شکل کلی ساختار :

```python
class ClassName:
    # Class attribute (optional)
    class_attribute = "some value"

    # Constructor to initialize instance attributes
    def __init__(self, param1, param2):
        self.instance_attribute1 = param1
        self.instance_attribute2 = param2

    # Instance method
    def method1(self):
        # Method implementation
        print(f"Instance Attribute 1: {self.instance_attribute1}")

    # Another instance method
    def method2(self, param):
        # Method implementation
        print(f"Instance Attribute 2: {self.instance_attribute2}, Param: {param}")

    # Special method for string representation
    def __str__(self):
        return f"ClassName(instance_attribute1={self.instance_attribute1},
instance_attribute2={self.instance_attribute2})"

    # Special method for object deletion (optional)
    def __del__(self):
        print(
            f"ClassName object with instance_attribute1={self.instance_attribute1} is
being deleted"
        )

    # Property for a read-only attribute
    @property
    def read_only_property(self):
        return self.instance_attribute1

    # Property with a setter for a read-write attribute
    @property
    def read_write_property(self):
        return self.instance_attribute2

    @read_write_property.setter
    def read_write_property(self, value):
        self.instance_attribute2 = value
```

قسمت ۱ : سازنده | `__init__`

```
    # Constructor to initialize instance attributes
    def __init__(self, param1, param2):
        self.instance_attribute1 = param1
        self.instance_attribute2 = param2
```

* برای تائین ویژگی شئی ، او را با self تائین ویژگی میکنیم و سپس پارامتر ها مقدار را تائین می کنیم . *

قسمت ۲ : متد نمونه | `method1` و `method2`

```
    # Instance method
    def method1(self):
        # Method implementation
        print(f"Instance Attribute 1: {self.instance_attribute1}")

    # Another instance method
    def method2(self, param):
        # Method implementation
        print(f"Instance Attribute 2: {self.instance_attribute2}, Param: {param}")
```

* متدهای نمونه داخل کلاس تعریف می‌شوند و روی ویژگی‌های نمونه عمل می‌کنند و مانند تابع برای کلاس عمل می کند .*

قسمت ۳ : متد ویژه | `__str__` و `__len__` و `__del__`

```
    # Special method to convert object to string (for display)
    def __str__(self):
        return f"Person(name={self._name}, age={self._age})"

    # Special method to determine the length (e.g., length of the name)
    def __len__(self):
        return len(self._name)

    # Special method to delete the object
    def __del__(self):
        print(f"Person {self._name} is being deleted")
```

`__str__` : چاپ اطلاعات شئی به شکل دلخواه شما

`__len__` : شمارش آبجکت مورد نظر شما

`__del__` : حذف شئی

برای حذف شئی از دستور `del` استفاده میکنیم مثلا: `del person`

```python
# Property for 'name' attribute
@property
def name(self):
    return self._name

# Setter for 'name' attribute
@name.setter
def name(self, value):
    self._name = value
```

* ویژگی‌ها و تزئین‌کننده‌ها به ما امکان می‌دهند تا به شیوه‌ای کنترل‌شده به ویژگی‌های یک شیء دسترسی پیدا کنیم.*