

LESSON

파이썬의 데이터시각화

```
animLength = toTime - fromTime

# Ask user for directory
filePath = c4d.storage.SaveDialog()
filePath, objName = os.path.split(filePath)
objName = objName + ". "
filePath = filePath + "\\ "

# Ask for confirmation
questionDialogText = "Obj Sequence will be saved as:\n\n" \
    "" + filePath + objName + "####.obj\n\n" \
    "From frame " + str(fromTime) + " to " + str(toTime) + " "
proceedBool = c4d.gui.QuestionDialog(questionDialogText)

if proceedBool == True:

    # Loop through animation and export frames
    for x in range(0,animLength):

        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

        # update status bar
        c4d.StatusSetText("Exporting " + str(x) + " of " + str(animLength))
        c4d.StatusSetBar(100.0*x/animLength)

        # add buffer 0.001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps))
        if len(bufferedNumber) < 3:
```

데이터시각화란?

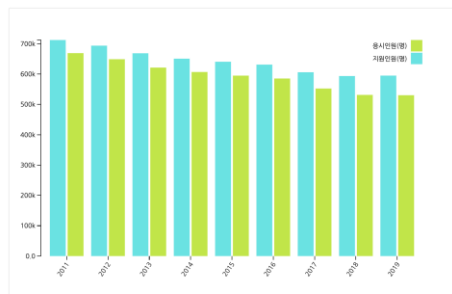
- ❖ 데이터 분석 결과를 쉽게 이해할 수 있도록 시각적으로 표현하고 전달하는 과정
- ❖ 디지털 시대의 학습자는 SNS, 영상, 클라우드 서비스 등 무한정의 데이터에 노출되어 있음
- ❖ 데이터의 표현, 수용 방식의 변화가 필요하며, 특히 **정보를 직관적이고 효율적으로** 제공하는 데이터 시각화의 필요성이 증가하는 추세

데이터시각화의 필요성

- 많은 양의 데이터를 한눈에 볼 수 있음
- 데이터 분석에 전문 지식이 없어도 누구나 쉽게 해당 데이터를 인지하고 활용할 수 있음
- 단순한 데이터의 요약, 통계보다 정확한 데이터 분석 결과를 도출해낼 수 있음
- 단순히 나열된 데이터에서는 확인하기 어려운 데이터의 의미와 새로운 정보를 파악할 수 있음

2011~2019 수능 지원 및 응시 현황 - 복사본

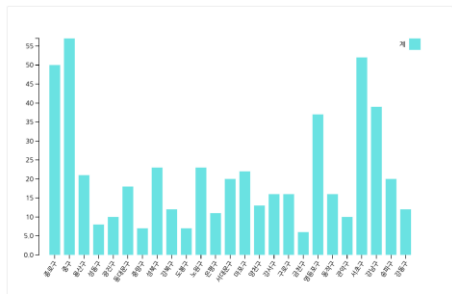
제작자 A 2019-11-14 02:01:49 | 한국교육과정평가원



| | |
|-----------|--|
| 시각화 유형 | 막대 차트 |
| 시각화 유형 설명 | 처트가 나타내는 값과 비교하는 값의 직사각형 막대가 표시됩니다. 둘 이상의 값을 비교하는 데 사용됩니다. |
| 사용 데이터 | 한국교육과정평가원 대학수학능력시험 정보 |

서울특별시 도서관 분포현황

제작자 이현정 2019-12-05 07:08:37



| | |
|-----------|--|
| 시각화 유형 | 막대 차트 |
| 시각화 유형 설명 | 처트가 나타내는 값과 비교하는 값의 직사각형 막대가 표시됩니다. 둘 이상의 값을 비교하는 데 사용됩니다. |
| 사용 데이터 | 서울특별시 도서관분포 |

파이썬에서의 데이터시각화 방법

- 파이썬에서는 데이터를 시각화하기 위한 다양한 라이브러리를 제공
 - 각 라이브러리마다 특징이 있기 때문에, 각자 사용하기 편한 것을 사용하면 됨
- 주로 그래프를 쉽게 그릴 수 있도록 도와주며, 3D, 원형, 막대 등 다양한 옵션 제공



파이썬의 주요 시각화 모듈

matplotlib, seaborn, plotnine,
folium, pyecharts, wordcloud
등



LESSON

Matplotlib 모듈 활용

```
animLength = toTime - fromTime

# Ask user for directory
filePath = c4d.storage.SaveDialog()
filePath, objName = os.path.split(filePath)
objName = objName + ". "
filePath = filePath + "\\ "

# Ask for confirmation
questionDialogText = "Obj Sequence will be saved as:\n\n" \
    "" + filePath + objName + "####.obj\n\n" \
    "From frame " + str(fromTime) + " to " + str(toTime) + " "
proceedBool = c4d.gui.QuestionDialog(questionDialogText)

if proceedBool == True:

    # Loop through animation and export frames
    for x in range(0,animLength):

        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

        # update status bar
        c4d.StatusSetText("Exporting " + str(x) + " of " + str(animLength))
        c4d.StatusSetBar(100.0*x/animLength)

        # add buffer 0.001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps))
        if len(bufferedNumber) < 3:
```

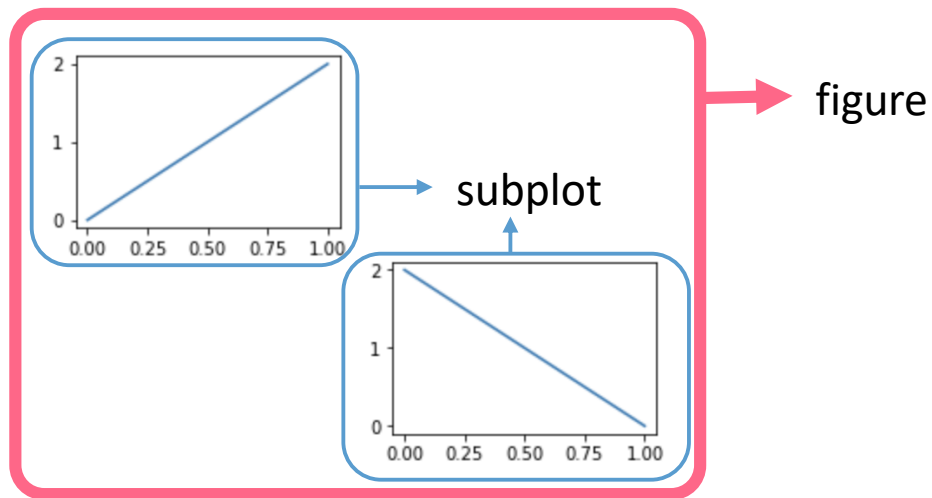
Matplotlib 모듈이란?

- ❖ 파이썬에서 데이터 시각화를 위해 가장 많이 사용하는 대표 라이브러리
- ❖ MATLAB 프로그램과 유사한 인터페이스
- ❖ 다양한 출력 형식 지원(PNG, JPG, SVG 등)
- ❖ 그래프의 종류, 축, 눈금선, 범례 등 상세한 설정 가능
- ❖ `pip install matplotlib` 명령어로 설치



figure와 subplot의 개념

- figure : subplot(그래프의 영역)을 생성하는 전체 틀
- subplot : 그래프를 그리기 위한 영역



그래프를 생성하기 위한 함수

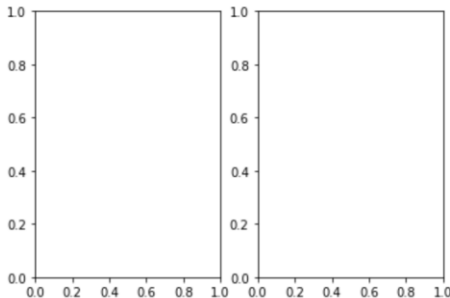
❖ add_subplot('특정 위치') : 특정 위치에 그래프 생성하는 함수

add_subplot([총 행의 수], [총 열의 수], [subplot 번호])

```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
figure = plt.figure()
axes1 = figure.add_subplot(121)
axes2 = figure.add_subplot(122)
```

```
plt.show()
```



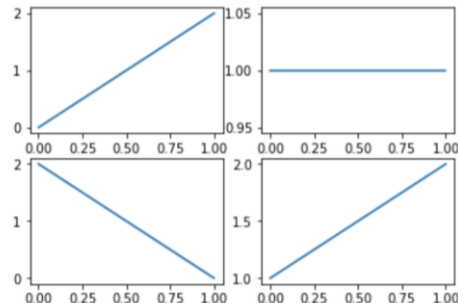
(행, 열, 번호) 순서

```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
figure = plt.figure()
axes1 = figure.add_subplot(2,2,1)
axes2 = figure.add_subplot(2,2,2)
axes3 = figure.add_subplot(2,2,3)
axes4 = figure.add_subplot(2,2,4)
```

```
axes1.plot([0,2])
axes2.plot([1,1])
axes3.plot([2,0])
axes4.plot([1,2])
```

```
plt.show()
```



Y축에 표시할 값

그래프를 생성하기 위한 함수

♣ subplots() : figure와 subplot 동시 생성

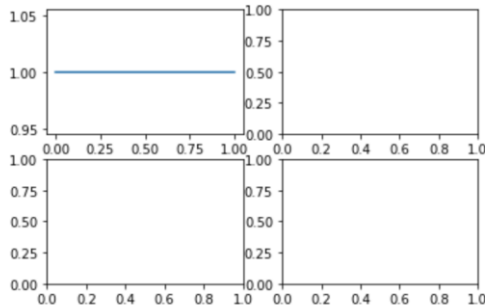
subplots(생성할 subplot의 행, 열)

```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
figure, axes = plt.subplots(2,2)
```

```
axes[0][0].plot([1,1])
```

```
plt.show()
```



```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
figure, axes = plt.subplots(2,2)
```

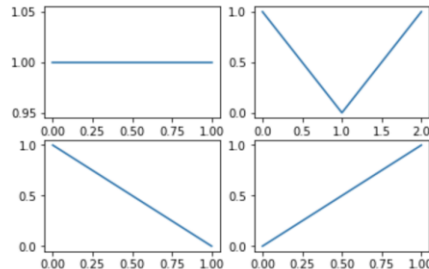
```
axes[0][0].plot([1,1])
```

```
axes[0][1].plot([1,0,1])
```

```
axes[1][0].plot([1,0])
```

```
axes[1][1].plot([0,1])
```

```
plt.show()
```



그래프를 생성하기 위한 함수

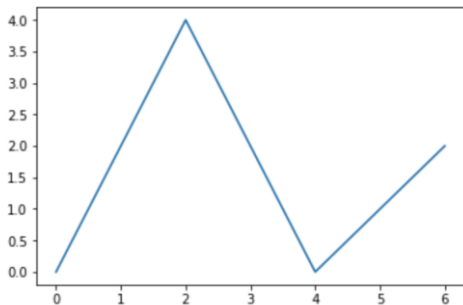
♣ plot () : 꺾은선 그래프를 생성하는 함수

- 파이썬의 리스트, numpy의 ndarray, pandas의 Series 모두 사용 가능함

```
import matplotlib.pyplot as plt
%matplotlib inline

figure = plt.figure()
axes = figure.add_subplot(111)

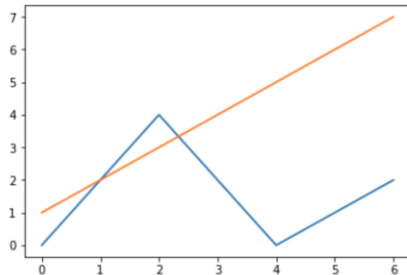
x = [0,2,4,6]
y = [0,4,0,2]
axes.plot(x,y)
plt.show()
```



```
import matplotlib.pyplot as plt
%matplotlib inline

figure = plt.figure()
axes = figure.add_subplot(111)

x = [0,2,4,6]
y = [0,4,0,2]
x2 = [0,1,2,3,4,5,6]
y2 = [1,2,3,4,5,6,7]
axes.plot(x,y)
axes.plot(x2,y2)
plt.show()
```



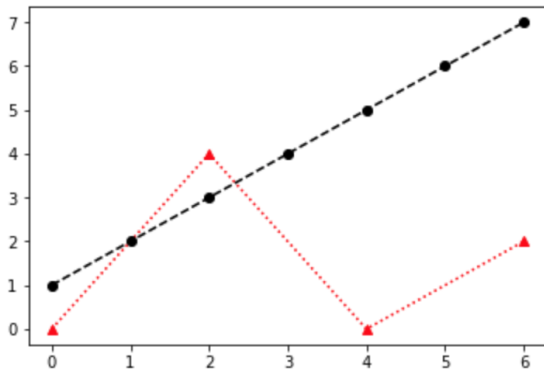
하나의 그래프에
plot을 여러 개
중첩하여 생성 가능

[참고] 꺾은선 그래프를 위한 다양한 스타일 옵션

```
import matplotlib.pyplot as plt
%matplotlib inline

figure = plt.figure()
axes = figure.add_subplot(111)

x = [0,2,4,6]
y = [0,4,0,2]
x2 = [0,1,2,3,4,5,6]
y2 = [1,2,3,4,5,6,7]
axes.plot(x,y, color='r', linestyle='dotted',marker='^')
axes.plot(x2,y2,linestyle='dashed',marker='o',color='k')
plt.show()
```



선 스타일(실선, 점선 등)과 선 색상,
마커 모양 등을 다양하게
설정할 수 있음

- **linestyle** : 선의 형태 지정 -- -
solid, dotted, dashdotted,
dashed 등
- **color** : 선 색상 지정
-black, red, blue , green 등
- **marker** : 마커 모양 지정
- 원 모양(o), 삼각형(^),
화살표(>) 등

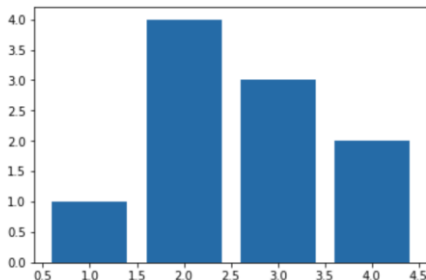
그래프를 생성하기 위한 함수

♣ `bar()` : 수량을 막대 길이로 표현하는 막대 그래프를 그리는 함수

```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
figure = plt.figure()
axes = figure.add_subplot(111)
x = [1,2,3,4]
y = [1,4,3,2]

axes.bar(x,y)
plt.show()
```

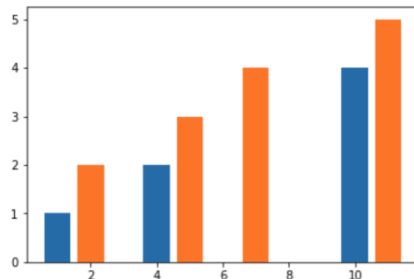


```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
figure = plt.figure()
axes = figure.add_subplot(111)
x = [1,4,7,10]
y = [1,2,3,4]
x2 = [2,5,7,11]
y2 = [2,3,4,5]

axes.bar(x,y)
axes.bar(x2,y2)

plt.show()
```



값이 중복되면
합쳐짐

그래프를 생성하기 위한 함수

♣ `twinx()`, `twiny()` : 서로 다른 단위의 2개의 x, y축을 표시할 경우 사용

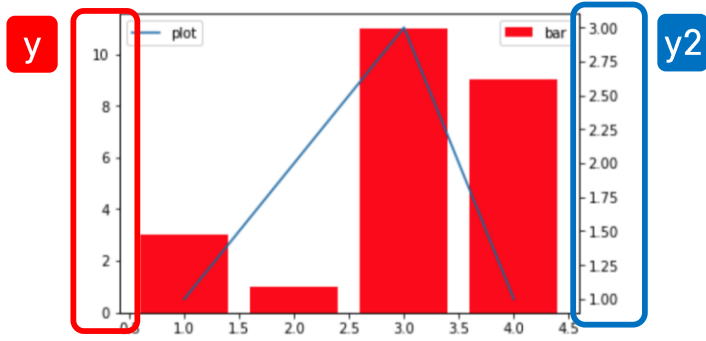
```
import matplotlib.pyplot as plt
%matplotlib inline

figure = plt.figure()
axes = figure.add_subplot(111)
axes2 = axes.twinx()

x = [1,2,3,4]
y = [3,1,11,9]
x2 = [1,2,3,4]
y2 = [1,2,3,1]

axes.bar(x,y, color = 'r',label = 'bar')
axes2.plot(x2,y2, label='plot')

axes.legend(loc=1)
axes2.legend(loc=2)
plt.show()
```



중첩된 그래프의 x, y값의 단위가
다른 경우 함께 표시할 수 있음

※ 그래프 생성 시 `label` 지정 및 `legend()`로 범례 생성 가능

그래프를 생성하기 위한 함수

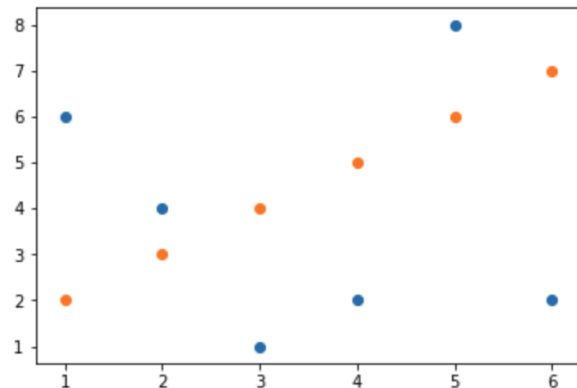
♣ scatter() : 산점도

- 그래프 상의 점의 위치로 데이터의 분포와 관계를 파악 할 수 있음

```
import matplotlib.pyplot as plt
%matplotlib inline

figure = plt.figure()
axes = figure.add_subplot(111)

x = [1,2,3,4,5,6]
y = [6,4,1,2,8,2]
x2 = [1,2,3,4,5,6]
y2 = [2,3,4,5,6,7]
axes.scatter(x,y)
axes.scatter(x2,y2)
plt.show()
```



그래프를 생성하기 위한 함수

♣ pie () : 원 그래프

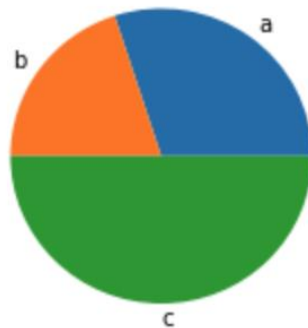
- 데이터 간 비율을 한눈에 파악 할 수 있음
- label과 데이터로 표현, figsize로 그래프의 크기 설정 가능

```
import matplotlib.pyplot as plt
%matplotlib inline

figure = plt.figure(figsize=(3,3))
axes = figure.add_subplot(111)

label = [ 'a', 'b', 'c' ]
data = [30,20,50]

axes.pie(data,labels=label)
plt.show()
```



LESSON

WordCloud 모듈 활용

```
animLength = toTime - fromTime

# prompt user for directory
filePath = c4d.storage.SaveDialog()
filePath, objName = os.path.split(filePath)
objName = objName + ".obj"
filePath = filePath + "\\ "

# check for confirmation
questionDialogText = "Obj Sequence will be saved as:\n\n" \
    "" + filePath + objName + "####.obj\n\n" \
    "From frame " + str(fromTime) + " to " + str(toTime) + " "
proceedBool = c4d.gui.QuestionDialog(questionDialogText)

if proceedBool == True:

    # loop through animation and export frames
    for x in range(0,animLength):

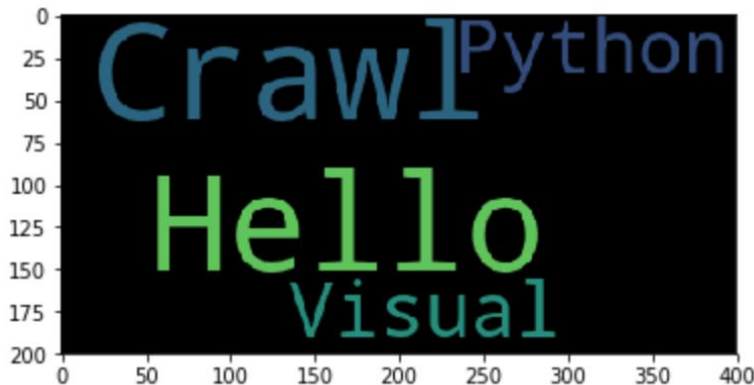
        # change frame, redraw view
        moveTime = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x)
        doc.SetTime(moveTime)
        c4d.EventAdd(c4d.EVENT_FORCEREDRAW)
        c4d.DrawViews(c4d.DRAWFLAGS_FORCEFULLREDRAW)

        # update status bar
        c4d.StatusSetText("Exporting " + str(x) + " of " + str(animLength))
        c4d.StatusSetBar(100.0*x/animLength)

        # get buffer 0001
        bufferedNumber = str(doc.GetTime().GetFrame(docFps))
        if len(bufferedNumber) < 4:
```


WordCloud 모듈이란?

- ❖ 파이썬에서 단어의 중요도에 따라 단어의 크기, 색 등을 변화시켜 어떤 단어가 중요한지 보여 줄 수 있는 데이터시각화 모듈
- ❖ 중요한 단어나 키워드를 직관적으로 보여 줄 수 있음
- ❖ `pip install wordcloud` 명령어로 설치



WordCloud의 생성과 출력

- WordCloud 모듈로 워드 클라우드를 생성
→ matplotlib 모듈로 생성한 워드 클라우드를 출력

```
import wordcloud
import matplotlib.pyplot as plt
%matplotlib inline

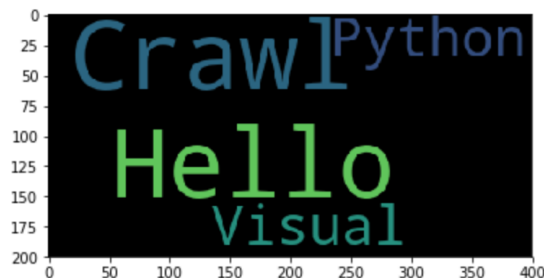
keywords = {'Hello':5, 'Python':3, 'Crawl':5, 'Visual':3}

wc = wordcloud.WordCloud()
cloud = wc.generate_from_frequencies(keywords)

plt.figure()
plt.imshow(cloud)
plt.show()
```

Wordcloud 생성

Wordcloud 출력



WordCloud의 생성과 출력

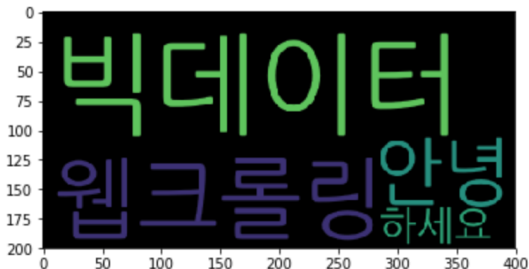
☛ 한글 출력 시 한글 깨짐을 방지하기 위해 font를 변경해야 함

```
import wordcloud
import matplotlib.pyplot as plt
%matplotlib inline

keywords = {'안녕':2, '하세요':1, '빅데이터':5, '웹크롤링':3}

wc = wordcloud.WordCloud(font_path='/Library/Fonts/NanumGothic.otf')
cloud = wc.generate_from_frequencies(keywords)

plt.figure()
plt.imshow(cloud)
plt.show()
```



font_path 옵션에
한글이 지원되는 폰트가
위치한 컴퓨터 경로를 입력

WordCloud 스타일 지정

♣ 배경 색상 및 크기 변경

```
import wordcloud
import matplotlib.pyplot as plt
%matplotlib inline

keywords = {'안녕':2, '하세요':1, '빅데이터':5, '웹크롤링':3}

wc = wordcloud.WordCloud(font_path='/Library/Fonts/NanumGothic.otf',
                          background_color='white', width=800, height=600)
cloud = wc.generate_from_frequencies(keywords)

figure = plt.figure()
plt.imshow(cloud)
plt.show()
figure.savefig('wordCloud.png')
```



background_color: 배경 색상 변경
colormap: 글씨 색상 변경(예: Blues, Reds ...)
width, height: 가로, 세로

WordCloud 이미지 저장

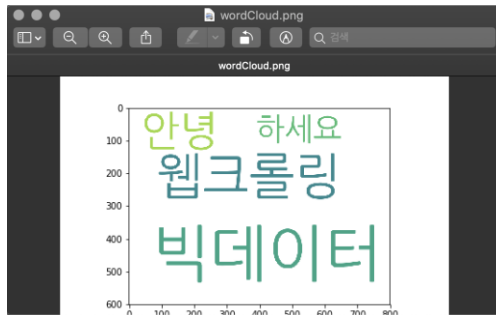
♣ matplotlib의 `savefig()`를 활용해 이미지를 파일로 저장할 수 있음

```
import wordcloud
import matplotlib.pyplot as plt
%matplotlib inline

keywords = {'안녕':2, '하세요':1, '빅데이터':5, '웹크롤링':3}

wc = wordcloud.WordCloud(font_path='/Library/Fonts/NanumGothic.otf',
                          background_color='white', width=800, height=600)
cloud = wc.generate_from_frequencies(keywords)

figure = plt.figure()
plt.imshow(cloud)
plt.show()
figure.savefig('wordCloud.png')
```



마스크 이미지를 이용해 WordCloud 만들기 예제

```
from PIL import Image
import numpy as np
import wordcloud
import matplotlib.pyplot as plt
import random
%matplotlib inline

heart_img = np.array(Image.open("heart.jpg"))

s = '파이썬'
for i in range(100):
    new_s = s + str(i)
    keywords[new_s] = random.randint(1,15)

wc = wordcloud.WordCloud(font_path='/Library/Fonts/NanumGothic.otf',mask=heart_img)

cloud = wc.generate_from_frequencies(keywords)

figure = plt.figure()
plt.imshow(cloud)
plt.axis("off")
plt.show()
figure.savefig('wordCloud.png')
```



PIL 모듈을 이용해 'heart.jpg' 이미지를 불러온 후
numpy 모듈을 이용해 이미지를 배열 형태로 변경하여 사용