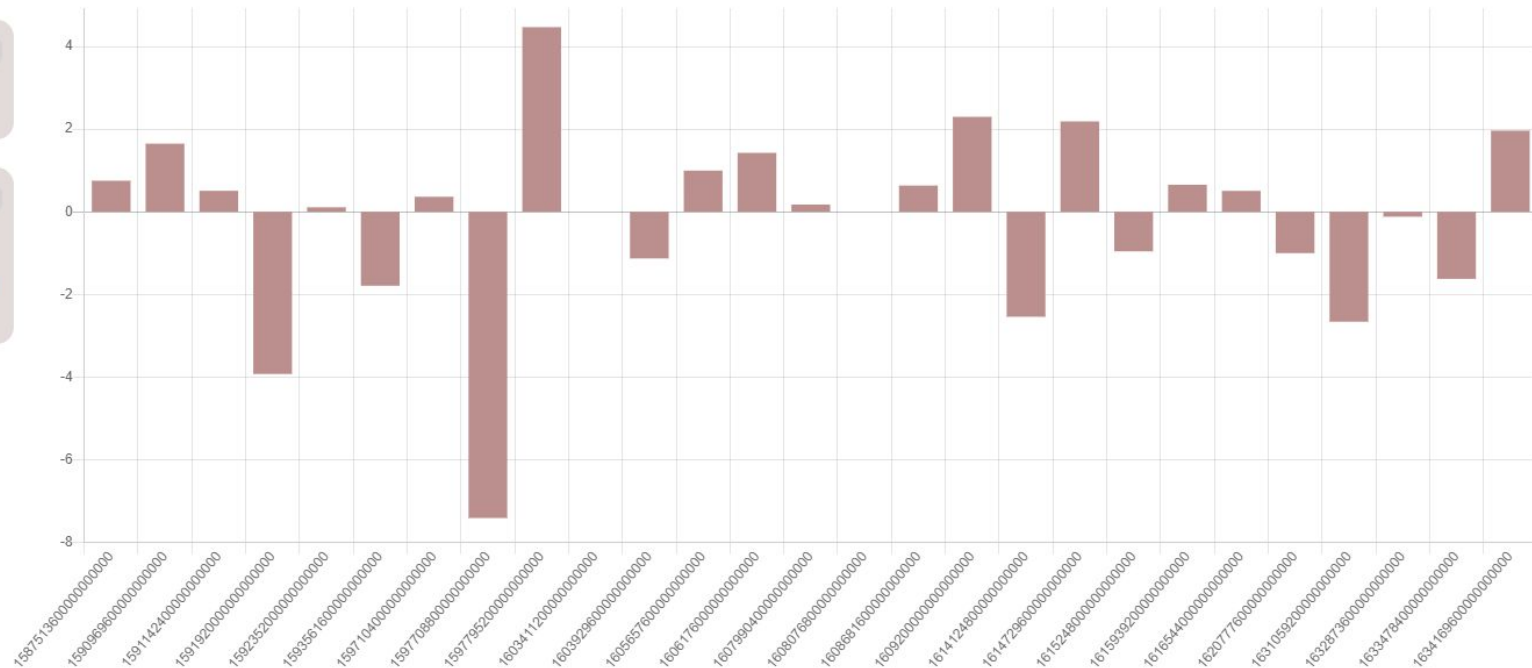
GO

\*특정 일자의 주가 및 코로나 데이터

종목 코드(ticker)

20200120~20220119 사이의 날짜

GO



해당 날짜의 코로나 신규 확진자는 전일 대비 25% 증감했습니다.

코로나 신규 확진자의 증감율이 20%와30% 사이 구간인 경우 해당 종목은 55.56%의 확률로 상승했습니다.

위의 그래프는, 코로나의 신규확진자 추세가 20%와30% 사이인 경우의 주가 증감율을 나타냅니다.

```

<form action="" method="post" name="fstock2" id="fstockfrom2" class='finputbox'>
    [% csrf_token %]
    <div id = 'fuserinput2' class= 'finputlabel'>
        <label>*특정 일자의 주가 및 코로나 데이터</label><br>
    </div>
    <label>종목 코드(ticker)</label><br>
    <input type="text" id="fstock2"><br>
    <label>20200120~20220119 사이의 날짜</label><br>
    <input type="text" id="fstock3">
    <input type="button" value="GO" id = 'fstock2btn'>
</form>
</div>
<div id = 'fchart'>
    <canvas id="myChart"></canvas>

```

```

function multiaxisChart(){
    // id myChart의 html 삭제
    $('#myChart').remove();
    // id fchart의 자식으로 <canvas id="myChart"></canvas> 생성
    $('#fchart').append('<canvas id="myChart"></canvas>');
    var ctx = document.getElementById('myChart').getContext('2d');
    var chart = new Chart(ctx, {
        type: 'bar',
        data: shapeChartData,
        options: chartOptions
    })
}

```

```

$("#fstock2btn").click(function(){
    // fstock2의 값 가져오기
    var code = $("#fstock2").val()
    // fstock3의 값 가져오기
    var date = $("#fstock3").val()
    $.ajax({
        url: "/fin/codeDate",
        type: "get",
        // 받아온 값들을 views의 request로 보내기
        data: {'code': code, 'date': date},
        dataType: "json",
        success: function(data){
            // alert(data.percent['rate'])
            shapeChartData = {
                labels: data.percent['Date'],
                datasets: [{
                    label: '2020-01-01종가 대비',
                    data: data.percent['Change'],
                    borderColor: 'blue',
                    backgroundColor: "skyblue",
                }]
            },
            chartOptions = {
                responsive: true,
                interaction: {
                    mode: 'index',
                    intersect: false,
                },
                stacked: false,
                plugins: {
                    title: {
                        display: true,
                    }
                },
                scales: {
                    yAxes: [{
                        id: 'A',

```

```

def codeDate(request):
    code = request.GET.get('code')
    date = request.GET.get('date')
    percent = pluspercent(code, date)
    context = {'msg': '메세지 성공', 'percent': percent}
    return JsonResponse(context)

```

```

def pluspercent(ticker, toDate):
    # 코로나가 있던 2년을 기준으로 주식 정보
    context = {'rate': rate, 'covidrate': covidrate}
    return context

```

```

multiaxisChart();
$("#flist").append('<li class="text"> " + '해당 날짜의 코로나 신규 확진자는 전일 대비 ' + data.percent['changePerfloor']*100 + '%와 ' + data.percent['rate'] + '%</li>')

```



**board- news (김재욱)**

Covidiamond

Seoul

Finance

Boards

## NEWS

언론사	제목	게시	썸네일
뉴시스	롯데홈쇼핑, 관광명소 랜선여행 '더 힐링 라이브' 출시	4분 전	
JTBC	방탄소년단, 현장감 가득... 영화관서 '라이브 뷰잉' 콘서트	15시간 전	
이데일리	휴넷, '2022 ATD TK 디프러핑' 온라인 라이브 개최	32분 전	



# 뉴스 스크래핑

```
def nlist(request):
    nowpage = request.GET.get('nowpage',1)
    url = 'https://search.naver.com/search.naver?where=news&sm=tab_jum8
    response = requests.get(url)
    soup = BeautifulSoup(response.text,"lxml")
    titles = soup.find_all("a",{"class":"news_tit"})
    subtitles = soup.find_all("a",{"class":"info_press"})
    times = soup.find_all("span",{"class":"info"})
    imgs = soup.find_all("img",{"class":"thumb_api_get"})

    publicData=[]
    for title in titles:
        name=title["title"]
        datalist={'title':name}
        publicData.append(datalist)

    for i,sub in enumerate(subtitles):
        subtitle=sub.get_text()
        data={'subtitle':subtitle}
        publicData[i].update(data)

    for i,time in enumerate(times):
        time1=time.get_text()
        data1={'time':time1}
        publicData[i].update(data1)

    context={'publicData':publicData}
    return render(request,'nlist.html',context)
```

```
<thead>
<tr>
<th>언론사</th>
<th>제목</th>
<th>게시</th>
<th>썸네일</th>
</tr>
</thead>
<!--게시글 등록부분-->
<tbody id="tbody">
{% if publicData %}
{% for pdata in publicData %}
<tr>
<td>{{pdata.subtitle}}</td>
<td><a href={{pdata.href}} onclick="window.open(this.href, '_blank',
<td>{{pdata.time}}</td>
<td></td>
</tr>
{% endfor %}
{% else %}
<tr>
<td colspan="4">데이터가 없습니다.</td>
</tr>
{% endif %}
</tbody>
</table>
</div>
```



**board-공지게시판(박혜정)**

# board-공지게시판(박혜정)

127.0.0.1:8000/board/blist/

Covidiamond

Seoul

Finance

Boards

번호	제목	등록일	조회수
17	내일은 휴일	2022-03-08 10:32	23
16	좋은 화요일	2022-03-08 10:32	235
15	글을 쓰자	2022-03-08 10:32	1
14	최강 2조 그런데 코로나를 결들인	2022-03-08 10:32	1
13	공지사항입니다	2022-03-08 10:32	1
12	페이지 나눌 수 있어요	2022-03-08 10:32	1
11	행운의 편지	2022-03-08 10:57	11
10	공지사항 10	2022-03-08 10:57	1
9	공지사항 9	2022-03-08 10:57	1
8	공지사항 8	2022-03-08 10:57	1
7	공지사항 7	2022-03-08 10:56	1
6	공지사항 6	2022-03-08 10:56	1
5	공지사항 5	2022-03-08 10:56	1
4	2022년을 시작하며..	2022-03-08 09:55	97
3	나라와 혜정은 사이가 좋다	2022-03-08 09:55	4

## 게시글 목록, 클릭 후 상세페이지

board > views.py > ...

```
1 from django.shortcuts import redirect, render
2 from django.http import JsonResponse
3 from django.core.paginator import Paginator
4 from board.models import Fboard
5 from django.db.models import Max, Min, Avg
6
7 def blist(request):
8     qs = Fboard.objects.all().order_by('-b_no')
9     paginator=Paginator(qs,15)
10    nowpage=int(request.GET.get('nowpage',1))
11    blist=paginator.get_page(nowpage)
12    context = {'blist':blist,'nowpage':nowpage}
13    return render(request,'blist.html', context)
14
15 def bview(request,b_no):
16     qs = Fboard.objects.get(b_no=b_no)
17     qs.b_hit += 1
18     qs.save()
19     context={'board':qs}
20     return render(request,'bview.html',context)
21
22 def bwrite(request):
23     return render(request, 'bwrite.html')
24
25 def bwriteOk(request):
26     title = request.POST.get('title')
27     content = request.POST.get('content')
28     no = Fboard.objects.aggregate(max_b_no=Max('b_no'))
29     max_no = no['max_b_no']
```

```
</thead>
<!--게시글 등록부분-->
<tbody class="tableBody">
    {% if blist %}
        {% for board in blist %}
            <tr>
                <td>{{board.b_no}}</td>
                <td><a href="{% url 'board:bview' board.b_no %}">{{board.b_title}}</a></td>
                <td>{{board.b_date|date:'Y-m-d H:i'}}</td>
                <td>{{board.b_hit}}</td>
            </tr>
        {% endfor %}
    {% else %}
        <tr>
            <td colspan='4'>등록된 게시물이 없습니다.</td>
        </tr>
    {% endif %}
</tbody>
```



## 분석 결과

분석 가설 1. 코로나 확진자율과 백신 접종율은 반비례 관계일 것이다.

-> 서울시 구 단위 지역별 실시간 확진자, 누적 확진자 수 시각화하여 제공했으나 코로나 확진자율에 영향 미치는 외부 요인 발견x.

분석 가설 2: 코로나 진행 추이에 따라 주식 섹터별 등락 영향이 다를 것이다.

-> 코로나 진행 추이에 따른 개별 주식의 변동 분석했으나 섹터별 비교 분석은 x.



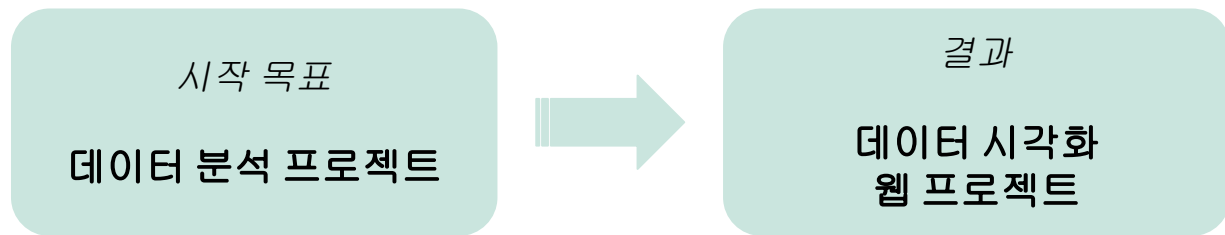
## 프로젝트 목표 달성 점검

api 활용 db 저장, 데이터 분석 및 시각화, 웹 페이지 구현

팀원끼리 의 상하지 않기

깃 사용법 숙지

## 프로젝트 목표 달성 점검



but 데이터 분석 제대로 배워보지 않은 상태에서 최선의 결과물.  
웹 다뤄보고 가시적 결과물 낸 것에 의의.



감사합니다.