

세상의 속도를  
따라잡고 싶다면



# 점프 투 파이썬

박응용 지음(위키독스 운영자)



# 파이썬 날개 달기



## 05-7 외부 라이브러리



## ■ pip

- 파이썬 모듈이나 패키지를 쉽게 설치할 수 있도록 도와주는 도구
  - 외부 라이브러리를 사용하려면 pip 도구를 먼저 설치
- 
- pip install
    - PyPI(python package index): 파이썬 소프트웨어가 모인 저장 공간
    - 현재 10만 건 이상의 파이썬 패키지가 등록되어 있으며, 누구나 내려 받아 사용 가능

- pip 설치 명령어

```
pip install SomePackage
```

- SomePackage는 내려받을 수 있는 특정 패키지

## ■ pip

### ■ pip uninstall

- 설치한 패키지를 삭제하는 명령어

```
pip uninstall SomePackage
```

### ■ 특정 버전으로 설치하기

- 특정 버전을 지정하여 설치

```
pip install SomePackage==1.0.4
```

- 버전을 생략하면 최신 버전을 설치

```
pip install SomePackage
```

## ■ pip

- 최신 버전으로 업그레이드하기
  - 패키지를 최신 버전으로 업그레이드하는 명령어

```
pip install --upgrade SomePackage
```

- --upgrade 옵션과 함께 사용

- 설치된 패키지 확인하기
  - pip을 이용하여 설치한 패키지 목록 출력

```
pip list
```

## ■ Faker

- 테스트용 가짜 데이터를 생성할 때 사용하는 라이브러리

```
C:\> pip install Faker
```

- Faker 사용해 보기
  - 다음과 같은 형식의 테스트 데이터 30건 만들기

```
[(이름1, 주소1), (이름2, 주소2), ..., (이름30, 주소30)]
```

```
>>> from faker import Faker
>>> fake = Faker()
>>> fake.name()
'Matthew Estrada' ← 무작위로 생성한 이름을 리턴
```

## ■ Faker

### ■ Faker 사용해 보기

#### ■ 한글 이름으로 만들기

```
>>> fake = Faker('ko-KR')
>>> fake.name()
'김하은' ← 무작위로 생성한 한글 이름을 리턴
```

#### ■ 주소 만들기

```
>>> fake.address()
'충청북도 수원시 잠실6길 (경자주이읍)' ← 무작위로 생성한 한국 주소를 리턴
```

#### ■ 실행 결과 예

```
>>> test_data
[('이예진', '인천광역시 동대문구 언주거리 (경자김면)'), ('윤도윤', '광주광역시 서초구 삼성로 (주원최박리)'), ('서동현', '인천광역시 관악구 잠실가 (민석엄김마을)'), ('김광수', '울산광역시 양천구 서초대로'), (...생략...) ('박성현', '전라남도 서산시 가락27길 (준영박문읍)'), ('김성호', '경상남도 영월군 학동거리'), ('백지우', '경기도 계룡시 서초대1로'), ('권유진', '경기도 양주시 서초중앙313가 (춘자나리)'), ('윤서준', '경상남도 청주시 서원 구 서초대64가')]
```

#### ■ 이름과 주소 쌍 최종 명령어

```
>>> test_data = [(fake.name(), fake.address()) for i in range(30)]
```

## ■ Faker

### ■ Faker 활용하기

항목	설명
<code>fake.name()</code>	이름
<code>fake.address()</code>	주소
<code>fake.postcode()</code>	우편 번호
<code>fake.country()</code>	국가명
<code>fake.company()</code>	회사명
<code>fake.job()</code>	직업명
<code>fake.phone_number()</code>	휴대전화 번호
<code>fake.email()</code>	이메일 주소
<code>fake.user_name()</code>	사용자명
<code>fake.pyint(min_value=0, max_value=100)</code>	0부터 100 사이의 임의의 숫자
<code>fake.ipv4_private()</code>	IP 주소
<code>fake.text()</code>	임의의 문장(한글 임의의 문장은 <code>fake.catch_phrase()</code> 사용)
<code>fake.color_name()</code>	색상명



## ■ sympy

- 방정식 기호(symbol)를 사용하게 해 주는 외부 라이브러리
- pip로 설치하기

```
C:\>pip install sympy
```

- sympy 사용해 보기
  - 예) 시윤이는 가진 돈의  $\frac{2}{5}$ 로 학용품을 샀다고 한다. 이때 학用品을 사는 데 쓴 돈이 1,760원이라면 남은 돈은 어떻게 구하면 될까?
    - fractions 모듈과 sympy 모듈 필요
    - 시윤이가 가진 돈:  $x$
    - 일차방정식:  $x * (\frac{2}{5}) = 1760$

## ■ sympy

### ■ sympy 사용해 보기

#### ■ 최종 코드 →

```
from fractions import Fraction
import sympy

# 가지고 있던 돈을 x라고 하자.
x = sympy.symbols("x")

# 가지고 있던 돈의 2/5가 1760원이므로 방정식은  $x * (2/5) = 1760$ 이다.
f = sympy.Eq(x*Fraction('2/5'), 1760)

# 방정식을 만족하는 값(result)을 구한다.
result = sympy.solve(f)    # 결과값은 리스트

# 남은 돈은 다음과 같이 가지고 있던 돈에서 1760원을 빼면 된다.
remains = result[0] - 1760

print('남은 돈은 {}원 입니다.'.format(remains))
```

**실행 결과**

남은 돈은 2640원입니다.

## ■ sympy

### ■ sympy 활용하기

- 예)  $x^2 = 1$ 과 같은 2차 방정식의 해 구하기

```
>>> import sympy
>>> x = sympy.symbols("x")
>>> f = sympy.Eq(x**2, 1)
>>> sympy.solve(f)
[-1, 1]
```

- 예) 연립방정식의 해 구하기

$$\begin{aligned}x + y &= 10 \\ x - y &= 4\end{aligned}$$

```
>>> import sympy
>>> x, y = sympy.symbols('x y')
>>> f1 = sympy.Eq(x+y, 10)
>>> f2 = sympy.Eq(x-y, 4)
>>> sympy.solve([f1, f2])
{x: 7, y: 3}
```



*"Life is too short,  
You need Python!"*

인생은 너무 짧으니,  
파이썬이 필요해!

감사합니다.