

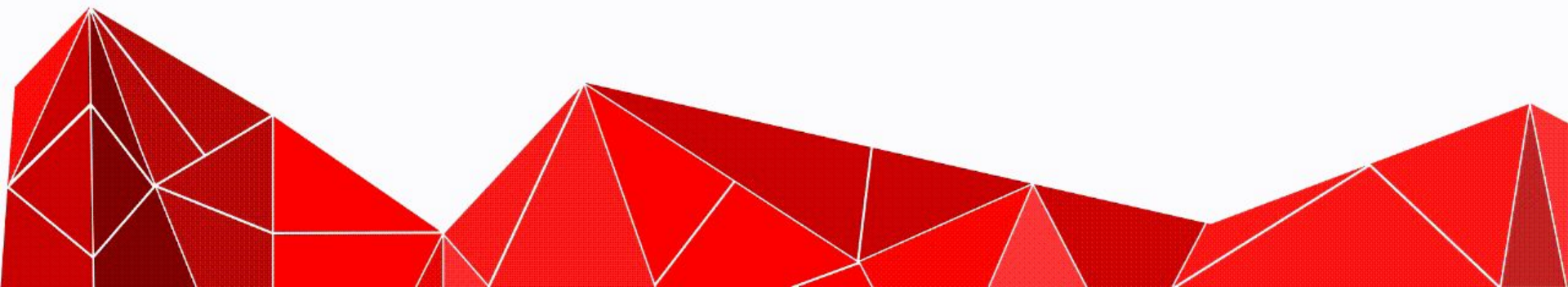


CODING

FUNÇÕES - PARTE 01

Press -> to continue

Powered by PET computação UFC

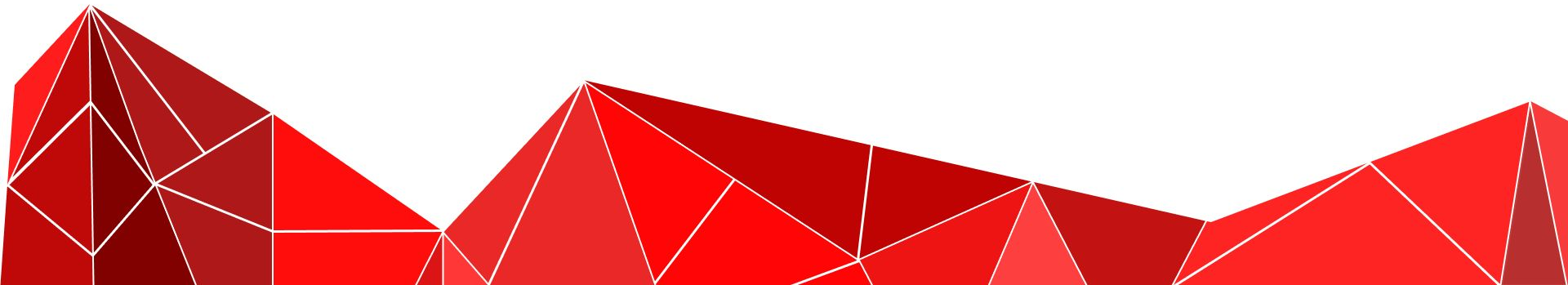


O'Que vamos ver hoje ?

Introdução

Como declarar

Argumentos e parâmetros

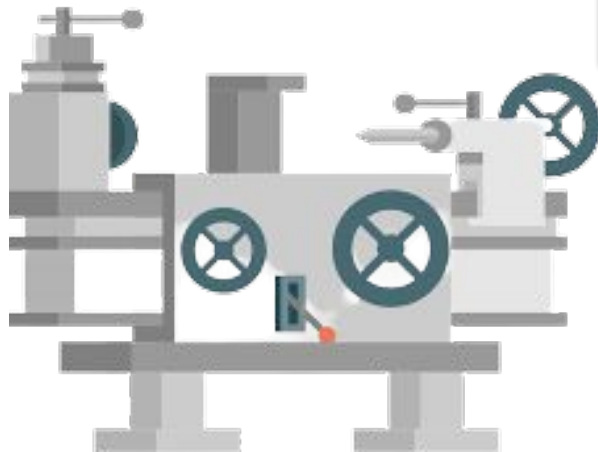
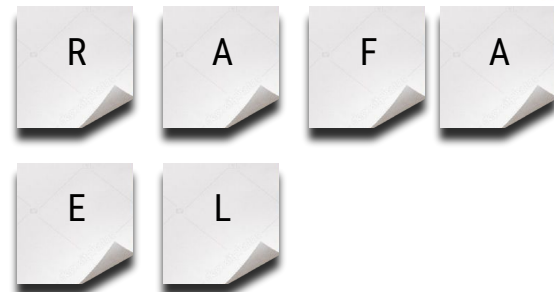




INTRODUÇÃO

O que é isso ?

- Como o nome já diz
- Uma parte do código que exerce uma FUNÇÃO específica
- Pode ou não, receber e/ou retornar algum valor
- Exemplo:



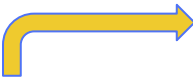



Como declarar




Como declarar

- No python é bem simples
- Usamos a seguinte estrutura:

 Keyword para definir uma função

`def nomeDaFunção ():`  Onde indicamos os parâmetros

 código

 Cuidado com a indentação

- Vejamos um exemplo

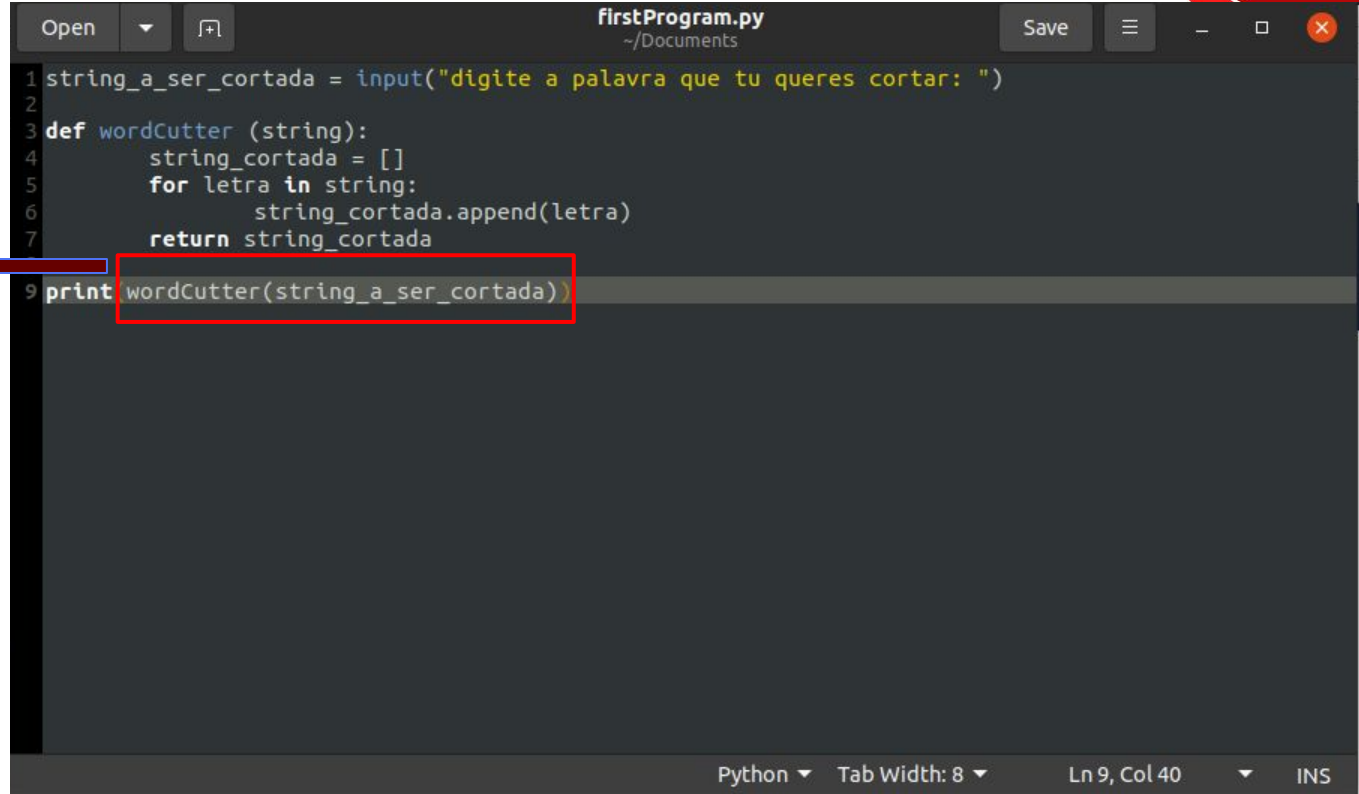


Exemplo

- Podemos transcrever nossa máquina corta-palavras para o Python
- O código seria assim

Veja que precisamos chamar a função para que ela consiga funcionar

É como se estivéssemos ligando a máquina, a diferença é que podemos chamar quantas vezes quisermos



```
1 string_a_ser_cortada = input("digite a palavra que tu queres cortar: ")
2
3 def wordCutter (string):
4     string_cortada = []
5     for letra in string:
6         string_cortada.append(letra)
7     return string_cortada
8
9 print(wordCutter(string_a_ser_cortada))
```

The screenshot shows a code editor window titled "firstProgram.py" with a dark theme. The code defines a function `wordCutter` that takes a string and returns a list of its characters. Line 9 shows the function being called with `print(wordCutter(string_a_ser_cortada))`. A red box highlights the function call, and a blue arrow points from the text "Veja que precisamos chamar a função para que ela consiga funcionar" to it. The status bar at the bottom indicates "Python", "Tab Width: 8", "Ln 9, Col 40", and "INS".

Exemplo

- Ela retornaria o seguinte resultado:

Aí sim meu
patrão

```
rafael@rafael-Lenovo-ideapad-330-15IKB: ~/Documents
rafael@rafael-Lenovo-ideapad-330-15IKB:~/Documents$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exec(open('firstProgram.py').read())
digite a palavra que tu queres cortar: rafael
['r', 'a', 'f', 'a', 'e', 'l']
>>>
```




ARGUMENTOS

Passando informação para a função

- Para “passarmos” algo para a função indicamos nos parênteses
- Indicamos na hora da construção da função
- Na hora de chamar a função passamos o argumento
- Atenção
 - O argumento passado precisa ser do mesmo tipo com o qual você construiu a função
- Vamos ver o código da máquina corta-palavras

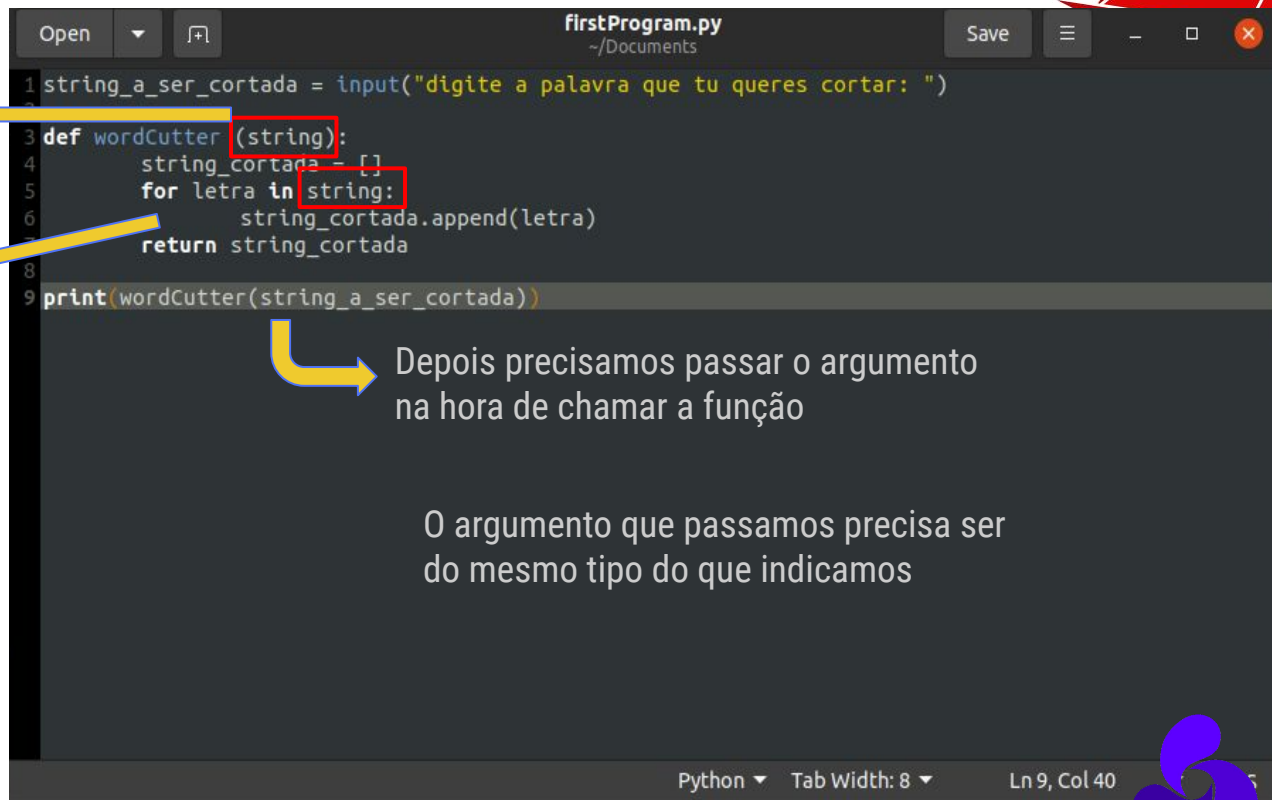


Passando informação para a função

Indicamos o parâmetro na construção

Depois trabalhamos com ela normalmente, indicando o que será feito

O nome do parâmetro é da sua preferência



```
1 string_a_ser_cortada = input("digite a palavra que tu queres cortar: ")
2
3 def wordCutter (string):
4     string_cortada = []
5     for letra in string:
6         string_cortada.append(letra)
7     return string_cortada
8
9 print(wordCutter(string_a_ser_cortada))
```

Depois precisamos passar o argumento na hora de chamar a função

O argumento que passamos precisa ser do mesmo tipo do que indicamos



Passando mais de um argumento

- Argumentos posicionais
 - Sabemos o que fazer com cada argumento dependendo da posição
 - Exemplo :

```
firstProgram.py
~/Documents

Open  Save  -  □  ×

1 nomeDoCliente = "Rafael"
2 valorApagarEmReais = 1500
3
4 def mostrarDividaEmDolares (cliente, divida):
5     dividaEmDolares = divida*5
6     print(f"O nome do cliente é {cliente}")
7     print(f"E o valor da sua divida em dolares é {dividaEmDolares}")
8
9 mostrarDividaEmDolares(nomeDoCliente, valorApagarEmReais)
10
```



Passando mais de um argumento

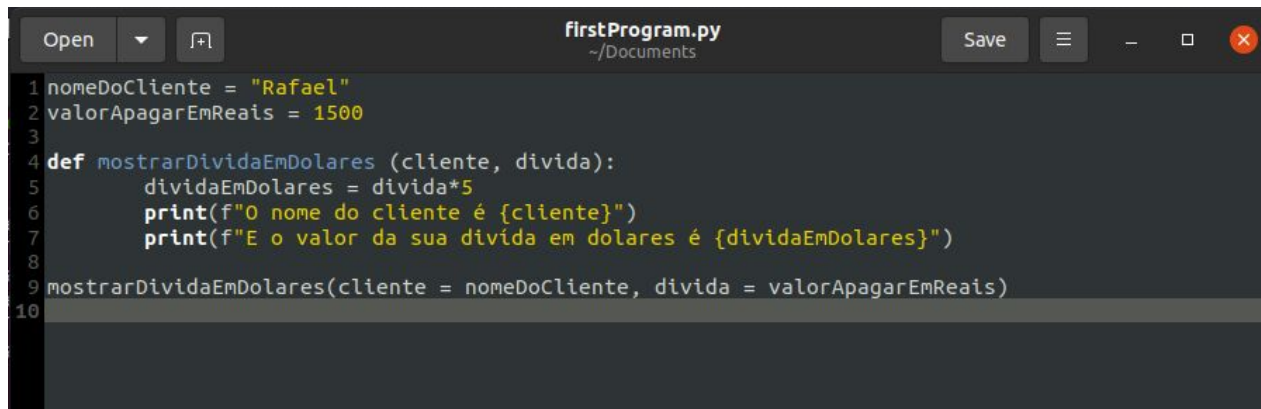
- Argumentos posicionais
 - Sabemos o que fazer com cada argumento dependendo da posição
 - Exemplo :

```
rafael@rafael-Lenovo-ideapad-330-15IKB: ~/Documents
rafael@rafael-Lenovo-ideapad-330-15IKB:~/Documents$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exec(open('firstProgram.py').read())
O nome do cliente é Rafael
E o valor da sua dívida em dolares é 7500
>>> 
```



Palavras chave

- Aqui não precisamos nos preocupar na ordem de passar os argumentos
- Indicamos o parâmetro e seu valor
- Evitamos trocar as ordens
- Exemplo:



```
firstProgram.py
~/Documents

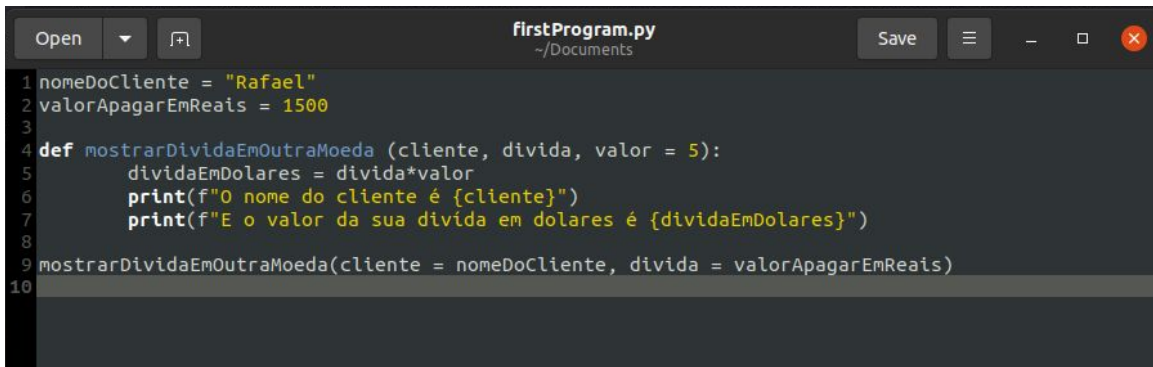
Open [dropdown] [icon] Save [icon] [icon] [icon] [icon]

1 nomeDoCliente = "Rafael"
2 valorApagarEmReais = 1500
3
4 def mostrarDividaEmDolares (cliente, divida):
5     dividaEmDolares = divida*5
6     print(f"O nome do cliente é {cliente}")
7     print(f"E o valor da sua divida em dolares é {dividaEmDolares}")
8
9 mostrarDividaEmDolares(cliente = nomeDoCliente, divida = valorApagarEmReais)
10
```



Valores padrões

- Definimos um valor padrão para o parâmetro
- Podemos usar para tornar um parâmetro opcional
- Caso não seja especificado
- O Python usará o valor padrão
- Exemplo:



```
firstProgram.py
~/Documents

1 nomeDoCliente = "Rafael"
2 valorApagarEmReais = 1500
3
4 def mostrarDividaEmOutraMoeda (cliente, divida, valor = 5):
5     dividaEmDolares = divida*valor
6     print(f"O nome do cliente é {cliente}")
7     print(f"E o valor da sua divida em dolares é {dividaEmDolares}")
8
9 mostrarDividaEmOutraMoeda(cliente = nomeDoCliente, divida = valorApagarEmReais)
10
```



Retornando valores

- Não é obrigatório uma função retornar algum valor
- Para retornar é bem simples
- Exemplo:

```
firstProgram.py
~/Documents
Save

1 primeiroNome = "Jimi"
2 segundoNome = "Hendrix"
3
4
5 def retornarNome (nome, sobrenome):
6     nomeCompleto = nome + ' ' + sobrenome
7     return nomeCompleto
8
9 nomeCompleto = retornarNome(primeiroNome, segundoNome)
10
11 print(nomeCompleto)
12
```



Retornando valores

- Resultado:

```
rafael@rafael-Lenovo-ideapad-330-15IKB: ~/Documents
rafael@rafael-Lenovo-ideapad-330-15IKB:~/Documents$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exec(open('firstProgram.py').read())
Jimi Hendrix
>>> 
```





Variáveis locais e globais



Variáveis locais e globais

- Imagine que você tem uma conta no banco C6, o banco C6 possui uma opção de conta internacional, logo você pode acessar o seu dinheiro em qualquer parte do planeta
- Agora imagine que você também tem uma conta no Nubank, porém no Nubank você só possui uma conta válida no Brasil, logo caso você viaje para outro país não pode acessar sua conta.



Variáveis locais e globais

- Podemos usar a mesma lógica para a programação.
- Os países podem ser comparados às funções
- Quando declaramos uma variável dentro de uma função, ela é chamada uma variável local, ou seja, só podemos acessar essa variável dentro dessa função.
 - Mas porque ?
- Já quando declaramos uma variável fora das funções, ela é declarada na estrutura principal do código, conseguimos acessar ela em qualquer parte e/ou função do código. Assim ela é chamada de função global.
- Vamos ver um exemplo que já vimos hoje.



Variáveis locais e globais



```
1 nomeDoCliente = "Rafael"
2 valorApagarEmReais = 1500
3
4 def mostrarDividaEmOutraMoeda (cliente, divida, valor = 5):
5     dividaEmDolares = divida*valor
6     print(f"O nome do cliente é {cliente}")
7     print(f"E o valor da sua divida em dolares é {dividaEmDolares}")
8
9 mostrarDividaEmOutraMoeda(cliente = nomeDoCliente, divida = valorApagarEmReais)
10
```

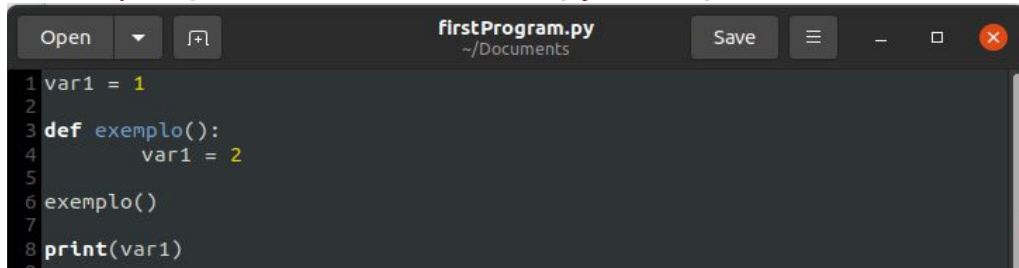
Variáveis globais

Variáveis locais



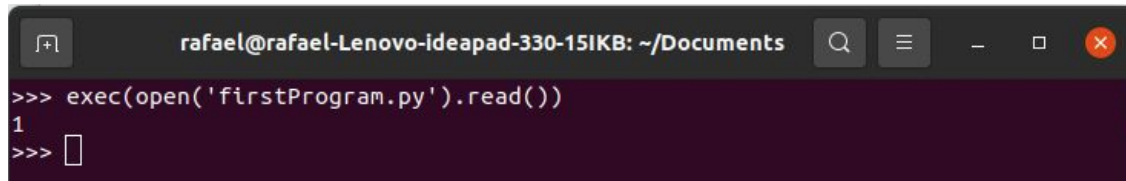
Variáveis locais e globais

- Mas veja bem, estamos utilizando a variável global, mas não estamos alterando o seu valor dentro da função.
- Para alterar o valor da função precisamos dizer ao python que estamos alterando uma variável global



```
1 var1 = 1
2
3 def exemplo():
4     var1 = 2
5
6 exemplo()
7
8 print(var1)
```

- Veja que o valor da variável var1 não se altera.

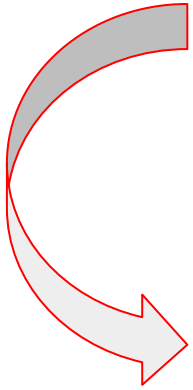


```
rafael@rafael-Lenovo-ideapad-330-151KB: ~/Documents
>>> exec(open('firstProgram.py').read())
1
>>> 
```



Variáveis locais e globais

- Para alterar o valor da variável, precisamos avisar ao python que a variável é uma Variável global.
- Fazemos isso da seguinte maneira.



```
firstProgram.py
~/Documents

1 var1 = 1
2
3 def exemplo():
4     global var1
5     var1 = 2
6
7 exemplo()
8
9 print(var1)
10
```

Nessa linha indicamos ao python que a variável var1 se refere à variável global já criada e que não estamos criando uma variável local.

```
rafael@rafael-Lenovo-ideapad-330-151KB: ~/Documents

>>> exec(open('firstProgram.py').read())
2
>>> 
```





Obrigado!!
Ainda com dúvida ?
Entre em contato



Exercícios

- Crie uma função que receba uma string como argumento e retorne a mesma string em letras maiúsculas. Faça uma chamada à função, passando como parâmetro uma string.



Exercícios

- Faça um programa que tenha uma função chamada `área()`, que receba as dimensões de um terreno retangular (largura e comprimento) e mostre a área do terreno.



Exercícios

- Crie um programa que tenha uma função chamada voto() que vai receber como parâmetro a idade de uma pessoa e retorna um valor literal indicando se uma pessoa tem voto NEGADO, OPCIONAL ou OBRIGATÓRIO nas eleições.



Exercícios

- Faça uma função que informe a quantidade de dígitos de um determinado número inteiro informado.



Exercícios

Faça um programa para imprimir:

```
1
2 2
3 3 3
....
n n n n n n ... n
```

para um n informado pelo usuário. Use uma função que receba um valor n inteiro e imprima até a n -ésima linha.

