

Prof. esp. Thalles Canela

- **Graduado:** Sistemas de Informação - Wyden Facimp
- **Pós-graduado:** Segurança em redes de computadores - Wyden Facimp
- **Professor:** Todo núcleo de T.I. (Graduação e Pós) - Wyden Facimp
- **Diretor:** SCS
- **Gerente de Projetos:** Motoca Systems

Redes sociais:

- **Linkedin:** <https://www.linkedin.com/in/thalles-canela/>
- **YouTube:** <https://www.youtube.com/aXR6CyberSecurity>
- **Facebook:** <https://www.facebook.com/axr6PenTest>
- **Instagram:** https://www.instagram.com/thalles_canela
- **Github:** <https://github.com/ThallesCanela>
- **Github:** <https://github.com/aXR6>
- **Twitter:** <https://twitter.com/Axr6S>

Nomes, Vinculações e Escopo

- Em Python, como em outras linguagens de programação, os nomes referem-se a locais na memória onde os dados são armazenados.
- A relação entre os nomes e os locais de memória é conhecida como vinculação.
- O local onde um nome é visível e pode ser utilizado é definido pelo seu escopo.

Declarando e atribuindo valores a variáveis:

- Em Python, não é necessário declarar explicitamente uma variável antes de atribuir um valor a ela.
- Uma variável é criada no momento em que você a atribui pela primeira vez.

Definição

- Uma variável é um nome dado a uma área de armazenamento na memória do computador.
- Em Python, não precisamos declarar o tipo de variável ao criar uma; a linguagem determina o tipo de variável automaticamente com base no valor atribuído a ela.

Atribuição

- A operação de dar um valor a uma variável é chamada de atribuição. Em Python, usamos o operador = para atribuir um valor a uma variável.

Tipos de Dados

- Como Python é uma linguagem de tipagem dinâmica, o tipo de uma variável pode mudar durante a execução do programa. As variáveis podem representar vários tipos de dados, incluindo:

Tipos de Dados

- Inteiros (int): Ex: 5, -3
- Pontos flutuantes (float): Ex: 5.0, -3.14
- Strings (str): Ex: "Hello", 'Python'
- Listas: Ex: [1, 2, 3], ["a", "b", "c"]
- Dicionários: Ex: {"nome": "João", "idade": 25}
- Booleanos (bool): True, False

Nomeação

- Os nomes podem começar com uma letra (a-z, A-Z) ou um sublinhado (`_`), mas não com números.
- O resto do nome pode conter letras, números e sublinhados.
- Os nomes são sensíveis a maiúsculas e minúsculas (ex., `idade` e `Idade` são duas variáveis diferentes).
- Evite usar palavras reservadas do Python como nomes de variáveis (ex., `while`, `if`, `print`).

Múltiplas Atribuições

- Python permite múltiplas atribuições em uma única linha:

python

 Copy code

```
x, y, z = 5, "Python", [1, 2, 3]
```

Variáveis e Memória

- Quando uma variável é criada em Python, um espaço de memória é reservado para ela. Usando a função `id()`, podemos obter o endereço de memória da variável.

python

 Copy code

```
x = 10  
print(id(x)) # Mostra o endereço de memória de x
```

Variáveis são Referências

- Em Python, as variáveis são na verdade referências a objetos. Isso significa que quando você cria uma variável que aponta para um objeto e, em seguida, atribui essa variável a outra, ambas as variáveis apontam para o mesmo objeto.

python

 Copy code

```
a = [1, 2, 3]
```

```
b = a
```

```
b[0] = 10
```

```
print(a) # Saída: [10, 2, 3] - a foi afetado porque b aponta para o me
```

Declarando e atribuindo valores a variáveis:

python

 Copy code

```
x = 10 # cria a variável x e atribui o valor 10  
nome = "Alice" # cria a variável nome e atribui a string "Alice"
```

Tipos dinâmicos:

python

 Copy code

```
x = 10          # x é do tipo int  
x = "Texto"     # x agora é do tipo str
```


ESCOPO.

Python tem quatro tipos básicos de escopo:

- **Escopo Local (L - Local):** Refere-se a nomes definidos dentro de uma função. A variável só é acessível dentro dessa função.
- **Escopo Encerrado (E - Enclosing):** Refere-se a nomes no escopo local de qualquer e todas as funções encapsuladas, de dentro para fora.
- **Escopo Global (G - Global):** Refere-se a nomes definidos no nível do módulo principal ou no corpo de um script.
- **Escopo Incorporado (B - Built-in):** Refere-se a nomes que estão predefinidos no Python.

ESCOPO.

Python tem quatro tipos básicos de escopo:

```
python Copy code

x = 10 # Global

def funcao_externa():
    y = 5 # Enclosing
    def funcao_interna():
        z = 3 # Local
        print(x, y, z) # acessa x (Global), y (Enclosing) e z (Local)
    funcao_interna()

funcao_externa()
```


Regras de LEGB:

- Quando uma variável é referenciada, o Python segue a regra LEGB para encontrar a variável.

Regras de LEGB:

- Primeiro, verifica-se o escopo local.
- Em seguida, verifica-se os escopos encerrados.
- Depois, verifica-se o escopo global.
- E, finalmente, verifica-se o escopo incorporado.
- Se uma variável não é encontrada em nenhum dos escopos, ocorrerá um 'NameError'.

LEGB

- A regra LEGB é uma diretriz que o Python usa para resolver nomes quando eles são referenciados no código.
- Essas letras representam a ordem em que o Python busca o nome de uma variável ou função.

L: Local

- Definição: Refere-se a nomes definidos diretamente dentro de uma função ou método. Eles não são visíveis fora da função em que são declarados.
- Aqui, 'variavel_local' só existe e é visível dentro de 'minha_funcao()'.

- **Exemplo:**

python

 Copy code


```
def minha_funcao():  
    variavel_local = "Estou no escopo local!"  
    print(variavel_local)
```

E: Enclosing (ou Enclosed)

- Definição: Este é um escopo intermediário que cobre funções ou métodos encapsulados. Se uma função estiver aninhada dentro de outra função, a função mais externa é considerada como o escopo "encerrado" da função mais interna.

• Exemplo:

python

 Copy code

```
def funcao_externa():  
    variavel_enclosing = "Estou no escopo enclosing!"  
  
    def funcao_interna():  
        print(variavel_enclosing)  
  
    funcao_interna()  
  
funcao_externa()
```

E: Enclosing (ou Enclosed)

- Aqui, 'funcao_interna()' pode acessar 'variavel_enclosing' porque 'variavel_enclosing' está no escopo de "encerramento" para 'funcao_interna()'.

G: Global

- Definição: Refere-se a nomes que são definidos no nível do módulo ou script. Eles são visíveis em todo o arquivo, mas não em outros módulos ou scripts, a menos que sejam explicitamente importados.

- **Exemplo:**

python

 Copy code

```
variavel_global = "Estou no escopo global!"
```

```
def minha_funcao():  
    print(variavel_global)
```

```
minha_funcao()
```

G: Global

- Aqui, 'minha_funcao()' pode acessar 'variavel_global' porque 'variavel_global' está no escopo global.

B: Built-in

- Definição: Estes são nomes que estão sempre disponíveis em Python porque são parte do módulo 'builtins'. Eles incluem funções como 'print()', 'len()', e tipos como 'list' e 'dict'.

- **Exemplo:**

python

 Copy code

```
def minha_funcao():  
    print(len("teste"))
```

```
minha_funcao()
```

B: Built-in

- Aqui, 'len' é uma função de escopo incorporado e está sempre disponível para uso.