

PROGRAMAÇÃO EM PYTHON



PROGRAMAÇÃO EM PYTHON



Por meio de uma parceria com o SENAI-SP, a Prefeitura de Santana de Parnaíba está disponibilizando um curso focado em Python, com o objetivo de preparar os alunos para o desenvolvimento de programas e aplicações.



PROGRAMAÇÃO EM PYTHON



Plano de Aula

- Conteúdo:**
- Linguagem Python;
 - Instalação do Python;
 - Programação em Python;
 - Variáveis;
 - Instruções de I/O;
 - Built-Ins;
 - Atividades;

Início:

As informações deste conteúdo visam compreender o conteúdo do curso.



PROGRAMAÇÃO EM PYTHON



Objetivo

Capacitar profissionais para desenvolver aplicações em linguagem Python, por meio de técnicas de programação, seguindo boas práticas, procedimentos e normas.

PROGRAMAÇÃO EM PYTHON



Linguagem Python



Python é uma **linguagem de programação de alto nível**, conhecida por sua **sintaxe clara e legibilidade**, o **que facilita tanto o aprendizado para iniciantes quanto a manutenção de projetos complexos**.

Desenvolvida inicialmente por Guido van Rossum, Python é uma linguagem interpretada, dinâmica e multi-paradigma, permitindo a utilização de programação orientada a objetos, funcional e imperativa.

Essa flexibilidade a torna uma escolha popular para diversas áreas, como desenvolvimento web, ciência de dados, automação e inteligência artificial.

Site: <https://www.python.org/>

PROGRAMAÇÃO EM PYTHON



Linguagem Python



Programação em Python

A programação em Python pode ser realizada em uma ampla variedade de ambientes e plataformas. Entre as possibilidades, destacam-se:

- **Ambientes de Desenvolvimento Integrados (IDEs):** Ferramentas como [PyCharm](#), [VS Code](#) e [Spyder](#) oferecem recursos avançados para escrever, depurar e gerenciar projetos em Python.
- **Notebooks Interativos:** Plataformas como [Jupyter Notebook](#) e [Google Colab](#) permitem a execução interativa de código, o que é especialmente útil para análises de dados e prototipagem.

PROGRAMAÇÃO EM PYTHON



Linguagem Python



- **Aplicações Web:** Com frameworks como **Django** e **Flask**, Python se destaca no desenvolvimento de sites e serviços web.
- **Scripts e Automação:** De simples scripts para automatizar tarefas rotineiras até aplicações robustas de automação de processos empresariais.
- **Ciência de Dados e Machine Learning:** Bibliotecas como NumPy, Pandas, Matplotlib e Scikit-learn fazem de Python uma das linguagens preferidas para análise e visualização de dados.

PROGRAMAÇÃO EM PYTHON



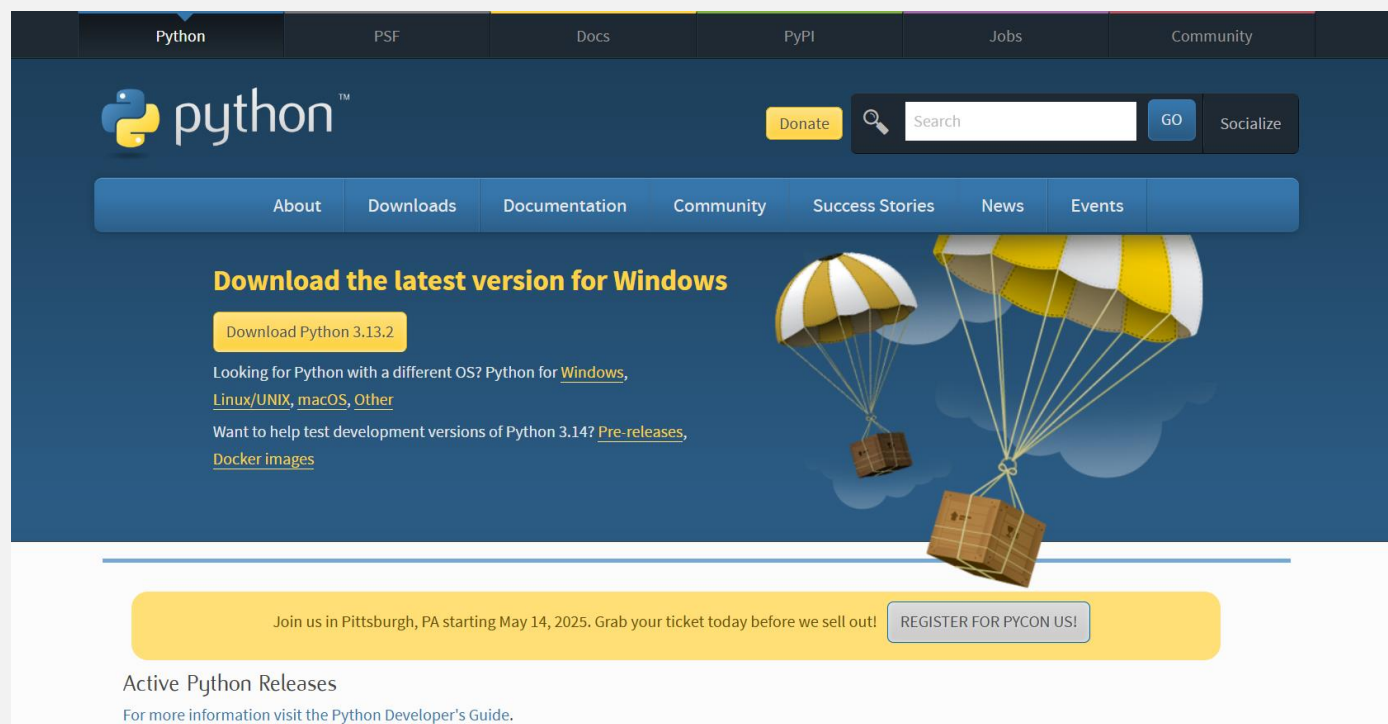
Linguagem Python



Para rodar Python no computador

Para rodar Python no seu computador, você precisa basicamente dos seguintes elementos:

Interpretador Python: Baixe e instale o Python a partir do site oficial.



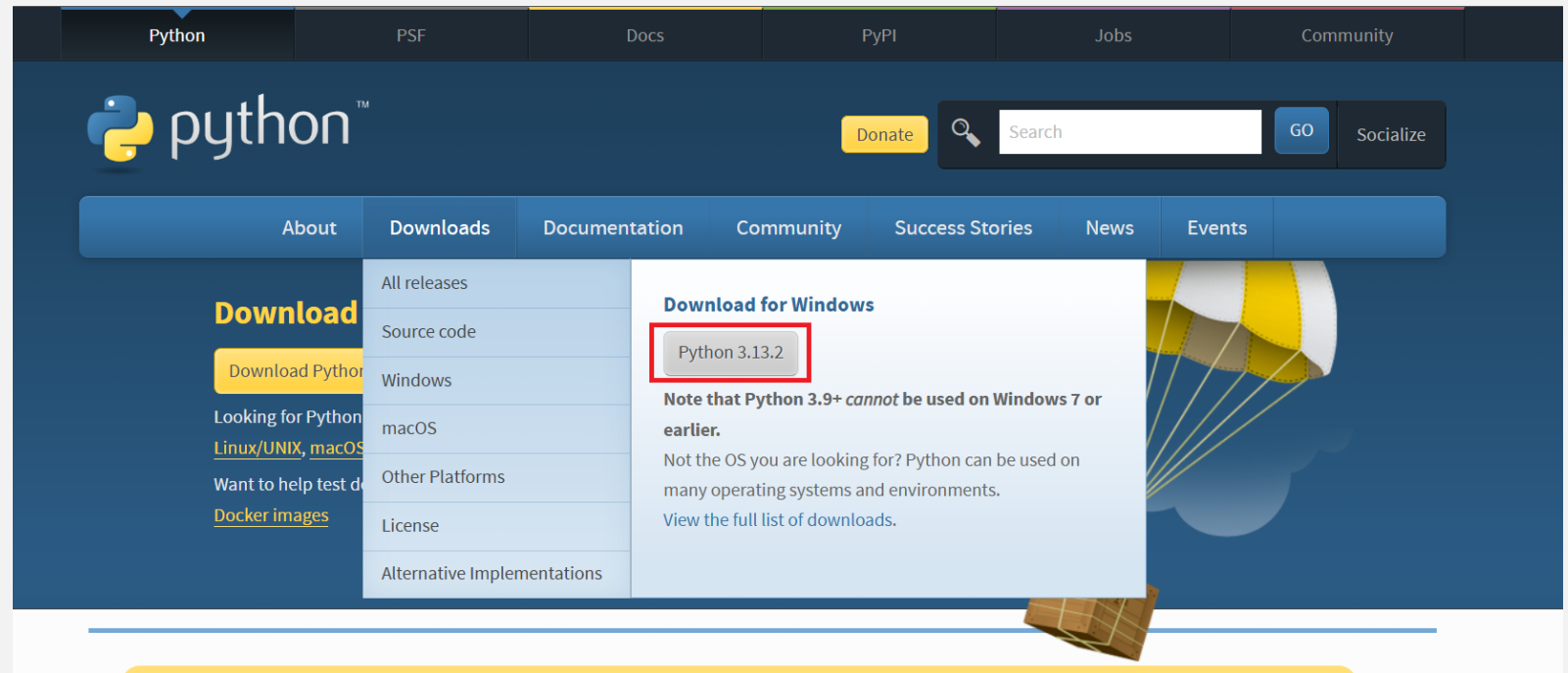
PROGRAMAÇÃO EM PYTHON



Linguagem Python



Vá em Downloads



Baixe o arquivo conforme ilustrado no exemplo acima.
Execute o instalador.

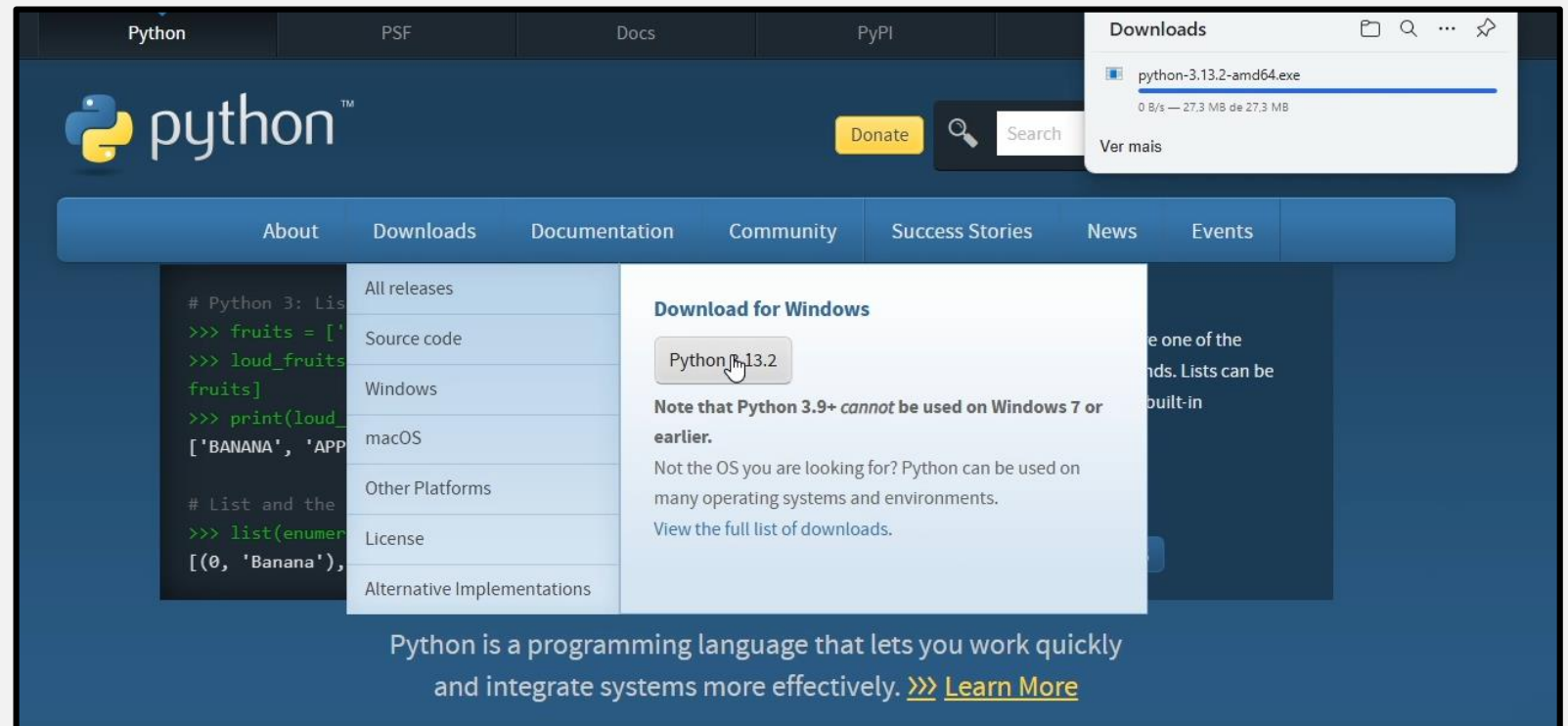
PROGRAMAÇÃO EM PYTHON



Linguagem Python



Executando o Instalador:



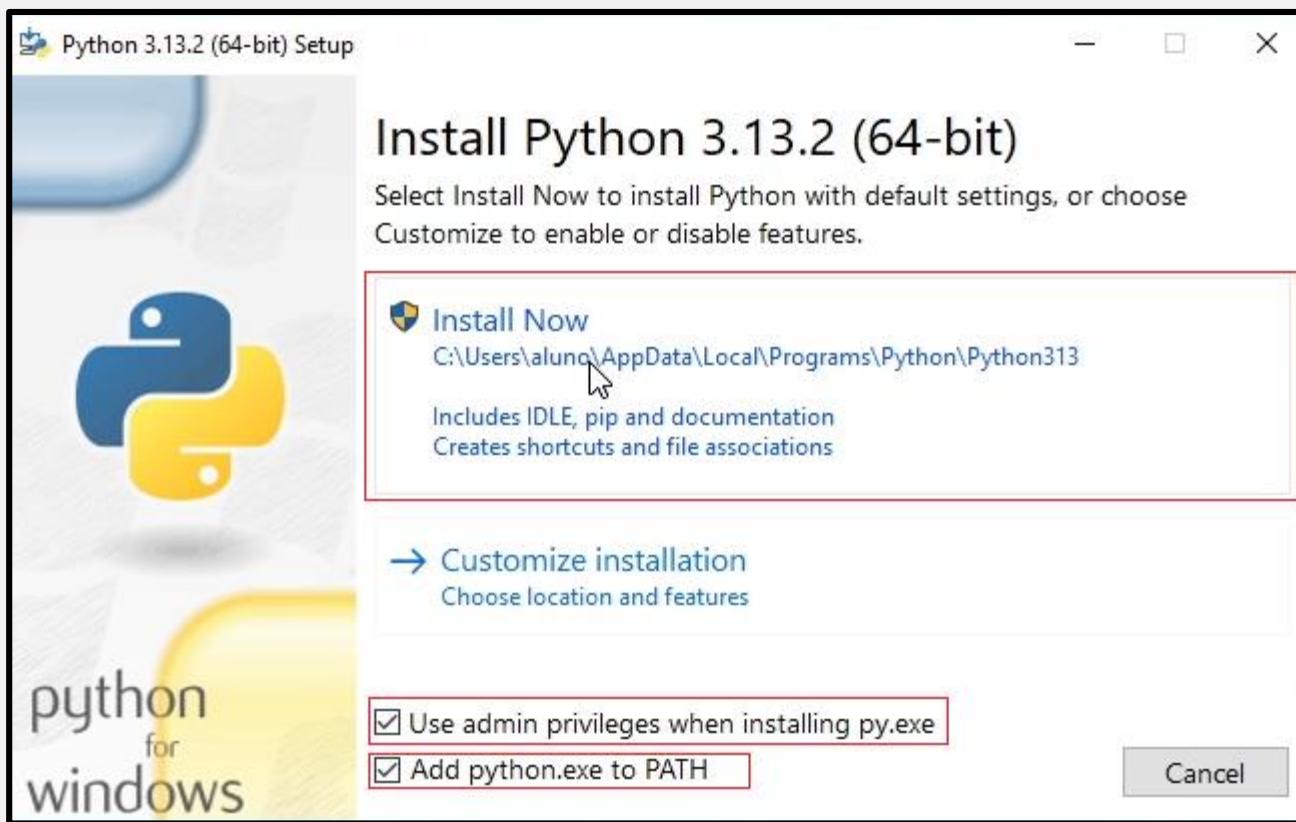
PROGRAMAÇÃO EM PYTHON



Linguagem Python



Procedimento de instalação:



Na parte inferior da tela, marque ambas as caixas e clique em **Install Now**

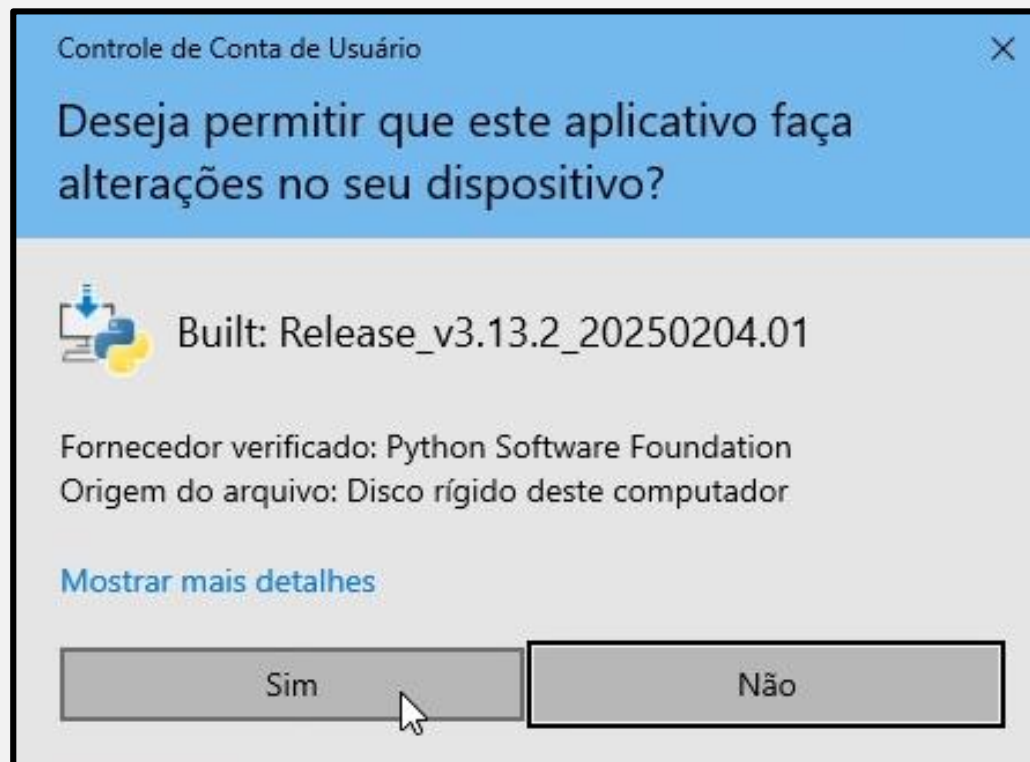
PROGRAMAÇÃO EM PYTHON



Linguagem Python



Procedimento de instalação:



Se o computador solicitar, clique em "**Sim**" para autorizar a instalação da linguagem. Essa autorização pode ser necessária mais de uma vez.

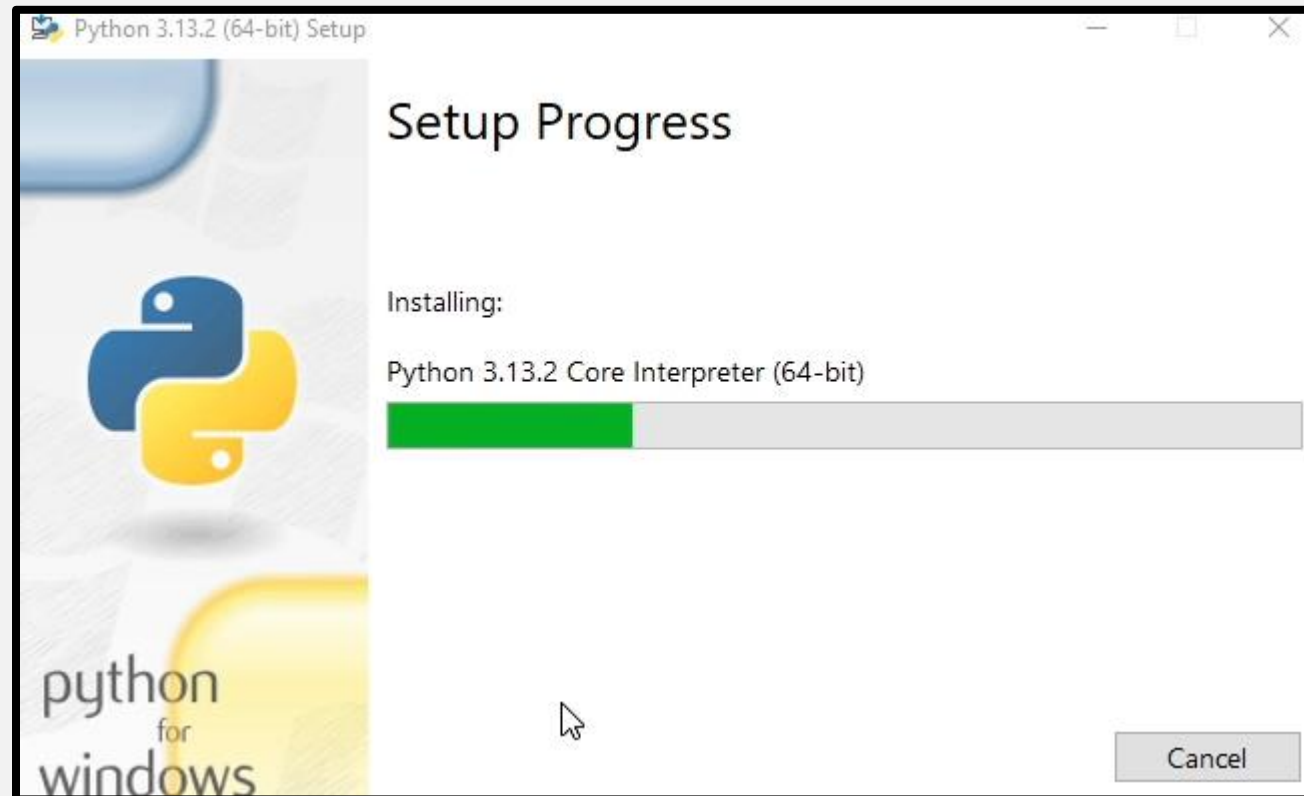
PROGRAMAÇÃO EM PYTHON



Linguagem Python



Procedimento de instalação:



A instalação ocorrerá normalmente.

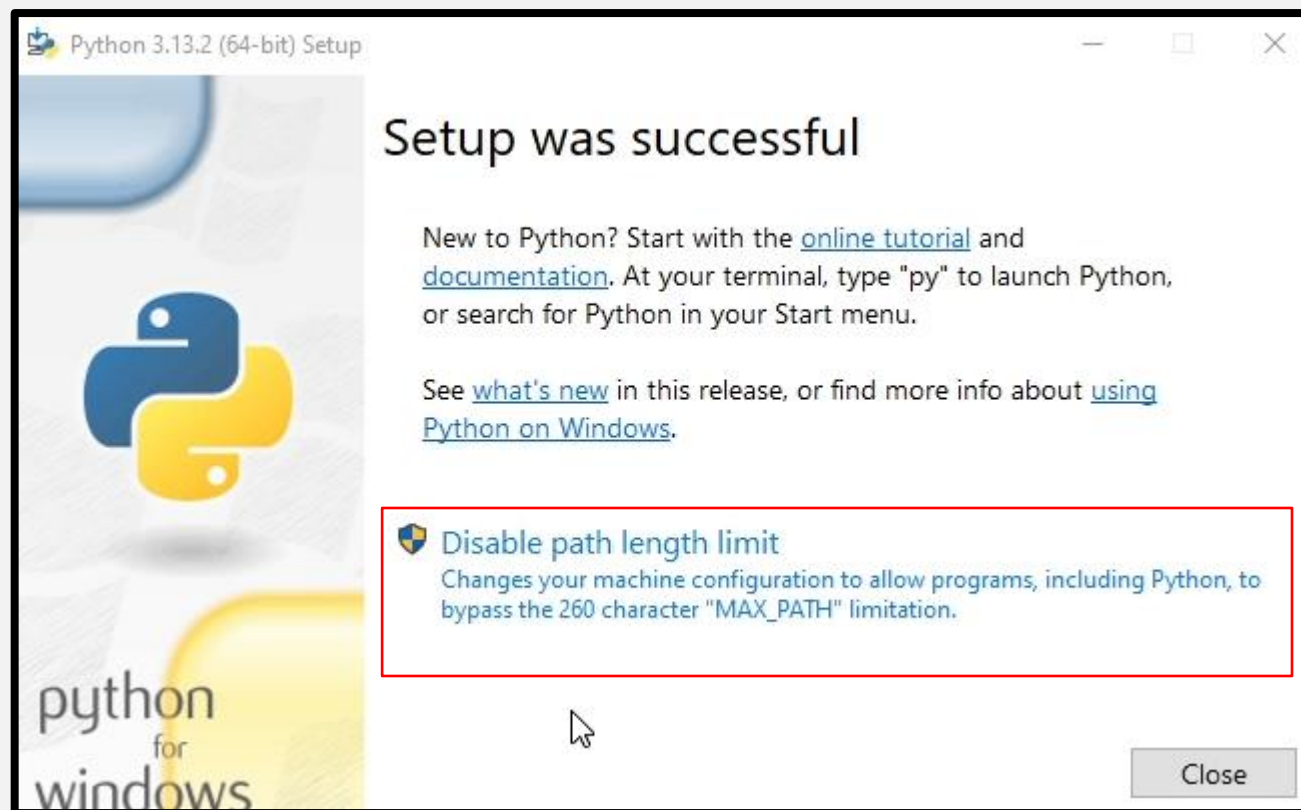
PROGRAMAÇÃO EM PYTHON



Linguagem Python



Procedimento de instalação:



Clique na opção “**Disable path length limit**”.

Opção remove a limitação de 260 caracteres para os caminhos de arquivos no Windows, permitindo que nomes e diretórios mais longos sejam usados sem gerar erros.

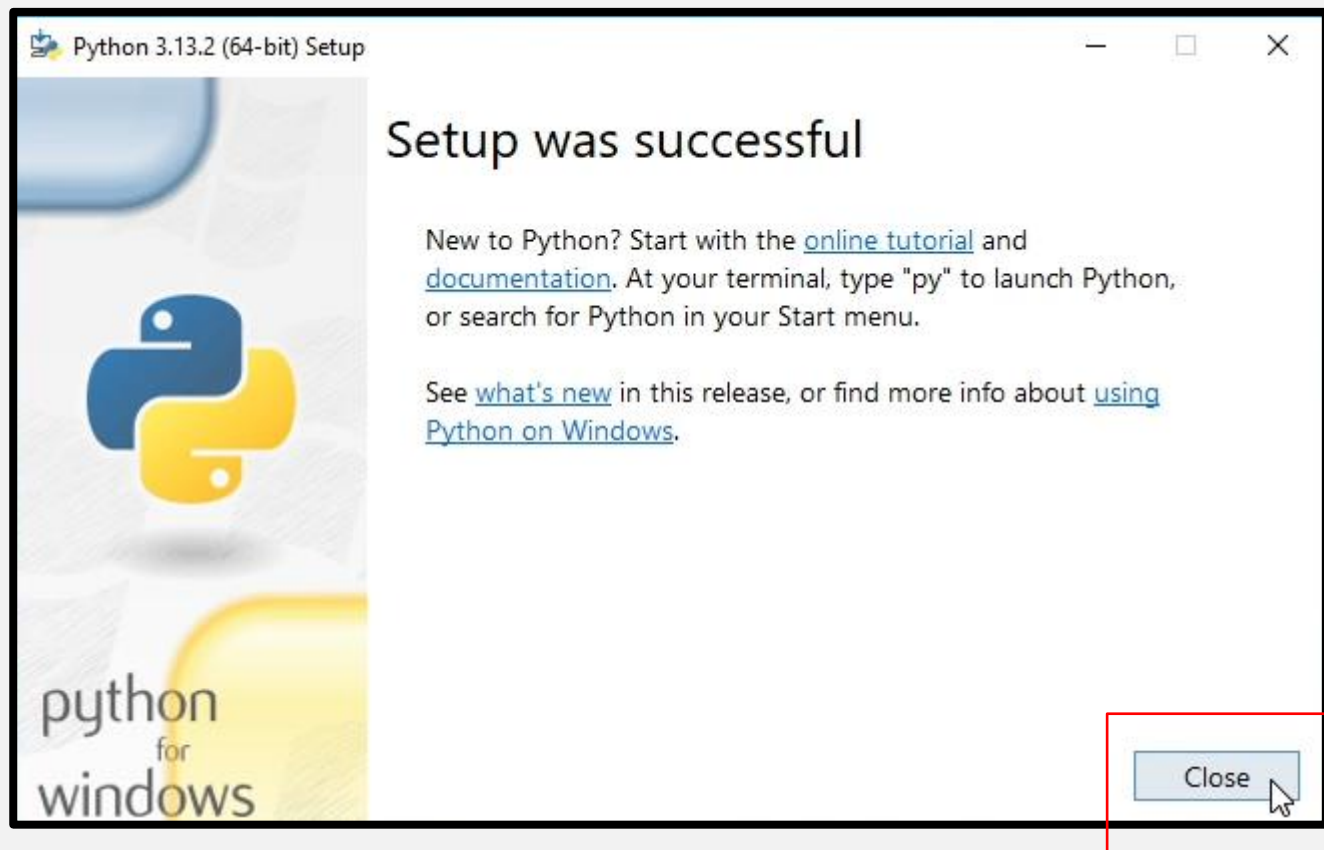
PROGRAMAÇÃO EM PYTHON



Linguagem Python



Procedimento de instalação:



Clique na opção “**Close**”.

Assim está instalado a linguagem de programação Python no seu computador.

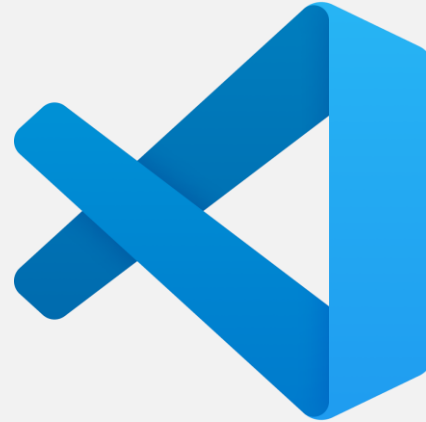
PROGRAMAÇÃO EM PYTHON



Programação Em Python

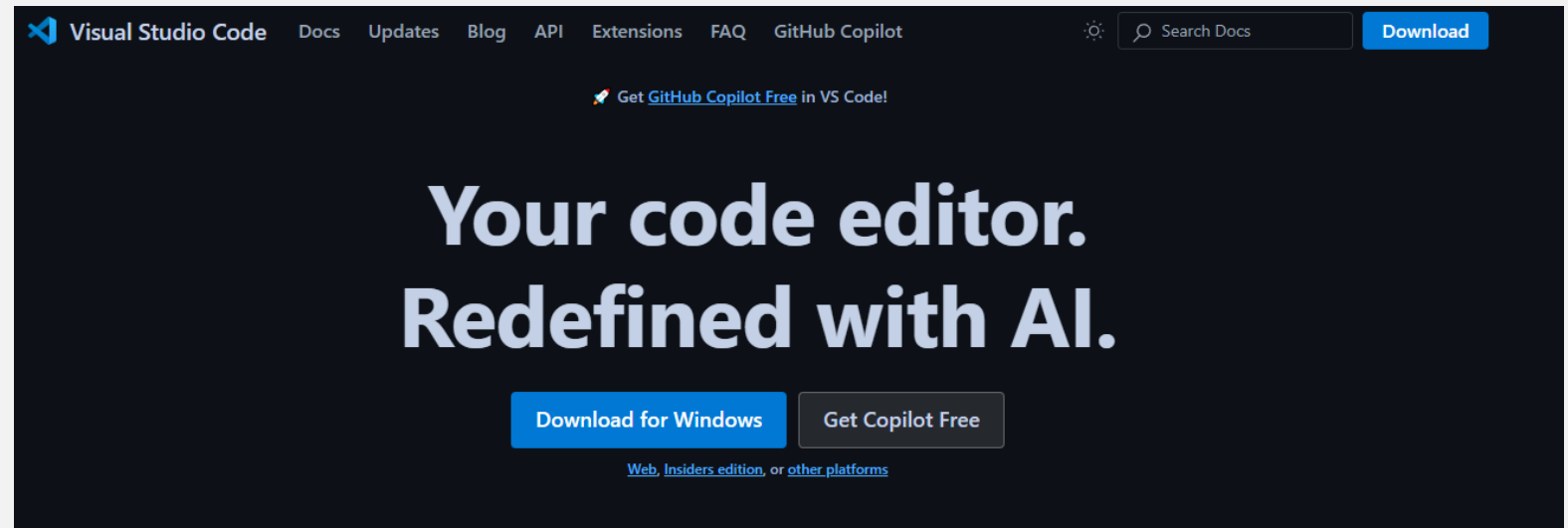


Para programar em Python, utilizaremos o VSCode.



SITE:

<https://code.visualstudio.com/>



PROGRAMAÇÃO EM PYTHON



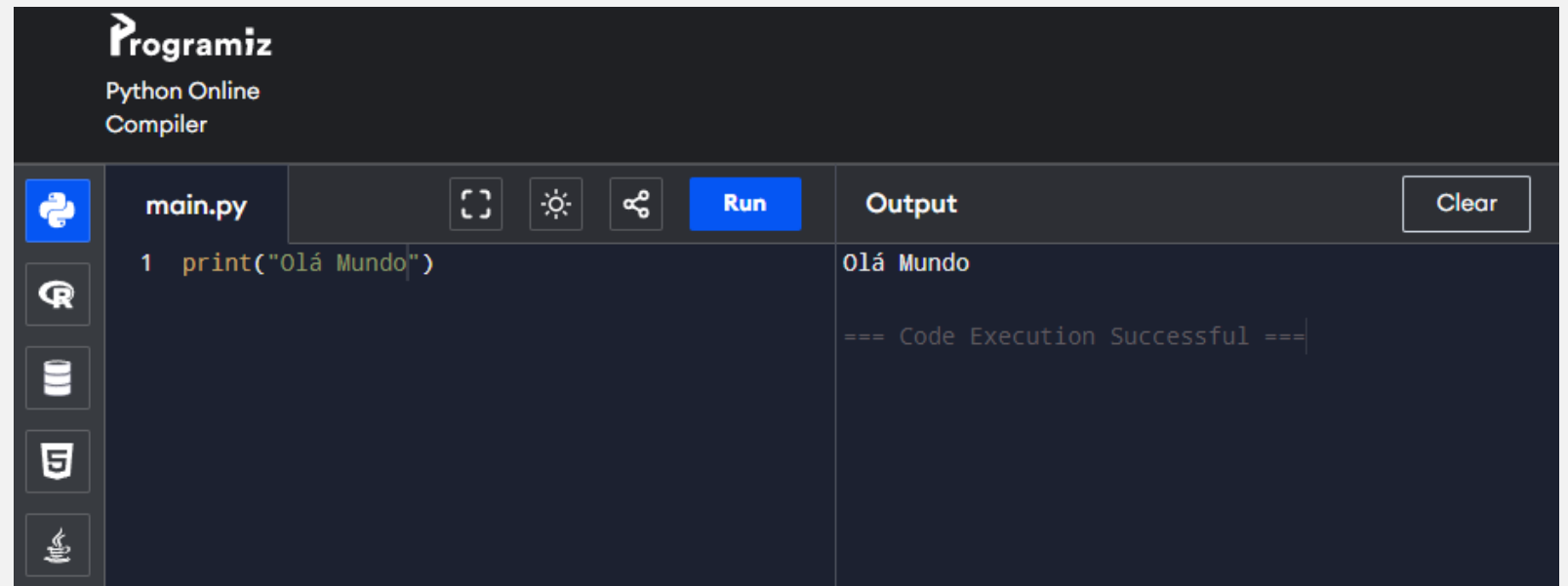
Programação Em Python



Alternativa:

O site [Programiz Python Online Compiler](https://www.programiz.com/python-programming/online-compiler/) é um compilador e interpretador online para a linguagem Python. Ele permite que você escreva, edite e execute código Python diretamente no navegador, sem precisar instalar nada no seu computador.

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



PROGRAMAÇÃO EM PYTHON



Programação Em Python



Variáveis

Em Python, a declaração de variáveis é simples e não requer a definição explícita de tipos, pois a linguagem utiliza tipagem dinâmica. Por exemplo, uma variável pode ser criada e modificada da seguinte forma:

main.py	Run	Output
<pre>1 nome = "Maria" 2 idade = 28 3 4 print("Nome da pessoa: ", nome) 5 print("Idade da pessoa: ", idade)</pre>		<pre>Nome da pessoa: Maria Idade da pessoa: 28 === Code Execution Successful ===</pre>

Professor, neste momento, explique ao aluno como inserir esses dados no VSCode, salvar o arquivo como main.py e executá-lo no terminal do VSCode ou do sistema operacional.

PROGRAMAÇÃO EM PYTHON



Programação Em Python



Variáveis

As variáveis possuem simplicidade e tipagem dinâmica, é importante seguir regras e convenções na nomeação dessas variáveis:

Regras de Nomenclatura:

- O nome da variável deve começar com uma letra (**a–z** ou **A–Z**) ou com um **underscore** (**_**).
- Pode conter **letras**, **números** e **underscores**.
- **Não pode começar com um número nem conter caracteres especiais, como `&`, `@`, `%`, etc.**
- Por exemplo, **'nome_completo'** é **válido**, enquanto **'&6varlo1'** não é permitido devido ao caractere **'&'** e por **iniciar com um número**.

PROGRAMAÇÃO EM PYTHON



Programação Em Python



Variáveis

Convenção snake_case:

- Em Python, a convenção recomendada é utilizar o **snake_case** para nomes de variáveis.
- Isso significa usar apenas letras minúsculas e separar palavras com underscores, como em 'idade_usuario' ou 'valor_total'.

Tipagem Dinâmica:

- Python **não exige declaração explícita** do tipo da variável.
- O **tipo é determinado pelo valor atribuído** à variável no momento da execução.

PROGRAMAÇÃO EM PYTHON



Programação Em Python



Instruções de I/O (Entrada/Saída)

As instruções de entrada e saída são essenciais para a interação entre o programa e o usuário ou outros sistemas. Em Python, as operações de I/O podem ser realizadas de diversas maneiras:

```
main.py  [Icons]  Run
1 nome = input("Qual é o seu nome? ")
2 idade = input("Qual sua idade? ")
3
4 print("Nome da pessoa: ", nome)
5 print("Idade da pessoa: ", idade)
```



```
Output  Clear
Qual é o seu nome? Maria
Qual sua idade? 28
Nome da pessoa:  Maria
Idade da pessoa:  28

=== Code Execution Successful ===
```

Para **entrada de dados**
usamos a função **input()**

Para **saída de dados**
usamos a função **print()**

PROGRAMAÇÃO EM PYTHON



Programação Em Python



Formas diferentes de usar o `print()`

main.py	Output
1 nome = input("Qual é o seu nome? ")	Qual é o seu nome? Maria
2 idade = input("Qual sua idade? ")	Qual sua idade? 25
3	-----
4 print("-----")	Nome da pessoa: Maria
5 print("Nome da pessoa:", nome)	Idade da pessoa: 25
6 print("Idade da pessoa:", idade)	-----
7 print("-----")	Nome da pessoa: Maria
8 print(f"Nome da pessoa: {nome}")	Idade da pessoa: 25
9 print(f"Idade da pessoa: {idade}")	=====
	=== Code Execution Successful ===

• Forma 1: `print("Nome da pessoa:", nome)`

- Na abordagem apresentada, a função `print` é utilizada com múltiplos argumentos separados por vírgula. Por padrão, o Python insere um espaço entre cada argumento ao exibir a saída, o que **permite imprimir diversos valores de forma clara e organizada**, sem a necessidade de concatenar manualmente as strings.

• Forma 2: `print(f"Nome da pessoa: {nome}")`

- Utilizamos f-strings (strings formatadas). Tudo que estiver entre chaves `{}` dentro de uma f-string será substituído pelo valor ou expressão correspondente. Essa abordagem permite **maior controle e legibilidade**, pois podemos inserir variáveis e até mesmo expressões diretamente no texto.

PROGRAMAÇÃO EM PYTHON



Programação Em Python

Vamos
realizar
juntos!



```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda

01-Carro.py X
01-Carro.py
1  nome_carro = input("Digite o nome do carro: ")
2  valor_carro = float(input("Digite o valor do carro: "))
3  consumo_por_litro = float(input("Digite o consumo por litro: "))
4  print("-----")
5  print("| Carro:", nome_carro)
6  print("| Valor:", valor_carro)
7  print("| Consumo por litro:", consumo_por_litro)
8  print("-----")

PROBLEMAS  SAÍDA  CONSOLE DE DEPURACÃO  TERMINAL  PORTAS  COMENTÁRIOS

● PS D:\Fabrica de Programadores> python.exe .\01-Carro.py
Digite o nome do carro: Honda Civic 1.5 Turbo
Digite o valor do carro: 45000
Digite o consumo por litro: 11.5
-----
| Carro: Honda Civic 1.5 Turbo
| Valor: 45000.0
| Consumo por litro: 11.5
-----
○ PS D:\Fabrica de Programadores> 
```

Professor, neste momento, oriente o aluno sobre como salvar o script e executá-lo no terminal do VSCode ou do sistema operacional. Explique o procedimento para execução do Python via linha de comando.

PROGRAMAÇÃO EM PYTHON



Programação Em Python

Vamos
realizar
juntos!



```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda

02-Produto.py X
02-Produto.py
1  nome_produto = input("Digite o nome do produto: ")
2  preco = float(input("Digite o preço do produto: "))
3  desconto = float(input("Digite o percentual de desconto: "))
4  valor_desconto = preco * desconto / 100
5  preco_final = preco - valor_desconto
6  print(f"Produto: {nome_produto} - Preço final: R$ {preco_final}")
7

PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL  PORTAS  COMENTÁRIOS

● PS D:\Fabrica de Programadores> python.exe .\02-Produto.py
  Digite o nome do produto: Mochila
  Digite o preço do produto: 120
  Digite o percentual de desconto: 20
  Produto: Mochila - Preço final: R$ 96.0
○ PS D:\Fabrica de Programadores> 
```

Professor, neste momento, destaque para o aluno a importância de converter dados nas variáveis utilizando **str**, **int** e **float** no Python, garantindo compatibilidade e precisão nos cálculos e manipulações.

PROGRAMAÇÃO EM PYTHON



Contexto sobre: Built-ins



Built-ins referem-se a funções, classes e objetos que estão disponíveis de **forma automática**, sem que seja necessário importar módulos adicionais. Eles são parte integrante da linguagem e fornecem ferramentas essenciais para a programação.

Disponibilidade imediata:

- Built-ins estão sempre disponíveis no ambiente. Funções como print(), len(), type(), float(), entre outras, podem ser utilizadas diretamente.

Facilitadores do desenvolvimento:

- Por oferecerem funcionalidades comuns, eles permitem que o programador foque na lógica do problema, sem precisar implementar do zero operações básicas como conversões de tipo, manipulação de coleções, cálculos matemáticos, etc.

PROGRAMAÇÃO EM PYTHON



Contexto sobre: Built-ins



Site sobre as funcionalidades Built-ins:

<https://docs.python.org/pt-br/3.13/library/functions.html>

Funções embutidas

A

[abs\(\)](#)
[aiter\(\)](#)
[all\(\)](#)
[anext\(\)](#)
[any\(\)](#)
[ascii\(\)](#)

B

[bin\(\)](#)
[bool\(\)](#)
[breakpoint\(\)](#)
[bytearray\(\)](#)
[bytes\(\)](#)

C

[callable\(\)](#)
[chr\(\)](#)
[classmethod\(\)](#)
[compile\(\)](#)
[complex\(\)](#)

D

[delattr\(\)](#)
[dict\(\)](#)
[dir\(\)](#)
[divmod\(\)](#)

E

[enumerate\(\)](#)
[eval\(\)](#)
[exec\(\)](#)

F

[filter\(\)](#)
[float\(\)](#)
[format\(\)](#)
[frozenset\(\)](#)

G

[getattr\(\)](#)
[globals\(\)](#)

H

[hasattr\(\)](#)
[hash\(\)](#)
[help\(\)](#)
[hex\(\)](#)

I

[id\(\)](#)
[input\(\)](#)
[int\(\)](#)
[isinstance\(\)](#)
[issubclass\(\)](#)
[iter\(\)](#)

L

[len\(\)](#)
[list\(\)](#)
[locals\(\)](#)

M

[map\(\)](#)
[max\(\)](#)
[memoryview\(\)](#)
[min\(\)](#)

N

[next\(\)](#)

O

[object\(\)](#)
[oct\(\)](#)
[open\(\)](#)
[ord\(\)](#)

P

[pow\(\)](#)
[print\(\)](#)
[property\(\)](#)

R

[range\(\)](#)
[repr\(\)](#)
[reversed\(\)](#)
[round\(\)](#)

S

[set\(\)](#)
[setattr\(\)](#)
[slice\(\)](#)
[sorted\(\)](#)
[staticmethod\(\)](#)
[str\(\)](#)
[sum\(\)](#)
[super\(\)](#)

T

[tuple\(\)](#)
[type\(\)](#)

V

[vars\(\)](#)

Z

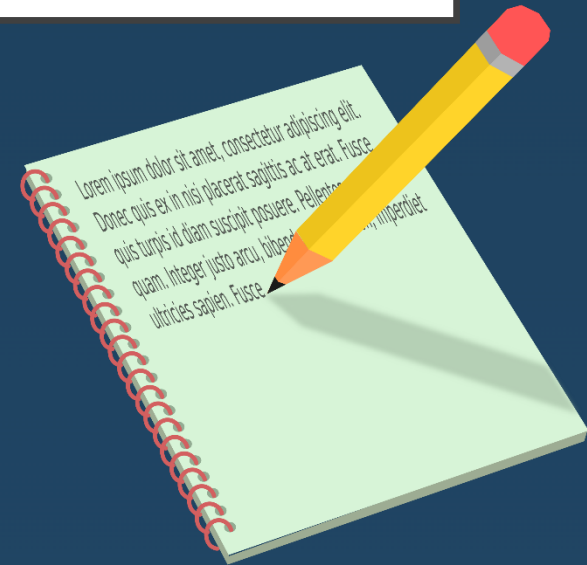
[zip\(\)](#)

[__import__\(\)](#)

Professor, sinte-se à vontade para explorar os conceitos das funções built-in do Python e realizar testes práticos em sala de aula.

EXERCÍCIOS

- Converta para Python as atividades previamente desenvolvidas em Portugol. Salve cada uma em sua pasta, nomeando os arquivos de forma adequada e não se esqueça de utilizar a extensão '.py'



REFERÊNCIAS

ALVES, William Pereira. Lógica de programação de computadores. São Paulo: Érica, 2012.

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. Fundamentos da programação de computadores. 3. ed. São Paulo: Pearson, 2012.

FARRELL, Joyce. Lógica e design de programação. São Paulo: Cengage Learning, 2010.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. Algoritmos: Lógica para desenvolvimento de programação de computadores. 27. ed. rev. São Paulo: Érica, 2014.

MEDINA, Marco; FERTIG, Cristina. Algoritmos e programação. Teoria e prática. São Paulo: Novatec, 2005.

PEREIRA, Silvio do Lago. Algoritmos e lógica de programação em C: Uma abordagem didática. São Paulo: Érica, 2010.

Importante:

Os conteúdos disponibilizados são específicos para este curso/turma, a divulgação ou reprodução do material para outras pessoas/organização não é autorizada.



daniel.petinice@sp.senai.br



Daniel Tadeu Petinice



Instrutor de Formação Profissional III

