

ویژگی‌ها و متدهای کلاس‌ها در پایتون

در پایتون، ویژگی‌ها (Attributes) و متدها (Methods) به دو دسته کلی تقسیم می‌شوند:

1. ویژگی‌های کلاس (Class Attributes) و ویژگی‌های نمونه (Instance Attributes)

2. متدهای کلاس (Class Methods) و متدهای نمونه (Instance Methods)

در این بخش، تفاوت این موارد را توضیح می‌دهیم و با مثال‌های عملی نحوه استفاده از آن‌ها را بررسی می‌کنیم.

۱. ویژگی‌های کلاس (Class Attributes) و ویژگی‌های نمونه (Instance Attributes)

ویژگی‌های نمونه (Instance Attributes) چیست؟

ویژگی‌های نمونه به هر شیء به صورت مستقل اختصاص داده می‌شوند و مقدار آن‌ها می‌تواند برای هر شیء متفاوت باشد.

ویژگی‌های کلاس (Class Attributes) چیست؟

ویژگی‌های کلاس متعلق به خود کلاس هستند و برای تمام اشیاء یکسان باقی می‌مانند.

مثال: تفاوت ویژگی‌های کلاس و نمونه

```
class Car:
    wheels = 4 # (تمام خودروها ۴ چرخ دارند)

    def __init__(self, brand, color):
        self.brand = brand # ویژگی نمونه (هر خودرو برند متفاوتی دارد)
        self.color = color # ویژگی نمونه (هر خودرو رنگ متفاوتی دارد)

# ایجاد دو شیء از کلاس Car
car1 = Car("BMW", "قرمز")
car2 = Car("Toyota", "آبی")

# دسترسی به ویژگی‌های نمونه
print(car1.brand) # خروجی: BMW
print(car2.brand) # خروجی: Toyota

# دسترسی به ویژگی کلاس
print(car1.wheels) # خروجی: 4
print(car2.wheels) # خروجی: 4

# تغییر مقدار ویژگی نمونه (تأثیر روی یک شیء خاص)
car1.color = "مشکی"
print(car1.color) # خروجی: مشکی
print(car2.color) # خروجی: آبی

# تغییر مقدار ویژگی کلاس (تأثیر روی کل کلاس)
Car.wheels = 6
print(car1.wheels) # خروجی: 6
```

خروجی: # 6 `print(car2.wheels)`

نکات:

- ویژگی `wheels` یک ویژگی کلاس است، که مقدار آن برای تمام اشیاء یکسان است.
- ویژگی‌های `brand` و `color` ویژگی‌های نمونه هستند و مقدار آن‌ها برای هر شیء متفاوت است.

۲. متدهای کلاس (Class Methods) و متدهای نمونه (Instance Methods)

متدهای نمونه (Instance Methods) چیست؟

متدهای نمونه با `self` کار می‌کنند و به ویژگی‌های هر شیء به صورت جداگانه دسترسی دارند.

متدهای کلاس (Class Methods) چیست؟

متدهای کلاس با `@classmethod` مشخص می‌شوند و به جای `self` از `cls` استفاده می‌کنند. این متدها روی کل کلاس اعمال می‌شوند.

مثال: متد نمونه و متد کلاس

```
class Person:
    species = "انسان" # ویژگی کلاس

    def __init__(self, name, age):
        self.name = name # ویژگی نمونه
        self.age = age # ویژگی نمونه

    def introduce(self): # متد نمونه
        print(f"سال دارم {self.age} هستم و {self.name} من")

    @classmethod
    def change_species(cls, new_species): # متد کلاس
        cls.species = new_species

# ایجاد اشیاء
person1 = Person("علی", 30)
person2 = Person("زهرا", 25)

# استفاده از متد نمونه
person1.introduce() # خروجی: من علی هستم و 30 سال دارم

# تغییر ویژگی کلاس با متد کلاس
Person.change_species("ربات")

# بررسی مقدار ویژگی کلاس برای تمام اشیاء
print(person1.species) # خروجی: ربات
print(person2.species) # خروجی: ربات
```

نکات:

- متد `introduce` یک متد نمونه است که اطلاعات هر شیء را جداگانه نمایش می‌دهد.
- متد `change_species` یک متد کلاس است که مقدار ویژگی کلاس را برای تمام اشیاء تغییر می‌دهد.

۳. استفاده از `self` برای دسترسی به ویژگی‌ها و متدهای نمونه

`self` چیست؟

- `self` به شیء جاری کلاس اشاره دارد.
- از `self` برای دسترسی به ویژگی‌های نمونه و متدهای نمونه استفاده می‌شود.

مثال: استفاده از `self` در متدهای نمونه

```
class Animal:
    def __init__(self, name, species):
        self.name = name
        self.species = species

    def speak(self):
        print(f"{self.name} از گونه {self.species} صدا تولید می‌کند")

# ایجاد شیء
dog = Animal("سگ", "توتو")

# استفاده از متد نمونه
dog.speak() # خروجی: توتو از گونه سگ صدا تولید می‌کند
```

نکات:

- `self.name` و `self.species` مربوط به همان شیء خاص هستند.
- در داخل متد `speak`، از `self.name` و `self.species` برای دسترسی به اطلاعات شیء استفاده می‌شود.

۴. متدهای استاتیک (Static Methods) در پایتون

متد استاتیک چیست؟

- متد استاتیک به کلاس یا نمونه وابسته نیست.
- برای تعریف آن از `@staticmethod` استفاده می‌شود.
- نیازی به `self` یا `cls` ندارد و مانند یک تابع عادی عمل می‌کند.

مثال: متد استاتیک

```
class MathOperations:
```

```
    @staticmethod
```

```
    def add(a, b):
```

```
        return a + b
```

فراخوانی متد استاتیک بدون نیاز به ایجاد شیء

خروجی: 8 # print(MathOperations.add(3, 5))

نکات:

- متد استاتیک به ویژگی‌های کلاس یا نمونه دسترسی ندارد.
- از متد استاتیک برای انجام عملیاتی که مستقل از ویژگی‌های کلاس هستند، استفاده می‌شود.

جمع‌بندی نهایی

نوع	دسترسی به ویژگی‌های نمونه	دسترسی به ویژگی‌های کلاس	پارامتر
متد نمونه (Instance Method)	دارد ✓	ندارد ✗	self
متد کلاس (Class Method)	ندارد ✗	دارد ✓	cls
متد استاتیک (Static Method)	ندارد ✗	ندارد ✗	بدون self و cls

ویژگی‌های نمونه مربوط به هر شیء جداگانه است.
ویژگی‌های کلاس برای همه اشیاء مشترک است.
متدهای نمونه روی یک شیء خاص عمل می‌کنند.
متدهای کلاس می‌توانند ویژگی‌های کلاس را تغییر دهند.
متدهای استاتیک برای انجام عملیات عمومی بدون وابستگی به کلاس یا شیء استفاده می‌شوند.

تمرین برای شما:

یک کلاس `Student` تعریف کنید که ویژگی‌های نام، سن و معدل داشته باشد.
متد `introduce` برای معرفی دانش‌آموز ایجاد کنید.
متد کلاس `set_passing_grade` برای تنظیم حداقل نمره قبولی ایجاد کنید.
متد استاتیک `is_adult` که بررسی کند آیا دانش‌آموز بالای ۱۸ سال است یا نه.