

파이썬 프로그래밍

모듈의 활용과 패키지

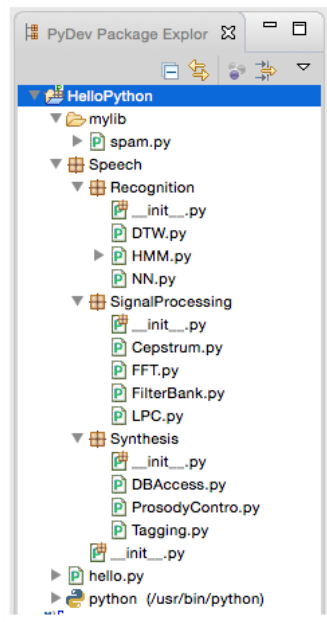


한국기술교육대학교
온라인평생교육원

■ 패키지

1. 패키지의 이해

- 패키지 (Package)
 - 여러 모듈들을 한데 묶어서 정리해 놓은 구조
 - 물리적으로 여러 모듈 파일을 모아 놓은 디렉토리에 해당
 - 최상위 디렉토리 이름이 패키지 이름이 된다.
 - 최상위 디렉토리 하위에 여러 서브 디렉토리는 해당 최상위 패키지의 하위 패키지가 된다.
- 예제: Speech 패키지
 - 각 폴더마다 `__init__.py` 파일 존재에 유의



- 모듈 = 파일, 패키지 = 디렉토리
- 최상위 패키지 이름 = 패키지 이름
- speech → 최상위 디렉토리, 패키지 이름
- 동일한 기능, 동일한 역할을 하는 모듈끼리 하위 패키지 내 묶음
- 패키지 = 모듈(라이브러리)이 많을 때 분류하여(디렉토리) 묶음
- `__init__.py` → 디렉토리를 패키지로 인식시키는 역할
- 서브 패키지에도 패키지와 마찬가지로 필요 (없으면 그냥 폴더)

■ 패키지

1. 패키지의 이해

```
import Speech
```

- Speech 디렉토리의 위치
 - sys.path(또는 PYTHONPATH 환경 변수)에 열거된 폴더 중 하나에 위치해야 한다.

- 이클립스의 경우 Python interpreter 창의 리스트 내 위치도 가능

- Speech/Recognition/HMM.py 코드 내용

```
def train():  
    print "Train"  
    pass  
  
def loadModel():  
    print "LoadModel"  
    pass  
  
def saveModel():  
    print "SaveModel"  
    pass
```

■ 패키지

2. __init__.py의 역할

- __init__.py의 역할

3. import 하기

```
import Speech.Recognition.HMM  
Speech.Recognition.HMM.train()
```

Train

- 패키지 내 특정 모듈, 모듈 내 함수 활용하는 방법
- import를 통해서 패키지 명 옆에 마침표(.)찍어 특정 모듈, 함수 활용

```
from Speech.Recognition import  
HMM HMM.train()
```

Train

- import 옆에 마침표 찍는 것이 불편 → from 구문 사용
- HMM이란 모듈 이름은 test.py 최상위 모듈에 import 됨
- HMM 이름 곧바로 써서 활용 가능

■ 패키지

3. import 하기

```
from Speech.Recognition.HMM import  
train train()
```

```
Train
```

3. import * 하기

```
from Speech.Recognition.HMM import *  
train() loadModel() saveModel()
```

```
Train  
LoadModel  
SaveModel
```

- import * 모듈 내 있는 변수 전부 다 가져옴