

파이썬 모듈(module)

- 파이썬 모듈은 변수, 함수, 클래스 등을 모아놓은 파이썬 파일 패키지이며, 다른 프로그램에서 불러와서 사용할 수 있다.
- 모듈 가져오기(import)
 - 내부 모듈이나 외부 모듈(pip로 설치한 모듈)를 사용하려면 먼저 모듈 가져오기를 해야 한다.
- 모듈의 종류
 - 표준 모듈 : 파이썬에서 제공하는 모듈
 - math, os, sys, random, time
 - 서드 파티 모듈 : 외부 개발자, 회사 등이 독립적으로 만들어 제공하는 모듈
 - numpy, pandas, scikit-learn, nltk, flask
 - 패키지 설치 후에 import (예, pip install numpy)
 - 사용자 정의 모듈 : 사용자가 직접 만들어 사용하는 모듈

프로그램 모듈 - myModule.py

```
def sum(n):  
    sum = 0  
    for I in range(1, n+1) :  
        sum = sum + 1  
    return sum
```

```
def power(x, n) :  
    prod = 1  
    for I in range(1, n+1) :  
        prod = prod + 1  
    return prod
```

파이썬 모듈 — 사용자 정의 모듈

(1) *import module* - *module* 전체를 불러온다

```
import myModule
```

모듈의 메서드를 호출하려면 모듈 이름을 메서드와 함께 사용

modulename.method()

```
myModule.sum()
```

(2) *from module import class/method*

```
from myModule import power
```

가져오기한 메서드 이름만 호출-*method()*

```
power()
```

파이썬 모듈 — 사용자 정의 모듈

(3) `from module import *` - `module`에 포함된 모든 `class`나 `method`를 가져온다

```
from myModule import *
```

모듈의 모든 속성이나 메서드 이름을 직접 사용할 수 있다-`method()`

모든 속성과 메서드를 불러오기 때문에 메모리를 많이 차지한다

```
sum()
```

(4) `import module as alias` - `module`를 축약 이름(`alias`)으로 불러온다

```
import myModule as my
```

축약 이름을 사용하여 메서드 호출-`alias.method()`

```
my.power()
```

예외 처리(Exception Handling)

- 파이썬은 예외가 발생하면 프로그램 실행을 중지한다.
- 예외가 발생하더라도 프로그램 실행을 계속하려면 예외 처리를 해야 한다.

- 대표적인 예외의 종류

ZeroDivisionError : 0으로 나눌 때

IndexError : 인덱스가 범위를 벗어날 때

FileNotFoundError : 존재하지 않는 파일을 열 때

ValueError : 원하는 값을 입력 받지 못할 때

NameError : 정의되지 않은 변수를 사용할 때

TypeError : 데이터형이 맞지 않는 연산을 할 때

예외 처리

□ 예외 처리 구문

- try/except, try/except/finally, try/except/else

```
try :  
    [실행 코드 블록] # 실행 문장  
  
except [예외1] : # 예외1 발생하면  
    [예외1 처리 코드 블록] # 예외1 발생시 처리  
  
except [예외2] (as 오류메시지) : # 예외2가 발생하면 오류메시지로 기록하고  
    [예외2 처리 코드2 블록] # 예외2의 오류메시지 알림  
  
except : # 위에 명시되지 않은 오류이면  
    [그 이외의 모든 예외 처리 코드 블록] # 그 이외의 오류 처리  
  
else : #오류가 없으면 다음 블록 실행  
    [정상 실행 코드 블록] # 이 부분 실행  
  
finally :  
    [처리 코드 블록] # 예외와 상관없이 무조건 실행되는 부분
```

예외처리

```
# try/except 예외 처리
import sys, os
base_path = "G:\강의원고\자동화통신\강의원고\lecture_notes" # 파일 디렉토리

try:
    fp = open(os.path.join(base_path, 'sample.txt')) # 파일 열기
    sl = fp.readline() # 파일에서 한 줄 읽기
    value = int(sl.strip())
    print('값 =', value)

# FileNotFoundError 예외 발생
except FileNotFoundError as err: # err = 에러 메시지
    print("파일 오류: ", err)

# ValueError 예외 발생
except ValueError:
    print("정수로 변환할 수 없습니다")

# 예외 내용을 알 수 없을 때
except:
    print("알 수 없는 오류가 발생하였습니다")
```

예외처리

```
# try/except/else 예외 처리

b = int(input("Type a number: "))
try:
    a=1/b

# 예외 발생
except ZeroDivisionError:
    print("0으로 나눌 수 없습니다")

# 예외가 발생하지 않았을 때
else:
    print(a)
```


예외 처리

#try/except/finally 예외 처리

x = 5

try:

y = int(input())

result = x / y

print("계산 결과:{:.1f}".format(result))

ZeroDivisionError 예외 발생

except ZeroDivisionError:

print("0으로 나눌 수 없습니다")

예외 원인을 알 수 없을 때

except Exception **as** err:

print("원인 불명 예외: ", err)

예외 발생 여부와 관계없이 실행

finally:

print("예외처리가 끝났습니다")