UNISTUDY

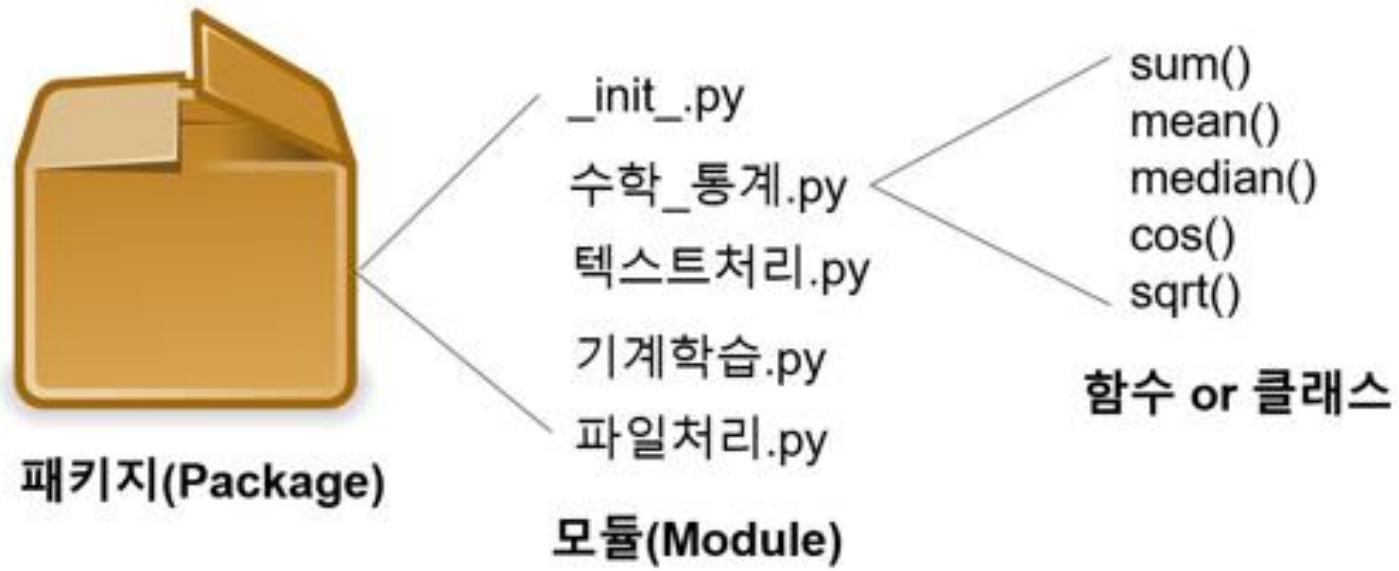
파이썬

24강. 객체지향 기법(패키지와 모듈)

1. 패키지와 모듈

- 파이썬을 설치하면 사용자 컴퓨터에 기본적으로 파이썬의 기본 라이브러리(Library)가 설치된 다.(C:₩Python37₩Lib) 라이브러리는 폴더의 모양을 띤 패키지(Package)와 파일(*.py)형식으로 제 공하는 모듈(Module)로 분류된다. 패키지는 관련 있는 모듈들을 하나의 꾸러미 형태로 묶어 주는 역할 이고, 모듈은 함수나 클래스를 파일로 작성해서 제공된다.
- 패키지와 모듈의 예를 나 타내고 있다. 패키지 안에는 ‘ init .py’ 라는 모듈을 포함하고 있어서 윈도우의 폴더와 구분할 수 있 다. 또한 모듈은 지금까지 실습을 위해서 PyCharm에서 작성한 파이썬 파일을 의미한다. 따라서 모듈 안에는 관련 있는 함수와 클래스들로 구성되어 있다. 예를 들면 ‘수학_통계.py’라는 모듈에는 수학과 통계를 처리하는 함수와 클래스들로 구성된다.

1. 패키지와 모듈

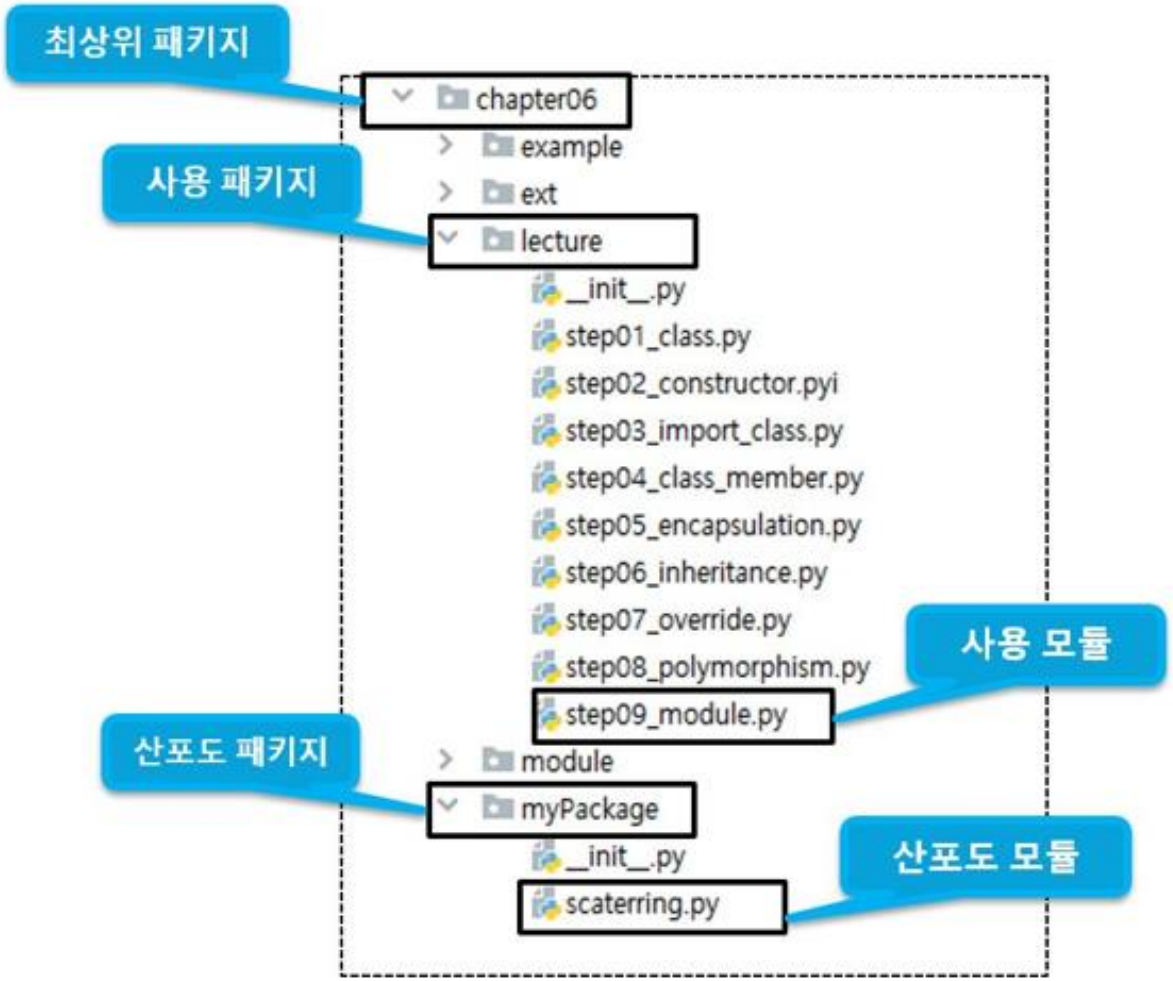

init.py
수학_통계.py
텍스트처리.py
기계학습.py
파일처리.py
sum()
mean()
median()
cos()
sqrt()
함수 or 클래스
패키지(Package)
모듈(Module)

2. 라이브러리 import

- 라이브러리를 사용하기 위해서는 먼저 패키지나 모듈을 import해야 한다. 다음은 패키지와 모듈을 import하는 대표적인 형식이다.
- 최상위 패키지(chapter06)와 2개의 하위 패키지(사용 패키지, 산포도 패키지)로 구성된 실습 환경이다.
- 산포도 패키지(myPackage)는 산포도 모듈(scattering.py)을 포함하고 있고, 산포도 모듈은 산포도 의 통계를 구하는 분산(variance)과 표준편차(standard deviation)를 계산하는 함수를 포함하고 있 다. 또한 사용 패키지(lecture)는 사용 모듈(step09_module.py)을 포함하고 있고, 사용 모듈은 산포 도 모듈의 분산과 표준편차 함수를 import하여 사용하는 모듈이다.

```
import 모듈    # 모듈 멤버 가져오기
import 패키지.모듈  # 패키지에 포함된 모듈의 멤버 가져오기
from 패키지.모듈 import 함수, 클래스, ... #  함수, 클래스 가져오기
```

2. 라이브러리 import



2. 라이브러리 import

- 산포도 패키지에 포함된 산포도 모듈(scattering.py) 의 내용이다. 산포도 모듈은 data를 대상으로 산술평균을 계산하는 함수와 산술평균을 이용하여 분산 과 표준편차를 계산하는 함수로 구성되어 있다.

chapter06.myPackage.scattering.py

```
# (1) 평균과 제곱근 모듈 import
from statistics import mean
from math import sqrt

# (2) 산술평균 함수
def Avg(data):
    avg = mean(data)
    return avg

# (3) 분산/표준편차 함수
def var_sd(data):  # [2,4,5,6,1,8] - avg
    avg = Avg(data)  # 함수 호출
    diff = [(d - avg) ** 2 for d in data] # list 내포
    var = sum(diff) / (len(data) - 1)
    sd = sqrt(var)

    return var, sd
```

Python Console

2. 라이브러리 import

- 산포도(Scatter)

- 평균을 중심으로 자료들이 얼마나 흩어져 있는지의 정도를 나타내는 통계학 용어이다. 산포도 값이 클수록 많이 흩어져 있고, 반대로 산포도 값이 작으면 자료들의 분포가 평균을 중심으로 고루 분포되었다는 의미이 다. 산포도의 통계량으로는 분산(variance), 표준편차(standard deviation)가 주로 많이 사용된다. 다음은 표본(sample)에 대한 분산과 표준편차의 수식이다.

2. 라이브러리 import

- 사용 패키지에 포함된 사용 모듈의 내용이다. 사용 모듈은 산포도 모듈의 함수를 사용하기 위해서 다음과 같이 두 가지 방법으로 import하여 분산과 표준편차를 사용한다.

- 방법1)은 import된 모듈의 전체 멤버(함수 or 클래스)를 가져오는 방법이고, 방법2)는 모듈의 멤버 중에서 특정 멤버만 가져오는 방법이다.
- 위 두 가지 모두 모듈의 함수를 가져오는 방법으로 다음과 같은 특징을 갖는다. 방법1)은 import 명 령문 형식은 간단 하지만 모듈에 포함된 멤버가 많은 경우 모든 멤버를 가져오기 때문에 소요되는 시간 이 길어진다.
- 방법2)는 import 명령문은 길지만 대신에 특정 멤버만 가져오기 때문에 소요되는 시간 이 비교적 짧다.

2. 라이브러리 import

- 두 가지 방법으로 어떻게 import하고 어떻게 해당 함수들을 호출하는 지에 대해서 정확히 구분할 필요가 있다.

chapter06.lecture.step09_module.py

```
# 1. 모듈 추가 (방법1)
# 형식) import 패키지명.모듈명
import chapter06.myPackage.scattering

# 데이터 셋
```

Python Console

2. 라이브러리 import

```
data = [1, 3, 1.5, 2, 1, 3.2]

# 산술평균 함수 호출
print('평균 : ', chapter06.myPackage.scattering.Avg(data))

# 분산과 표준편차 함수 호출
var, sd = chapter06.myPackage.scattering.var_sd(data)
print('분산 :', var)
print('표준편차 :', sd)



# 2. 모듈 추가 (방법2)
# 형식) from 패키지명.모듈명 import 함수명
from chapter06.myPackage.scattering import Avg, var_sd

print('평균 : ', Avg(data))

var, sd = var_sd(data)
print('분산 :', var)
print('표준편차 :', sd)
```

```
평균 :  1.95
분산 : 0.9350000000000
표준편차 : 0.966953980290
평균 :  1.95
분산 : 0.9350000000000
표준편차 : 0.966953980290
```

2. 라이브러리 import

- # 1. 모듈 추가 (방법1)
- import 명령어로 다른 패키지의 모듈을 가져오기 위해서는 다음 형식과 같이 구분자(.)을 이용하여 상위패키지, 하위 패키지명, 모듈 순서대로 나열하여 import 한다.
- import 상위패키지명.하위패키지명.모듈명
- 방법1에 의해서 가져온 모듈의 함수를 호출하기 위해서는 '상위패키지.하위패키지.모듈명.함수명()' 형식으로 작성한다. 만약 패키지와 모듈의 경로가 복잡하고 긴 경우 코딩하기 어렵기 때문에 다음과 같이 import 명령문에서 별칭을 지정하고, '별칭.함수명()' 형식으로 함수를 호출할 수 있다.
- import 상위패키지명.하위패키지명.모듈명 as 별칭
- 예를 들면 import chapter06.myPackage.scattering as scattering으로 패키지 경로에 scattering 별칭을 지정하면 var_sd()함수를 호출할 때 scattering.var_sd(data) 형식으로 호출할 수 있다.

2. 라이브러리 import

- # 2. 모듈 추가 (방법2)
- from 명령어로 다른 패키지의 모듈을 가져오기 위해서는 다음 형식과 같이 from 다음에 상위패키지, 하위 패키지명, 모듈 순서대로 나열하고 import 다음에 호출할 함수명을 작성한다.
- from 상위패키지명.하위패키지명.모듈명 import 함수명
- 방법2에 의해서 가져온 함수를 호출하기 위해서는 '함수명()' 형식으로 작성한다.

3. 시작점(main) 만들기

- 모듈 안에는 변수, 함수, 클래스 그리고 명령문 등을 포함할 수 있다. 이러한 모듈 안에서 프로그램이 시작되는 시작점(메인(main)이라고도 부름)을 만들 수 있다. 다음은 모듈 안에서 시작점을 만드는 형식이다.

```
if_name == "__main__" : # 프로그램 시작점
        명령문
```

- 분산과 표준편차 함수를 포함하고 있는 scattering 모듈 안에서 프로그램 시작점을 기 준으로 data를 만들고 data를 실인수로 하여 평균, 분산, 표준편차 함수를 호출한 다음 결과를 출력 한 예문이다. 만약 다른 모듈에서 scattering 모듈을 import하는 경우 프로그램 시작점 다음 문장 은 실행되지 않는다.

3. 시작점(main) 만들기

chapter06.myPackage.scattering.py

```
# (1) 평균과 제곱근 모듈 import
from statistics import mean
from math import sqrt

# (2) 산술평균 함수
def Avg(data):
    avg = mean(data)
    return avg

# (3) 분산/표준편차 함수
def var_sd(data):  # [2,4,5,6,1,8] - avg
    avg = Avg(data)  # 함수 호출
    diff = [(d - avg) ** 2 for d in data] #
list 내포
    var = sum(diff) / (len(data) - 1)
    sd = sqrt(var)
```

Python Console

3. 시작점(main) 만들기

```
    return var, sd

# 프로그램 시작점
if __name__ == "__main__" :
    data = [1, 3, 5, 7]
    print('평균=', Avg(data))
    var, sd = var_sd(data)
    print('분산=', var)
    print('표준편차=', sd)
```

```
평균= 4

분산= 6.666666666666667
표준편차= 2.581988897471611
```

- scattering 모듈에서 프로그램 시작점이 없는 예문이다. 프로그램 시작점을 지정하지 않고 평균, 분산, 표준편차 함수를 호출하고, print()함수를 이용하여 결과를 출력했다면 다른 모듈 에서 scattering 모듈을 import하면 print()함수의 결과도 함께 출력된다.

3. 시작점(main) 만들기

chapter06.myPackage.scattering.py

```
# (1) 평균과 제곱근 모듈 import
from statistics import mean
from math import sqrt

# (2) 산술평균 함수
def Avg(data):
    avg = mean(data)
    return avg


# (3) 분산/표준편차 함수
def var_sd(data):  # [2,4,5,6,1,8] - avg
    avg = Avg(data)  # 함수 호출
    diff = [(d - avg) ** 2 for d in data] #
list 내포
    var = sum(diff) / (len(data) - 1)
    sd = sqrt(var)


    return var, sd


# 프로그램 시작점 없음
data = [1, 3, 5, 7]
print('평균=', Avg(data))
var, sd = var_sd(data)
print('분산=', var)
print('표준편차=', sd)
```

Python Console

```
평균= 4

분산= 6.666666666666667
표준편차= 2.581988897471611
```

3. 시작점(main) 만들기

- 시작점이 없는 scattering 모듈을 module_import 모듈에서 import하는 예이다. import 명령문으로 scattering 모듈을 import하면 scattering 모듈의 함수도 가져 오지만 data와 print()함수도 함께 실행된다.
- 다른 모듈에서 scattering 모듈의 함수만 가져오고, 나머지는 실행되지 않도록 하기 위해서는 프로 그램 시작점을 만들어야 한다.

chapter06.myPackage.module_import.py	Python Console
`import chapter06.myPackage.scattering`	평균= 4 분산= 6.666666666666667 표준편차= 2.581988897471611

THANK YOU