
CHATBOT FOR STUDENT

:파이썬을 이용하여 간단한 챗봇 만들기

INDEX

1 기능

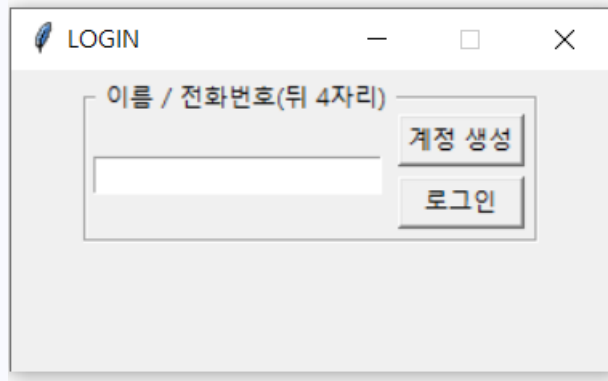
2 주의사항

3 주요 함수 소개

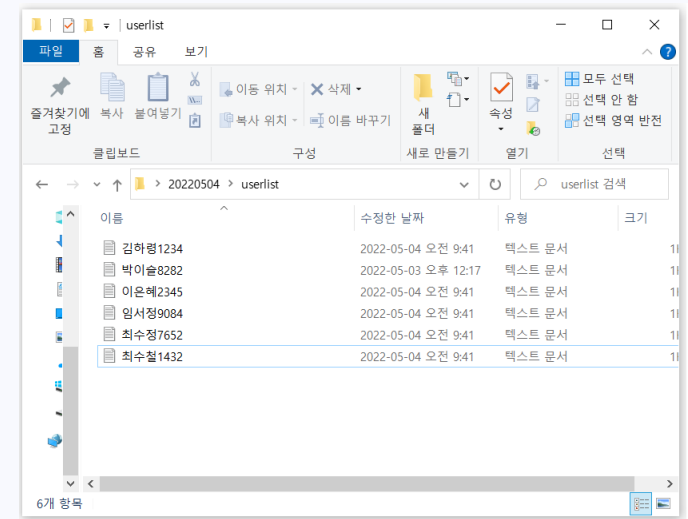
01

LOGIN PAGE

실행시 첫 화면



이름과 전화번호(뒤 4자리)를 넣어 계정을 만든다
Ex) 최수정7652 공백 없이 입력



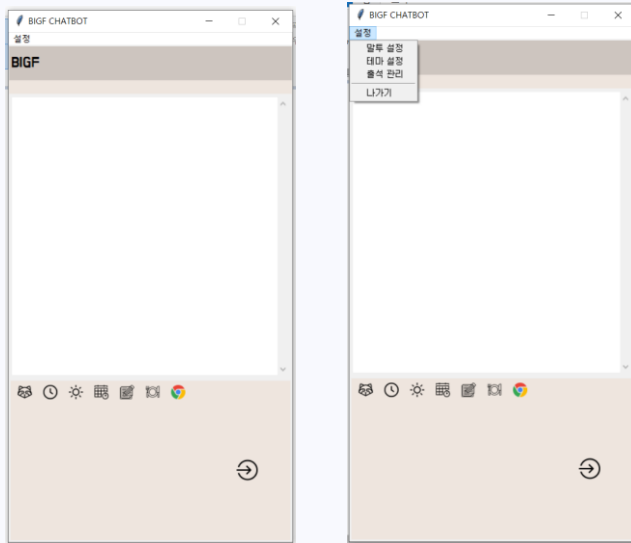
계정이름으로 생성된 사용자 파일

- 계정 생성 버튼을 누르면 계정 생성 후 챗봇 화면으로 넘어간다..
- 빈 텍스트로 계정 생성을 하거나 로그인을 누르면 경고 박스가 뜨고 파일이 생성되지 않는다.
- 이미 사용자 파일에 계정이 존재하면 계정 입력 후 로그인 버튼을 눌러 챗봇을 실행한다.

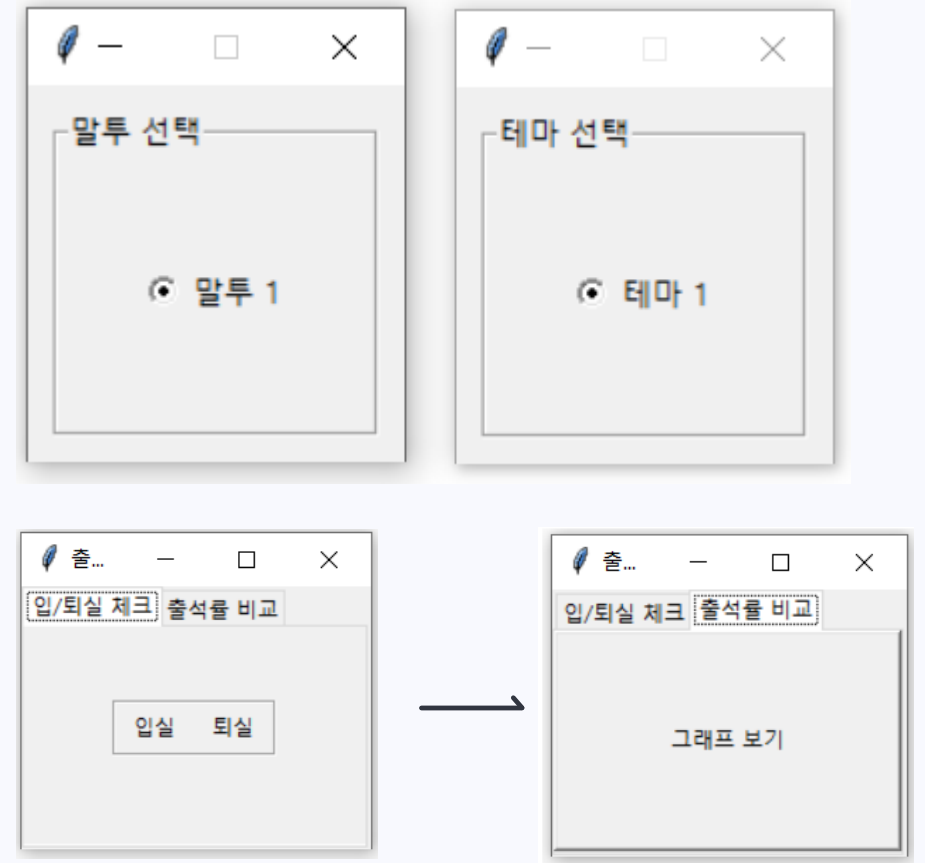
02

CHATBOT PAGE

-메뉴 탭



메뉴 탭의 설정 버튼

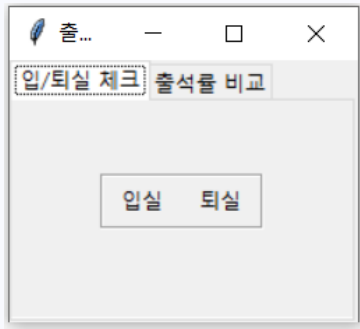


- 메뉴 탭의 설정을 누르면 말투 설정, 테마 설정, 출석 관리가 뜬다.
- 말투 선택과 테마 선택은 라디오 버튼 선택으로 옵션을 선택할 수 있다.
- 출석 관리는 입/퇴실 체크와 출석률 비교 그래프를 볼 수 있다.

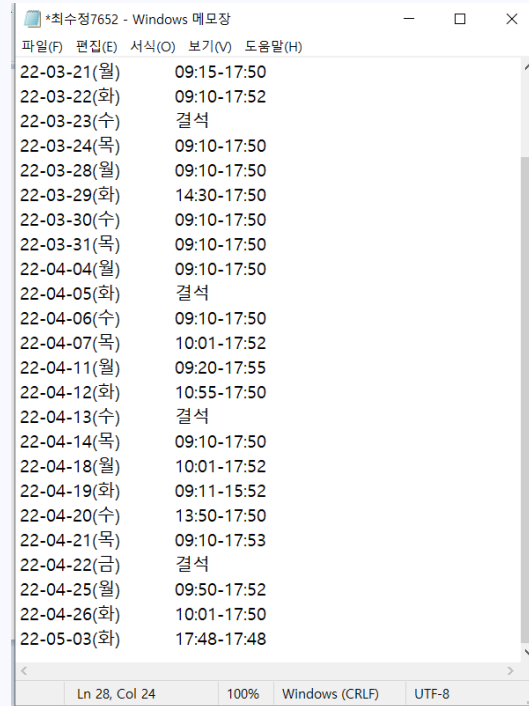
02

CHATBOT PAGE

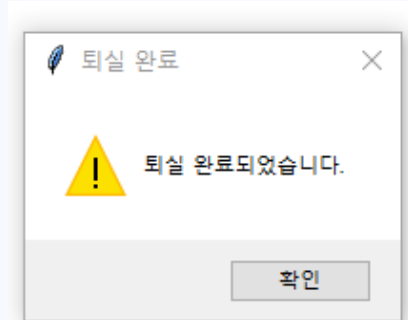
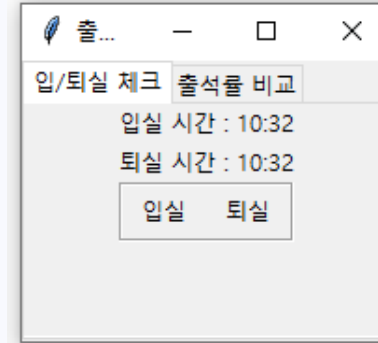
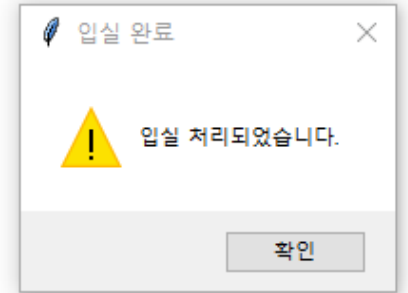
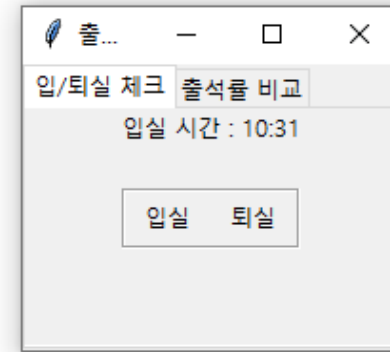
-메뉴 탭의 출석 관리 (입/퇴실 체크)



출석 관리의 입/퇴실 체크 화면



다음과 같은 텍스트 파일이 생성된다.



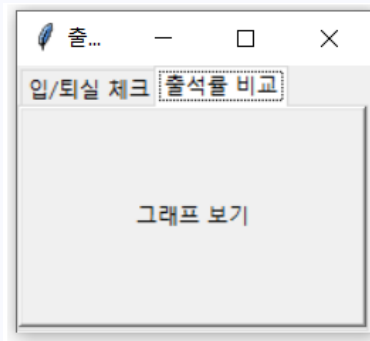
입/퇴실 버튼 클릭 시 보이는 화면

- 입실 버튼을 누르면 버튼을 누른 날짜와 시간이 텍스트 파일 마지막 라인에 들어간다
- 퇴실 버튼을 누르면 버튼을 누른 시간이 입실 시간 옆에 추가된다.
- 입/퇴실 버튼을 누르고 입/퇴실 처리가 완료되면 완료 안내 창이 뜨게 된다.

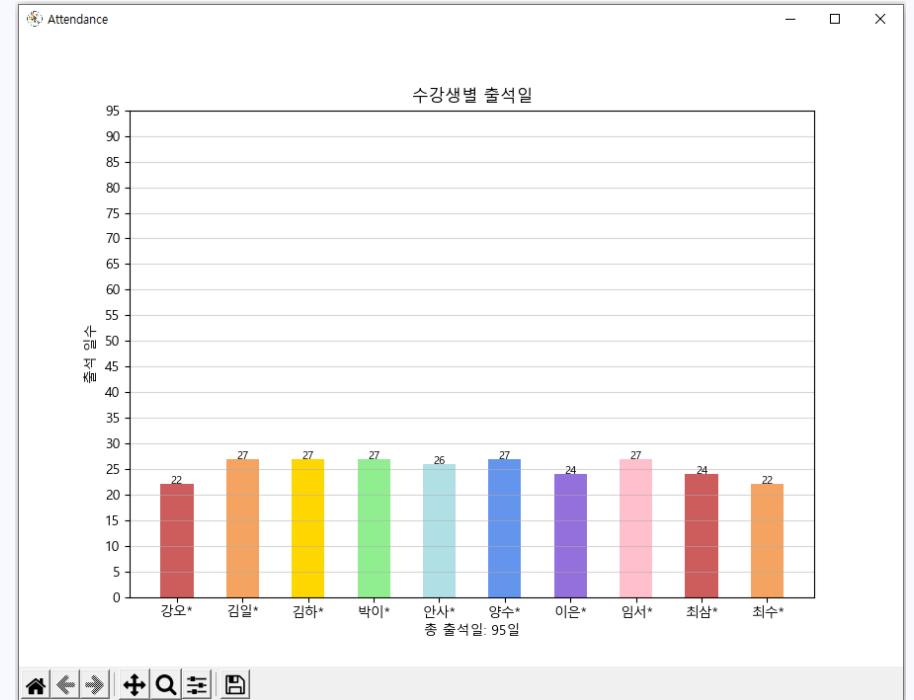
02

CHATBOT PAGE

- 메뉴 탭의 출석 관리 (출석률 비교)



출석률 비교 탭의 그래프 보기 버튼 클릭



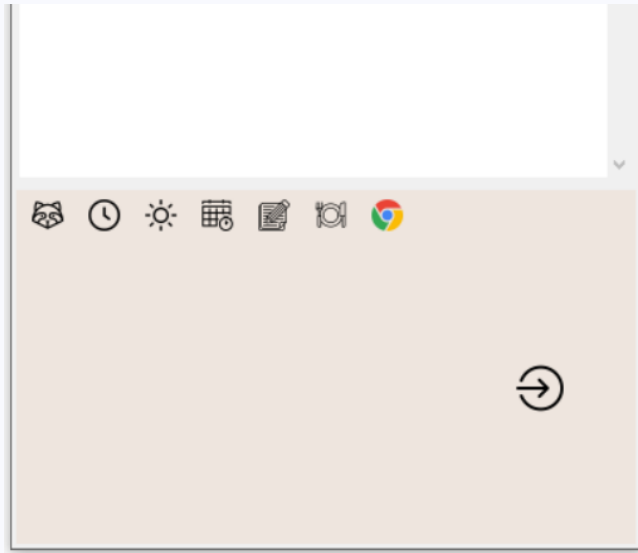
입/퇴실 시간 체크로 만들어진 userlist 폴더의 파일을 기반으로 만들어진 수강생 별 출석률 그래프

앞 페이지에서 언급된 입/퇴실 체크 기능을 이용하여 userlist에 계정 별 텍스트 파일을 생성하고 생성된 파일들을 비교하여 만들어진 그래프이다.

03

BUTTON

- 챗봇 메인 화면의 버튼들



(왼쪽부터 순서대로)

1. 메시지 출력 창에 인사 출력
(안녕하세요, 부르셨나요 등등).

2. 현재 시간 출력

3. 현재 날씨와 내일 날씨 출력

4. 시간표 창 켜기

5. 메모장 켜기

6. 식당 추천 출력
(둔산, 탄방, 갈마 입력)

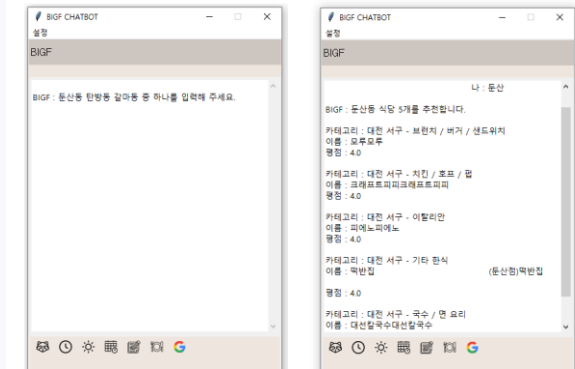
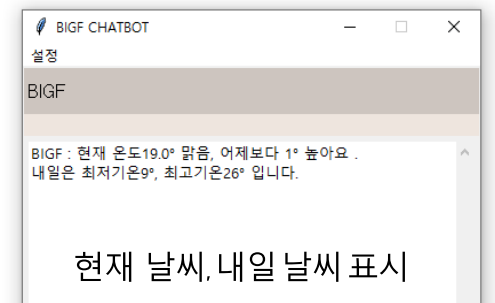
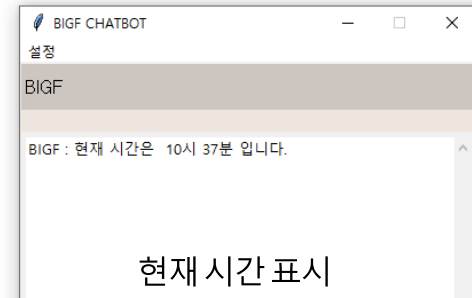
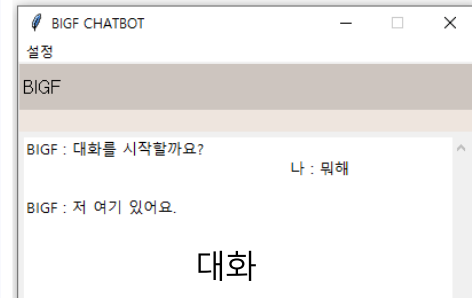
7. 엣지 구글 창 켜기

- 각 버튼 별로 연결된 기능이 실행되고, 어떤 기능이 실행되었는지 메시지 출력 창에 표시한다.

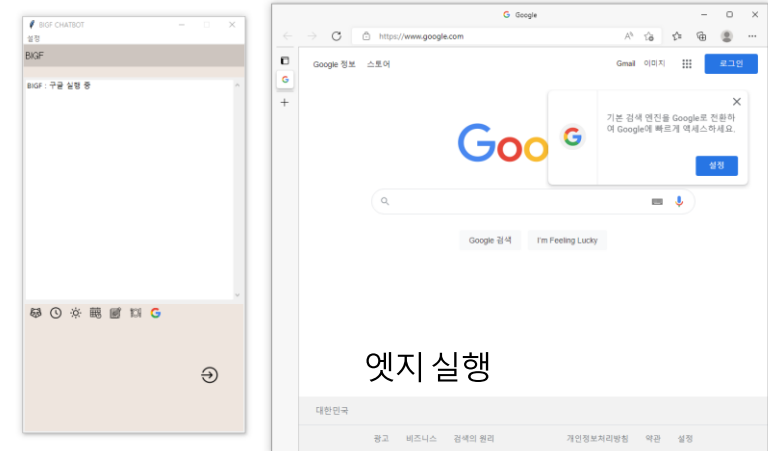
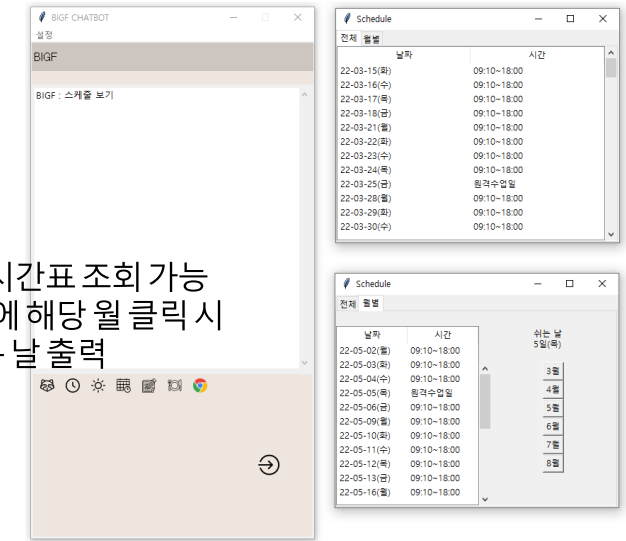
03

BUTTON

-버튼 클릭시 출력 화면



전체, 월 별 시간표 조회 가능
월 별 시간표에 해당 월 클릭시
쉬는 날 출력



각 버튼 별로 연결된 기능이 실행되고, 어떤 기능이 실행되었는지 메시지 출력 창에 표시한다

START CHAT EXAMPLE

```

thx=('고마워','고마워','고맙습니다','감사','감사합니다','고맙습니다','고맙습니다','고맙습니다','고맙습니다','고맙습니다')
thx_1=('네','네','네','네','네','네','네','네','네','네')

local1=('돈산들','돈산','돈들')
local1_1=('돈산들 식당을 보여드릴게요','돈산들 식당 5개를 추천합니다','추천 리스트예요')

local2=('한방들','한방')
local1_2=('한방들 식당을 보여드릴게요','한방들 식당 5개를 추천합니다','추천 리스트예요')

local3=('갈마들','갈마')
local1_3=('갈마들 식당을 보여드릴게요','갈마들 식당 5개를 추천합니다','추천 리스트예요')

start_1=('안녕','저기','하이','아','','','','','','','','','','','','','','','')
start_rn=('안녕하세요','부르셨나요?','왜 그러시나요?','저 여기 있어요','네?','그냥 가만히 있어요')
start_6=('네?','뭐요?','뭐라고요','다시 일하게 주세요','다시 말해봐')

start_1w=('날씨가','밖에','온도','비','몇도','날씨가','추위','더워?','날씨가?','날씨가')
start_rnw=('날씨가 어떤가 하면','날씨를 알려드릴게요')

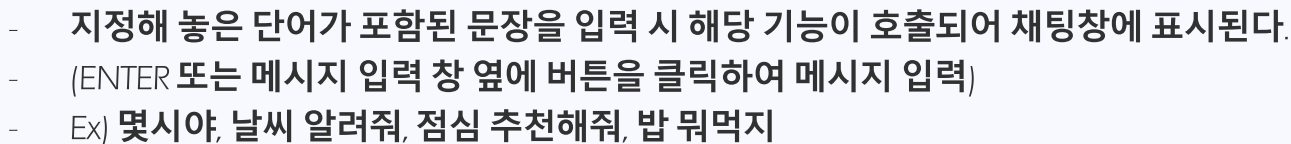
start_1t=('시간','몇시','시각','타입','몇시야','시간은?','시간은?','몇시?','몇시야?','몇시지','시간알려줘')
start_rnt=('시간이 궁금하시군요','시간을 알려드릴게요')

start_1m=('메오장','메오','투두','투두리스트','메오장려집','투두려집','메오장려여집','투두여여집','메오장','적어야겠다','적을까?','적을까')
start_rnm=('메오장을 알려드릴게요','메오장을 실행합니다')

start_1i=('밥','점심','저녁','카레','튀김','튀김지','점메추','배고프다','튀김을까?','튀김을까?','배고파','배고픈걸')
start_rli=('어디에서 드실 건가요?','식당을 추천해 드릴게요','밥 먹을 시간이지')

start_1s=('박보','정영이','개새끼','롯데','x','지랄','롯데','죽어','죽인다','죽일거야','죽고싶다','롯데','새끼','사발','죽일거다')
start_2s=('나쁜 말은 쓰지 마세요','그런 말은 하지 마세요','예쁜말을 쓰는 건 어떨까요?','무슨 일 있으신가요?')

```



CAUTION

주의사항

사용 시 주의사항

1. 아무 단어 입력 후 로그인 버튼을 눌러도 챗봇 화면으로 넘어간다.
(아무 단어 입력 후 넘어가면 출석 관리 메뉴를 사용하지 못함)
2. 계정 삭제는 텍스트 파일을 직접 지워야 한다.
3. 입실 시간만 존재하면 출석률 그래프 출력 불가
4. 지정 단어가 여러 번 들어간 문장을 입력시 빅프가 단어가 포함된 수 만큼 대응 텍스트를 표시한다.

-
- 계정 생성 버튼을 누르면 계정 생성 후 챗봇 화면으로 넘어간다..
 - 빈 텍스트로 계정 생성을 하거나 로그인을 누르면 경고 박스가 뜨고 파일이 생성되지 않는다.
 - 이미 사용자 파일에 계정이 존재하면 계정 입력 후 로그인 버튼을 눌러 챗봇을 실행한다.

06

SELECTED FUNCTION 1

챗봇을 만들며 사용한 함수2

CODE

이용 된 함수: BeautifulSoup4(bs4)

```
def weather():  
    from urllib.request import urlopen  
    import bs4  
  
    w_url='https://search.naver.com/search.naver?where=nexearch&sm=top_hty&fbm=0&ie=utf8&query=%EB%82%A0%EC%94%A8'  
    html=urlopen(w_url).read()  
  
    soup= bs4.BeautifulSoup(html, 'html.parser')  
  
    temperature=soup.find(class_='temperature_text')  
    w_condition=soup.find(class_='before_slash')  
    before_y=soup.find(class_="temperature")  
  
    return (temperature.text+w_condition.text,"어제보다 "+before_y.text)  
  
def t_weather():  
    from urllib.request import urlopen  
    import bs4  
  
    w_url = 'https://search.naver.com/search.naver?where=nexearch&sm=top_hty&fbm=0&ie=utf8&query=%EB%82%A0%EC%94%A8'  
    html = urlopen(w_url).read()  
  
    soup = bs4.BeautifulSoup(html, 'html.parser')  
  
    t_ltemp=soup.find(class_="lowest")  
    t_htemp=soup.find(class_="highest")  
    return (t_ltemp.text,t_htemp.text)
```

사용자 위치 기반 날씨 표시

RESULT

결과물

BIGF : 날씨를 알려드릴게요.
BIGF : 현재 온도22.0° 맑음, 어제보다 2° 높아요 .
내일은 최저기온9°, 최고기온26° 입니다.



06

SELECTED FUNCTION 2

챗봇을 만들며 사용한 함수2

CODE

이용된 함수: matplotlib.pyplot

```
#모든 사용자의 출석부 리스트 만드는 함수
def total_rollbook():
    Alist=[]
    for i in range(len(namedata)):
        Alist.append(rollbook(namedata[i],numdata[i]))
    return Alist

#userlist'디렉토리의 모든 출석부 리스트
rlist=total_rollbook()

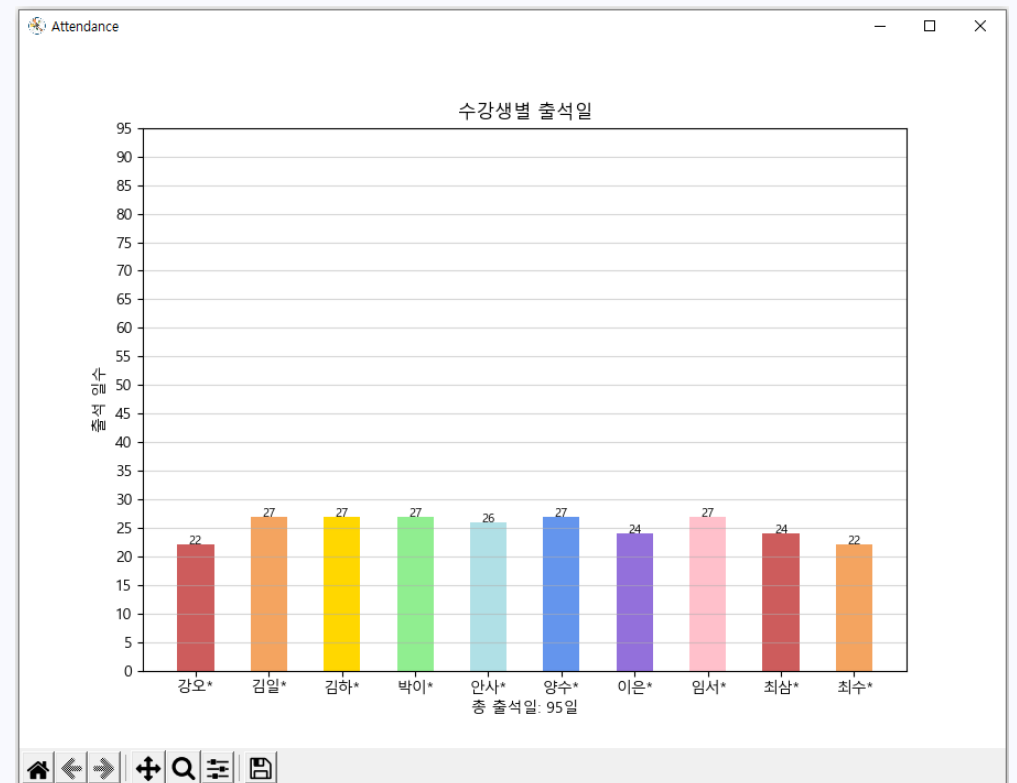
#사용자별 출석일 리스트
def total_Adays():
    y_data_Adays=[]
    for i in rlist:
        r=Adays(i)
        y_data_Adays.append(r)
    return y_data_Adays

#사용자별 출석일 비교 그래프
def compare_Adays():
    x_data = anonymity() # x축, 사용자 이름
    y_data = total_Adays() # y축, 사용자별 출석일
    x=np.arange(len(x_data)) # 주어진 범위와 간격에 따라 균일한 값을 갖는 어레이를 반환
    plt.figure('Attendance')
    plt.title('수강생별 출석일')
    plt.xticks(x, label=x_data) # .xticks(x축 눈금 레이블, label=x축 데이터)
    plt.xlabel('총 출석일: 95일')
    plt.ylabel('출석 일수')
    plt.grid(True, axis='y', alpha=0.5)
    for i in range(1, len(x_data) + 1):
        plt.text(i - 1, y_data[i - 1], y_data[i - 1], ha='center', size=8)
    plt.bar(x_data, y_data, color=barcolors, width=0.5)
    plt.show()
```

수강생 별 출석일 비교 그래프 만들기

RESULT

결과물



THANK YOU