**윤성우의 열혈 파이썬: 기초 편**

Chapter 13. '클래스와 객체'에 대한 보충 자료

본 자료를 배포하는 이유

파이썬 '기초편'에서 어루만지고 '중급편'에서 완성하고자 했던 내용이 있습니다. 그런데 중급편까지 공부하지 않으시는 분들에게는 완성되지 않은 지식으로 남을까 걱정스러워서 추가 설명을 담은 이 자료를 배포해 드립니다. 참고로 본 내용은 13장을 학습하신 이후에 보실 것을 권해드립니다. 그렇지 않으면 오히려 많이 헷갈릴 수도 있습니다.

13장에서는 그 제목처럼 클래스와 객체에 대해 설명을 합니다. 그리고 전통적인 객체지향 이론 관점에서 ‘클래스’와 ‘객체’는 서로 구분이 됩니다. 혹 C++, Java와 같은 객체지향 언어를 이미 알고 계신분들은 당연한 것 아니야? 라고 말씀하실 겁니다. 즉 다음과 같은 설명에 전혀 문제가 없습니다.

클래스 데이터와 기능(메소드)를 기반으로 만들어진 일종의 '틀'

객체 클래스라는 틀을 기반으로 만들어 낸 메모리에 존재하는 실제 대상

그런데 말이죠. 파이썬은 좀 많이 다릅니다. 파이썬은 클래스도 객체이기 때문입니다.

"파이썬은 클래스도 객체이다."

물론 위의 내용은 ‘기초편’에서 언급하지 않습니다. 위의 설명까지 보태서 풀다 보면 내용이 엄청 길어지고 어렵기 때문이지요. 특히 이런 내용은 첫 객체지향 언어로 파이썬을 공부하는 분들께 혼란을 줄 수 있습니다. 그래요. 파이썬은 쉬운 언어가 맞습니다. 그런데요 조금 정확히 그리고 구체적으로 알려고 들어가다 보면 의외로 복잡한 부분이 제법 있습니다. 그래서 저는 다음과 같은 접근 순서를 구성했습니다.

기초편에서의 설명 ⇒ 객체 생성시 '데이터'와 '기능'이 함께 채워져서 만들어진다.

중급편에서의 설명 ⇒ 사실은 객체 속 데이터는 나중에 채워진다.

기초편에서의 설명은 전통적인 객체지향 관점에서의 설명입니다. 그리고 우선은 이렇게 이해하는 것이 중요합니다. 그래서 13장에서는 다음 예제를 보면서 객체 속에 이미 변수가 존재하는 것으로 설명합니다. 잠시 다음 예제를 보시죠.

# class\_object.py

class AgeInfo: # 클래스 AgeInfo의 정의

def up\_age(self):

self.age += 1

def get\_age(self):

return self.age

def main():

fa = AgeInfo() # AgeInfo의 객체를 생성하고 이를 변수 fa에 저장

fa.age = 39 # fa에 저장된 객체의 변수 age에 39를 저장

. . . .

위에서 다음 문장 실행 시 객체 생성이 완료됩니다.

fa = AgeInfo()

그리고 클래스를 보면 변수 age가 필요하다는 것이 보이지요? 그래서 객체 안에 age도 있는 것처럼 설명합니다. 하지만 실제 객체 속에 age가 채워지는 시점은 main 함수에서 다음 문장을 실행할 때입니다.

fa.age = 39 # 실제 객체 속에 age가 저장되는 시점

모든 변수는 초기화를 해야 합니다. 그리고 파이썬은 바로 그 초기화 시점에 실제 객체에 해당 변수를 넣어줍니다. 즉 파이썬의 객체는 다른 언어의 객체에 비해 매우 유연합니다. 그리고 이러한 유연함이 많은 장점을 주지만 처음 공부하는 분들에게는 혼란을 주기도 하지요. 정리하면 이렇습니다.

“파이썬의 객체 속에 변수가 생성되는 시점은 첫 대입연산을 진행하는 시점이다.”

그래서 \_\_init\_\_ 메소드가 있는 경우에는 그 안에서 변수를 초기화할 때 해당 변수가 만들어 집니다. 다음 예를 보시죠.

# ctor2.py

class Const:

def \_\_init\_\_(self, n1, n2):

self.n1 = n1 # self.n1은 인스턴스 변수, n1은 매개변수

self.n2 = n2 # self.n2는 인스턴스 변수, n2는 매개변수

def show\_data(self):

print(self.n1, self.n2)

def main():

o1 = Const(1, 2) # 생성자에 1과 2를 전달

. . . .

위 예제를 보면 \_\_init\_\_ 메소드가 다음과 같이 정의되어 있습니다.

self.n1 = n1 # 이 문장 실행되면서 객체에 n1이 생긴다.

self.n2 = n2 # 이 문장 실행되면서 객체에 n2가 생긴다.

즉 n1, n2의 첫 대입 연산이 \_\_init\_\_ 메소드 내에서 이뤄집니다. 때문에 모든 객체 내에는 n1과 n2가 존재하게 되겠지요. \_\_init\_\_ 메소드는 객체 생성시에 자동으로 호출이 되니까요. 어쨌든 바로 이 \_\_init\_\_ 메소드 내에서 객체 내에 필요한 모든 변수를 적절히 초기화한다는 가정하에서 객체 생성 시 객체 내에 데이터도 함께 생성된다고 봐도 괜찮겠지요? 일단 보충 설명은 여기까지 하겠습니다.

지금부터는 그냥 서비스로 조금 더 언급합니다.

일반적으로 클래스는 메모리 공간을 차지하거나 실제 접근이 가능한 대상이 아니라고 설명합니다. 실체가 아니라는 표현을 쓰기도 하지요. 그런데요 파이썬에서는 좀 얘기가 다릅니다. 클래스와 객체 모두 상당히 유연합니다. 클래스라는 틀에서 구성하지 않은 변수를(데이터를) 객체에 뒤 늦게 추가할 수도 있고요. 클래스도 정의 이해에 변수를 추가할 수 있습니다. 중급편에서는 이런 내용도 설명합니다. 하지만 기초편에서는 거기까지 신경 쓰지 않아도 됩니다.

파이썬 저자 윤성우 드림.