

# LÓGICA E LINGUAGEM DE PROGRAMAÇÃO

Curso Técnico Integrado em Informática  
Lucas Sampaio Leite



# Funções

- O escopo de uma variável determina em quais partes do código ela pode ser acessada ou modificada.
- Escopo Local:
  - Variáveis declaradas dentro de uma função pertencem ao seu escopo local.
  - Só podem ser utilizadas dentro da própria função onde foram criadas.
- Escopo Global:
  - Variáveis declaradas fora de funções, no corpo principal do programa.
  - Podem ser acessadas em qualquer parte do código, tanto no escopo global quanto dentro de funções.

# Funções

```
var_global = "Sou uma variável global \o/"

def show_me_the_code():
    var_local = "Sou uma variável local o/"
    print(var_local)
    print(var_global)

show_me_the_code()
print(var_global)
print(var_local)
```

Qual a saída do programa?

# Funções

```
var_global = "Sou uma variável global \o/"

def show_me_the_code():
    var_local = "Sou uma variável local o/"
    print(var_local)
    print(var_global)

show_me_the_code()
print(var_global)
print(var_local)
```



```
Sou uma variável local o/
Sou uma variável global \o/
Sou uma variável global \o/
Traceback (most recent call last):
  File "/home/lucas/Dropbox/IF_Baiano/logica_subsequente/codes/exemplo2.py", line 10, in <module>
    print(var_local)
NameError: name 'var_local' is not defined. Did you mean: 'var_global'?
```

# Funções

- Escopo local:
  - Criado dentro de uma função.
  - Só pode ser utilizado na função onde foi declarado.

```
def minha_funcao():  
    x = 10  
    print("Dentro da função:", x)  
  
minha_funcao()  
print(x)
```

Qual a saída do programa?

# Funções

- Escopo local:
  - Criado dentro de uma função.
  - Só pode ser utilizado na função onde foi declarado.

```
def minha_funcao():  
    return 10  
  
minha_funcao()  
print(x)
```

Qual a saída do programa?  
Resolveu o problema anterior?  
Como resolver?

# Funções

- Escopo local:
  - Criado dentro de uma função.
  - Só pode ser utilizado na função onde foi declarado.

```
def calcular_area_quadrado(lado):  
    area = lado * lado  
    return area  
  
print(calcular_area_quadrado(4))  
print(area)
```

Qual a saída do programa?

# Funções

- Escopo global:
  - Criado fora de funções, no corpo principal do programa.
  - Pode ser acessado tanto globalmente quanto dentro de funções.

```
y = 20

def outra_funcao():
    print("Dentro da função:", y)

outra_funcao()
print("Fora da função:", y)
```

Qual a saída do programa?



# Funções

- Escopo global:
  - Criado fora de funções, no corpo principal do programa.
  - Pode ser acessado tanto globalmente quanto dentro de funções.

```
taxa_juros = 0.05

def calcular_juros(valor):
    return valor * taxa_juros

print("Juros sobre R$1000:", calcular_juros(1000))
print("Taxa utilizada:", taxa_juros)
```

Qual a saída do programa?

# Funções

```
z = 5

def altera_global():
    z = 15
    print(z)
    return z

print(altera_global())
print(z)
```

Qual a saída do programa?

# Funções

Quando criamos uma variável local com o mesmo nome de uma variável global, não estamos alterando a global; apenas criamos uma nova variável independente, restrita ao escopo local.

```
z = 5

def altera_global():
    z = 15
    print(z)
    return z

print(altera_global())
print(z)
```



15  
15  
5

# Funções

- É possível modificar uma variável global dentro de uma função usando a palavra-chave global.

```
z = 5

def altera_global():
    global z
    z = 15
    print(z)
    return z

print(altera_global())
print(z)
```

Qual a saída do programa?

# Funções

- É possível modificar uma variável global dentro de uma função usando a palavra-chave global.

```
z = 5

def altera_global():
    global z
    z = 15
    print(z)
    return z

print(altera_global())
print(z)
```



```
15
15
15
```

# Funções

- É possível modificar uma variável global dentro de uma função usando a palavra-chave `global`.

```
contador = 0

def registrar_acesso():
    global contador
    contador += 1
    print("Acesso registrado. Total:", contador)

registrar_acesso()
registrar_acesso()
print("Total de acessos:", contador)
```

Qual a saída do programa?

# Funções

```
contador = 0

def registrar_acesso():
    global contador
    contador += 1
    print("Acesso registrado. Total:", contador)

registrar_acesso()
registrar_acesso()
print("Total de acessos:", contador)
```



```
Acesso registrado. Total: 1
Acesso registrado. Total: 2
Total de acessos: 2
```



# Exercícios

1. Crie uma função `calcular_dobro(numero)` que:
  - Declare uma variável local chamada `resultado`;
  - Calcule o dobro do número recebido como parâmetro;
  - Retorne o valor de `resultado`.

Teste chamando a função com diferentes números e tente acessar `resultado` fora dela. O que acontece?



## Exercícios

2. Implemente uma função `celsius_para_fahrenheit(celsius)` que:
- Use uma variável local `fahrenheit` para armazenar o resultado da conversão;
  - Retorne esse valor.

Confirme que a variável `fahrenheit` não existe fora da função.

## Exercícios

3. Crie uma variável global  $\text{PI} = 3.14159$ . Depois, implemente uma função `calcular_circunferencia(raio)` que:
- Use a variável global `PI` para calcular a circunferência ( $2 * \text{PI} * \text{raio}$ ).
  - Retorne o valor calculado.

Mostre que `PI` pode ser acessada tanto dentro da função quanto fora dela.

## Exercícios

4. Implemente uma variável global `acessos = 0`. Depois, crie uma função `registrar_login()` que:
- Use a palavra-chave `global`;
  - A cada chamada, incremente `acessos` em 1 e exiba o número total de logins registrados.

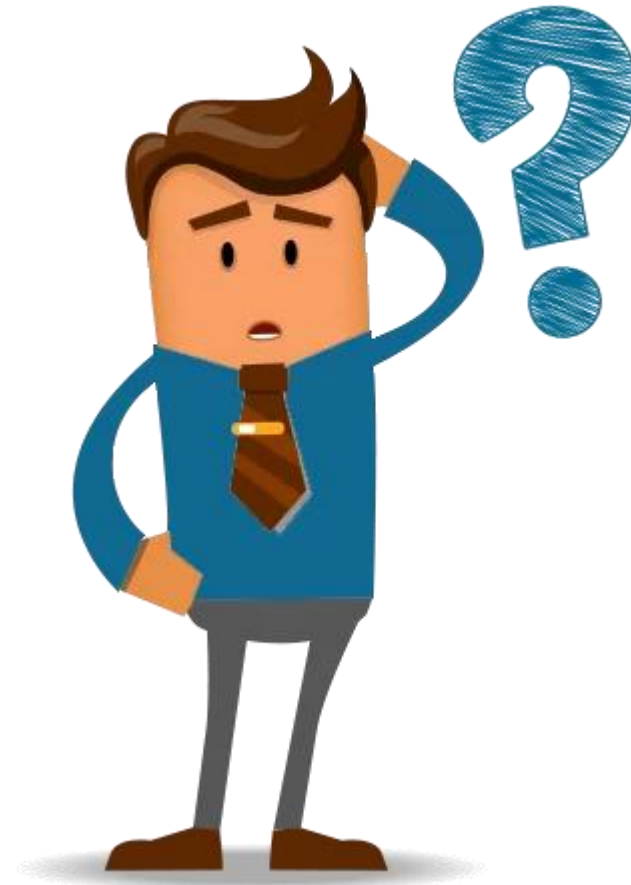
Teste chamando a função várias vezes e verifique o valor atualizado de `acessos`.

## Exercícios

5. Crie uma variável global `saldo = 1000`. Implemente uma função `sacar(valor)` que:
- Verifique se há saldo suficiente;
  - Se sim, use global para atualizar o saldo global;
  - Use uma variável local mensagem para retornar se a operação foi bem-sucedida ou não.
  - Implemente também a função `consultar_saldo()` que apenas exibe o valor atual de saldo.

Teste as funções criadas.

# Dúvidas



# LÓGICA E LINGUAGEM DE PROGRAMAÇÃO

Curso Técnico Integrado em Informática  
Lucas Sampaio Leite

