

학습 내용

2부. 프로그래밍 언어 활용

6장. 모듈과 패키지



7장. 객체지향 프로그래밍

- 1. 객체와 클래스
- 2. 상속과 재정의

8장. 예외 처리

9장. 파일 입/출력 프로그래밍

10장. 데이터베이스 연동

1절 객체와 클래스

- 클래스 : 객체를 만들기 위한 틀(template, blueprint)
- 객체 : 클래스의 인스턴스
- 객체의 명사적 특성 : 데이터(Data), 변수(Variable), 필드(Field), 속성(Attribute)
- 객체의 동사적 특성 : 행위(Behavior), 함수(Function), 메서드(Method), 기능(Operation)

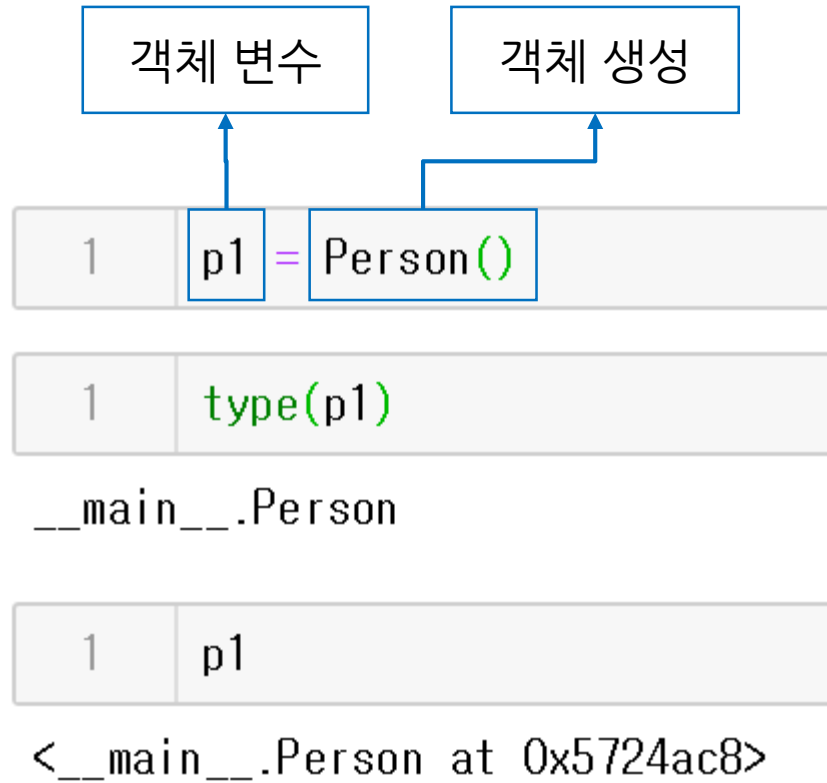
클래스 정의

- class 키워드 사용
- 카멜 표기법

```
class ClassName:  
    class_body
```

```
1  ▼ class Person:  
2      pass  
3
```

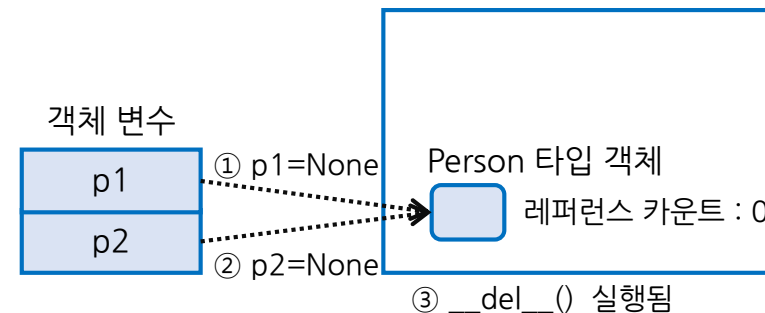
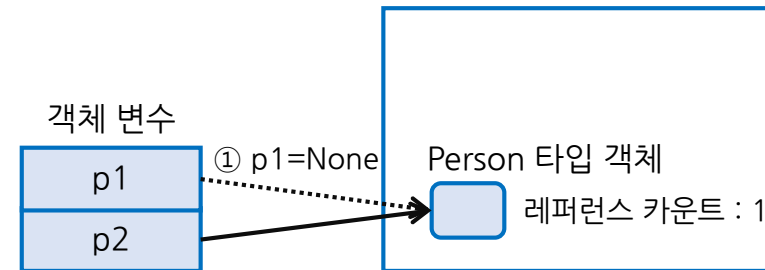
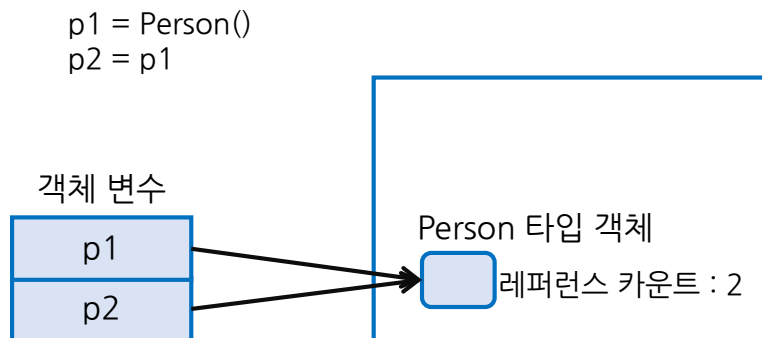
객체 생성



생성자와 소멸자

생성자는 객체가 생성될 때 자동으로 실행되며, 생성 시 필요한 코드를 포함할 수 있습니다. 생성자의 이름은 `__init__()`입니다.

소멸자는 객체가 소멸될 때 자동으로 실행되며, 소멸 시 필요한 코드를 포함할 수 있습니다. 소멸자의 이름은 `__del__()`입니다. 객체는 인스턴스 객체의 레퍼런스 카운트가 0이 될 때 소멸됩니다.



생성자

- 생성자

- 생성자(Constructor)는 객체가 생성될 때 자동으로 실행
- 생성 시 필요한 코드를 포함할 수 있음
- 생성자의 이름은 `__init__()`

생성자를 갖는 클래스

```
1 class Person:
2     'Person은 이름과 성별 멤버변수가 있는 객체타입입니다'
3     def __init__(self, name='홍길동', gender='남자'):
4         print('{} , {} Person 객체를 생성시킵니다'.format(name, gender))
5         self.name = name
6         self.gender = gender
7     def print_info(self):
8         print('{}님은 {}입니다'.format(self.name, self.gender))
```

[36]:

```
1 p1 = Person('유관순', '여자')
```

유관순, 여자 Person 객체를 생성시킵니다

[43]:

```
1 print(p1.name , p1.gender)
2 p1.print_info()
```

홍길동 남자
홍길동님은 남자입니다

[44]:

```
1 p2 = Person()
```

홍길동, 남자 Person 객체를 생성시킵니다

[45]:

```
1 print(p2.name , p2.gender)
```

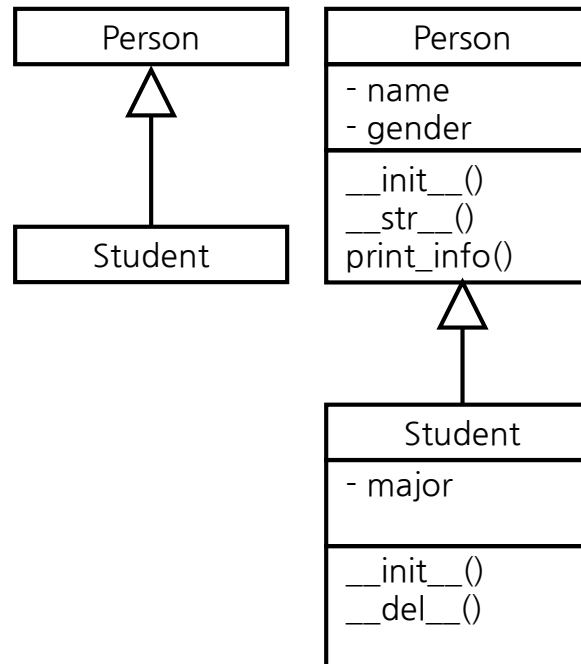
홍길동 남자

2절. 상속과 재정의

- 상속(Inheritance)은 객체 재사용의 한 방법
- 상속을 이용하면 부모 클래스의 모든 속성들을 자식 클래스로 물려줄 수 있다

```
class SubClassName(SuperClassName) :
```

Student is a Person



상속

```

1  ▼ class Person:
2  ▼     def __init__(self, name, gender):
3         self.name = name
4         self.gender = gender
5
6  ▼     def __str__(self):
7         return "name: {0}, gender: {1}".format(self.name, self.gender)
8
9  ▼     def print_info(self):
10        print("{}님은 {}입니다.".format(self.name, self.gender))
11

```

```

1  ▼ class Student(Person):
2  ▼     def __init__(self, name, gender, major):
3         self.name = name
4         self.gender = gender
5         self.major = major
6
7  ▼     def __del__(self):
8         pass

```

```

1  issubclass(Student, Person)

```

True

super()

- super()는 부모 클래스의 멤버를 참조

```

1  ▼ class Student(Person):
2  ▼     def __init__(self, name, gender, major):
3         Person.__init__(self, name, gender)
4         self.major = major
5
6  ▼     def __del__(self):
7         pass
8
9  ▼     def __str__(self):
10         return super().__str__() + ", major:{}".format(self.major)
11
12 ▼     def print_info(self):
13 ▼         print("{}님은 {}이며, 전공은 {}입니다."
14             .format(self.name, self.gender, self.major))
15

```

__str__() 메서드에서
super().__str__()을 통해 Person의
__str__() 메서드를 호출 가능

```
1 s3 = Student("홍길북", "남자", "심리학")
```

```
1 print(s3)
```

name: 홍길북, gender: 남자, major:심리학

3절. 연습문제

1. 다음 조건을 만족하는 클래스를 작성하세요

- ✓ 도형(Shape)클래스와 삼각형(Triangle) 클래스를 만들어야 합니다.
- ✓ 도형(Shape)클래스
 - 생성자를 통해 width, height를 초기화할 수 있습니다. width, height의 기본값은 0, 0입니다.
 - width, height를 조절할 수 있는 modify 함수가 있습니다.
 - width, height 값 정보를 문자열로 리턴하는 __str__ 함수가 있습니다
 - 도형의 면적을 구하는 함수(calc_area)가 있습니다.
- ✓ '삼각형(Triangle)클래스는 도형(Shape)클래스이다'관계가 성립해야 합니다
 - ✓ 생성자를 통해, width, height, x, y좌표를 초기화하고 x, y 좌표의 기본값은 0, 0입니다
 - ✓ 삼각형의 면적을 구하는 메소드 calc_area를 재정의합니다
 - ✓ 삼각형의 정보를 문자열로 반환하는 __str__() 함수가 있습니다.