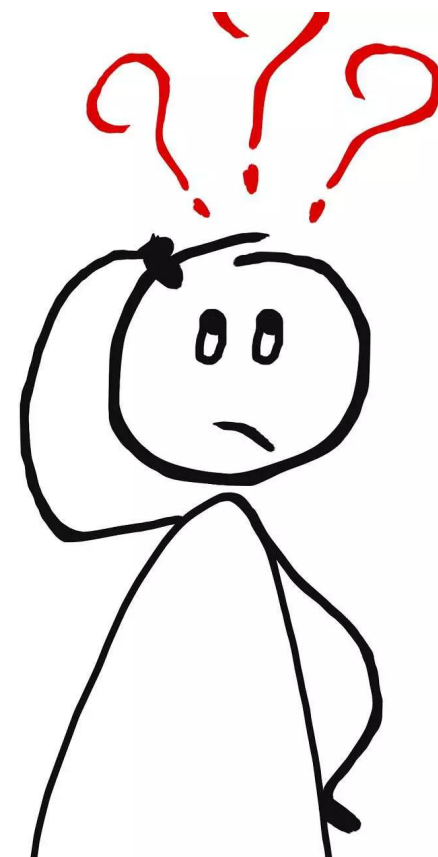


# PROGRAMAÇÃO I

Curso Técnico Subsequente em Informática  
Lucas Sampaio Leite



# O que são variáveis?



# Variáveis

- Programas manipulam dados (valores) que são armazenados em variáveis.
- Uma variável é uma posição na memória associada a um identificador (nome) que permite acessar e manipular dados.
- Em Python, não é necessário declarar variáveis explicitamente; elas são criadas automaticamente quando recebem um valor. O tipo da variável é definido dinamicamente com base no valor atribuído, podendo ser inteiro (int), real (float), booleano (bool) ou string (str), por exemplo.
- A atribuição de valores a variáveis é feita utilizando o operador de atribuição (=), que realiza as seguintes operações:
  - Atribui um valor a uma variável.
  - Armazena o valor na posição de memória associada à variável.

# Variáveis

```
main.py > ...  
1      x = 10  
2      nome = 'Lucas'  
3      y = 2.5  
4      sn = True
```

## Variáveis

- Em Python, as variáveis devem seguir algumas regras de nomenclatura conforme estabelecido pela PEP 8, que são as convenções oficiais de estilo para o código Python. Essas regras incluem:
  - Sequência de caracteres válidos: Um nome de variável pode ser composto por letras ( $a \rightarrow z$ ,  $A \rightarrow Z$ ), números ( $0 \rightarrow 9$ ) e o caractere de sublinhado (`_`). No entanto, o nome da variável deve sempre começar com uma letra ou um sublinhado.
  - Proibição de caracteres especiais: Caracteres especiais como acentos, cedilhas, espaços e símbolos como `$`, `#`, `@` não são permitidos. Apenas o caractere de sublinhado (`_`) é aceito.
  - Uso de letras minúsculas: Por convenção, os nomes das variáveis devem começar com letras minúsculas e, se necessário, usar o estilo `snake_case` (letras minúsculas separadas por sublinhados) para melhorar a legibilidade. Por exemplo, `idade_usuario` ou `nome_completo`.

# Variáveis

Descrição	É válido?	Exemplo
Iniciar por letra		
Iniciar por número		
Conter letras e números		
Conter underline		
Conter espaço		
Conter caracter especial		
Conter acentuação		

# Variáveis

Descrição	É válido?	Exemplo
Iniciar por letra	Sim	variavel_teste
Iniciar por número	Não	1variavel
Conter letras e números	Sim (desde que não inicie por número)	variavel1
Conter underline	Sim	_variavel_
Conter espaço	Não	variavel teste
Conter caracter especial	Não	variavel*teste
Conter acentuação	Sim (v3) / Não (v2)	variável

## Palavras reservadas da linguagem

- Palavras reservadas são palavras que têm um significado especial em uma linguagem de programação e não podem ser usadas como identificadores (nomes de variáveis, funções, classes, etc.).
- É importante também estar atento às palavras reservadas da linguagem, que não podem ser utilizadas para nomear variáveis:

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	



## Atribuição de valores

- Exemplo de atribuição:

```
nota_p1 = 7.5
```

- No exemplo, é criada uma variável de nome `nota_p1` que passa a armazenar o valor real 7.5.
- A partir do momento em que uma variável é criada, ela pode ser usada em expressões aritméticas e atribuídas a outras variáveis. Exemplo:

```
nota_p1 = 7.5  
nota_p2 = 8.0  
nota_final = (nota_p1 + nota_p2) / 2
```

## Variáveis e tipo de dados

- Em Python, é possível determinar o tipo de uma variável utilizando o interpretador. Para isso, basta usar a função `type()` e passar a variável como argumento dentro dos parênteses.

```
print(type(nota_p1))  
print(type(nota_final))  
print(type(10))  
print(type("IF Baiano"))  
print(type(["Lógica de Programação", 175, 82.5]))
```



```
<class 'float'>  
<class 'float'>  
<class 'int'>  
<class 'str'>  
<class 'list'>
```

# Constantes

- A regra de nomeação das constantes em Python segue um padrão semelhante ao das variáveis, com a diferença de que todas as letras são maiúsculas e as palavras são separadas por underline (\_).
- No entanto, devido à tipagem dinâmica de Python, o valor atribuído a uma constante pode ser alterado sem problemas, apesar da convenção de mantê-las imutáveis.

```
MINHA_CONSTANTE = 10  
print(MINHA_CONSTANTE)  
  
MINHA_CONSTANTE = 15  
print(MINHA_CONSTANTE)
```



```
10  
15
```

# Impressão de variáveis e constantes em Python

- Para imprimir variáveis e constantes em Python usa-se a função `print ()`.

- Exemplo: 

```
mensagem = "minha primeira mensagem em Python"
print(mensagem)
```



Minha primeira mensagem em Python

- Exemplo: 

```
nota_p1 = 7.5
nota_p2 = 8.0
nota_final = (nota_p1 + nota_p2) / 2
print("A média entre as notas", nota_p1, "e", nota_p2, "é", nota_final)
```

Como se pode ver, ao receber uma variável, o comando `print` exibe o valor da variável e não seu nome.



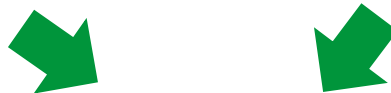
A média entre as notas 7.5 e 8.0 é 7.75

# Impressão de variáveis e constantes em Python

- É possível imprimir o resultado de uma expressão diretamente na função print.

```
PI = 3.14  
raio = 5.0  
area = PI * raio ** 2  
print(area)
```

```
PI = 3.14  
raio = 5.0  
print(PI * raio ** 2)
```



78.5

## Exercícios rápido

- Crie uma constante chamada TAXA\_JUROS com valor 0.05.
- Crie uma variável valor\_inicial e peça ao usuário para informar um número.
- Calcule  $\text{valor\_final} = \text{valor\_inicial} + (\text{valor\_inicial} * \text{TAXA\_JUROS})$ .
- Mostre:
  - O valor inicial
  - O valor final
  - O tipo de cada variável

## Exercícios rápido

- Peça ao usuário o preço de um produto (float).
- Peça a porcentagem de desconto (float).
- Calcule o valor com desconto:  $\text{preco\_final} = \text{preco} - (\text{preco} * \text{desconto} / 100)$
- Mostre o preço original, o desconto e o preço final formatados com duas casas decimais. Exemplo de saída:
  - Preço original: R\$ 100.00
  - Desconto: 10%
  - Preço final: R\$ 90.00

# PROGRAMAÇÃO I

Curso Técnico Subsequente em Informática  
Lucas Sampaio Leite

