

02.파이썬 기본 문법-2

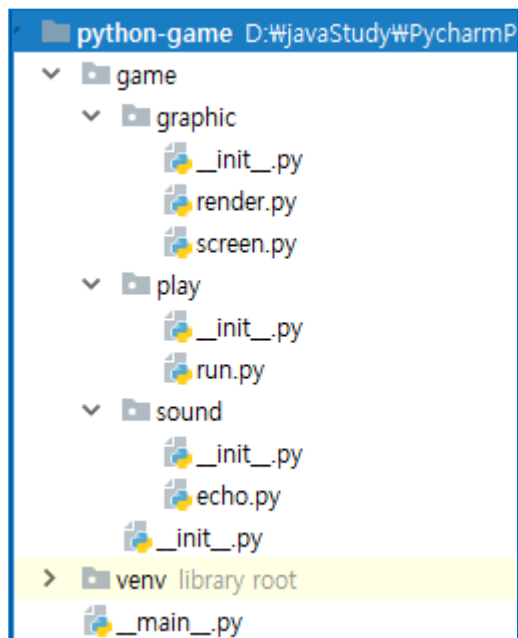
06. 함수

07. 모듈

08. 패키지

8.1 패키지란?

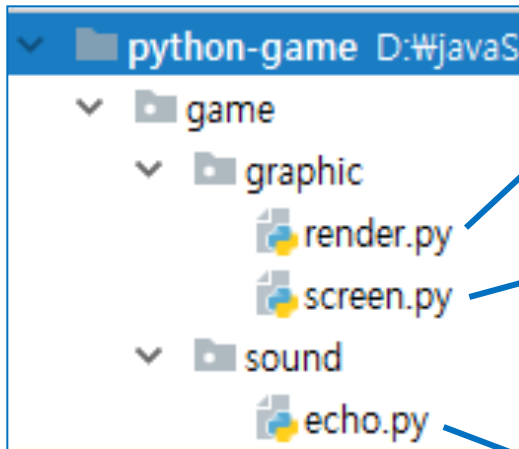
- 하나의 대형 프로젝트를 만드는 코드의 묶음
- 다양한 모듈들의 합, 폴더로 연결됨
- `__init__`, `__main__` 등이 사용됨
- 다양한 오픈소스들이 패키지로 관리된다.



- `sound`, `graphic`, `play` 는 디렉토리(패키지) 이다
- 파일명.py 파일은 파이썬 모듈이다.
- `__init__.py` 는 패키지를 인식시켜주는 역할을 하는 파일이다.
디렉토리가 패키지로 인식되기 위해 필요한 파일이다.
v3.3 이후에는 필요 없으나 호환을 위해 작성한다

[실습: 패키지 모듈 만들기]

- 1)python-game 프로젝트를 만듭니다.
- 2)game 패키지를 만듭니다.
- 3)game 패키지 아래에 graphic 패키지와 render 모듈, screen 모듈을 만듭니다.
- 4)game 패키지 아래에 sound 패키지와 echo 모듈을 만듭니다.



[render.py]

```
def test_render():
    print("test_render()")
```

[screen.py]

```
def test_screen():
    print("test_screen()")
```

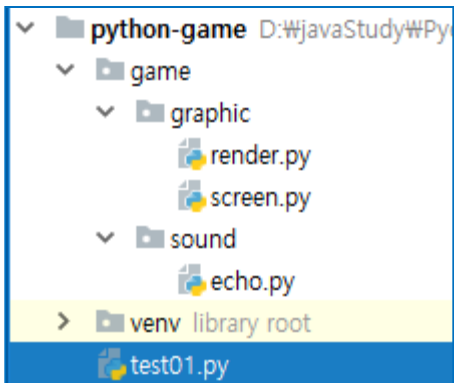
[echo.py]

```
def test_echo():
    print("test_echo()")
```

[실습: 함수 실행해 보기]

1) 최상위에 test01.py 를 추가 합니다.

2) test01.py 에서 render.py 모듈의 test-render() 함수를 실행해 봅니다.



[test01.py]

모듈을 import 해서 함수를 사용합니다.

```
import game.graphic.render  
game.graphic.render.test_render()
```

이경우 함수 경로까지 이름이 너무 길기 때문에 from ~ import 문을 사용합니다.

```
from game.graphic import render  
render.test_render()
```

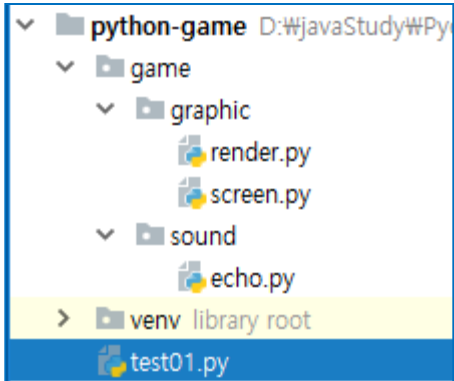
함수를 직접 import 해서 사용 할 수 있다.

```
from game.graphic.render import test_render  
test_render()
```

[실습: 함수 실행해 보기]

1) 최상위에 test01.py 를 추가 합니다.

2) test01.py 에서 render.py 모듈의 test-render() 함수를 실행해 봅니다.



[test01.py]

함수를 직접 import 할 수 없다. (import 는 모듈까지만)

```
import game.graphic.render.test_render  
test_render()
```



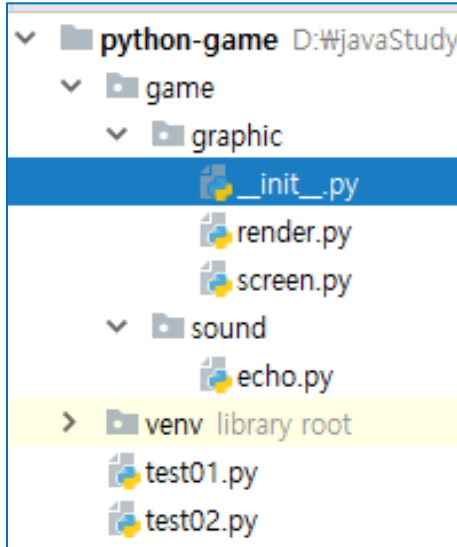
패키지까지만 import 한 경우 사용할 수 없다.
__init__ 파일을 정의 해야 사용할 수 있다.

```
import game.graphic  
game.graphic.render.test_render()  
game.graphic.screen.test_screen()
```



[실습: __init__ 파일 정의]

- __init__.py 파일은 해당 디렉토리가 패키지 일부임을 알리는 역할을 한다.
- v3.3 버전 이후에는 사용하지 않아도 되지만, 하위버전 호환을 위해 사용하는 것이 좋다.



[__init__.py]

.(점) 은 현재 디렉토리

```
from . import render  
from . import screen
```

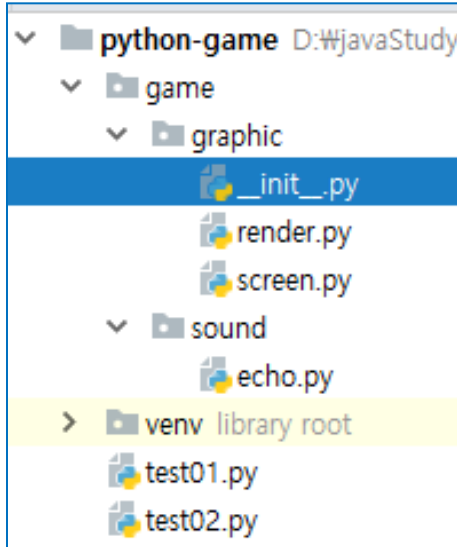
[test02.py]

```
import game.graphic  
game.graphic.render.test_render()  
game.graphic.screen.test_screen()
```



[실습: `__init__` 파일 정의 `__all__=[]`]

- `__init__.py` 파일은 해당 디렉토리가 패키지 일부임을 알리는 역할을 한다.
- v3.3 버전 이후에는 사용하지 않아도 되지만, 하위버전 호환을 위해 사용하는 것이 좋다.



[test02.py]

아래와 같이 * 은 `__init__.py` 파일에 정의 된 것만 사용 가능하다.

```
from game.graphic import *
render.test_render()
screen.test_screen()
```



[__init__.py]

```
__all__=["render", "screen"]

from . import render
from . import screen
```

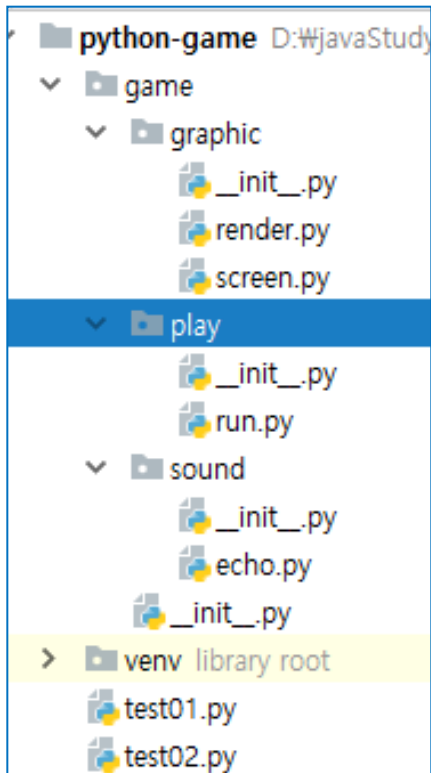
보통 모든 패키지에 `__init__.py` 을 위와 같은 형태로 정의한다.

[실습]

모든 패키지에
`__init__.py` 을 생성
한 후 정의해보세요

[패키지 내에서의 참조]

- 1) game 패키지 아래에 play 패키지와 run 모듈을 만듭니다.
- 2) __init__.py 을 만든 후 내용을 작성합니다.
- 3) run.py 모듈에서 start() 함수를 작성합니다.
- 4) start()함수는 다른 패키지의 test_render(), test_screen(), test_echo() 함수를 호출합니다.



[run.py]

```
from game.graphic import render
from game.graphic import screen
from game.sound import echo
```

```
def start():
    render.test_render()
    screen.test_screen()
    echo.test_echo()
```

```
# start()
```

[__init__.py]

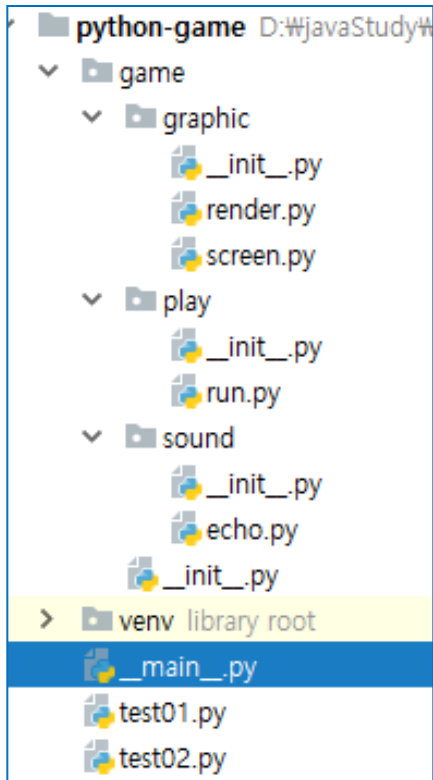
```
__all__=["run"]

from . import run
```

실습
상대경로로 호출하기
.. 사용

[_main_ 사용]

- 1) 루트에 `__main__.py` 를 만듭니다.
- 2) `play` 패키지의 `run` 모듈의 `start()` 함수를 실행합니다.



[_main_.py]

```
from game.play.run import start
```

```
if __name__ == "__main__":  
    start()
```