

# Python 07. 객체 지향 프로그래밍



## | Background

- ✓ 클래스의 활용
- ✓ 클래스, 인스턴스, 메서드

## | Goal

- ✓ 외부 라이브러리를 통한 클래스 구조 이해
- ✓ 클래스 기반의 코드 해석

## | Problem

Faker는 개발시 활용할 수 있는 가상의 데이터를 생성해주는 파이썬 패키지이다.

워크샵에 등장하는 코드는 모두 Github(<https://github.com/joke2k/faker>) 문서의 예제이다.

지금까지 배운 파이썬 개념을 활용하여 해석 하시오.

### 1. pip

아래 명령어는 (1) 무엇을 위한 명령인지 (2) 실행은 어디에서 해야하는지 작성 하시오.

```
$ pip install faker
```

### 2. Basic Usages(<https://github.com/joke2k/faker#basic-usage>)

Faker는 다양한 메서드를 통해 임의의 결과값을 반환해준다.

임의의 영문 이름을 반환하는 아래 코드에서 **라인별 의미**를 주석을 참고하여 작성하시오.

```
from faker import Faker # 1 _____을 하기 위한 코드이다.
fake = Faker()           # 2 Faker는 _____, fake는 _____이다.
fake.name()              # 3 name()은 fake의 _____이다.
```

# Python 07. 객체 지향 프로그래밍



## 3. Localization(<https://github.com/joke2k/faker#localization>)

Faker는 다양한 언어의 Locale을 지원한다.

1. 인자 없이 호출 시에는 **영문**이 기본 설정이다. (en\_US)

```
fake = Faker()
fake.name()
# => 'Shelly Wilcox' (랜덤이므로 결과 값이 다를 수 있음)
```

2. Locale 정보를 포함하여 호출 시에는 **해당 언어 설정**을 따른다.

```
fake_ko = Faker('ko_KR')
fake_ko.name()
# => '박진우' (랜덤이므로 결과 값이 다를 수 있음)
```

직접 해당하는 기능을 구현한다고 하였을 때, 빈칸 (a), (b), (c)에 들어갈 코드로 적절한 것을 작성하시오. (힌트: 생성자 메서드와 함수의 개념)

```
class Faker():
    def __(a)__(b), (c)):
        pass
```

# Python 07. 객체 지향 프로그래밍



## 4. Seeding the Generator(<https://github.com/joke2k/faker#seeding-the-generator>)

컴퓨터 프로그래밍에서 임의의 값을 반환하는 경우(난수 생성 등) 시드라는 개념이 있다. 시드를 설정하게 되면 동일한 순서로 난수를 발생시킬 수 있어 일반적으로 디버깅을 위하여 활용 된다.

```
import random

random.random() # => 임의의 수
random.random() # => 임의의 수

random.seed(7777)
random.random() # => 0.8170477907294282

random.seed(7777)
random.random() # => 0.8170477907294282
```

아래의 코드를 실행 했을 때, #1과 #2에서 출력되는 결과를 각각 작성하고, seed()는 어떤 종류의 메서드인지 작성하시오.

```
fake = Faker('ko_KR')
Faker.seed(4321)

print(fake.name()) # 1

fake2 = Faker('ko_KR')
print(fake2.name()) # 2
```

# Python 07. 객체 지향 프로그래밍



## 4. Seeding the Generator

아래의 코드를 실행 했을 때, #1과 #2에서 출력되는 결과를 각각 작성하고, `seed_instance()`는 어떤 종류의 메서드인지 작성하시오.

```
fake = Faker('ko_KR')
fake.seed_instance(4321)

print(fake.name())      # 1

fake2 = Faker('ko_KR')
print(fake2.name())     # 2
```

`seed()`와 `seed_instance()`는 각각 어떠한 용도로 쓰일 수 있는지 작성하시오.