

# 파이썬 프로그래밍

30차시

외부 모듈과  
turtle 모듈



## ⚠ 학습개요

- ... 외부 모듈 `numpy(numpy.org)`
- ... 외부 모듈 `matplotlib(matplotlib.org)`
- ... 프로젝트 lab1, lab2

## ⚠ 학습목표

- ... 외부 모듈 `numpy`를 설치하고 활용할 수 있다.
- ... 외부 모듈 `matplotlib`를 설치하고 활용할 수 있다.
- ... 표준 모듈 `datetime`을 활용할 수 있다.
- ... 표준 모듈 `turtle`을 활용할 수 있다.

## Chapter 1.

# 외부 모듈 `numpy` (`numpy.org`)

P Y T H O N   P R O G R A M M I N G

## ⚠ 써드 파티 모듈 numpy 설치

### + pypi

- 구글 플레이나 앱 스토어와 비슷하지만 무료

### + numpy(numpy.org)

- 과학 계산을 위한 기본 패키지
- 배열과 행렬을 기본으로 수학 연산을 위한 가장 기본적인 라이브러리를 제공

## ⚠ 써드 파티 모듈 numpy 설치

### + > pip install 패키지 이름

- 파이썬 설치 폴더 하부 python37-32\Scripts에서 윈도우의 명령 프롬프트 또는 윈도우 10의 파워셸(PowerShell)



```
Windows PowerShell

C:\Python\Python37-32\Scripts>pip install numpy
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/07/46/656c25b39fc152ea525eef14b641993624a6325a8ae815b200de57cff0b
c/numpy-1.16.4-cp37-cp37m-win32.whl
Installing collected packages: numpy
Successfully installed numpy-1.16.4

C:\Python\Python37-32\Scripts>
```

[그림 30-1] 모듈 numpy 설치



## ⚠ 모듈 numpy의 기본 자료형 ndarray

[코딩실습] 1차원 배열의 사칙연산

난이도 기본

```
1. import numpy as np
2.
3. aryA = np.array([1, 2, 3, 4, 5])
4. aryB = np.array([6, 7, 8, 9, 10])
5.
6. print(aryA + aryB)
7. print(aryA - aryB)
8. print(aryA * aryB)
9. print(aryA / aryB)
```

결과

```
[ 7  9 11 13 15]
[-5 -5 -5 -5 -5]
[ 6 14 24 36 50]
[0.16666667 0.28571429 0.375      0.44444444 0.5      ]
```

## ⚠ 모듈 numpy의 기본 자료형 ndarray

```
>>> import numpy as np
>>> data = np.array([1, 2, 3, 4, 5])
>>> data
array([1, 2, 3, 4, 5])
>>> print(data)
[1 2 3 4 5]
>>> data.shape
(5,)

>>> lst = [[1, 2, 3], [4, 5, 6], [6, 8, 9]]
>>> mat = np.array(lst)
>>> mat.shape
(3, 3)
```

## Chapter 2.

# 외부 모듈 matplotlib (matplotlib.org)

P Y T H O N   P R O G R A M M I N G



## ⚠ 데이터 시각화에 활용되는 matplotlib 설치

+ 데이터 시각화(data visualization) 모듈

+ matplotlib(matplotlib.org)

■ 아래를 비롯해 다양한 차트와 플롯 스타일을 지원

- 라인 플롯
- 바 차트
- 파이 차트
- 히스토그램
- 박스 플롯
- 스캐터 플롯 등

## ⚠ 데이터 시각화에 활용되는 matplotlib 설치

### + 모듈 matplotlib

```
Windows PowerShell
C:\Python\Python37-32\Scripts>pip install matplotlib
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/76/e2/f87bf36a2e460857ec9c28535e00102eb6903a727e427adc6427c5dc5453/matplotlib-3.1.0-cp37-cp37m-win32.whl (8.9MB)
    100% |#####| 8.9MB 3.0MB/s
Requirement already satisfied: numpy>=1.11 in c:\python\python37-32\lib\site-packages (from matplotlib) (1.16.4)
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/dd/d9/3ec19e966301a6e25769976999bd7bbe552016f0d32b577dc9d63d2e0c49/pyparsing-2.4.0-py2.py3-none-any.whl (62kB)
    100% |#####| 71kB 4.8MB/s
Collecting cycler>=0.10 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/f7/d2/e07d3ebb2bd7af696440ce7e754c59dd546ffe1bbe732c8ab68b9c834e61/cyclar-0.10.0-py2.py3-none-any.whl
Collecting kiwisolver>=1.0.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/20/6a/e5fff2ed776ab0cd11d7c1d5d3e5e549952464a6f1b9084b7ecbd8341352/kiwisolver-1.1.0-cp37-none-win32.whl (44kB)
    100% |#####| 51kB 6.4MB/s
Collecting python-dateutil>=2.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/41/17/c52facbfbdb163c7f57f3844689e3a78bae1f403648a6afb1d066d87fbb/python-dateutil-2.8.0-py2.py3-none-any.whl (226kB)
    100% |#####| 235kB 5.7MB/s
Collecting six (from cyclar>=0.10->matplotlib)
  Downloading https://files.pythonhosted.org/packages/73/fb/00a976f728d0d1fecfe898238ce23f502a721c0ac0ecfedb80e0d88c64e9/six-1.12.0-py2.py3-none-any.whl
Requirement already satisfied: setuptools in c:\python\python37-32\lib\site-packages (from kiwisolver>=1.0.1->matplotlib) (40.6.2)
Installing collected packages: pyparsing, six, cyclar, kiwisolver, python-dateutil, matplotlib
Successfully installed cyclar-0.10.0 kiwisolver-1.1.0 matplotlib-3.1.0 pyparsing-2.4.0 python-dateutil-2.8.0 six-1.12.0
C:\Python\Python37-32\Scripts>
```

[그림 30-2] 모듈 matplotlib

## ⚠ matplotlib로 기본 시각화 표현

### + matplotlib.pyplot을 별칭 plt로 import

- plt.plot()은 라인 플롯을 그리는 함수
- 실제 그림을 표시하는 함수인 plt.show()을 호출로 마무리

### + X축 값 2, 4, 6, 8에 대응하는 Y축 값 8, 10, 3, 6의 좌표를 연결하는 라인 플롯 **예제**

[코딩실습] 모듈 matplotlib으로 그래프 그리기

난이도 기본

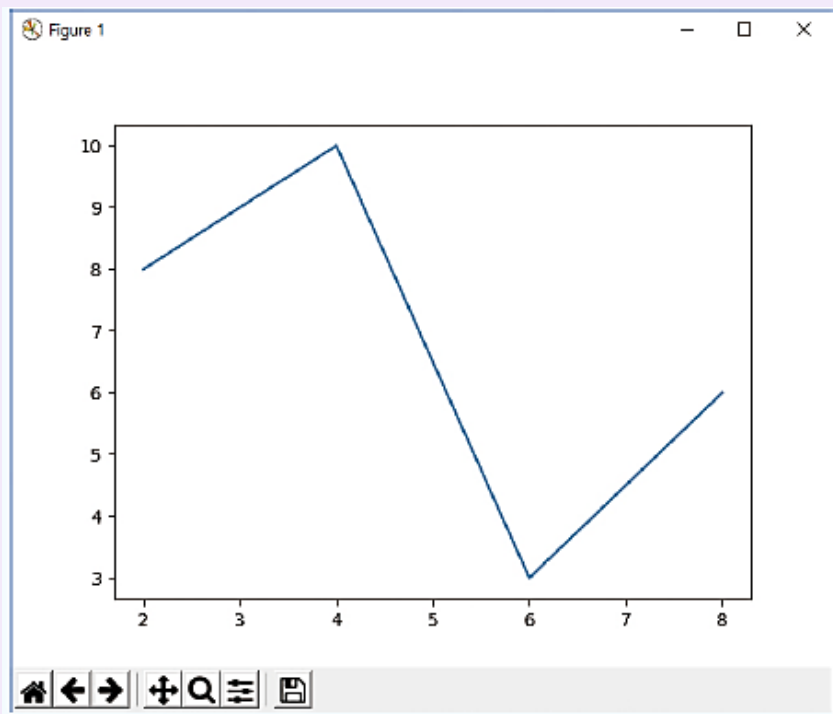
```
1. import matplotlib.pyplot as plt
2.
3. plt.plot([2, 4, 6, 8], [8, 10, 3, 6])
4. plt.show()
```

⚠ matplotlib로 기본 시각화 표현

+ X축 값 2, 4, 6, 8에 대응하는 Y축 값 8, 10, 3, 6의 좌표를  
연결하는 라인 플롯 **예제**

[코딩실습] 모듈 matplotlib으로 그래프 그리기 난이도 기본

결과



## ⚠ 가로 막대그래프 그리기

[코딩실습] 가로 막대그래프 그리기

난이도 응용

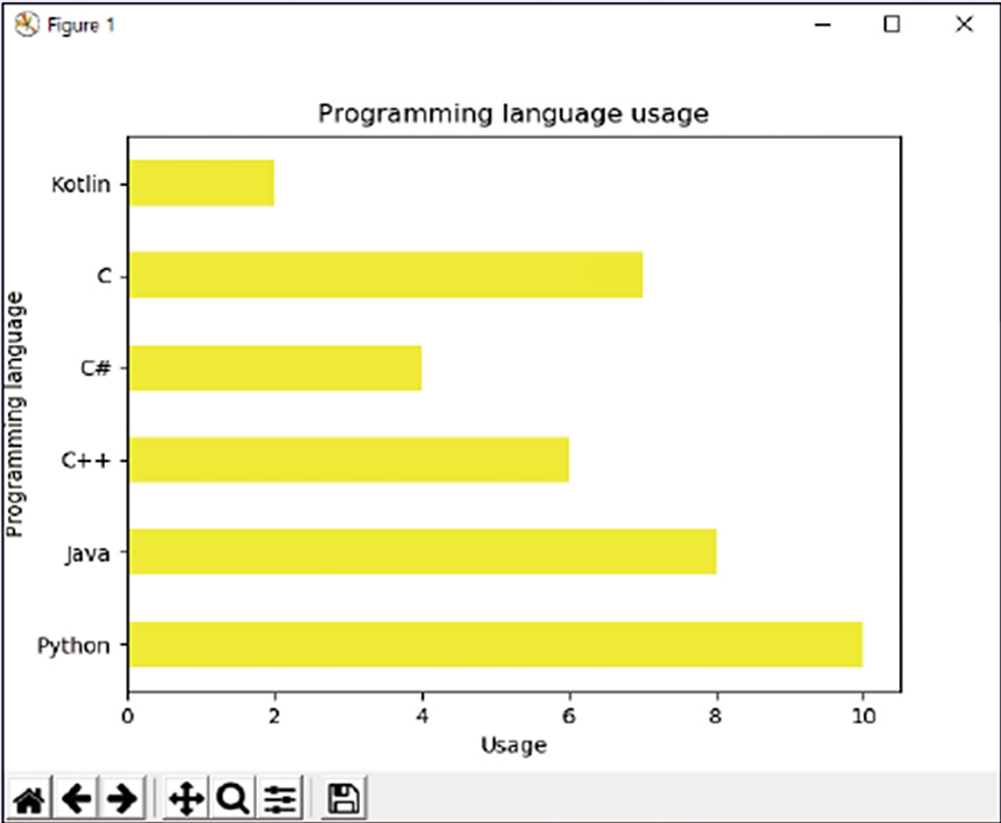
```
1. import matplotlib.pyplot as plt
2.
3. objects = ('Python', 'Java', 'C++', 'C#', 'C', 'Kotlin')
4. performance = [10, 8, 6, 4, 7, 2]
5.
6. plt.title('Programming language usage') # 제목
7. plt.xlabel('Usage') # x축 레이블
8. plt.ylabel('Programming language') # y축 레이블
9. plt.barh(objects, performance, height=0.5, color='yellow')
   # 가로바 그리기
10. plt.show()
```

! 가로 막대그래프 그리기

[코딩실습] 가로 막대그래프 그리기

난이도 응용

결과





## Chapter 3.

# 프로젝트 lab1, lab2

P Y T H O N   P R O G R A M M I N G

## 프로젝트 lab1

### 모듈 datetime 불러 날짜 계산

난이도 응용

표준 모듈 datetime 패키지는 datetime, date, time, timedelta 클래스를 제공한다.  
특히 timedelta는 시간 구간 정보에 활용된다.  
클래스 date와 timedelta를 사용해 오늘부터 크리스마스까지 남은 일수를 알아보고,  
오늘부터 100일 이후의 날짜도 알아보자.

### 문제 이해(Understanding)

오늘과 특정 일자를 저장한다.  
이 두 날짜의 차이를 출력한다. 또한 오늘부터 100일 이후의 날짜를 출력한다.  
모듈 datetime의 클래스 date는 특정 일자를 저장하는데 적합하다.  
또한 클래스는 timedelta는 구간 정보 활용에 적합하다.



## 프로젝트 lab1

### 설계(Design)

#### 알고리즘(Algorithm)

- ① 클래스 date의 매소드 today()와 date()를 사용해 오늘 날짜와 크리스마스를 저장한다.
  - 오늘 날짜와 두 날의 차이를 출력
- ② 클래스 datedelta로 100일 지정한 후 + 연산자로 100일 이후의 날짜를 저장해 출력한다.
  - `date100 = timedelta(days = 100)`  
# 100일 후 지정
  - `after100 = today + date100`  
# 오늘 이후 100일 연산

#### 표준 입출력 샘플

오늘: 2020년 6월 1일

크리스마스까지 남은 일수: 207

오늘 이후 100일 이후 날짜:

2020-09-09



## 프로젝트 lab1

[코딩실습]

난이도 응용

```
1. #%% 프로젝트 Lab 09-pl01-datetime.py 모듈 datetime 불러 날짜 계산
2. from datetime import date
3.
4. #오늘부터 크리스마스까지 남은 일수 계산
5. today = date.today() # 오늘 날짜
6. print('오늘: %d년 %d월 %d일' % (today.year, today.month, today.day))
7.
8. xmasday = date(today.year, 12, 25) #크리스마스 일 지정
9. delta = xmasday - today #크리스마스까지 남은 일수를 계산
10. print('크리스마스까지 남은 일수: {}'.format(delta.days)) #남은 일수 출력
11.
12. from datetime import timedelta
13.
14. #오늘부터 100일 이후 날짜 출력
15. date100 = timedelta(days = 100) #100일 후 지정
16. after100 = today + date100 #오늘 이후 100일 연산
17. print('오늘 이후 100일 이후 날짜: {}'.format(after100)) #오늘 이후 100일 출력
```

결과

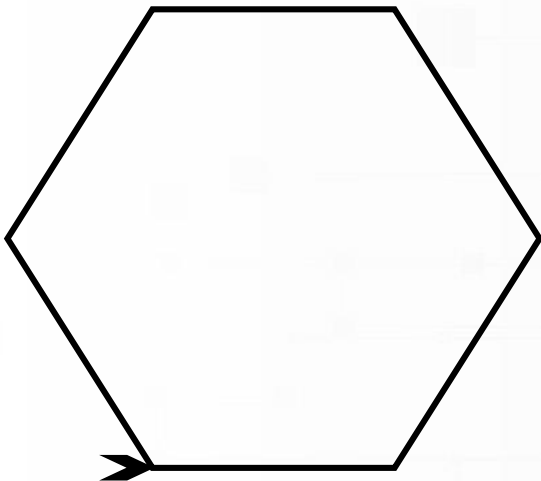
오늘: 2022년 1월 25일  
크리스마스까지 남은 일수: 334  
오늘 이후 100일 이후 날짜: 2022-05-05

## 프로젝트 lab2

### 터틀로 변형된 육각형 그리기

난이도 응용

터틀 예제에서 다각형을 그리는 함수를 구현해봤다.  
육각형인 경우, 한 변을 그리고 각도를  $360/6$ 인 60도씩 틀어 6번을 반복한다.  
한 변의 길이를 1에서 180픽셀까지 계속 증가시키면서 육각형의 내각인 60도를 약간 변형해 59도로 하면 어떤 모습일까?  
여섯 가지 색상을 한 번 마다 지정해 그려보자.



## 프로젝트 lab2

### 문제 이해(Understanding)

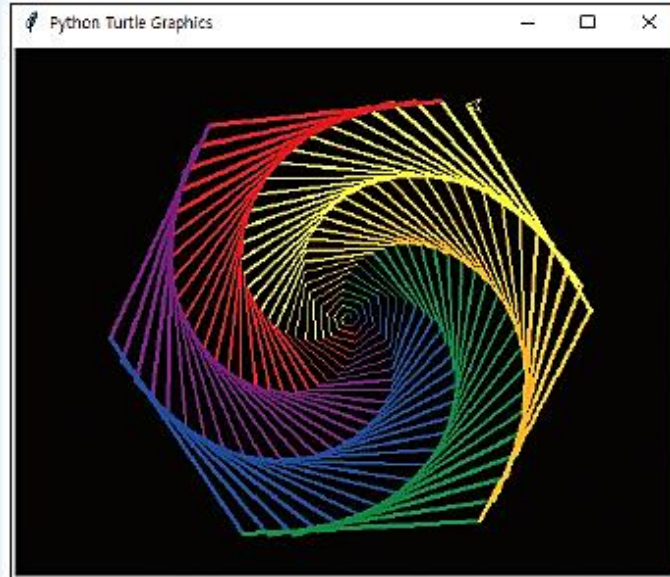
다음 결과가 그림이 우리가 구현하고자 하는 그림이다. 한 변의 길이를 1에서 180까지 증가시키면서 거북이를 전진시키고, 59도의 각도를 왼쪽으로 돌린다. 이 과정을 반복한다.

### 설계(Design)

#### 알고리즘(Algorithm)

- ① 한 변의 색을 지정할 6개 색상을 리스트로 만든다
- ② 터틀 윈도우 크기를 500, 400으로 지정하고 윈도우 바탕색을 검은색으로 수정한다.
- ③ 반복 for i in range(180) 구문으로 180번의 반복을 수행하면서 다음을 구현한다.
  - 함수 pencolor()로 색상 6개를 순서대로 지정
  - 함수 width(i/100 + 1)로 펜의 두께를 조금씩 증가
  - 함수 forward(i)로 i만큼씩 선을 그림
  - 함수 left(59)로 방향을 전환

#### 표준 입출력 샘플







## 프로젝트 lab2

[코딩실습]

난이도 실전

```
1. #%% 프로젝트 Lab 09-pl02-drawmodhexagon.py
2. # 터틀로 변형된 육각형 그리기
3. import turtle as t
4.
5. #선 색상에 사용할 이름 리스트
6. colors = ['red', 'purple', 'blue', 'green', 'orange', 'yellow']
7. t.setup(500, 400) #초기 윈도우의 크기 조정
8. t.bgcolor('black') #바탕색 변경

9. for i in range(180):
10.     t.pencolor(colors[i % len(colors)])
11.     t.width(i/100 + 1)
12.     t.forward(i)
13.     t.left(59)
```

결과

# SUMMARY



## ⚠ 외부 모듈 **numpy**([numpy.org](http://numpy.org))

... 과학 계산을 위한 기본 패키지

## ⚠ 외부 모듈 **matplotlib**([matplotlib.org](http://matplotlib.org))

... 데이터 시각화(data visualization) 모듈

## ⚠ 프로젝트 **lab1, lab2**

... 표준 모듈 **datetime**

... 표준 모듈 **turtle**