클래스 (2)

PDF	비어 있음
비고	비어 있음
# 숫자	15
실습문제	비어 있음
실습문제답안	비어 있음

02. 인스턴스 (Instance)

02-01. 인스턴스 개요

02-01-01. 인스턴스란

💡 클래스로 객체를 생성하면 메모리에 매번 서로 다른 주소를 가진 인스턴스가 할당된다. 인스턴스는 각각 독립적인 공간이 되며, 필드가 있을 경우 서로 다른 상태 값을 가진다.

02-01-02. 인스턴스 생성

- 클래스명 뒤에 소괄호(())를 붙여 객체를 생성한다.
  - 클래스 내에 생성자(__init__ 메서드)가 정의되어 있는 경우, 생성자에서 사용하는 매개변수를 전달하면서 객체를 생성할 수 있다.

```
squirrel = Person('다람쥐', 900) Gorilla = Person('고릴라', 20) print(squirrel) # <__main__.Person object at 0x000002E16EE94560> print(Gorilla) # <__main__.Person object at 0x000002E16EE94BF0> print(squirrel.national) # korea print(squirrel.greeting()) # 안녕하세요 print(Gorilla.information()) # I'm from korea and I use korean. My name is 고릴라. I'm 20 print(Gorilla.favorite('black')) # I love black
```

02-02. 인스턴스 속성

02-02-01. 클래스 속성과 인스턴스 속성

- 인스턴스 변수는 인스턴스별 데이터를 위한 것이고, 클래스 변수는 그 클래스의 모든 인스턴스에서 공유되는 어트리뷰트와 메서드를 위해 사용한다.
- 클래스 속성은 공유되는 속성으로 변경이 발생하면, 전체 객체에서 변경될 수 있다.

```
class Character: skills = [] def __init__(self, nickname, type): self.nickname = nickname self.type = type def add_skill(self, skill): self.skills.append(skill) def show_skill(self): return self.type + ' ' + self.nickname + '님의 보유 스킬 ' + str(self.skills) my_character = Character('산골짜기다람쥐', '전사') your_character = Character('흑염룡의고릴라', '법사') my_character.add_skill('곤봉 휘두르기') your_character.add_skill('마법진 그려서 날려버리기') print(my_character.show_skill()) # 전사 산골짜기다람쥐님의 보유 스킬 ['곤봉 휘두르기', '마법진 그려서 날려버리기']
```

- 따라서 객체마다 다른 데이터를 관리해야 하는 경우에는 인스턴스 속성으로 설정해야 한다.

```
class Character: def __init__(self, nickname, type): self.nickname = nickname self.type = type self.skills = [] def add_skill(self, skill): self.skills.append(skill) def show_skill(self): return self.type + ' ' + self.nickname + '님의 보유 스킬 ' + str(self.skills) my_character = Character('산골짜기다람쥐', '전사') your_character = Character('흑염룡의고릴라', '법사') my_character.add_skill('곤봉 휘두르기') your_character.add_skill('마법진 그려서 날려버리기') print(my_character.show_skill()) # 전사 산골짜기다람쥐님의 보유 스킬 ['곤봉 휘두르기']
```

- 인스턴스 속성과 클래스 속성으로 같은 이름을 사용하면, 인스턴스 속성을 우선한다.

```python
class Character: nickname = '땅파먹는개미핥기' def __init__(self, nickname, type): sel
f.nickname = nickname self.type = type my_character = Character('산골짜기다람쥐', '전
사') print(my_character.nickname) # 산골짜기다람쥐
```