CLASSES E OBJETOS

Python é uma linguagem de programação orientada a objetos.

Quase tudo em Python é um objeto, com suas propriedades e métodos.

Uma classe é como um construtor de objeto ou um "projeto" para a criação de objetos.

Criação de uma classe

Para criar uma classe, use a palavra-chave `class`.

Exemplo:

```
class Myclass:
 pass #Palavra-chave para classe vazia não apresentar erro.
print(Myclass)

#Resultado:
<class '__main__.Myclass'>
```

💡 As definições de `class` não podem estar vazias, mas se por algum motivo você tiver uma classe sem conteúdo, coloque a instrução `pass` para evitar um erro.

Criação de um objeto

Agora podemos usar a classe chamada MyClass para criar objetos.

Exemplo:

```
class Myclass:
 x = 5
p1 = MyClass()
```

```
print(p1.x)

#Resultado:
5
```

Os exemplos acima são classes e objetos em sua forma mais simples e não são realmente úteis em aplicatições reais.

A função __init__()

Todas as classes possuem uma função chamada `__init __ ()`, que sempre é executada quando a classe está sendo iniciada. Ela é usada para atribuir valores às propriedades do objeto ou outras operações que são necessárias quando o objeto está sendo criado.

Exemplo:

Crie uma classe chamada Person, use a função __init __ () para atribuir valores para nome e idade:

```
class Personagem:
     def __init__(self, nome, idade):
        self.nome = nome
        self.idade = idade
p1 = Personagem("John", 36)
print(p1.nome)
print(p1.idade)
```

Nota: A função `__init__()` é chamada automaticamente toda vez que a classe está sendo usada para criar um novo objeto.

Observação: o parâmetro `self` é uma referência à instância atual da classe e é usado para acessar variáveis que pertencem à classe. vamos explicar melhor nesse documento.

Métodos de Objeto

Os objetos também podem conter métodos. Métodos em objetos são funções que pertencem ao objeto.

Vamos criar um método na classe Personagem:

Exemplo:

Insira uma função que imprima uma saudação e execute-a no objeto p1:

```
class Personagem:
     def __init__(self, nome, idade):
        self.nome = nome
        self.idade = idade
     def funcao(self):
print("Meu nome é " + self.nome)
p1 = Personagem("John", 36)
p1.funcao()
```

O parâmetro self

O parâmetro `self` é uma referência à instância atual da classe e é usado para acessar variáveis que pertencem à classe.

Não precisa ser nomeado `self`, você pode chamá-lo do que quiser, mas deve ser o primeiro parâmetro de qualquer função na classe. Basta observar o exemplo anterior. Faremos uma demonstração agora.

Exemplo:

Use as palavras strogonoff e fome *em* vez de *self*:

```
class Comida:
     def __init__(strogonoff, tipo, acomp):
        strogonoff.tipo = tipo
        strogonoff.acomp = acomp
     def funcao(fome):
         a = ("Eu adoro comer um " + fome.tipo)
         b = (" com " + fome.acomp)
         print(a + b)
prato = Comida("Strogonoff de frango", "coquinha gelada")
prato.funcao()

#Resultado:
Eu adoro comer um Strogonoff de frango com coquinha gelada
```

Modificar Propriedades do Objeto

Você pode modificar propriedades em objetos.

Exemplo:

No código anterior, substitua a propriedade “tipo” no objeto “prato” por “Strogonoff de Camarão”

```
class Comida:
    def __init__(strogonoff, tipo, acomp):
        strogonoff.tipo = tipo
        strogonoff.acomp = acomp

    def funcao(fome):
        a = ("Eu adoro comer um " + fome.tipo)
        b = (" com " + fome.acomp)
        print(a + b)

prato = Comida("Strogonoff de frango", "coquinha gelada")
prato.tipo = str("Strogonoff de Camarão") #mudança
prato.funcao()

#Resultado:
Eu adoro comer um Strogonoff de Camarão com,coquinha gelada
```

Excluir Propriedades do Objeto

Você pode excluir propriedades em objetos usando a palavra-chave `del`:

Exemplo:

No código anterior, exclua a propriedade “acomp” do objeto “prato”:

```
class Comida:
    def __init__(strogonoff, tipo, acomp):
        strogonoff.tipo = tipo
        strogonoff.acomp = acomp

    def funcao(fome):
        a = ("Eu adoro comer um " + fome.tipo)
        b = (" com " + fome.acomp)
        print(a + b)

prato = Comida("Strogonoff de frango", "coquinha gelada")
del acomp
prato.funcao()

#Resultado:
NameError: name 'acomp' is not defined
#Exatamente um erro pois exigimos que acomp seja impresso, porém foi deletada.
```

Você também pode excluir objetos com essa mesma sintaxe.