05-3 패키지

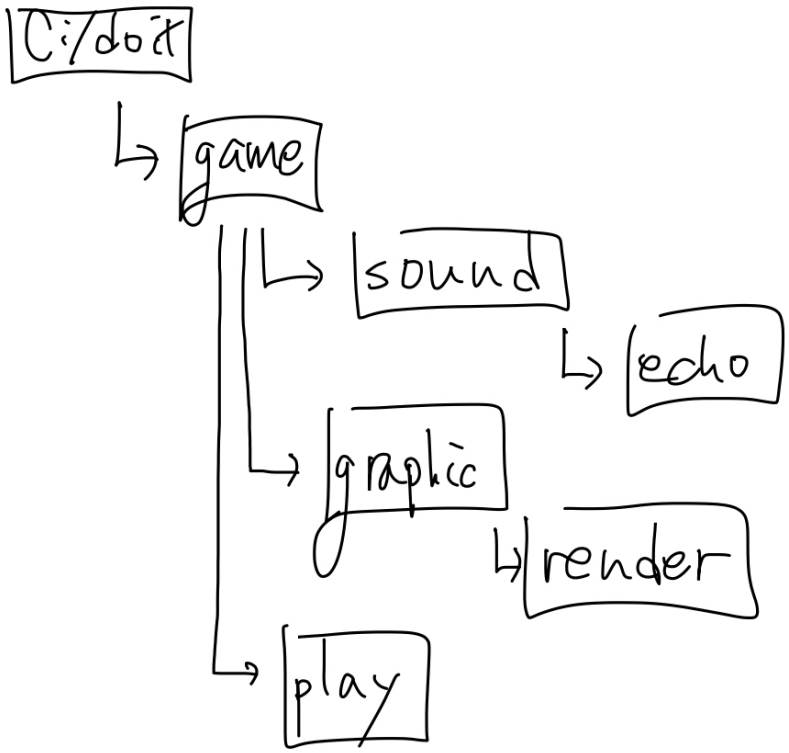
패키지 시작하기 전)

파이썬 디렉토리 생성 방법: os모듈 사용하기

os.mkdir(): 괄호 안에 원하는 경로 + 새로운 디렉토리 이름

패키지란 무엇인가

도트(.)를 사용하여 모듈을 계층적으로 관리할 수 있게 함



패키지 구조

패키지의 장점) 다른 모듈과 이름이 겹쳐도 더 안전하게 사용 가능

패키지 만들기

패키지 기본 구성 요소: 메인 디렉토리, 서브 디렉토리, .py파일들, __init __.py파일들

패키지 안의 함수 실행_3ways

```
#0
import game.sound.echo
game.sound.echo.echo_test()
```

```
#1
import game.sound import echo
echo.echo_test()
```

```
#2
import game.sound.echo import echo_test
echo_test()
```

▼ 불가능한 함수 실행법

```
import game
game.sound.echo.echo_test()
```

```
import game.sound.echo.echo_test
```

__init __.py의 용도

해당 디렉토리가 패키지의 일부임을 알려주는 역할

디렉토리에 __init __.py이 없다면 패키지로 인식되지 않을 수 있음

> **__all __**이란 특정 디렉토리의 모듈을 *을 이용하여 import할 때 __init __파일에 __all __이라는 변수를 설정하고 import할 수 있는 모듈을 정의해줘야함

relative 패키지

만약 graphic 디렉토리의 render모듈이 sound 디렉토리의 echo모듈을 사용하고 싶다면?

→

```
from game.sound.echo import echo_test
  def render_test():
    print("render")
    echo_test()

$ from game.sound.echo import echo_test 문장을
추가하여 graphic 디렉토리의 render 모듈에서
sounde 디렉토리의 echo모듈을 사용할 수 있도록 해줌
```

또는

```
from ..sound.echo import echo_test

$ 여기서 ..은 부모 디렉토리를 의미함
grapgic과 sound는 같은 depth이므로 부모디렉토리(..)
를 이용해서 import 가능
```