

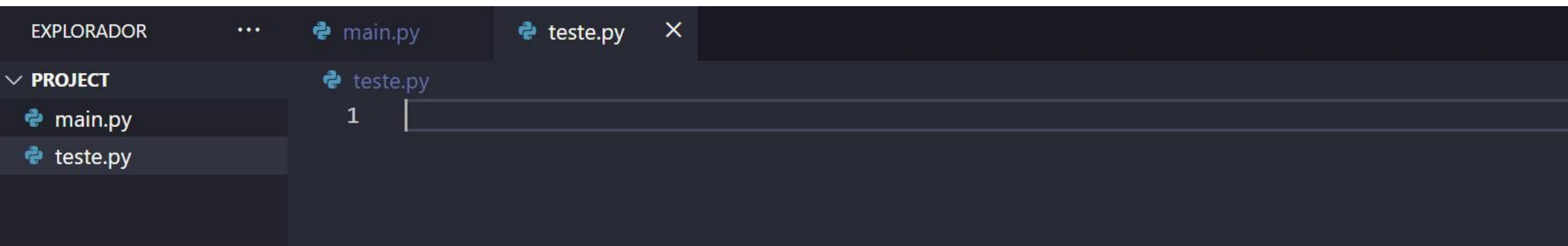
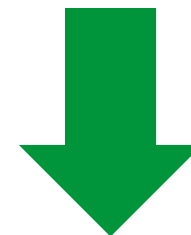
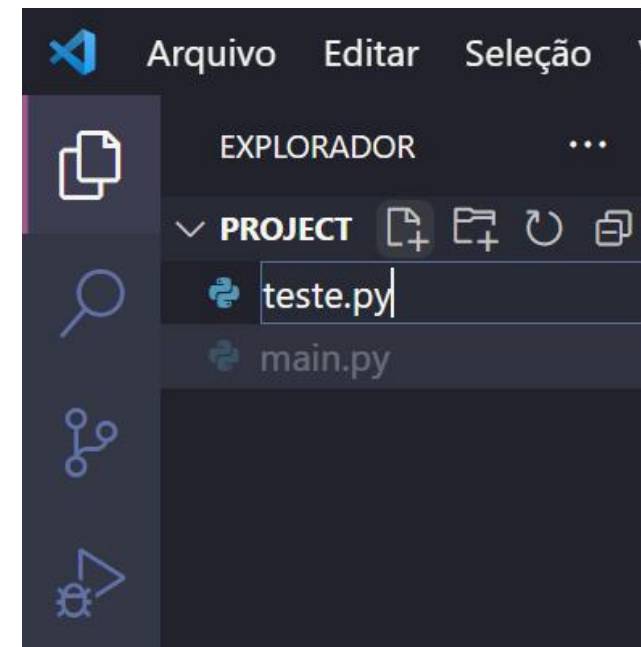
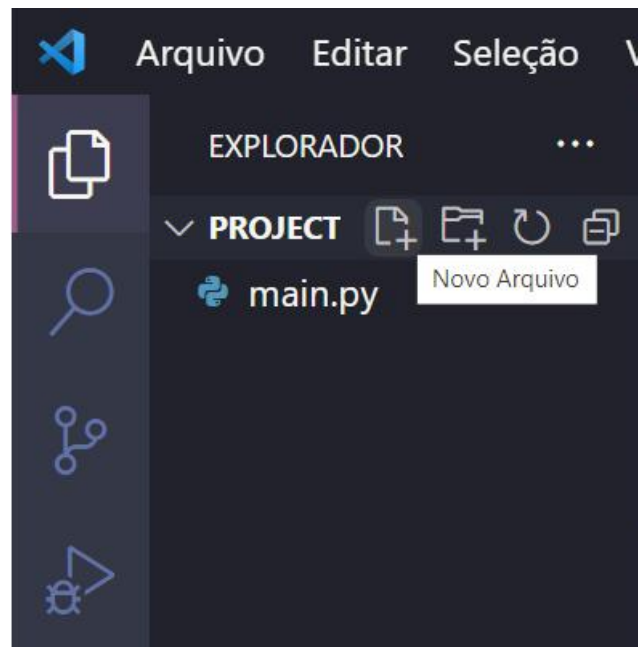
# LÓGICA E LINGUAGEM DE PROGRAMAÇÃO

Curso Técnico Integrado em Informática  
Lucas Sampaio Leite



# Criando um primeiro programa

- Criar um arquivo com a extensão .py
- Ex: teste.py



# Hello World em Python

- Em Python 3, é necessário o uso da função print():

```
print ('Hello World!')
```

- Em Python 2, a instrução de impressão não é função:

```
print 'Hello World!'
```

# Identação

- Python usa identação como delimitação de blocos. A seguir, podemos ver dois exemplos de identação, apresentando dois modos:

```
main.py > ...  
1  def f():  
2  x = 42  
3  return x  
4  print(f())  
5
```



```
main.py > ...  
1  def f():  
2      x = 42  
3      return x  
4  print(f())  
5
```



# Identação

- Se o bloco tem apenas um comando, pode-se escrever tudo em uma linha:

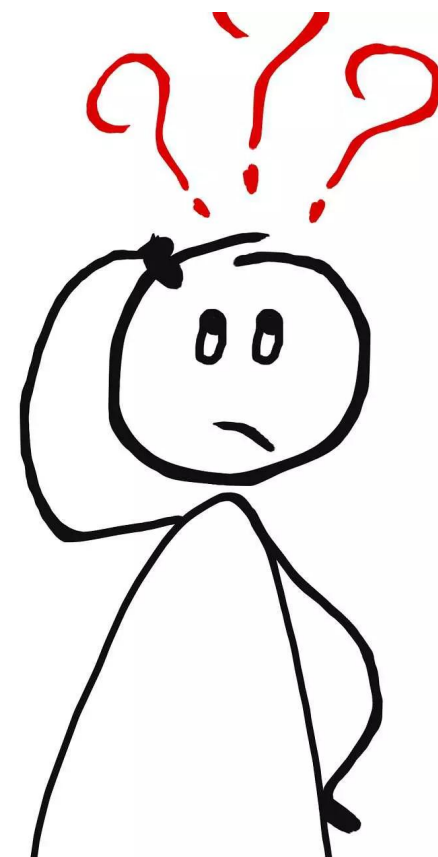
```
main.py > ...  
1  resposta = int(input("Digite uma opção: "))  
2  if resposta == 42: print('Opção válida!')  
3
```

# Comentários em Python

- Para colocar comentários no código, utiliza-se hashtag (comentários de uma linha), aspas simples e aspas duplas (comentários de várias linhas).
- Se o bloco tem apenas um comando, pode-se escrever tudo em uma linha.

```
main.py
1  # Exemplo de comentário em uma linha!
2
3  '''
4  Exemplo de
5  comentário utilizando
6  aspas simples.
7  '''
8
9  """
10 Exemplo de
11 comentário utilizando
12 aspas duplas.
13 """
```

# 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

- As variáveis em Python possuem um tipo, que é determinado automaticamente no momento em que a variável é criada, através de um comando de atribuição.
- Cada tipo de dado define os valores que a variável pode armazenar e as operações que podem ser realizadas sobre ela.
- Além disso, cada tipo ocupa uma quantidade específica de memória, dependendo de suas características.

## 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)
```



A média entre as notas 7.5 e 8.0 é 7.75

# 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

## Alguns exercícios rápidos...

1. Que tipo de dado seria usado para representar a idade de uma pessoa?
2. Que tipo de dado seria usado para representar a altura de uma pessoa?
3. Que tipo de dado seria usado para representar o sexo de uma pessoa?
4. Que tipo de dado seria usado para representar se uma pessoa está presente ou não?

## Alguns exercícios rápidos...

5. Considere as seguintes afirmativas:

- I. Uma variável é um elemento sintático do algoritmo que tem como finalidade armazenar um determinado valor.
- II. Uma constante é um elemento sintático do algoritmo que tem como finalidade armazenar mais de um determinado valor.
- III. O valor associado à variável depende do tipo de informação que se deseja manipular.
- IV. O valor associado à constante não depende do tipo de informação que se deseja manipular.

Alternativas:

- a) apenas o item I
- b) apenas o item III
- c) apenas os itens I e III
- d) apenas os itens II e IV
- e) os itens I, II e III.

# Operadores aritméticos

Operação	Operador
Adição	+
Subtração	-
Multiplicação	*
Divisão	/

# Operadores aritméticos

Operação	Operador
Exponenciação	<b>**</b>
Parte inteira do resultado da divisão	<b>//</b>
Módulo	<b>%</b>



# Ordem de precedência dos operadores aritméticos

Operador	Ordem de resolução na expressão
()	1º
**	2º
*, /, //, %	3º
+, -	4º

## Construção de sentenças

- $a + \frac{b}{b+c} \Rightarrow a + b/(b + c)$

- $b^2 + c^2 \Rightarrow b * b + c * c \Rightarrow b ** 2 + c ** 2$

- $\frac{x}{a+\frac{b}{c}} \Rightarrow x/(a + b/c)$

# Vamos exercitar?

a)  $5+3*2$

b)  $5/2$

c)  $5//3$

d)  $4*(5+2)**3$

e)  $420**339$

f)  $5\%2$

g)  $9**(1/2)$

h)  $8**(1/3)$

i)  $\text{pow}(2,5)$

## Vamos exercitar mais?

1. Como você pode consultar o tipo de uma variável em python?
  - a) Imprimindo o valor e determinando o tipo de baseado no valor exibido.
  - b) Usando a função `type()`.
  - c) Usando o valor em uma equação conhecida e imprimindo o valor resultante.
  - d) Olhando para a declaração da variável.
  
2. Indique como verdadeiro ou falso:
  - Duas variáveis definidas como “teste” e “Teste” são consideradas como idênticas.

## Vamos exercitar mais?

3. Classifique o itens abaixo como nome de variável válido (V) ou inválido (F).
- a) teclado\_lidinho
  - b) germa66
  - c) 1º\_lugar
  - d) PlayerID
  - e) int
  - f) class

## Vamos exercitar mais?

4. Faça um Programa que dado o número de horas trabalhadas e o valor da hora, calcule e mostre o total do salário no referido mês.
5. Você estudou muito pra sua prova de lógica de programação e conseguiu terminar a prova em 1 hora e 34 minutos. Faça um programa que calcula e exibe o tempo de prova decorrido, em minutos e em segundos.
6. Faça um Programa que converta metros para centímetros.

# Dúvidas



# LÓGICA E LINGUAGEM DE PROGRAMAÇÃO

Curso Técnico Integrado em Informática  
Lucas Sampaio Leite

