

تعریف کلاس‌ها و ساختار آنها در پایتون

در پایتون، کلاس‌ها (Classes) ساختاری برای ایجاد اشیاء هستند که ویژگی‌ها (Attributes) و متدها (Methods) را در خود جای می‌دهند. در این بخش، با نحوه تعریف کلاس، ویژگی‌ها و متدها، دسترسی به ویژگی‌ها و متدها با شیء و چندین مثال عملی آشنا می‌شویم.

۱. نحوه تعریف یک کلاس با استفاده از کلمه‌کلیدی `class`

در پایتون، برای تعریف یک کلاس از کلمه‌کلیدی `class` استفاده می‌کنیم.

ساختار کلی یک کلاس

```
class ClassName:
    def __init__(self, attribute1, attribute2):
        self.attribute1 = attribute1 # ویژگی اول
        self.attribute2 = attribute2 # ویژگی دوم

    def method1(self):
        print("این یک متد است")

    def method2(self, value):
        self.attribute1 = value # تغییر مقدار ویژگی اول
```

نکات مهم:

- کلاس با `class ClassName`: تعریف می‌شود.
- `__init__` یک سازنده (Constructor) است که ویژگی‌های کلاس را مقداردهی می‌کند.
- ویژگی‌ها در داخل `self.attribute` ذخیره می‌شوند.
- متدها (توابع داخل کلاس) می‌توانند ویژگی‌های کلاس را بخوانند یا تغییر دهند.

۲. تعریف ویژگی‌ها (Attributes) و متدها (Methods) در کلاس‌ها

ویژگی‌ها (Attributes) و متدها (Methods) دو جزء اصلی کلاس‌ها هستند.

◆ ویژگی‌ها (Attributes) چیستند؟

ویژگی‌ها، متغیرهای داخلی کلاس هستند که اطلاعات مربوط به شیء را ذخیره می‌کنند.

◆ متدها (Methods) چیستند؟

متدها، توابع داخل کلاس هستند که عملیات خاصی را انجام می‌دهند و می‌توانند ویژگی‌های شیء را تغییر دهند یا مقداردهی کنند.

مثال: تعریف کلاس با ویژگی‌ها و متدها

```
class Car:
    def __init__(self, brand, color, speed):
        self.brand = brand # ویژگی برند خودرو
        self.color = color # ویژگی رنگ خودرو
        self.speed = speed # ویژگی سرعت خودرو

    def drive(self):
        print(f"{self.brand} با سرعت {self.speed} کیلومتر بر ساعت حرکت می‌کند.")

    def change_color(self, new_color):
        self.color = new_color
        print(f"رنگ جدید {self.brand}: {self.color}")

# ایجاد یک شیء از کلاس Car
car1 = Car("BMW", "220", "مشکی")
car1.drive()

# تغییر رنگ خودرو
car1.change_color("قرمز")
```

خروجی:

با سرعت 220 کیلومتر بر ساعت حرکت می‌کند BMW.
قرمز BMW: رنگ جدید

نکات مهم:

- در متد `drive`، اطلاعات خودرو را نمایش می‌دهیم.
- در متد `change_color`، رنگ خودرو را تغییر می‌دهیم.

۳. تفاوت بین ویژگی‌ها (Attributes) و متدها (Methods)

ویژگی‌ها (Attributes)	متدها (Methods)
مشخصات و داده‌های شیء را ذخیره می‌کنند.	عملکرد و رفتار شیء را تعریف می‌کنند.
معمولاً در <code>__init__</code> تعریف می‌شوند.	معمولاً توابع داخل کلاس هستند.
مقدار آنها می‌تواند تغییر کند.	می‌توانند ویژگی‌ها را تغییر دهند یا عملیات خاصی انجام دهند.
مثال: <code>self.color = "مشکی"</code>	مثال: <code>def change_color(self, new_color):</code>

۴. دسترسی به ویژگی‌ها و متدها با استفاده از شیء

پس از ایجاد یک شیء از کلاس، می‌توان به ویژگی‌ها و متدهای آن دسترسی داشت.

مثال: خواندن و تغییر ویژگی‌ها

```
class Person:
    def __init__(self, name, age):
        self.name = name # ویژگی نام
        self.age = age # ویژگی سن

    def greet(self):
        print(f"سال دارم {self.age} هستم و {self.name} سلام، من")

# ایجاد یک شیء از کلاس Person
person1 = Person("علی", 25)

# دسترسی به ویژگی‌ها
print(person1.name) # خروجی: علی
print(person1.age) # خروجی: 25

# استفاده از متدها
person1.greet()

# تغییر مقدار ویژگی
person1.age = 26
print(f"خروجی: 26 # {person1.age}: سن جدید")
```

نکات:

- مقدار ویژگی‌ها را می‌توان مستقیماً تغییر داد: `person1.age = 26`
- برای استفاده از متدها، نام شیء را همراه با متد فراخوانی می‌کنیم: `person1.greet()`

۵. مثال‌های کاربردی از ایجاد کلاس‌ها و اشیاء

مثال: کلاس `BankAccount` برای مدیریت حساب بانکی

```
class BankAccount:
    def __init__(self, owner, balance=0):
        self.owner = owner # نام صاحب حساب
        self.balance = balance # موجودی حساب

    def deposit(self, amount):
        self.balance += amount
        print(f"تومان {self.balance}: تومان به حساب اضافه شد. موجودی جدید {amount}")

    def withdraw(self, amount):
        if amount > self.balance:
            print("موجودی کافی نیست")
        else:
            self.balance -= amount
            print(f"تومان {self.balance}: تومان برداشت شد. موجودی جدید {amount}")

# ایجاد حساب جدید
account1 = BankAccount("محمد", 1000)
```

واریز پول

```
account1.deposit(500)
```

برداشت پول

```
account1.withdraw(1200)
```

تلاش برای برداشت بیش از موجودی

```
account1.withdraw(500)
```

خروجی:

تومان به حساب اضافه شد. موجودی جدید: 1500 تومان

تومان برداشت شد. موجودی جدید: 300 تومان

!موجودی کافی نیست

نکات:

- از `self.balance += amount` برای اضافه کردن پول به حساب استفاده کردیم.
- قبل از برداشت، مقدار موجودی بررسی می‌شود.

جمع‌بندی نهایی

برای تعریف کلاس، از `class` استفاده می‌کنیم.

ویژگی‌ها (Attributes) مشخصات شیء را ذخیره می‌کنند.

متدها (Methods) رفتارهای شیء را مشخص می‌کنند.

از `self` برای دسترسی به ویژگی‌های شیء در داخل کلاس استفاده می‌شود.

می‌توان ویژگی‌های شیء را خواند، تغییر داد و با متدها آن‌ها را مدیریت کرد.

تمرین برای شما:

یک کلاس `Book` تعریف کنید که دارای عنوان، نویسنده و تعداد صفحات باشد. متدی برای نمایش اطلاعات کتاب و متدی برای اضافه کردن صفحات جدید به آن بنویسید.