

07. 객체지향 프로그래밍

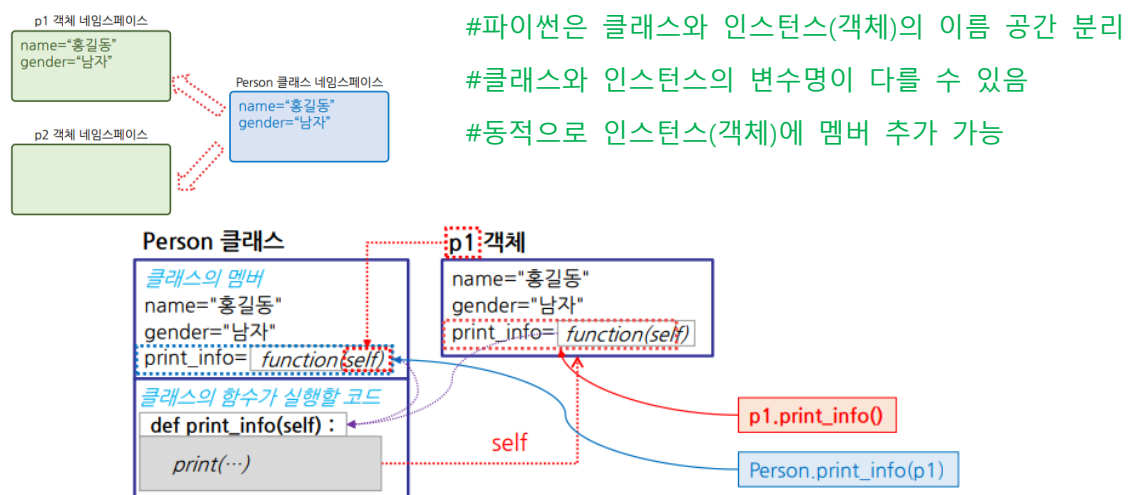
#src: 8_Python/ch07_객체지향프로그래밍 참고

1) 객체와 클래스

- (1) 클래스: 객체를 만들기 위한 틀. #클래스 생성: `class ClassName:`
- (2) 객체: 클래스의 인스턴스 (객체 특성: 데이터, 변수, 필드, 속성, 행위, 함수, 메소드, 기능 등)

2) 변수와 메소드

- (1) 클래스 사용 이유: 객체에 데이터를 저장(변수)하고, 객체 고유의 기능(메소드) 갖기 위함
- (2) 객체를 이용한 참조와 클래스를 이용한 참조:



- (3) **인스턴스 메소드**; 객체를 이용해 참조할 수 있는 메소드로 첫번째 인자는 **self**
- (4) **class 메소드**; @classmethod를 통해서 선언한 메소드 첫번째 인자는 **cls** or **clazz**
- (5) **static 메소드**; @staticmethod를 통해서 선언한 메소드 **인자 없음**

3) 생성자(__init__)와 소멸자(__del__)

- (1) 생성자함수: `__init__(self, 매개변수1, ...)` #중복정의(오버로딩) 불가(마지막 정의만 유효)
객체가 생성될 때 자동 실행, 생성시 필요한 코드(멤버 변수 초기화)를 포함
- (2) 소멸자함수: `__del__(self)` #소멸시 필요한 코드 포함할 수 있음
객체가 소멸될 때 자동 실행, **인스턴스 객체의 레퍼런스 카운트가 0이 될 때 실행**

4) 상속과 재정의 class ClaaName(SuperClassName):

- (1) 부모 클래스의 생성자 사용: `SuperClassName.__init__(self, 매개변수)`
- (2) `super()`; 부모 클래스의 멤버를 참조 ex) `super().__str__()`
- (3) 정적(static)변수: `클래스명_클래스명_변수명`; 여러 객체들 사이에서 데이터 공유
↳#classmethod에서는 `cls._변수명`도 사용 가능