مفاهیم اولیه شی گرایی در پایتون

مفاهیم اولیه کلاس ، شی ، صفت و رفتار در شی گرایی :

برنامه‌نویسی شی‌گرا (Object-Oriented Programming) یا به اختصار OOP یک الگو یا شیوه تفکر در برنامه‌نویسی است که برگرفته از دنیای واقعی بوده و از دهه ۱۹۶۰ میلادی مطرح گشته است. به زبانی که از این الگو پشتیبانی کند، «زبان شی‌گرا» گفته می‌شود.

برنامه‌نویسی شی‌گرا در قالب دو مفهوم کلاس (Class) و شی (Object) ارایه می‌گردد. هر کلاس واحدی از برنامه است که تعدادی داده و عملیات را در خود نگه‌داری می‌کند و هر شی نیز حالتی (State) مشخص از یک کلاس می‌باشد.

قصه از آنجایی شروع شد که برنامه نویسان نیاز به اشیا زیادی با صفات مشترک پیدا کردند، برای مثال دانش آموزان یک مدرسه.
به جای اینکه دانش آموزان به ترتیب هر کدام در چندین خط تعریف شوند ، مفهومی به اسم کلاس به وجود آمد تا تمام خصوصیات و رفتارهای دانش آموزان را در خود داشته باشد ، و اگر نیاز به تعریف دانش آموز جدیدی بود فقط کافی بود یک شی از آن کلاس ساخته شود.

در برنامه‌نویسی شی‌گرا، هر برنامه در قالب موجودیت‌های کوچکی که در واقع همان اشیا هستند و با یکدیگر تعامل دارند در نظر گرفته می‌شود. برای داشتن این اشیا می‌بایست ابتدا کلاس‌های برنامه را تعریف نماییم؛ هر کلاس «رفتار» (Behavior) و «صفات» (Attributes) اشیایی که قرار است از آن ایجاد شوند را تعریف می‌کند. از یک کلاس می‌توان هر تعداد که بخواهیم شی ایجاد نماییم. هر شی بیانگر یک «حالت» یا یک «نمونه» (Instance) از کلاس خود است.
به هر شی کلاس، یک نمونه از آن کلاس گفته می‌شود و هر زمان که یک شی از کلاسی ایجاد می‌گردد در واقع یک نمونه از آن ساخته می‌شود. به این عمل در شی‌گرایی «نمونه‌سازی» (Instantiation) گفته می‌شود.

اعضای داده در واقع همان متغیرهای درون کلاس هستند که خصوصیات یا صفات شی را بیان می‌کنند و در شی‌گرایی با عنوان «فیلد» (Field) یا «صفت» (Attribute) از آن‌ها یاد می‌شود. توابع عضو نیز عملیات یا کارهایی هستند که یک شی از کلاس قادر به انجام آن‌ها می‌باشد؛ می‌توان توابع عضو را بیانگر رفتار اشیا کلاس دانست. در شی‌گرایی به این توابع «متد» (Method) گفته می‌شود.

متدهای اینترنال و اکسترنال :

اگر در برنامه نویسی ، یک متد از بدنه اصلی خود کلاس باشد و بصورت نام تابع فقط صدا زده شود به این متد ، متد اینترنال یا داخلی میگوییم ، اما اگر این تابع برای صدا زدن ، نیاز به object خارجی داشته باشد ، آنرا متد اکسترنال یا خارجی میگوییم.

کلاس دیاگرام :

در رسم صفت ها و همچنین رفتارهای یک کلاس از یک نمایش کاربردی به نام کلاس دیاگرام استفاده میکنیم که در این قسمت بصورت مختصر فقط خواهیم آموخت که در مستطیل بالا ، نام کلاس قرار میگیرد ، سپس در مستطیل وسط صفت های کلاس نوشته میشوند و در مستطیل پایین ، رفتارهای کلاس نوشته میشوند .
جلوتر روابط بین کلاس دیاگرام ها هم رسم خواهیم کرد

مقدار None :

در برنامه نویسی پایتون ، مانند دیگر زبان های برنامه نویسی یک مقدار برای خالی نشان دادن تعریف شده است.
از None برای خالی نشان دادن یک متغیر استفاده میشود.

```
x = None
```

شی گرایی در کد :

- تعریف کلاس :

برای تعریف کلاس از class استفاده میکنیم و روبروی آن نام کلاس را وارد میکنیم.

```
class Student :
```

تعاریف مربوط به کلاس در پایین این خط کد نوشته میشوند با این شرط که حتما باید به اندازه یک tab جلوتر باشند.

با ایجاد هر نمونه از کلاس یک متد خاص در آن به صورت خودکار اجرا می‌گردد. این متد «سازنده» (Constructor) نام دارد و کار آن «مقداردهی اولیه» (Initialization) شی است. این کار موجب اطمینان از مقداردهی تمامی اعضای داده پیش از استفاده شی در برنامه می‌گردد.

- تعریف Constructor :

برای تعریف کانستراکتور از دستور __init__ استفاده میکنیم.
یکی از متغیرهای مهم که حتما حتما باید اولین ورودی تابع init قرار بگیرد ، self است. متغیر self به این معناست که هر object از این کلاس میتواند جای آن بنشیند و مقداردهی شود.
در مثال زیر ، مقادیر اسم ، فامیل ، کدملی و سن به عنوان ورودی های حتمی کانستراکتور تعریف شده اند که در ادامه ی کد برای هر کدام از شی ها که قرار است به جای self بنشینند ، مقداردهی شده اند.

```
def __init__(self, name, family, idNumber, age):
        self.name = name
        self.family = family
        self.idNumber = idNumber
        self.age = age
```

- تعریف یک متد دلخواه :

در مثال زیر یک متد تعریف شده است که ورودی آن یک شی است. در بعضی از نرم افزارها مانند pycharm بصورت پیشفرض همیشه مقدار self به عنوان ورودی اول در متدهای کلاس قرار میگیرد.
این تابع وظیفه چاپ کردن نام و نام خانوادگی شی مورد نظر در کنار همدیگر را دارد.

```
def fullname(self):
        print(self.name, self.family)
```

- ساختن object و استفاده از آن :

برای ساخت شی به طریقه زیر عمل میکنیم و میبایست هنگام تعریف ، ورودی های مربوط به کانستراکتور را هم وارد کنیم.
در انتها با نوشتن نام شی و سپس بعد از یک نقطه، نوشتن رفتار (متد) و یا صفت موردنظر میتوانیم به آن دسترسی داشته باشیم:

```
s1 = Student("Keykavoos", "Bouazzar", 123, 17)
s2 = Student("ali", "hossini", 122, 15)

s1.fullname()
s2.fullname()
```