

# Programação Orientada à Objetos: Parâmetros Opcionais em métodos

Tomamos como exemplo novamente a nossa classe gato:

```
class Gato:
    def __init__(self, peso, idade, nome="sem nome", raça="sem raça"):
        self.peso=peso
        self.idade=idade
        self.nome = nome
        self.raça = raça
    def mudar_nome(self, nome):
        self.nome=nome
    def engordar(self, peso):
        self.peso+=peso
    def envelhecer(self):
        self.idade +=1
```

parâmetros opcionais

**mas o que são esses parâmetros?**

# Programação Orientada à Objetos: Parâmetros Opcionais em métodos

- Nem sempre precisaremos passar todos os parâmetros para uma função ou método;
- Podemos utilizar um valor previamente escolhido como padrão, mas deixando a possibilidade de alterá-lo, caso necessário.
- Ex:

```
def barra():  
    print(40 * '*')
```

```
def barra(n=40, caractere='*'):  
    print(n * caractere)
```

# Programação Orientada à Objetos: Parâmetros Opcionais em métodos

Exemplos de execução:

```
barra()
```

```
barra(10)
```

```
barra(20, '+')
```

```
barra('$')
```

**Qual o resultado da execução dessas linhas?**

# Programação Orientada à Objetos:

## Parâmetros Opcionais em métodos

```
***** # barra()
***** # barra(10)
+++++++ # barra(20,'+') # barra('$')
Traceback (most recent call last):
  File "/home/rogerio/PycharmProjects/P00_266/ex_funções1.py",
    main()
  File "/home/rogerio/PycharmProjects/P00_266/ex_funções1.py",
    barra('$')
  File "/home/rogerio/PycharmProjects/P00_266/ex_funções1.py",
    print(n * caractere)
TypeError: can't multiply sequence by non-int of type 'str'
```



Outro exemplo. Função soma com argumentos obrigatórios e opcionais.

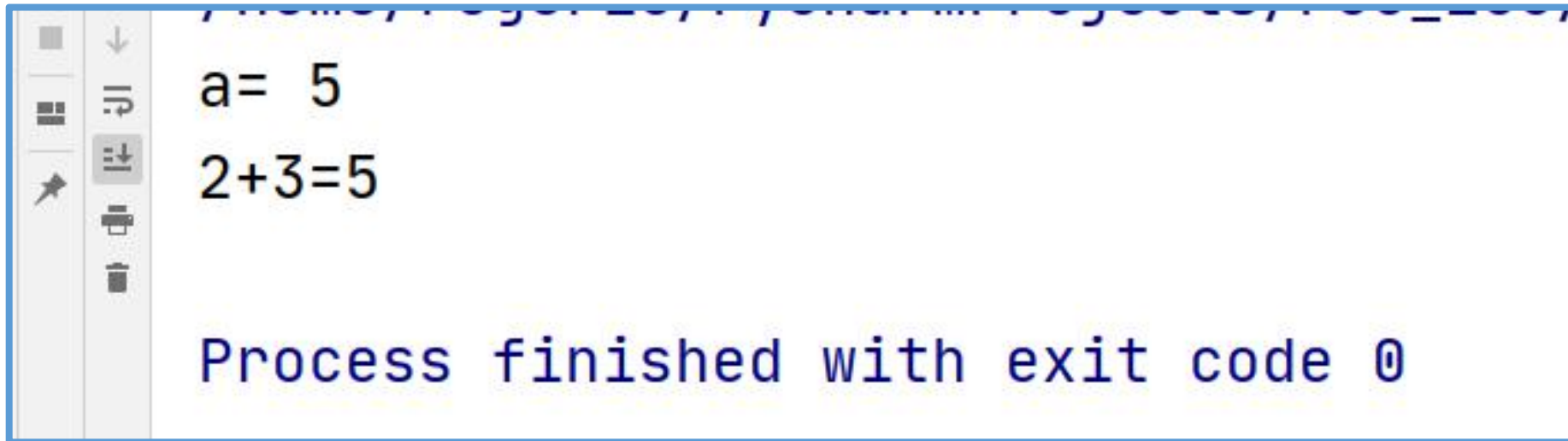
```
def soma(a,b,imprime=False):  
    if imprime:  
        print('{}+{}={}'.format(a,b,(a+b)))  
    return a+b
```

**chamada da função:**

```
a=soma(2,3)  
print("a=",a)  
soma(2,3,True)
```

# Programação Orientada à Objetos: Parâmetros Opcionais em métodos

Execução:

A screenshot of a terminal window with a blue border. On the left side, there is a vertical toolbar with icons for window management (minimize, maximize, close), a search icon, and a list of open files. The terminal output is as follows:

```
a= 5  
2+3=5  
  
Process finished with exit code 0
```

# Programação Orientada à Objetos: Parâmetros Opcionais em métodos

voltando ao nosso exemplo inicial da classe gato...:

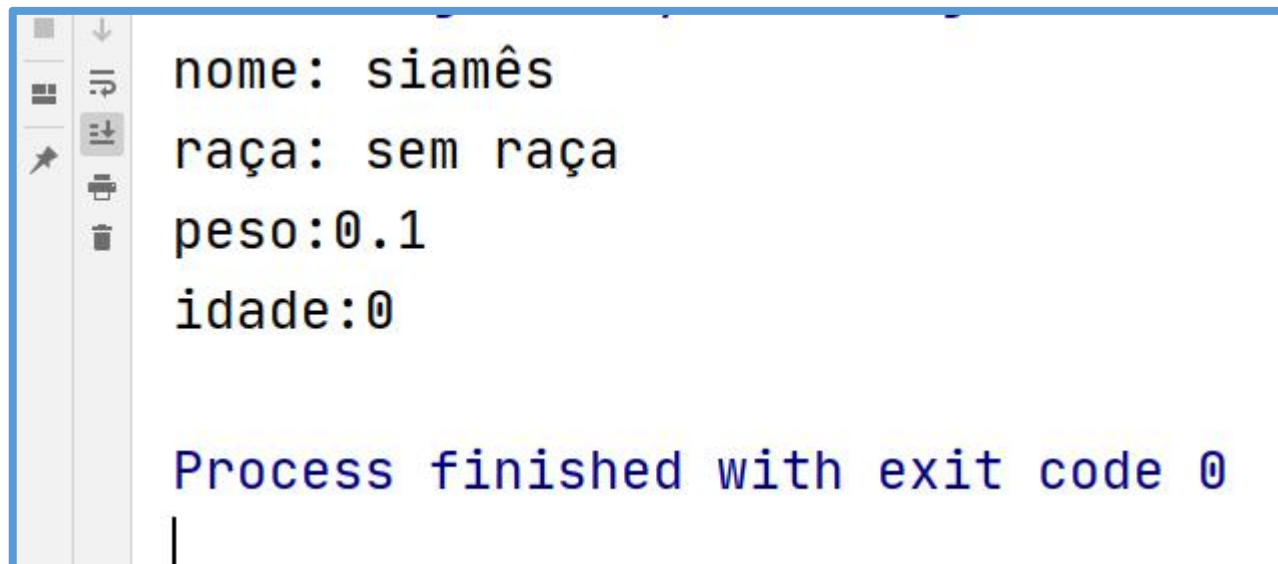
```
class Gato:
    def __init__(self, peso, idade, nome="sem nome", raça="sem raça"):
        self.peso=peso
        self.idade=idade
        self.nome = nome
        self.raça = raça
    def mudar_nome(self, nome):
        self.nome=nome
    def engordar(self, peso):
        self.peso+=peso
    def envelhecer(self):
        self.idade +=1
```

**Agora faz mais sentido a nova definição do construtor?**

Como definir um objeto gato de peso: 0.1 kg, 0 anos de idade e de raça: siamês?

```
meu_gato=Gato(0.1,0,"siamês") ?
```

```
print("nome:",meu_gato.nome)
print("raça:",meu_gato.raça)
print("peso:{}".format(meu_gato.peso))
print("idade:{}".format(meu_gato.idade))
```

A terminal window with a blue border. On the left is a vertical toolbar with icons for file operations (new, open, save, print, delete) and window management (maximize, close). The main area shows the output of the Python script: 'nome: siamês', 'raça: sem raça', 'peso:0.1', and 'idade:0'. Below this, it says 'Process finished with exit code 0' and a cursor is visible on a new line.

```
nome: siamês
raça: sem raça
peso:0.1
idade:0

Process finished with exit code 0
|
```

**Errado!**



## resposta correta...

```
meu_gato=Gato(0.1,0,raça="siamês")
```



```
nome: sem nome  
raça: siamês  
peso:0.1  
idade:0
```

```
Process finished with exit code 0
```

# Programação Orientada à Objetos: Parâmetros Opcionais em métodos

## **outras possibilidades...**

- Crie um objeto gato com peso:1.5 kg e 2 anos de idade
- Crie um objeto gato com peso: 1.0 kg, 1 ano de idade e nome: Tom
- Crie um objeto gato com peso: 2.0 kg, 3 anos de idade, nome='Mimi' e raça: siamês.

# Programação Orientada à Objetos: Construtores

## Exercícios