

# Desenvolvimento de Sistemas



Recife  
2024

# Fundamentos Python

1. Criado pelo **Holandês Guido Van Rossum**, em 1989. O objetivo de Rossum era criar uma linguagem mais flexível e acessível para programadores de todos os níveis;
2. **Tipagem dinâmica** (interpretador infere o tipo dos dados de uma variável). O que significa que você não precisa declarar o tipo das variáveis explicitamente ;
3. Sua logo são duas cobras **Piton**;
4. Mas, o nome Python surgiu em homenagem a série: **Monty Python Flying Circus** (programa de comédia britânico).



# Principais Vantagens e Características do Python



1. **Simples, Fácil e Intuitivo:** A sintaxe de Python é clara e direta, tornando o processo de aprendizado mais acessível tanto para iniciantes quanto para programadores experientes.
2. **Compatível com Web, Aplicativos e Mobile:** O Python é versátil, sendo utilizado em diferentes áreas, como desenvolvimento de aplicações web, apps móveis e até sistemas desktop.
3. **Funciona em Várias Plataformas:** Python pode ser executado em diversos dispositivos, como TV, celulares, computadores e até mesmo sistemas embarcados.
4. **Código Aberto:** Sendo uma linguagem de código aberto, Python permite que qualquer pessoa contribua com o desenvolvimento, além de ser totalmente gratuito.



# Principais Vantagens e Características do Python



5. **Organização Lógica:** Python organiza o fluxo de execução de forma sequencial, facilitando a compreensão de como o código será processado no sistema ou na web.
6. **Orientada a Objetos:** Python organiza o código em objetos, que podem ter dados e funções. Isso facilita a reutilização e organização do código. Com herança e encapsulamento, é possível criar estruturas mais simples e fáceis de manter.
7. **"Baterias Incluídas":** Python já vem com uma vasta coleção de bibliotecas e ferramentas essenciais no pacote padrão, permitindo que você comece a desenvolver imediatamente, como:
  - Desenvolvimento de telas para aplicativos móveis;
  - Criação de jogos;
  - Desenvolvimento de interfaces para sistemas e aplicações complexas.



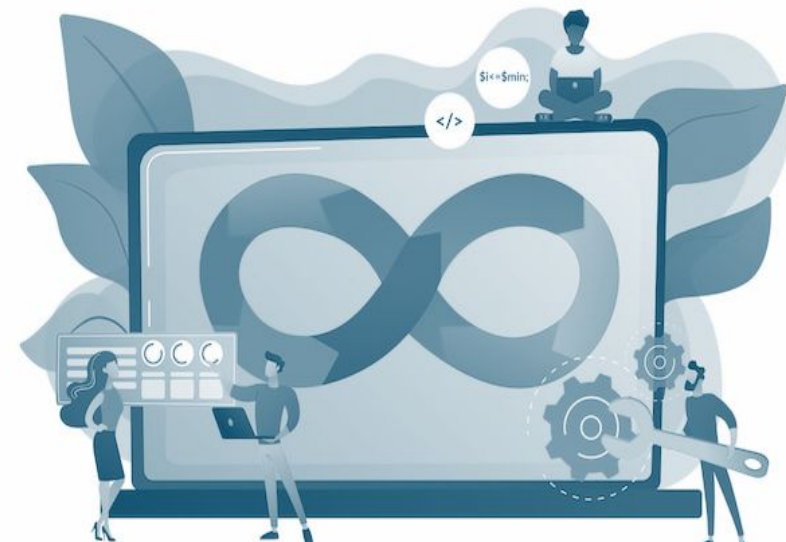
# Principais Áreas

- 1. Desenvolvimento Web:** Frameworks como Django e Flask facilitam a criação de sites e aplicações web.
- 2. Ciência de Dados:** Python é a linguagem preferida para análise de dados, machine learning e big data, com bibliotecas como Pandas, NumPy e TensorFlow.
- 3. Automação:** Python é excelente para automação de tarefas repetitivas, como scripts de sistema e testes automatizados.
- 4. Inteligência Artificial e Machine Learning:** Com bibliotecas como Keras e Scikit-learn, Python é muito usado para IA, aprendizado de máquina e redes neurais.



# Principais Áreas

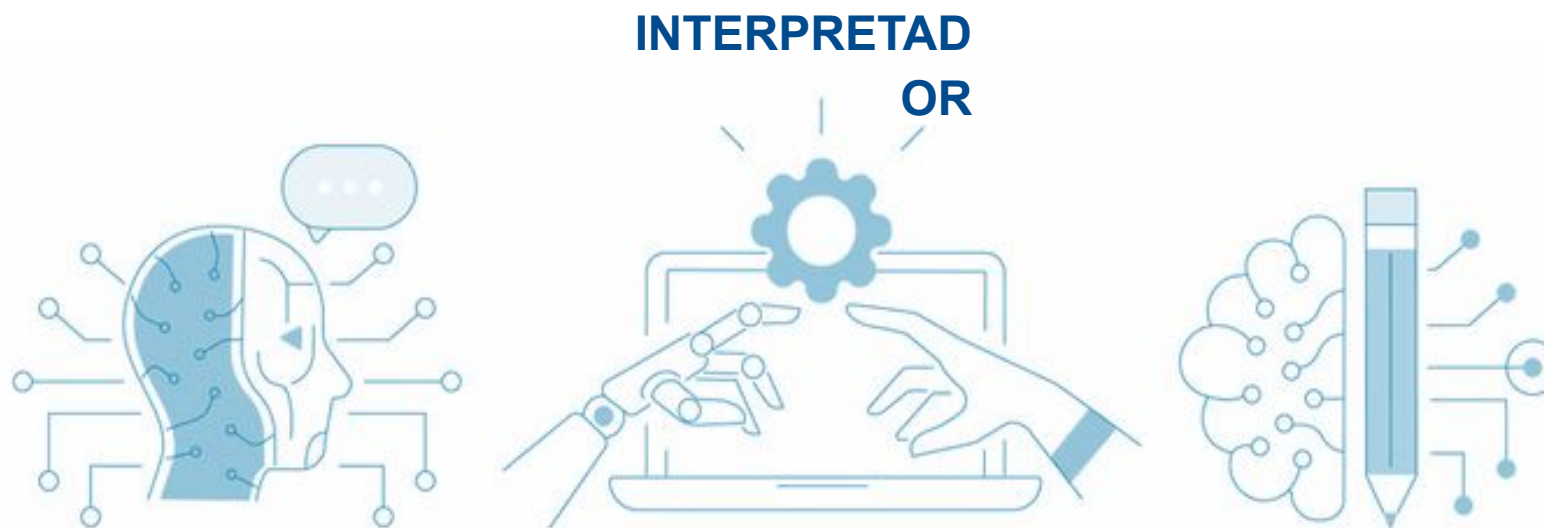
5. **Desenvolvimento de Jogos:** Pygame é uma biblioteca popular para criar jogos 2D com Python.
6. **Desenvolvimento de Aplicações Desktop:** Python permite criar interfaces gráficas com Tkinter, PyQt, entre outros.
7. **Sistemas Embarcados:** Python é utilizado em plataformas como Raspberry Pi para controlar hardware em sistemas embarcados.
8. **Desenvolvimento Científico:** Muito usado em pesquisa Científica para simulações, modelagens e cálculos matemáticos com SciPy e Matplotlib.











# O que é e como funciona o Python?

**Python** é uma linguagem de programação interpretada, o que significa que o código é executado linha por linha por um interpretador. Ao escrever um programa em Python, você salva o código em um arquivo com extensão .py, e o interpretador Python executa esse código diretamente, sem a necessidade de compilação prévia.

A execução pode ocorrer diretamente no terminal (linha de comando), em ambientes de desenvolvimento integrados (IDEs) ou através de scripts. O Python também inclui uma vasta biblioteca padrão, permitindo que você execute muitas tarefas sem precisar instalar pacotes adicionais.



# Principais empresas que usam o Python

1. **Google:** Usa Python em vários serviços, como sistemas de busca e automação. 
2. **Netflix:** Python é usado para automação e análise de dados. 
3. **Instagram:** Back-end construído com Django, um framework Python.  Instagram
4. **Spotify:** Utiliza Python para análise de dados e recomendações  Spotify®
5. **Dropbox:** Usa Python no cliente e servidor.  Dropbox
6. **Amazon:** Python é usado em automação e gerenciamento de dados na nuvem 
7. **Facebook:** Usa Python para ferramentas internas de infraestrutura 
8. **NASA:** Python é usado para simulações e controle de missões 



# Conhecendo os Ambientes de Desenvolvimento



Caso você tenha preferências por Ambientes de Desenvolvimento Integrado mais robustos, você pode optar por:

**Thonny**  
Python IDE para iniciantes

**Para baixar acesse:**  
<https://thonny.org/>



PyCharm

**Para baixar acesse:**  
<https://www.jetbrains.com/pycharm/download/#section=windows>



**Para baixar acesse:** <https://www.spyder-ide.org/>

# Python Web Browser

Caso você tenha preferências por Ambientes online, usando o Web Browser, pode optar por:



Google Colab

**Para acessar:**

<https://colab.research.google.com/>



**Para acessar:** <https://replit.com/languages/python3>

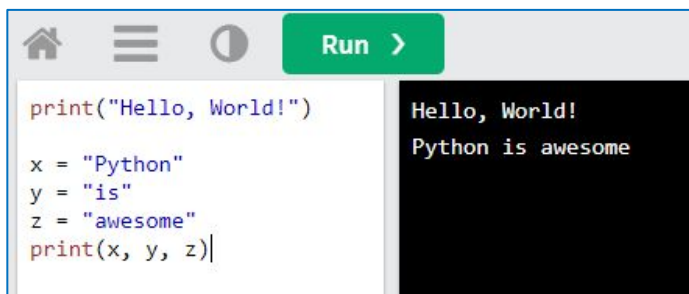


**Para acessar:**

<https://www.programiz.com/python-programming/online-compiler/>

# Python Web Browser

Caso você tenha preferências por Ambientes online, usando o Web Browser, pode optar por:



**Para acessar:**

[https://www.w3schools.com/python/trypython.asp?filename=demo\\_compiler](https://www.w3schools.com/python/trypython.asp?filename=demo_compiler)



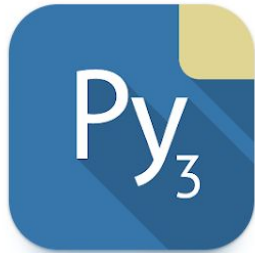
**Para acessar:**

[https://www.onlinegdb.com/online\\_python\\_interpreter](https://www.onlinegdb.com/online_python_interpreter)

Caso você tenha preferências por Ambientes mobile,  
pode optar por:



**Para acessar:** <https://replit.com/mobile>



**Pydroid  
3**

**Para acessar:**  
<https://play.google.com/store/apps/details?id=ru.iiec.pydroid3>



**QPython  
3L**

**Para acessar:**  
[https://www.onlinegdb.com/online\\_python\\_interpreter](https://www.onlinegdb.com/online_python_interpreter)



Google Colab

O Colaboratory ou também chamado de “Colab” é uma ferramenta na nuvem que permite realizar a escrita e a execução de códigos em python.

É destinado para o aprendizado de máquina, análise de dados e para a educação (áreas científicas do Google).



O Pycharm é uma ferramenta com recursos avançados de desenvolvimento. Suporta a linguagem Python e Javascript. Tem como diferencial disponibilizar recursos como:

- Depuração e testes;
- Integração com Git;
- Ferramentas de banco de dados;
- Frameworks web como Django;
- Console interativo e etc.

# Variáveis





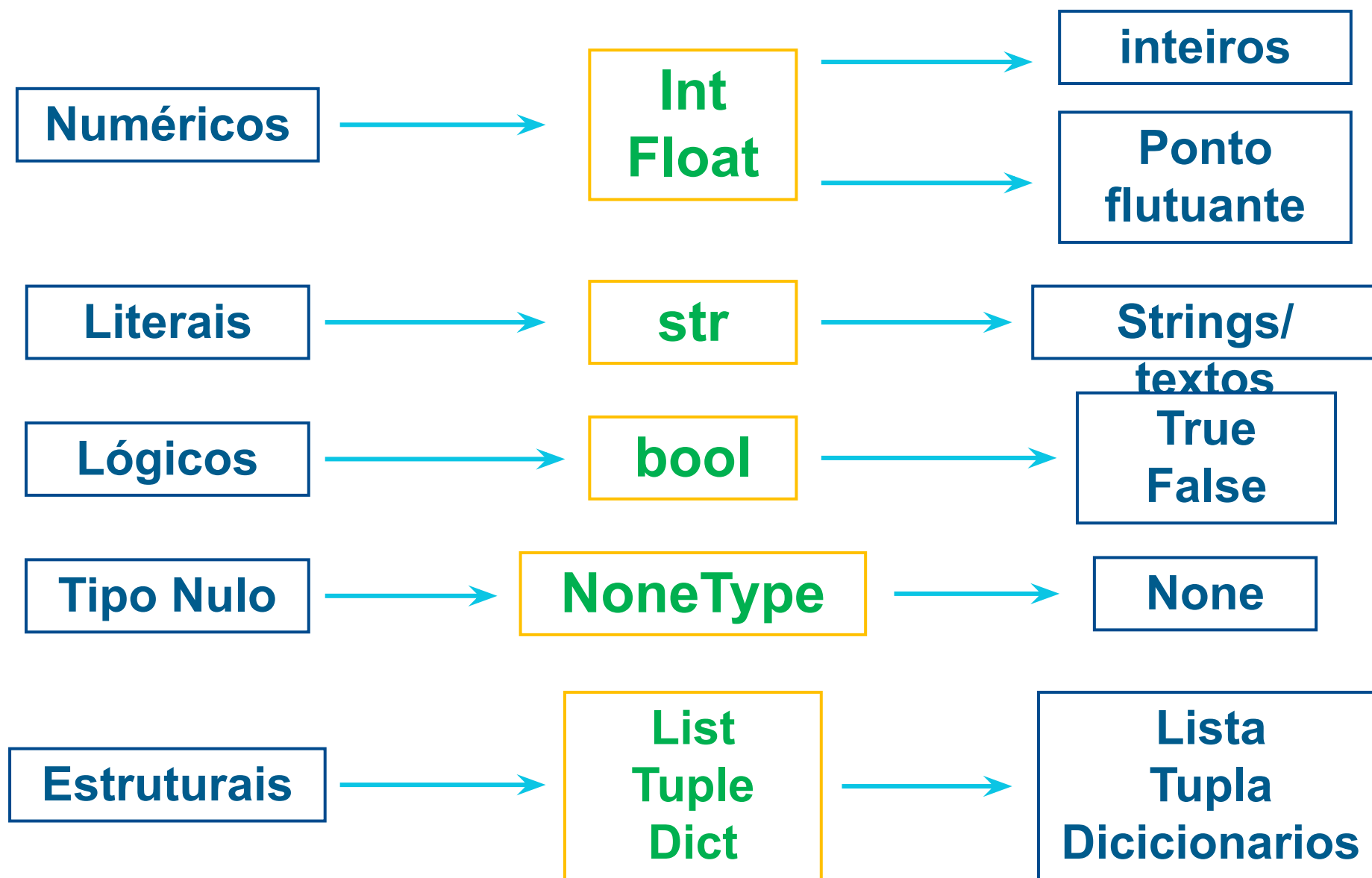
Para declarar alguma variável em Python com um valor definido, não é necessário especificar o tipo primitivo dessa variável. Porque o Python tem uma tipagem dinâmica. A tipagem dinâmica significa que o interpretador consegue automaticamente inferir o tipo de dado que a variável recebe, sem a necessidade do programador coloque o tipo primitivo.

Exemplo:

```
nome = 'Paulo'  
idade = 25
```

Dados declarados pelo programador

# Tipos de Dados Primitivos:



# Primeiros comandos

1. O Python ele aceita você escrever tanto com aspas simples como com aspas duplas. Porém, comunidade Python recomenda fortemente que você padronize o tipo de aspas usadas nos seus códigos, a mais utilizada são aspas simples.
2. Palavras reservadas em linguagens de programação, não podem ser usadas para criar nomes de variáveis;
3. Para **funções** usamos os (parênteses);
4. A primeira função que aprendemos é o: **print('Olá Mundo!');**
5. Quando usamos **números** usamos “apenas” eles. Ex.:  $2 + 2 = 4$
6. Podemos juntar dois números considerando textos.  
Ex.: `print('2' + '2') = 22`  
pois nesse caso não estamos somando e sim “**concatenando**”;

```
saudacao = "Olá, " + "mundo!"  
print(saudacao)  
# Saída: Olá, mundo!
```

**Concatenação** é a operação de unir duas ou mais cadeias de texto em uma única cadeia.

# Conhecendo o NoneType



O tipo none pertence a classe noneType, é utilizado para indicar que um valor é nulo. Ou seja, que uma variável não tem nenhum valor atribuído.

Não confunda, none não é uma string vazia, não é uma variável inteira com o número zero e nem um booleano do tipo False.

(Trybe, 2022): “Podemos atribuir o valor None a qualquer variável para indicar que ela não tem um valor específico.”

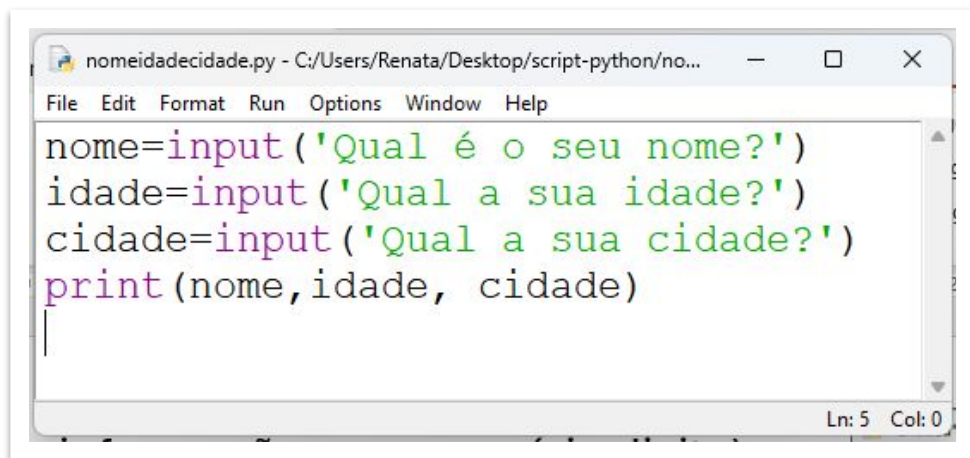
# **Conhecendo os comandos Entrada e Saída de Dados**



# Comando de Entrada e Saída

- Para você receber um valor como entrada, utiliza-se o comando `input`, mas, lembre-se de criar uma variável para receber o valor.
- Tudo que é digitado no `input` é uma `string`.
- Recebendo um número inteiro, é necessário fazer a conversão.
- Para mostrar na tela, chamamos o comando `print`.

```
nome = str(input('Digite seu nome: '))
```



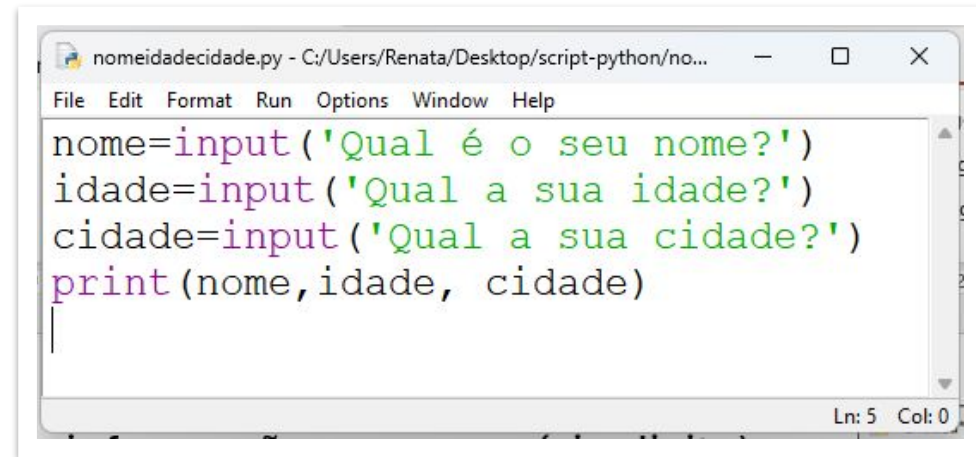
```
nomeidadecidade.py - C:/Users/Renata/Desktop/script-python/no...
File Edit Format Run Options Window Help
nome=input('Qual é o seu nome?')
idade=input('Qual a sua idade?')
cidade=input('Qual a sua cidade?')
print(nome, idade, cidade)
Ln: 5 Col: 0
```

```
idade = int(input('Digite sua idade: '))
print('A sua idade é: ', idade)
```

# Ainda sobre input

```
Renata Recife
>> nome=input('Qual é o seu nome?')
Qual é o seu nome?Renata
>> idade=input('Qual a sua idade?')
Qual a sua idade?38
>> cidade=input('Qual a sua cidade?')
Qual a sua cidade?Recife
>> print(nome,idade, cidade)
Renata 38 Recife
>> print(cidade+ ' ' + nome)
Recife Renata
>> print('Nome: ' + nome + ' Idade: ' +
idade + ' Cidade: ' + cidade)
Nome: Renata Idade: 38 Cidade: Recife
>>
```

- Use **vírgula** para separar e automaticamente adicionar espaços em funções como print().
- Use + para concatenar strings ou somar números.



```
nomeidadecidade.py - C:/Users/Renata/Desktop/script-python/no...
File Edit Format Run Options Window Help
nome=input('Qual é o seu nome?')
idade=input('Qual a sua idade?')
cidade=input('Qual a sua cidade?')
print(nome,idade, cidade)
Ln: 5 Col: 0
```

```
>>
===== RESTART: C:\Users\Renata\Desktop\script-python\nomeidadecidade.py =====
Qual é o seu nome?Renata
Qual a sua idade?38
Qual a sua cidade?Recife
Renata 38 Recife
>>
```

# Comandos de Saída





# Comando de Saída

Formas de exibir na tela

```
nome = input('Digite seu nome: ')
```

```
print('Seu nome é: ', nome) #vírgula
```

```
print('Seu nome é: ' + nome) #concatenação
```

```
print(f'Seu nome é: {nome}') #f-string
```

```
print('Seu nome é: {}'.format(nome)) #format
```

# Exercitando

```
num1 = float(input('Diga um número:'))
num2 = float(input('Diga outro número:'))
soma = num1 + num2
#print('A soma dos dois números é: ', soma)
print('A soma entre', num1, 'e', num2, 'é:', soma)
print('A soma entre {} e {} é: {}'.format(*args: num1,num2,soma))
print('A soma entre {1} e {0} é: {2}'.format(*args: num1,num2,soma))
```

```
print('A soma entre {1} e {0} é: {2}'.format(*args: num1,num2,soma))
print('A soma entre {} e {} é: {}'.format(*args: num1,num2,soma))
```

```
Diga um número:5.3
Diga outro número:4.7
A soma entre 5.3 e 4.7 é: 10.0
A soma entre 5.3 e 4.7 é: 10.0
A soma entre 4.7 e 5.3 é: 10.0
```

```
A soma entre 4.7 e 5.3 é: 10.0
```

# Diferenças entre comentários e docString



Comentários são linhas de textos que são desconsideradas pelo interpretador.

`#Aqui é um comentário`

Já o docString não é reconhecido como um comentário, pois ele lido pelo interpretador. Ele é considerado como um documento/ou bloco de notas.

`““`

`DOCSTRING  
ASPAS SIMPLES`

`””`

# Como descobrir o tipo de dado

```
num = 15
```

```
print(num)
```

```
print(type(num))
```

Resultado: <class 'int'>

Como conferir se esse número é inteiro?

```
num1 = input('Diga um número:')
print(type(num1))
```

```
Diga um número:2
<class 'str'>
```

```
num1 = int(input('Diga um número:'))
print(type(num1))
```

```
Diga um número:2
<class 'int'>
```

```
num1 = float(input('Diga um número:'))
print(num1)
print(type(num1))
```

```
Diga um número:4
4.0
<class 'float'>
```

```
num1 = bool(input('Diga um número:'))
print(num1)
print(type(num1))
```

```
Diga um número:4
True
<class 'bool'>
```

```
Diga um número:
False
<class 'bool'>
```

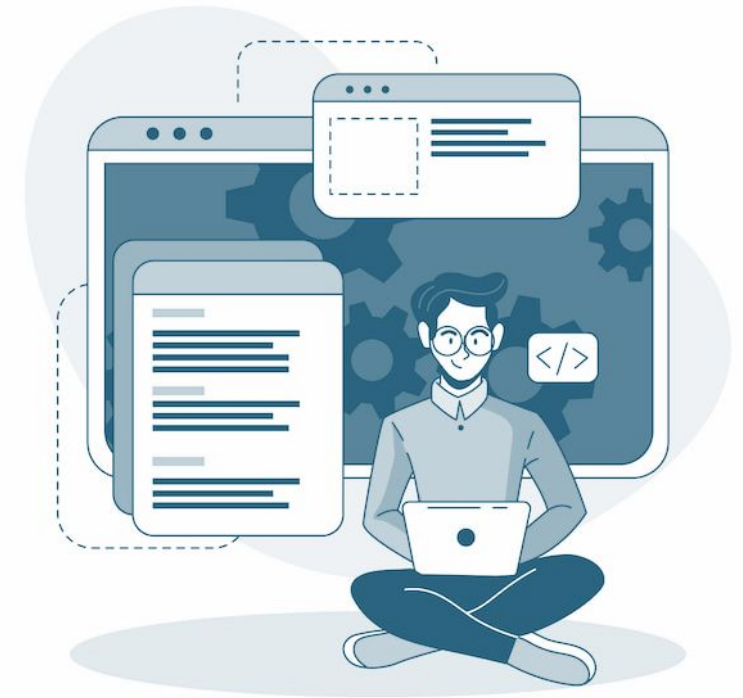
# Conversão de Tipo



Chamados de : coerção, type conversion, typecasting ou conversão, são funções de conversões para converter um dado em outro tipo.

Existem dois tipos de conversão de tipo em Python:

- Conversão de tipo implícita;
- Conversão de tipo explícita.



# Conversão de tipo implícita

O interpretador converte automaticamente sem qualquer envolvimento do usuário.

Exemplo:

num = 12

<class 'int'>

num = 15.6

print(type(num))

<class 'float'>

# Conversão de tipo **explícita**

A conversão é realizada manualmente pelo o programador.

Exemplo:

```
valor = '145' → <class 'str'>  
print(int(valor))
```

```
Resultado: 145 → <class 'int'>
```



# EXERCÍCIOS



# Exercício

Crie um programa que calcule o índice de massa corporal de uma pessoa.

## RESULTADO ESPERADO

```
Informe seu nome: Maria
Informe sua altura: 1.57
Informe seu peso: 57
----- resultado do IMC -----
Olá Maria
Sua altura é de: 1.57 seu peso é: 57.0
IMC: 23.124670372023203
```

# Exemplo

Crie um programa que calcule o índice de massa corporal de uma pessoa.

```
nome = str(input('Informe seu nome: ')) #comentário
altura = float(input('Informe sua altura: '))
peso = float(input('Informe seu peso: '))
imc = peso/(altura *altura)
print('-'*30, 'resultado do IMC', '-'*30)
print('Olá', nome)
print('Sua altura é de: ', altura, 'seu peso é: ', peso)
print('IMC: ', imc)
```