

# 비주얼프로그래밍

## 과제 5.

20196014 고지흔

# 1번

## 문제의 해결 방안

from openpyxl import load\_workbook를 사용해 엑셀 파일을 열었습니다.

matplotlib을 사용해 파이 차트를 만들었습니다.

```
plt.draw()
```

```
plt.pause(10)
```

```
plt.clf()
```

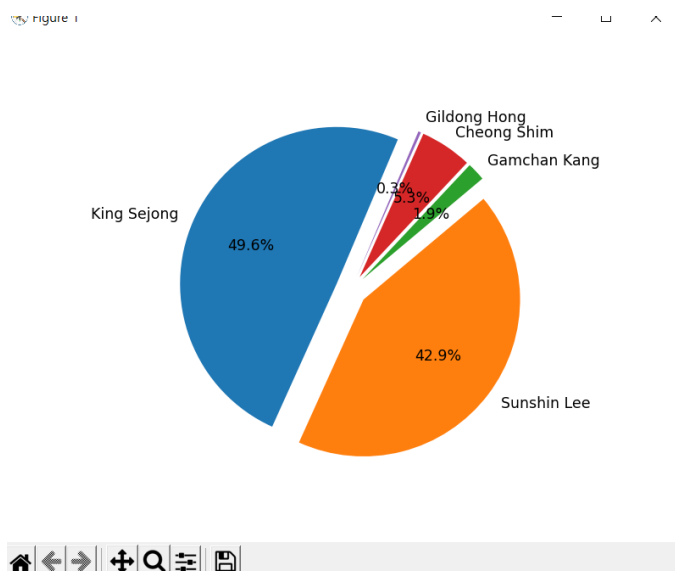
를 while 문으로 무한 반복해 10초마다 계속 업데이트 시켰습니다.

## 소스코드

```
from openpyxl import load_workbook
import matplotlib.pyplot as plt

while True:
    load_wb = load_workbook("elec.xlsx", data_only=True)
    load_ws = load_wb['Sheet1']
    labels=load_ws.cell(1, 1).value, load_ws.cell(1, 2).value, load_ws.cell(1, 3).value, load_ws.cell(1, 4).value, load_ws.cell(1, 5).value
    num=[load_ws.cell(2, 1).value,load_ws.cell(2, 2).value,load_ws.cell(2, 3).value,load_ws.cell(2, 4).value,load_ws.cell(2, 5).value]
    explode = (0.1, 0.1, 0.1, 0.1, 0.1)
    plt.pie(num, explode=explode, labels=labels, autopct='%1.1f%%', startangle=67)
    plt.draw()           # 그리기
    plt.pause(10)        # 잠시 기다리기
    plt.clf()            # figure 지우기
```

## 결과 화면 캡처



## 2번

### 문제의 해결 방안

from datetime import datetime을 사용해 시와 분을 가져왔습니다.

Matplotlib의 subplot기능을 이용해 파이차트를 두 개 그리고 하나는 시를 하나는 분을 나타냈습니다.

1번과 같은 방법으로 60초마다 업데이트 되게 했습니다.

시와 분의 각도를 계산해서 startangle에 반영했습니다

### 소스코드

```
import matplotlib.pyplot as plt
from datetime import datetime
import numpy as np

colors = ['#71797E', '#B2BEB5']
wedgeprops = {
    'edgecolor': 'black',
    'linestyle': '-',
    'linewidth': 1.0
}
fig = plt.figure(figsize=(10,10)) ## 캔버스 생성
fig.set_facecolor('white') ## 캔버스 배경색을 하얀색으로 설정
```

```
while True:
    ax = fig.add_subplot(121) ## 프레임 생성
    bx = fig.add_subplot(122) ## 프레임 생성
    ax.set_title("HOUR")
    bx.set_title("MINUTE")

    hour=datetime.now().hour
    Minute=datetime.now().minute
    Hlabels = [hour, None] ## 라벨
    Hfrequency = [1, 300] ## 빈도
    hourangle=hour/12*360##angle
    explode = [-0.5, 0] ## 튀어나오는 정도를 결정한다.
    ax.pie(Hfrequency, ## 파이차트 출력
           explode = explode,
           labels=Hlabels, ## 라벨 출력
           startangle=-hourangle+90,
           counter-clock=False ## 시계 방향으로 그린다.
           , colors=colors, wedgeprops=wedgeprops
           )
```

```

Mlabels = [Minute, None] ## 라벨
Mfrequency = [1, 1000] ## 빈도
Minuteangle = Minute / 60 * 360 ## angle

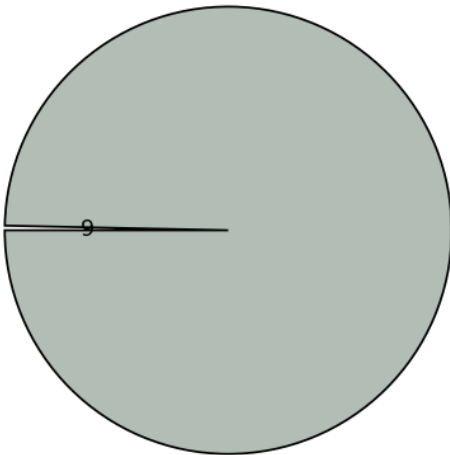
explode = [0.05, 0] ## 튀어나오는 정도를 결정한다.
bx.pie(Mfrequency, ## 파이차트 출력
       explode = explode,
       labels = Mlabels, ## 라벨 출력
       startangle = -Minuteangle + 90,
       counterclock = False ## 시계 방향으로 그린다.
       , colors = colors, wedgeprops = wedgeprops
       )

plt.draw()
plt.pause(60)
plt.clf()

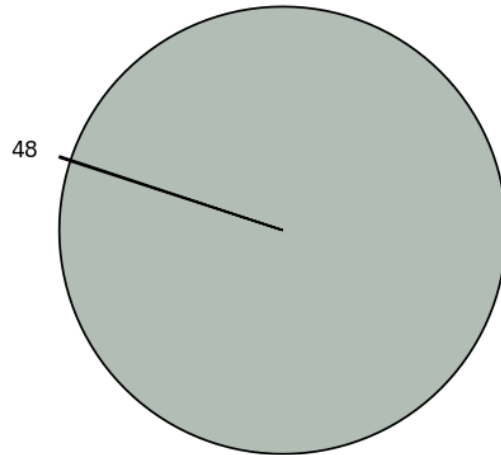
```

결과 화면 캡처

HOUR



MINUTE



## 3번

### 문제의 해결 방안

지난과제와 같이 csv파일에서 데이터를 가져왔고 국가가 변하는 순간이 가장 최신 데이터기 때문에 이를 이용해서 정리했습니다.

국가가 아닌 소득별 대륙별 데이터들은 따로 제외하는 if문을 넣었습니다

#12/1일 기준 구글 데이터와 일치하는 것을 확인했습니다

국가와 총 확진자 수만 있는 리스트를 만들었고 확진자 수 기준으로 sort했습니다. 그리고 이 데이터 중 상위 20개를 따로 리스트에 저장했습니다.

이 데이터를 matplotlib를 이용해 수평 바 차트를 만들었습니다.

### 소스코드

```
import matplotlib.pyplot as plt

covidf=open('owid-covid-data.csv','r')
firstline=covidf.readline()
i=0
linelist=[]
while True:
    line = covidf.readline()
    if not line: break
    linelist.append(list(line.split(',')))

totallist=[]#국가, 총확진자 저장
for i in range(len(linelist)-1):
    if linelist[i][2]!='Africa' or linelist[i][2]!='Asia' or linelist[i][2]!='Europe' or linelist[i][2]!='European Union' or \
        linelist[i][2]!='High income' or linelist[i][2]!='International' or linelist[i][2]!='Low income' or \
        linelist[i][2]!='Lower middle income' or linelist[i][2]!='North America' or linelist[i][2]!='Oceania' or \
        linelist[i][2]!='South America' or linelist[i][2]!='Upper middle income' or linelist[i][2]!='World':
        continue
    if linelist[i][2]!=linelist[i+1][2]:
        totallist.append([linelist[i][2],linelist[i][4]])
    totallist.append([linelist[len(linelist)-1][2],linelist[len(linelist)-1][4]])
```

```

for i in totallist:
    if i[1]=='':
        i[1]=0
    i[1]=int(i[1])
totallist.sort(key=lambda x :x[1], reverse=True)
totallist=totallist[0:20]#12/1일 기준 구글 순위와 일치하는 것 확인

nation = [totallist[i][0] for i in range(len(totallist))]
total_cases= [totallist[i][1] for i in range(len(totallist))]

plt.barh(nation, total_cases, height=0.4)
plt.xlabel('TOTAL CASES')
plt.ylabel('NATION')

plt.xticks(total_cases[::5],label=total_cases[::5])
plt.ticklabel_format(axis='x',useOffset=False, style='plain')

plt.show()

```

## 결과 화면 캡처

