**به نام خدا**

**زهره پیروز**

تعریف کلاس در پایتون

فرض کنید می‌خواهیم کلاسی به نام Person ایجاد کنیم. این کلاس نشان‌دهنده موجودیت افراد در سیستم ما است.

ساختار کلی تعریف کلاس در پایتون به صورت زیر است. ابتدا کلمه کلیدی class را نوشته و سپس نام کلاس را با حرف بزرگ می‌نویسیم و با : آغاز محدوده کلاس را تعیین می‌کنیم.

مشابه هر ساختار دیگری در پایتون، دندانه گذاری (یا فرورفتگی) کدها در ساختار کلاس دارای اهمیت است.

class Person:

# class body here...

برای تعریف توضیحات کلاس، توضیحات توصیف‌کننده را در خط اول بدنه به شکل یک رشته متنی می‌نویسیم؛ مثل کلاس زیر:

class Person:

"represent person identity"

# class body here ....

ویژگی کلاس در پایتون

ویژگی‌های یک کلاس (property) وضعیت و مقادیری مرتبط با نمونه‌های ایجاد شده از آن کلاس را نگهداری می‌کند. در حقیقت این ویژگی‌ها متغیرهایی هستند که اطلاعات مورد نیاز را در آن‌ها قرار می‌دهیم.

برای اینکه یک ویژگی کلاس در پایتون را از همان ابتدا تعریف کنیم، مشابه تعریف متغیر پایتون عمل می‌کنیم. به این صورت که در بدنه کلاس، نام متغیر (ویژگی) را نوشته و مقداردهی می‌کنیم.

class Person:

name = None

nationality = "iranian"

score = 100

# some codes

البته گاهی اوقات ممکن است بنا به نیازمان، متغیری را در یکی از متدهای کلاس ایجاد کنیم. با کلمه کلیدی self که در ادامه یاد می‌گیریم می‌توان چنین کاری انجام داد.

تعریف متدهای کلاس

متدهای کلاس در پایتون همان رفتارهای یک شی از کلاس است. برای تعریف متدها دقیقاً مشابه تعریف تابع در پایتون عمل می‌کنیم. یعنی پس از کلمه کلیدی def نام متد مورد نظر را نوشته و در صورت نیاز، پارامترهای ورودی آن را مشخص می‌کنیم.

در قطع کد زیر، من یک متد به نام say\_hello() تعریف کردم که «سلام» در خروجی چاپ می‌کند.

class Person:

name = None

score = 100

def say\_hello(self):

print( "Hello!" )

دقت کنید که برای تعریف متدها در کلاس‌های پایتون، می‌بایست حداقل یک آرگومان ورودی که معمولاً به نام self تعریف می‌شود برای آن در نظر گرفت.

متغیر self در پایتون

برای اینکه در متدهای کلاس به ویژگی‌های همان شئ دسترسی داشته باشیم، از self استفاده می‌کنیم. این مقدار به همان شئ ما اشاره کرده و می‌توانیم مشابه یک شئ با آن رفتار کنیم.

در قطعه کلاس زیر، متد set\_name() را تعریف کرده‌ایم. این متد یک ورودی از کاربر می‌گیرد و ویژگی name شئ را تغییر می‌دهد.

class Person:

name = None

score = 100

def set\_name(self, name):

self.name = name

در بدنه متدهای کلاس پایتون می‌توانیم پردازش‌های مورد نظرمان را انجام دهیم. اجازه دهید کلاس بالا را کمی تغییر دهم. می‌خواهم وقتی متد set\_name() صدا زده شد، اگر نام کاربر None بود، تغییر نام انجام شود، در غیر اینصورت خطایی در خروجی چاپ کنم. (شرط در پایتون)

class Person:

name = None

score = 100

def set\_name(self, name):

if name == None:

self.name = name

else:

print("You Cann't Change name Attribute!")

**مثال خودرو در کلاس پایتون**

بیایید یک کلاس به نام Car ایجاد کنیم. هر کدی که در زیر کلاس قرار گرفته باشد، بخشی از بدنه کلاس محسوب می شود:

class Car:

pass

توجه: نام کلاس های پایتون باید به شکل CapitalizedWords نوشته شده است. به عنوان مثال، MySedanCar

در کد بالا یک کلاس به نام Car ایجاد کردیم. این کلاس مشخص میکند که ما حالت کلی اتومبیل را داریم. میتوانیم به این کلاس ویژگی‌ها و رفتارهای مشخصی را اضافه کنیم. کلاس Car ما فعلا خالی است و چیزی درون آن وجود ندارد.

**# ایجاد آبجکت از یک کلاس در پایتون**

یک آبجکت یا شی نمونه ای از یک کلاس است. ما می توانیم از کلاس Car که در بالا تعریف شده استفاده کرده و از آن برای ایجاد یک شی یا نمونه استفاده کنیم. برای ساخت یک آبجکت از کلاس کافیست کلاس مورد نظر را در یک متغیر ذخیره کنید. به ساخت آبجکت از یک کلاس instantiating میگویند. نام آبجکت هر چیزی میتواند باشد اما سعی کنید نامی معنی دار انتخاب کنید. آبجکت‌ها میتوانند به تمام اطلاعاتی که در کلاس وجود دارند دسترسی کامل داشته باشند.

**ما یک شیء Car به نام benz خواهیم ساخت:**

benz = Car()

benz.price = 800

print(benz.price)

آبجکت benz که از کلاس Car ایجاد کردیم، میتواند به تمام اطلاعاتی که در کلاس Car وجود دارد،‌ دسترسی پیدا کرده و حتی تغییر دهد. همانطور که در کد بالا میبینید، توانستیم با روش نقطه گذاری(dot notation) مقدار price را در آبجکت benz برابر با 800 قرار دادیم. در آخر توانستیم به مقدار price از آبجکت benz دسترسی پیدا کنیم.

**# ساخت متد در شی گرایی پایتون**

متد به فانکشن‌هایی گفته میشود که درون یک کلاس قرار دارند. هر متد وظیفه انجام کار مشخصی را دارد. متدها رفتارهای یک کلاس هستند که آبجکت‌ها میتوانند از آنها برای کار روی داده‌های خود استفاده کنند. برای فراخوانی متدها باید از آبجکتی که از کلاس ساخته اید استفاده کنید. در کد پایین یک متد به نام drive در کلاس Car ایجاد کرده‌ایم.

class Car:

def drive(self):

print('i drive a car')

a = Car()

a.drive()

# OUTPUT

i drive a car

در کد بالا بعد از ساختن متد drive، یک آبجکت از کلاس Car به نام a ایجاد کرده‌ایم. سپس توانستیم با آبجکت a به متد drive دسترسی داشته باشیم.

پایان