

PROGRAMAÇÃO EM PYTHON

Fernando F. Santos



python



AULA 16

24/03/2023

CLASSES ABERTAS

No Python podemos alterar nossas classes e objetos em tempo de execução adicionando novos métodos e atributos.

Isso é possível porque o Python é uma linguagem dinâmica.

Somente não podemos modificar classes built-ins (ou seja, as classes embutidas do Python).

Podemos adicionar novos atributos e métodos de forma dinâmica quando não quisermos modificar uma classe existente ou quando não possuímos o código-fonte da classe, tendo apenas seu bytecode, por exemplo.

Vamos criar uma classe e gerar o bytecode para simular um caso real.

CRIAR UMA CLASSE E COMPILAR SEU CÓDIGO

1- Crie um projeto novo

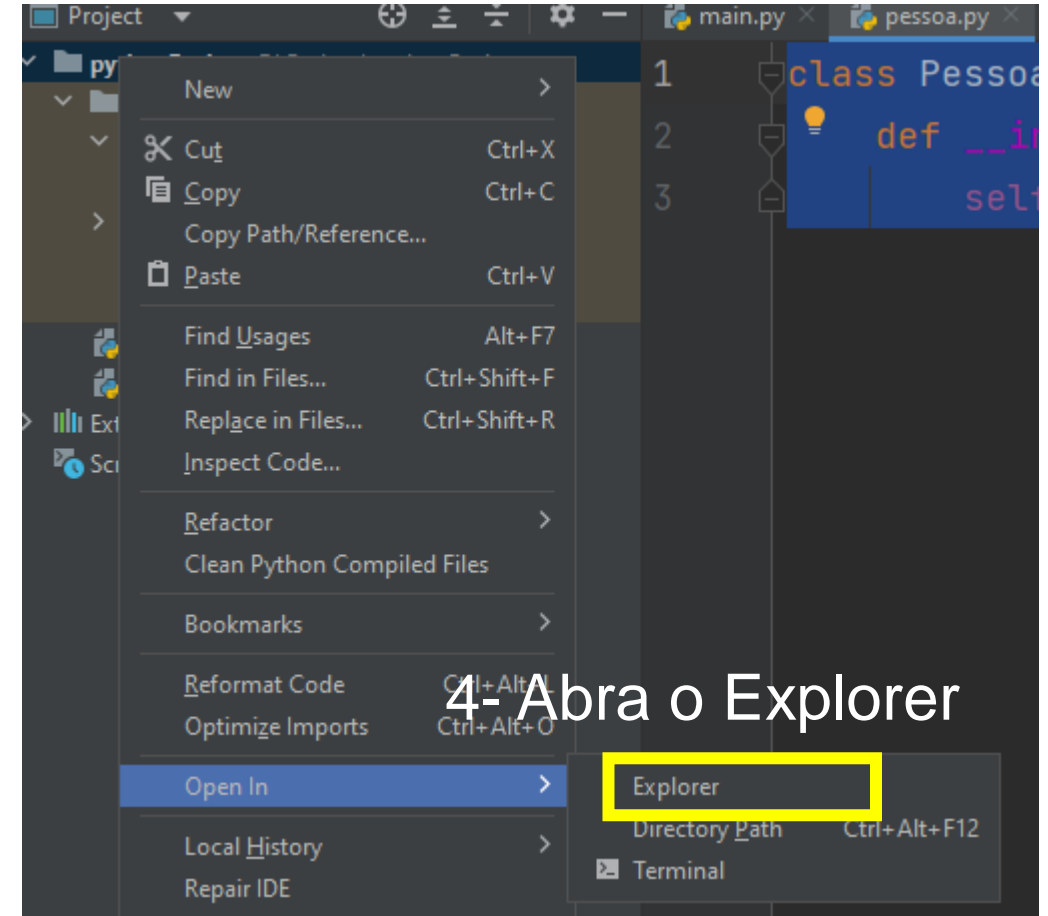
2- Crie a classe `pessoa.py`

```
class Pessoa(object):  
    def __init__(self, nome):  
        self.nome = nome
```

3- Abra o console e digite os comandos

```
>>> import py_compile  
>>> py_compile.compile("pessoa.py")  
'__pycache__\\pessoa.cpython-310.pyc'  
  
>>>
```

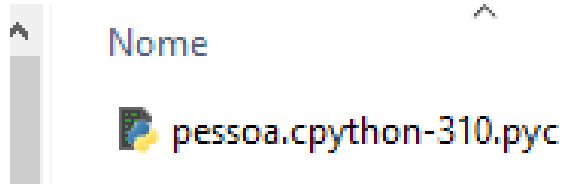
Python Console



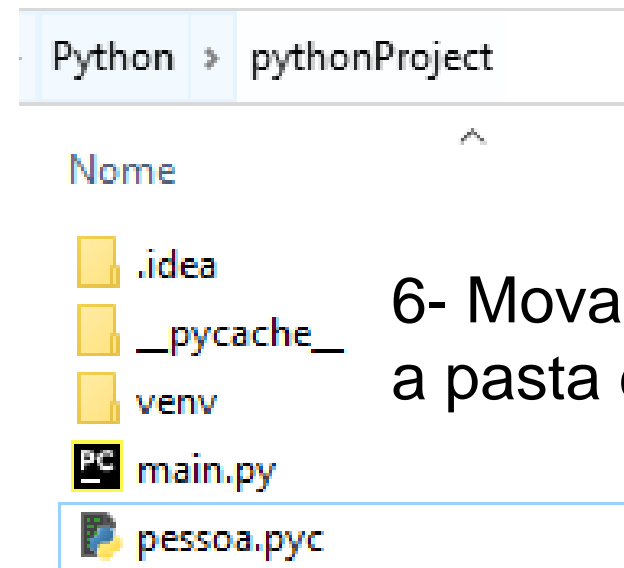
4- Abra o Explorer

CRIAR UMA CLASSE E COMPILAR SEU CÓDIGO

Python > pythonProject > __pycache__



5- Renomear o **pessoa.cpython-310.pyc** para **pessoa.pyc**



6- Mova o arquivo para a pasta do projeto

7- Apague o arquivo **pessoa.py**

IMPLEMENTANDO O PROGRAMA

```
from pessoa import Pessoa
# Criado objeto p da classe pessoa passando 'José' para nome
p = Pessoa('José')
print(p.nome) # imprimindo o valor do atributo nome do objeto p
# Criando um método chamado retornar_nome que retorna a propriedade nome (de self)
def pegar_nome(self):
    return self.nome
# Criando o método pegar_nome na classe Pessoa com base no método pegar_nome que criei
# acima
Pessoa.pegar_nome = pegar_nome
# Executando o método pegar nome do objeto p
print(p.pegar_nome())
# Criando o método mudar_nome
def mudar_nome(self, novo_nome):
    self.nome = novo_nome
```

IMPLEMENTANDO O PROGRAMA

```
# Criando o método mudar_nome na classe Pessoa
Pessoa.mudar_nome = mudar_nome
# Executando o método mudar_nome no objeto p
p.mudar_nome('Maria')
# Criando um método denominado imprimir_ola para imprimir "Olá x", onde x é o
valor da propriedade nome
def imprimir_ola(self):
    print("Olá", self.nome)
# Criando o método ola na classe Pessoa com base no método imprimir_ola
Pessoa.ola = imprimir_ola
# Executando o método ola do objeto p
p.ola()
```

EXERCÍCIOS DE FIXAÇÃO

- Resolva os exercícios da lista em anexo.