Python

김선녕(sykim.lecture@gmail.com)

패키지
예외처리

패키지란?

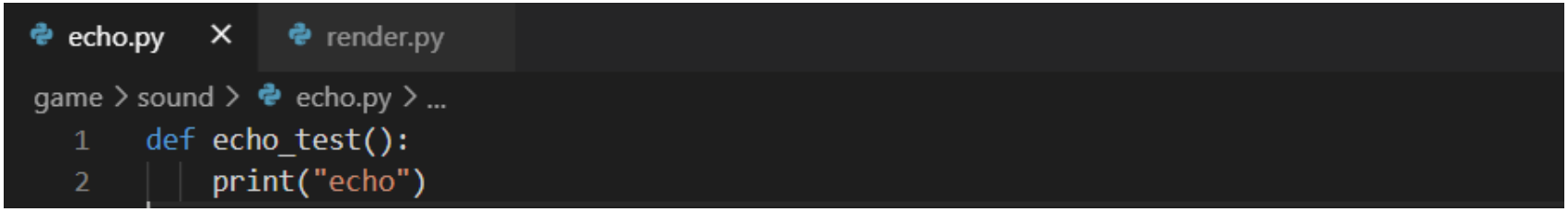
- 도트(.)를 이용하여 파이썬 모듈을 계층적으로 관리할 수 있다
- 파이썬 패키지는 디렉터리와 파이썬 모듈로 이루어진다.
- 패키지 구조로 파이썬 프로그램을 만드는 것이 공동 작업이나 유지 보수 등 여러 면에서 유리하다
- 패키지 구조로 모듈을 만들면 다른 모듈과 이름이 겹치더라도 안전하게 사용할 수 있다

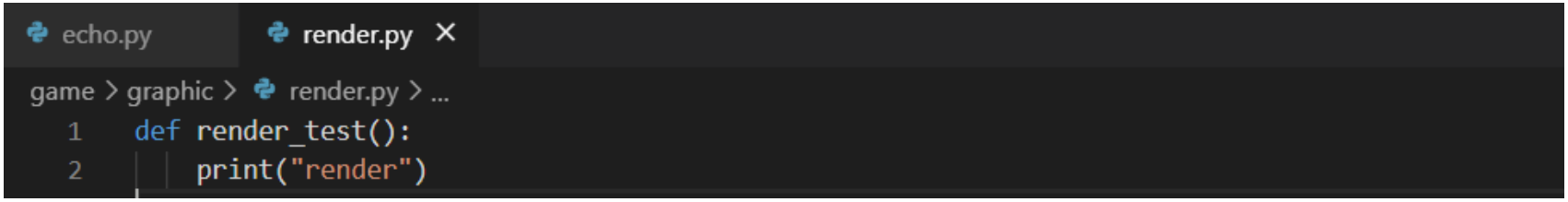
패키지 기본 요소

- 디렉토리와 파일 생성

```
/game/__init__.py
/game/sound/__init__. py
/game/sound/echo.py
/game/graphic/__init__.py
/game/graphic/render. py
```

- __init__.py 의 용도
  - __init__.py 파일은 해당 디렉터리가 패키지의 일부임을 알려주는 역할을 한다.
  - 만약 game, sound, graphic등 패키지에 포함된 디렉터리에 __init__.py 파일이 없다면 패키지로 인식되지 않는다.
  - python3.3 버전부터는 __init__.py 파일 없이도 패키지로 인식이 된다(PEP 420)
  - 하지만 하위 버전 호환을 위해 __init__.py 파일을 생성하는 것이 안전한 방법이다.





echo 모듈을 import하여 실행하는 방법(3가지)

echo.py | render.py | pakage1.py

pakage1.py > ...

```python
# 첫번째
from game.sound.echo import echo_test
from game.sound import echo
import game.sound.echo
game.sound.echo.echo_test()

# 두번째
echo.echo_test()

# 세번째
echo_test()

```

터미널 문제 출력 디버그 콘솔 1: powershell

```
PS G:\workspace_script\study\python> python pakage1.py
echo
echo
echo
PS G:\workspace_script\study\python>
```

불가능한 코드

echo.py | render.py | pakage1.py | pakage2.py

pakage2.py

```
1   '''
2   특정 디렉토리의 모듈을 *를 이용하여 import할 때에는
3   해당 디렉토리의 __init__.py 파일에 __all__이라는 변수를 설정하고
4   import할 수 있는 모듈을 정의해 주어야 한다.
5   '''
6   from game.sound import *
7   echo.echo_test()
8
```

터미널 문제 1 출력 디버그 콘솔 1: powershell

```
PS G:\workspace_script\study\python> python pakage2.py
Traceback (most recent call last):
  File "pakage2.py", line 7, in <module>
    echo.echo_test()
NameError: name 'echo' is not defined
PS G:\workspace_script\study\python>
```

_all_의 용도

echo.py | render.py | pakage1.py | pakage2.py | __init__.py

game > sound > __init__.py > ...

```
1   '''
2   sound 디렉토리에서 *를 이용하여 import할 경우
3   이곳에 정의된 echo 모듈만 import된다는 의미
4   '''
5   __all__ = ['echo']
```

pakage2.py를 재 실행해 보면 다음과 같은 결과를 확인할 수 있다.

터미널 문제 1 출력 디버그 콘솔

```
PS G:\workspace_script\study\python> python pakage2.py
echo
PS G:\workspace_script\study\python>
```

relative1.py

```python
from game.sound.echo import echo_test


def render_test():
    print("render")
    echo_test()
```

relative2.py

```python
import relative1
relative1.render_test()
```

터미널 문제 출력 디버그 콘솔 1: powershell

```
PS G:\workspace_script\study\python> python relative2.py
render
echo
PS G:\workspace_script\study\python>
```

패키지

예외처리

오류발생


```
1 f = open("noFile","r")
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-1-2867dc91f75f> in <module>()
----> 1 f = open("noFile","r")

FileNotFoundError: [Errno 2] No such file or directory: 'noFile'
```

```
1 4/0
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-2-221068dc2815> in <module>()
----> 1 4/0

ZeroDivisionError: division by zero
```

```
1 a = [1,2,3]
2 a[4]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-3-59fd82b1af02> in <module>()
      1 a = [1,2,3]
----> 2 a[4]

IndexError: list index out of range
```

오류 예외 처리 기법


```
try:
    ...
except:
    ...
```

```
try:
    ...
except  발생 오류:
    ...
```

```
try:
    ...
except 발생 오류 as 오류 메시지 변수:
    ...
```

try 블록 수행 중 오류가 발생하면 except 블록이 수행된다.
하지만 try블록에서 오류가 발생하지 않는다면 except 블록은 수행되지 않는다.

오류 예외 처리 기법


```
try:
    4/0
except ZeroDivisionError as e:
    print(e)
```

```
division by zero
```

```
# try ... else
# foo.txt라는 파일이 없다면 except절이 수행되고 foo.txt파일이 있다면 else절이 수행
try:
    f = open('foo.txt', 'r')
except FileNotFoundError as e:
    print(str(e))
else:
    data = f.read()
    f.close()
```

```
[Errno 2] No such file or directory: 'foo.txt'
```

```
# try .. finally
# try문 수행 도중 예외발생여부 상관없이 항상 수행된다.
f = open('foo.txt', 'w')
try:
    # 로직 수행
    pass
finally:
    print("finally")
    f.close()
```

```
finally
```

오류 회피하기


```
# 오류 회피하기
try:
    f = open("nofile", 'r')
except FileNotFoundError:   # 파일이 없더라도 오류를 발생시키지 않고 통과한다
    pass
```

```
# 오류 일부러 발생시키기(raise 이용)
class Bird:
    def fly(self):
        raise NotImplementedError
```

```
class Eagle(Bird):
    pass

eagle = Eagle()
eagle.fly()  # Bird class에 raise문에 의해 NotImplementedError발생 유발
```

```
---------------------------------------------------------------------------
NotImplementedError                       Traceback (most recent call last)
<ipython-input-19-01256a26320c> in <module>()
      3
      4 eagle = Eagle()
----> 5 eagle.fly()

<ipython-input-18-feb0189ed584> in fly(self)
      2 class Bird:
      3     def fly(self):
----> 4         raise NotImplementedError

NotImplementedError:
```