

# 패키지

))모듈은 클래스의 집합, .py파일 이라고 한다.

))패키지는 모듈의 집합, 생성자

))패키지의 생성자는 길잡이—>

/home/code/test/point

A.py init.py

init

B.py

C.py

→생성자는 길잡이 파일마다 다 있어야 함.

import 경로 print.py

파일이 깨지는 경우

파이썬 파일이 가지는 위치가 코드로 잡혀서 깨짐

생성자에서 경로 초기화 시켜주거나

os.chdir(로 상대경로 지정(내기준지정))

os.pwd(내가지금있는경로가 출력)→터미널 기준

주피터노트북실행시키면 시스템기준

★

웬만하면 절대경로쓰기,폴더이름은 복수명 안붙이고 파일이름은 복수로 하기,파일  
폴더이름 서로 다르게

import os

os.chdir(시작경로지정)

wheel은 경로만들어줌

오류가 나도 실행되는 문 (예외처리)(한국인-외국인)

try:

except:

→while과 같은 무한 반복문

warning) 경고문

error)오류

→둘다 오류품음

코드 신버전을 쓰는데 옛날함수가 없을경우

try(신규

except(구버전 코드)

warning (더이상 지원안됨)

))except:라고만 쓰면 모든 오류를 다 받아들임.

))특정 오류만 났을때 처리하고싶으면  
발생오류as발생변수

- 1.exception:모든 오류
- 2.exception 발생오류 as 오류변수
- 3.exception 발생오류 :지정한 발생오류

2. except zerodivisionerror as e

```
    L -> e
    print (0으로 나누지 마세요)
    print(original.error.e)
    print(설명:)
    오류외우기!!!
```

try

else ??

에러날 곳을 미리 탐지하고 정확한 구문 쓰는것이 중요

지

## 질문 과 답

[—] [오전 10:41] ))패키지의 생성자는 길잡이—>

/home/code/test/point

A.py init.py

init

B.pyC.py

—>생성자는 길잡이 파일마다 다 있어야 함.

import 경로 print.py

파일이 깨지는 경우

파이썬 파일이 가지는 위치가 코드로 잡혀서 깨짐

[—] [오전 10:41] 이부분 부터 놓쳐서 그러는데 한번만 설명해주면 감사하겠습니다.

[

—] [오전 10:41] 그 코드 /home/code/test/point A.py init.py 여기서 부터 놓쳐서

[이준기010 7172 7013] [오전 10:42] 아넵넵

[이준기010 7172 7013] [오전 10:42] 잠시만요

[이준기010 7172 7013] [오전 10:47] 모든 내용을 간단하게 설명하자면 생성자 (init.py)안에 상대경로의 기준이 되는 위치를 os.chdir(상대경로의 기준점경로)를

[이준기010 7172 7013] [오전 10:47] 넣어주면 해당하는 파일이 패키지의 시작지점을 알리는 길잡이가 될수 있다 입니다

[이준기010 7172 7013] [오전 10:48] 쉽게 설명하면 파일이 깨지는 경우가 아니라. python /home/code/test/point A.py 로 실행을 하면 pointA.py 파일은 상대경로의 기준점 경로가 home이 되는거죠

[이준기010 7172 7013] [오전 10:48] 원래는 python point.py 라고 실행하면 상대경로의 기준점 경로가 test가 됩니다

[이준기010 7172 7013] [오전 10:49] 즉 그러면 안에 import (상대 경로 )를 써버리면, 기준점 경로가 달라져서 오류가 날수 있다 라는겁니다.

[－] [오전 10:53] 그럼 import안에 웬만하면 절대경로를 쓰라는 말씀이신가요?

[이준기010 7172 7013] [오전 10:53] 네 맞습니다~ 쓰더라도 절대 경로를 쓰고 만약에 쓰면 경로를 os.chdir로 지정하면 됩니다

[－] [오전 10:54] 절대경로를 or.chdir로요?

[이준기010 7172 7013] [오전 10:54] 아뇨 상대경로를 쓰면

[－] [오전 10:54] 아

[이준기010 7172 7013] [오전 10:54] os.chdir로 지정해주는게 맞습니다

[－] [오전 10:54] 만약 상대경로를 지정하게 되면 생성자파일(init.py)을 만들어서 os.chdir로 지정하나요??

[이준기010 7172 7013] [오전 10:56] 네 보통 패키지를 임포트 하는 경우가 많아서 생성자 파일을 만들어주고 난다음에 생성자 파일안에다가 경로를 지정합니다

[－] [오전 10:57] 제가 이해한게 맞는지 모르겠는데 파일 a,b,c마다 생성자 파일 만들어 주면 경로 오류가 덜 난다고 한것같은데 맞나요?ㄸ

[이준기010 7172 7013] [오전 10:58] 파일 마다 아니구 폴더 마다 생성자 파일을 만들어주는 거구요. 만약 생성자 파일을 폴더마다 만들어주면 상대경로를 실수로 적게 되는 경우라도 오류에 안전할수 있죠

[－] [오전 10:59] 네~

에러를 발생시키는 것

```
raise  
ErrorofError
```

라이즈는 에러오브 에러 값을 반환

```
def errors()
```

**raise ErrorofErrors**

```
errors()
```

→raise 쓰게 되면 에러가 남

ex) raise name as e

```
def setname(이름)
```

```
    if 이름 == 병신
```

```
        raise 병신은 안됩니다.
```

```
        print(e) →에러가 발생
```

→다른사람이 쓸때 병신이라는 구문을 에러를 시키고 싶다.

**raise ErrorofErrors는 에러 규칙을 만들때(일부러 에러를 발생시키고 싶을때)**

try-except구문을 잘 활용하면 좋다.

```
class 이름( Exception)
    def __str__(self)
        return "에러가 발생"
```

→위의 raise ErrorofError랑 다른점은:

raise는 에러라는 클래스만 올 수 있음(예를 들어 ZeroDivisionerror)

```
class zeroDivisionError(exception)
    def __str__(self)
        return (결과)
raise뒤에는 (에러 클래스)
```

기억)

1. 에러는 class이다.
2. 에러만들때 exception class상속
3. 에러결과함수는 `__str__`를 오버라이딩한다.
4. raise는 항상옆에 error class가 온다.

```
def set name
    if 이름 ==병신
```

```
# 사용자 정의 예외 클래스
class ErrorofErrors(Exception):
    def __str__(self):
        return "병신이라는 단어는 사용할 수 없습니다." # 예외가 발생했

# 입력을 검사하고 특정 단어가 있으면 예외 발생
def check_input(user_input):
    if "병신" in user_input:
        raise ErrorofErrors()

try:
```

```

# 사용자 입력을 받음
user_input = input("단어를 입력하세요: ")
check_input(user_input) # 입력된 단어를 검사
except ErrorofErrors as e:
    print(e) # "병신이라는 단어는 사용할 수 없습니다." 출력

```

```

# 사용자 정의 예외 클래스
class InvalidNameError(Exception):
    def __str__(self):
        return "병신은 안됩니다."

# 이름을 설정하는 함수
def setname(이름):
    if 이름 == "병신":
        raise InvalidNameError() # 사용자 정의 예외 발생

    print(f"이름이 정상적으로 설정되었습니다: {이름}")

# 예외 처리
try:
    setname("병신") # "병신"을 입력하면 예외가 발생
except InvalidNameError as e:
    print(e) # 예외 메시지 출력

```

```

def try (조건):
    if 조건=error

elif

```