

# 으뜸 파이썬



## 6강 외부 모듈의 설치와 활용

- 본 강의노트는 으뜸 파이썬(박동규, 강영민 著) 1판의 강의자료를 활용하여 교양수업에 맞게 편집되었습니다.

- 외부 모듈의 설치

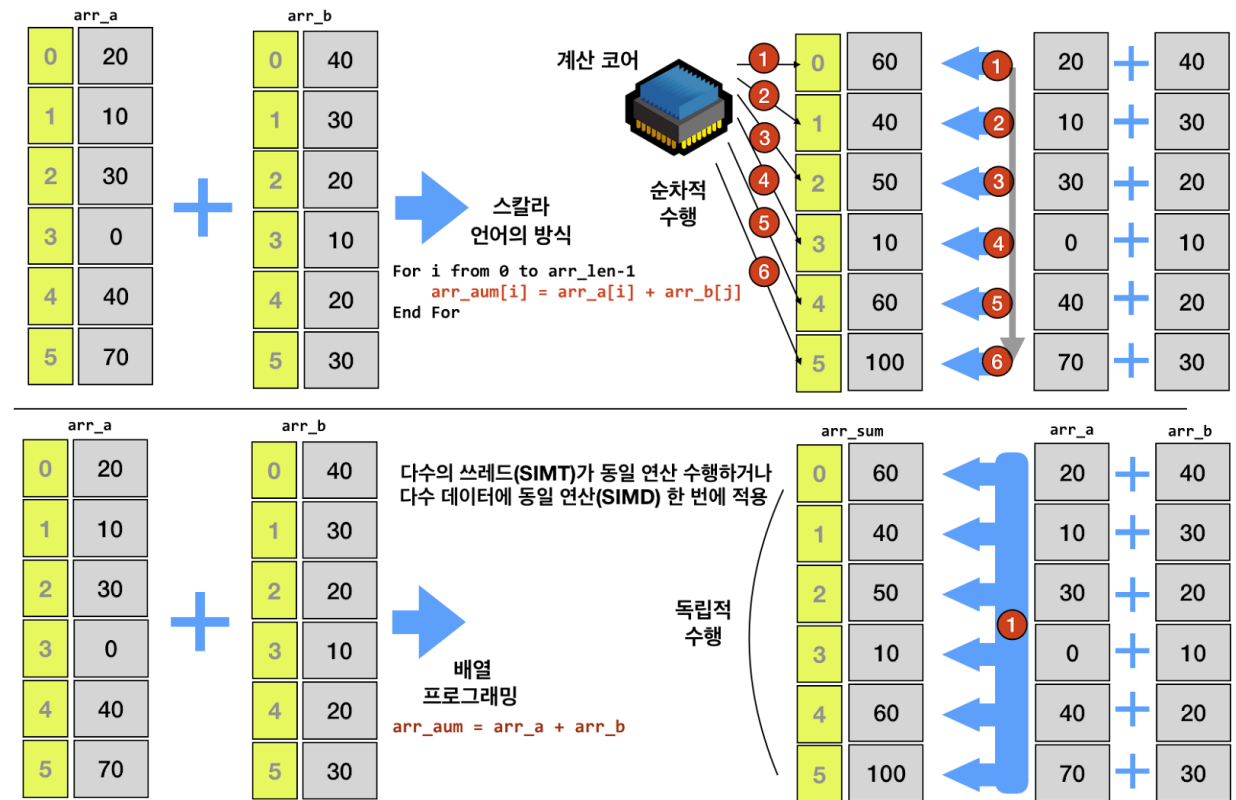
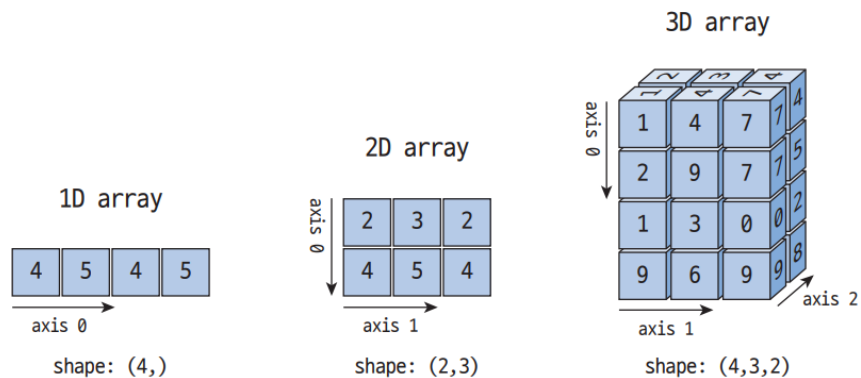
C:\> pip install **\*\*외부모듈명\*\***

PIP는 “Package Installer for Python”

파이썬 패키지를 설치하고 관리하기 위한 표준 패키지 관리자(package manager)  
표준 라이브러리에 포함되지 않은 외부 모듈(예: numpy, pandas, flask 등)의 다운로드와 설치를 지원

- 수치를 다루는 데에는 numpy

C:\> pip install **numpy**



## • 강력한 기능 내장

$$\begin{cases} x + 2y + z = 9 \\ 2x + 3y + 4z = 21 \\ 3x + y + 2z = 10 \end{cases}$$

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 3 & 4 \\ 3 & 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 9 \\ 21 \\ 10 \end{pmatrix}$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

```
import numpy as np
```

```
# 계수 행렬 A (3x3: Ax = b에서 A 부분)
```

```
A = np.array([[1, 2, 1], # 첫 번째 방정식: 1x + 2y + 1z  
              [2, 3, 4], # 두 번째 방정식: 2x + 3y + 4z  
              [3, 1, 2]]) # 세 번째 방정식: 3x + 1y + 2z
```

```
# 상수 벡터 b (Ax = b에서 b 부분)
```

```
b = np.array([9, 21, 10])
```

```
# 방정식 풀기: x = A^{-1} b
```

```
x = np.linalg.solve(A, b)
```

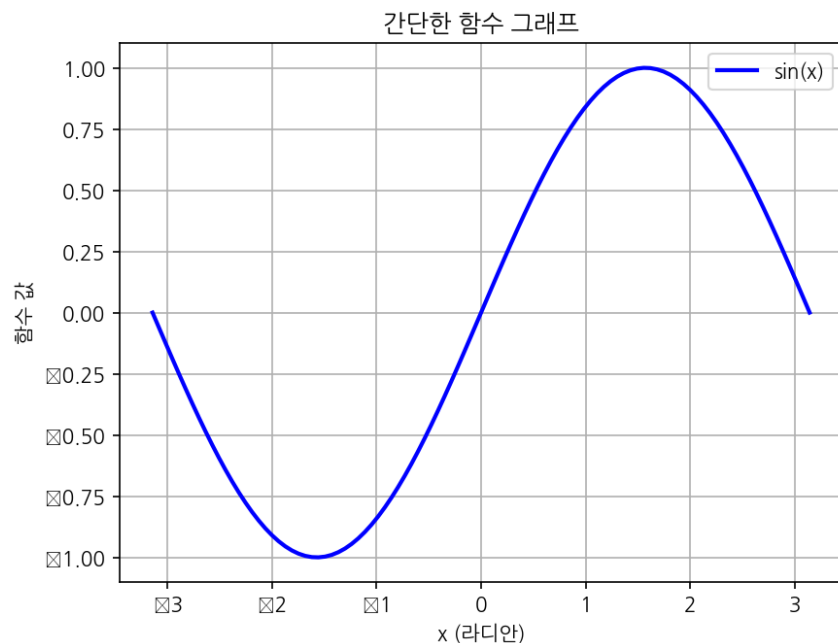
```
# 결과 출력
```

```
print("해: x =", x[0], ", y =", x[1], ", z =", x[2])
```

```
print("전체 벡터:", x)
```

## • 차트를 그려보자

C:\> pip install **matplotlib**



```
import matplotlib.pyplot as plt
import numpy as np # NumPy를 사용해 x, y 값을 계산합니다.
```

```
plt.rcParams['font.family'] = ['NanumGothic', 'DejaVuSans']
# NanumGothic이 없으면 DejaVuSans fallback
```

```
# x 값 생성 (0부터  $2\pi$ 까지 100개의 점)
x = np.linspace(-np.pi, np.pi, 100)
```

```
# y 값: 사인 함수
y = np.sin(x)
```

```
# 그래프 그리기
plt.plot(x, y, label='sin(x)', color='blue', linewidth=2) # 선 그래프
```

```
# 그래프 제목과 라벨 설정
plt.title('간단한 함수 그래프')
plt.xlabel('x (라디안)')
plt.ylabel('함수 값')
plt.legend() # 범례 표시
plt.grid(True) # 격자 표시
```

```
# 그래프 표시
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np # NumPy를 사용해 x, y 값을 계산합니다.
```

```
plt.rcParams['font.family'] = ['NanumGothic', 'DejaVuSans']
```

```
# x 값 생성 (0부터  $2\pi$ 까지 100개의 점)
x = np.linspace(-np.pi, np.pi, 100)
```

```
y = np.sin(x)
y2 = x ** 2
y3 = np.cos(x)
y4 = np.tan(x)
```

```
# 그래프 그리기
```

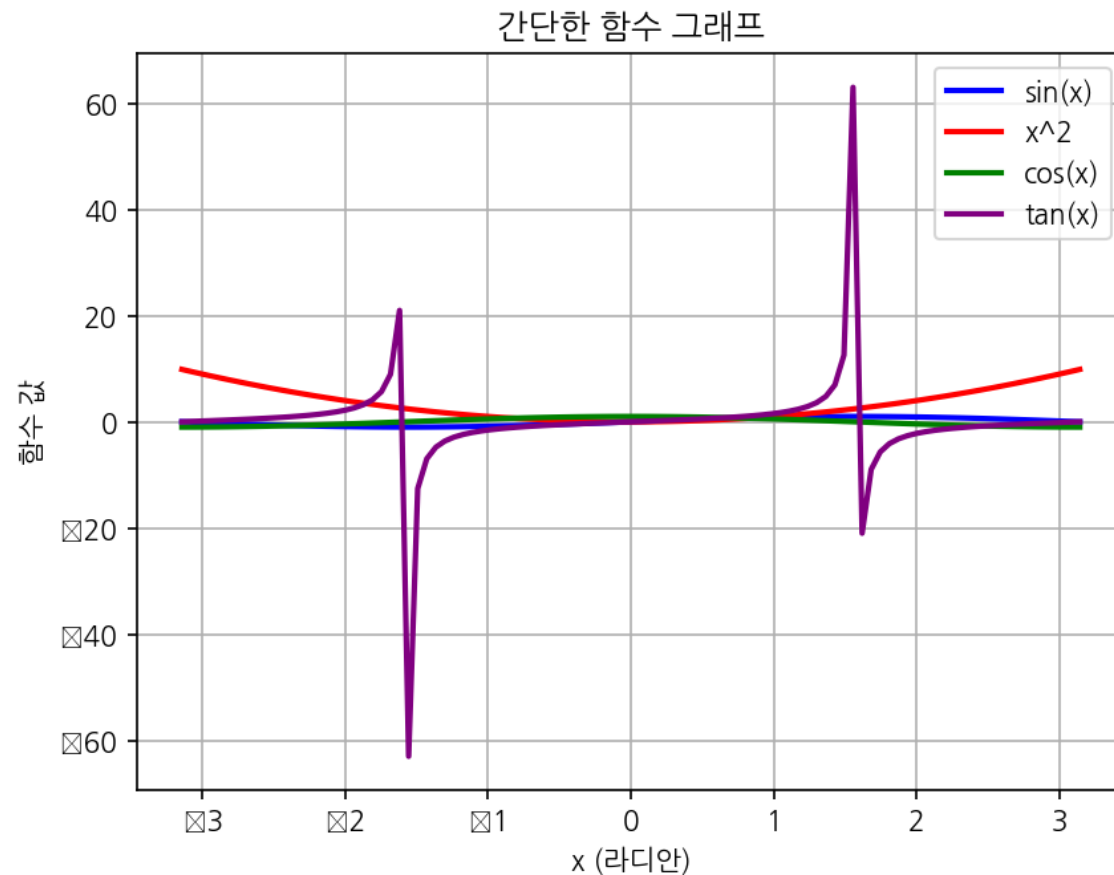
```
plt.plot(x, y, label='sin(x)', color='blue', linewidth=2) # 선 그래프
plt.plot(x, y2, label='x^2', color='red', linewidth=2) # 선 그래프
plt.plot(x, y3, label='cos(x)', color='green', linewidth=2) # 선 그래프
plt.plot(x, y4, label='tan(x)', color='purple', linewidth=2) # 선 그래프
```

```
# 그래프 제목과 라벨 설정
```

```
plt.title('간단한 함수 그래프')
plt.xlabel('x (라디안)')
plt.ylabel('함수 값')
plt.legend() # 범례 표시
plt.grid(True) # 격자 표시
```

```
# 그래프 표시
```

```
plt.show()
```



## • QR코드 만들기

C:\> pip install qrcode



```
import qrcode
import matplotlib.pyplot as plt
```

```
# URL 입력 & QR 생성
url = input("URL 입력: ").strip()
if not url.startswith(('http://', 'https://')):
    url = 'https://' + url
```

```
qr = qrcode.QRCode(version=1,
                    error_correction=qrcode.constants.ERROR_CORRECT_L,
                    box_size=10, border=4)
```

```
qr.add_data(url); qr.make(fit=True)
img = qr.make_image(fill_color="blue", back_color="yellow")
img.save('qr_code.png')
```

```
print(f"QR 저장됨: {url}")
```

```
# 표시
plt.figure(figsize=(6,6)); plt.imshow(img, cmap='gray'); plt.axis('off')
plt.show()
```

- 프로젝트 1
  - 웹 사이트 내용 추출해 보기 (위키피디아)

```
C:\> pip install requests beautifulsoup4 ltk
```



```
import requests
from bs4 import BeautifulSoup
import re
```

```
url = "https://en.wikipedia.org/wiki/CORTIS"
```

```
def scrape_wikipedia_words(url):
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/58.0.3029.110'}
    response = requests.get(url, headers=headers)
```

```
    if response.status_code != 200:
        print(f"Failed to retrieve page: {response.status_code}")
        return []
```

```
    soup = BeautifulSoup(response.text, 'html.parser')
```

```
    paragraphs = soup.find_all('p')
    full_text = ""
    for para in paragraphs:
        text = para.get_text()
        if text.strip():
            full_text += text + "\n"
```

```
    # 단어 추출: 영어 단어 기준 (\w+ 패턴으로 알파벳/숫자 추출, 소문자 변환)
    # 단어 경계 (\b) + 1개 이상의 단어 문자 (\w+) + 단어 경계 (\b)"
    # 개별 단어(공백이나 구두점으로 구분된 단어들)를 찾음.
    # 예: "Hello, world! 123" ["hello", "world", "123"] (소문자 변환 후).
    words = re.findall(r'\b\w+\b', full_text.lower())
    return words
```

```
word_list = scrape_wikipedia_words(url)
print("추출된 단어 리스트 (처음 50개):", word_list[:50])
print(f"총 단어 수: {len(word_list)}")
```

```
PS D:\...\2025_2_Python_문제해결\Codes\Lec06> & beautifulsoupTest.py
추출된 단어 리스트 (처음 50개): ['cortis', 'korean', '코르티스', 'stylised', 'in', 'all', 'caps', 'is', 'a',
'south', 'korean', 'boy', 'band', 'formed', 'by', 'big', 'hit', 'music', 'the', 'group', 'consists', 'of', 'five',
'members', 'james', 'juhoon', 'martin', 'seonghyeon', 'and', 'keonho', 'they', 'debuted', 'on', 'august',
'18', '2025', 'with', 'the', 'digital', 'single', 'what', 'you', 'want', 'the', 'lead', 'single', 'of', 'their', 'first',
'extended']
총 단어 수: 213
```

- 프로젝트 2
  - 워드 클라우드

```
C:\> pip install wordcloud nltk
```

```
from collections import Counter # 빈도 계산용 추가
from wordcloud import WordCloud # 워드 클라우드 생성용 추가
import matplotlib.pyplot as plt # 그래프 표시용 추가
```

```
import matplotlib.font_manager as fm # 폰트 경로 찾기용 추가
font_path = fm.findfont('NanumGothic')
```

```
...
```

```
word_list = scrape_wikipedia_words(url)
print("추출된 단어 리스트 (처음 50개):", word_list[:50])
print(f"총 단어 수: {len(word_list)}")
```

```
# 워드 클라우드 생성 부분 추가 (기존 코드 유지)
if word_list:
    # 단어 빈도 계산 (자주 등장하는 단어 위주)
    word_freq = Counter(word_list)
```

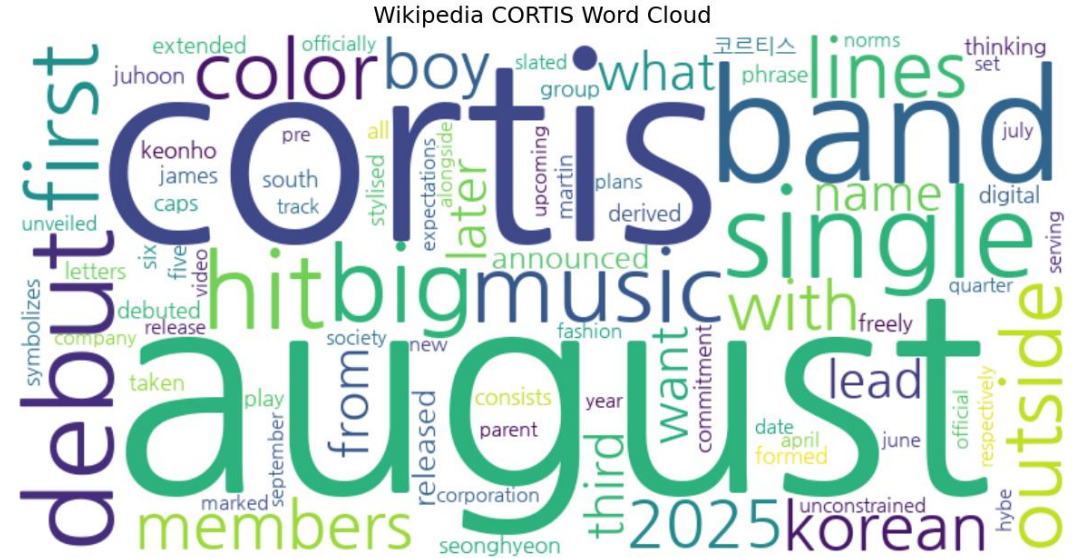
```
# 불용어(Stopwords) 제거 (예: 'the', 'and' 등 흔한 단어, 간단히 하드코딩)
stopwords = {'the', 'and', 'or', 'but', 'in', 'on', 'at', 'to', 'for', 'of', 'a', 'an', 'is', 'are', 'was', 'were', 'be', 'been', 'have', 'has', 'had', 'do', 'does', 'did', 'will', 'would', 'could', 'should',
'may', 'might', 'must', 'can', 'this', 'that', 'these', 'those', 'i', 'you', 'he', 'she', 'it', 'we', 'they', 'me', 'him', 'her', 'us', 'them', 'my', 'your', 'his', 'its', 'our', 'their'}
filtered_freq = {word: freq for word, freq in word_freq.items() if word not in stopwords and len(word) > 2}
```

```
# 워드 클라우드 생성
```

```
wc = WordCloud(width=800, height=400, background_color='white', max_words=100, font_path=font_path).generate_from_frequencies(filtered_freq)
```

```
# Matplotlib으로 표시
```

```
plt.figure(figsize=(10, 5))
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.title('Wikipedia CORTIS Word Cloud')
plt.show()
```





Questions?