



Programação Orientada a Objetos

Aula 01 – Apresentação da Disciplina

Prof. Paulo Vinicius Vieira
paulo.vieira@faculdadeimpacta.com.br

Sumário

- Apresentação da Disciplina
- Plano de Ensino
 - Objetivos
 - Ementa
 - Conteúdo Programático
 - Bibliografia
 - Critérios de Avaliação
 - Metodologia
 - Recursos
- Instalação do Ambiente de Desenvolvimento
 - Python e Visual Studio Code
- Exercícios de Revisão

Plano de Ensino

- Objetivos:
 - Desenvolver software orientado a objetos.
 - Codificar utilizando boas práticas de estilo de programação.
 - Utilizar testes automatizados para aumento de qualidade de código.
 - Usar o ambiente integrado de desenvolvimento (IDE) Visual Studio Code.
 - Conhecer as estruturas de dados básicas.
 - Persistir dados em arquivos e banco de dados.

Plano de Ensino

- Ementa:
 - Estruturas de dados básicas:
 - Listas, pilhas, filas e dicionários
 - Introdução à lista encadeada e árvore
 - Programação orientada a objetos:
 - Objetos e classes
 - Encapsulamento
 - Relacionamento entre classes
 - Herança e polimorfismo
 - Classe abstrata
 - Tratamento de Exceções
 - Testes de unidade automatizados
 - Estratégias de persistência em arquivos e banco de dados

Plano de Ensino

- Conteúdo Programático:

Semana 1
<p>Apresentação da disciplina.</p> <p>Apresentação e preparação do ambiente de programação (Visual Studio Code).</p> <p>Revisão da linguagem Python (estruturas básicas e funções).</p>
Semana 2
<p>Revisão da linguagem Python (listas e tuplas).</p>
Semana 3
<p>Dicionários</p> <p>Criação e utilização de Módulos.</p> <p>AC01 – Módulos, funções, listas, tuplas e dicionários.</p>

Plano de Ensino

- Conteúdo Programático:

Semana 4
<p>Padrão de Código PEP8.</p> <p>Tratamento de Exceções.</p> <p>Testes de Unidade.</p>
Semana 5
<p>Programação Orientada a Objetos: introdução.</p> <p>Conceitos básicos de POO: abstração, classes, atributos, métodos.</p>
Semana 6
<p>Exercícios de Implementação de classes.</p> <p>AC02 – Exceções, testes unitários, implementação de classes, atributos, métodos</p>

Plano de Ensino

- Conteúdo Programático:

Semana 7
Relacionamento entre classes.
Semana 8
Encapsulamento, atributos/métodos públicos e privados. <i>AC03 – Relacionamento entre classes, Encapsulamento.</i>

Plano de Ensino

- Conteúdo Programático:

Semana 9
Herança simples.
Semana 10
Herança múltipla.
Semana 11
Polimorfismo. Classes abstratas e métodos abstratos. <i>AC04 – Herança, polimorfismo, classes abstratas e métodos abstratos.</i>

Plano de Ensino

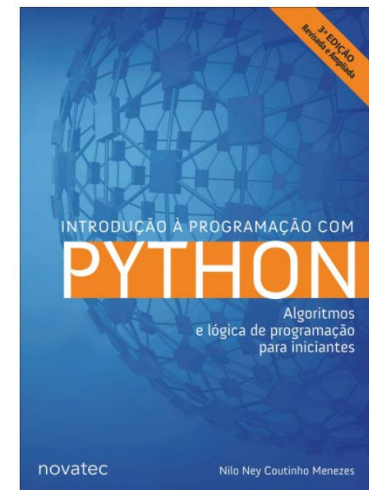
- Conteúdo Programático:

Semana 12
Persistência de dados em arquivos.
Semana 13
Persistência de dados: integração com banco de dados.
Semana 14
Persistência de dados: mapeamento objeto-relacional (ORM).
Semana 15
<i>AC05 – Persistência de Dados.</i>

Plano de Ensino

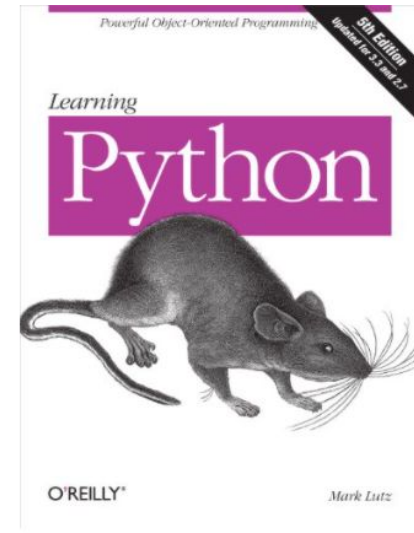
- Bibliografia

- RAMALHO, Luciano. **Python Fluente: Programação Clara, Concisa e Eficaz.** Novatec, 2015.
- MENEZES, Nilo Ney Coutinho. **Introdução à Programação com Python: algoritmos e lógica de programação para iniciantes.** Novatec, 2019.



Plano de Ensino

- Bibliografia
 - LUTZ, Mark. **Learning Python: Powerful Object-Oriented Programming**. O'Reilly, 2011.
 - PHILLIPS, Dusty. **Python 3: Object Oriented Programming: Harness the power of Python 3 objects**. Packt Publishing, 2010.



Plano de Ensino

- Critérios de Avaliação:
 - ***5 Atividades Contínuas (ACs):***
 - A menor nota das ACs é descartada
 - Os critérios de avaliação de cada AC será definido na aplicação da atividade
 - ACs devem obedecer o prazo estipulado de entrega e não serão aceitas em atraso
 - Não há SUB de AC
 - ***1 Prova Semestral:***
 - Conteúdo de todo o semestre

Plano de Ensino

- Critérios de Avaliação:
 - ***Média Final = (50% Média das ACs) + (50% Prova)***
 - Se ($\text{Média Final} \geq 6,0$ e $\text{Frequência} \geq 75\%$)
 - o aluno é ***aprovado***
 - Caso contrário
 - o aluno é ***reprovado***
 - *Direito a uma prova substitutiva*

Plano de Ensino

- Critérios de Avaliação:
 - ***Controle da Frequência (Aulas Remotas):***
 - Exercícios e Atividades individuais devem ser realizadas durante a aula para validar a presença
 - Caso o aluno não entregue a atividade, ficará com falta na aula.

Plano de Ensino

- Metodologia:
 - Aulas expositivas nas quais se apresenta e discute os tópicos da disciplina.
 - Resolução de exercícios e atividades práticas individuais e em grupos, realizados em ambiente de desenvolvimento.
 - Atividades contínuas (AC) para acompanhamento do processo de ensino aprendizagem.
 - Utilização do ambiente G Suite com Classroom e Meet.

Plano de Ensino

- Recursos Adicionais:
 - Videoaulas:
 - ***Curso de Python*** (Gustavo Guanabara):
<https://youtu.be/S9uPNppGsGo>
 - ***Python Orientado a Objetos*** (Otávio Miranda):
<https://youtu.be/RLVbB91A5-8>
 - ***Curso de Python*** (eXcript):
<https://youtu.be/j94IGZmwtYI>

Plano de Ensino

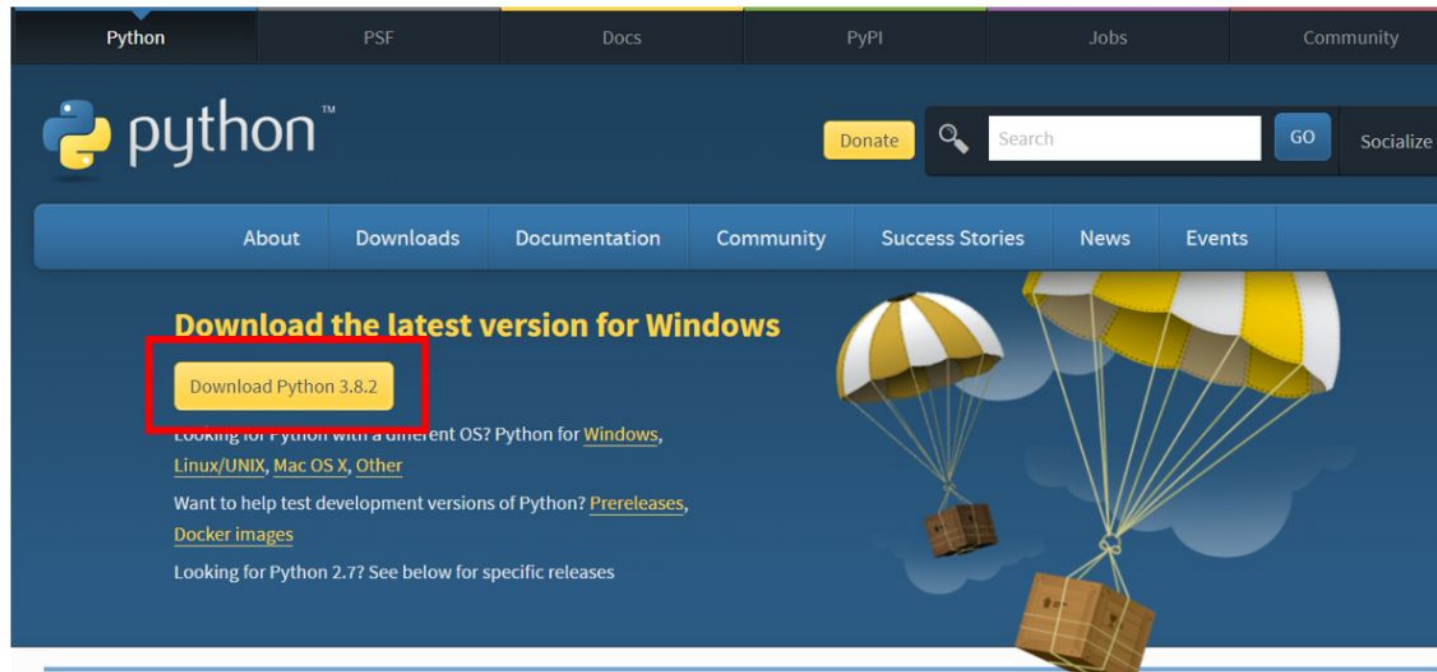
- Recursos Adicionais:
 - Ferramentas:
 - **Python:** <https://www.python.org/downloads/>
 - **Visual Studio Code:** <https://code.visualstudio.com/download>
 - **Editor Online Repl.it:** <https://repl.it/languages/python3>
 - **Python Tutor:** <http://pythontutor.com/>

Ambiente de Desenvolvimento

Python

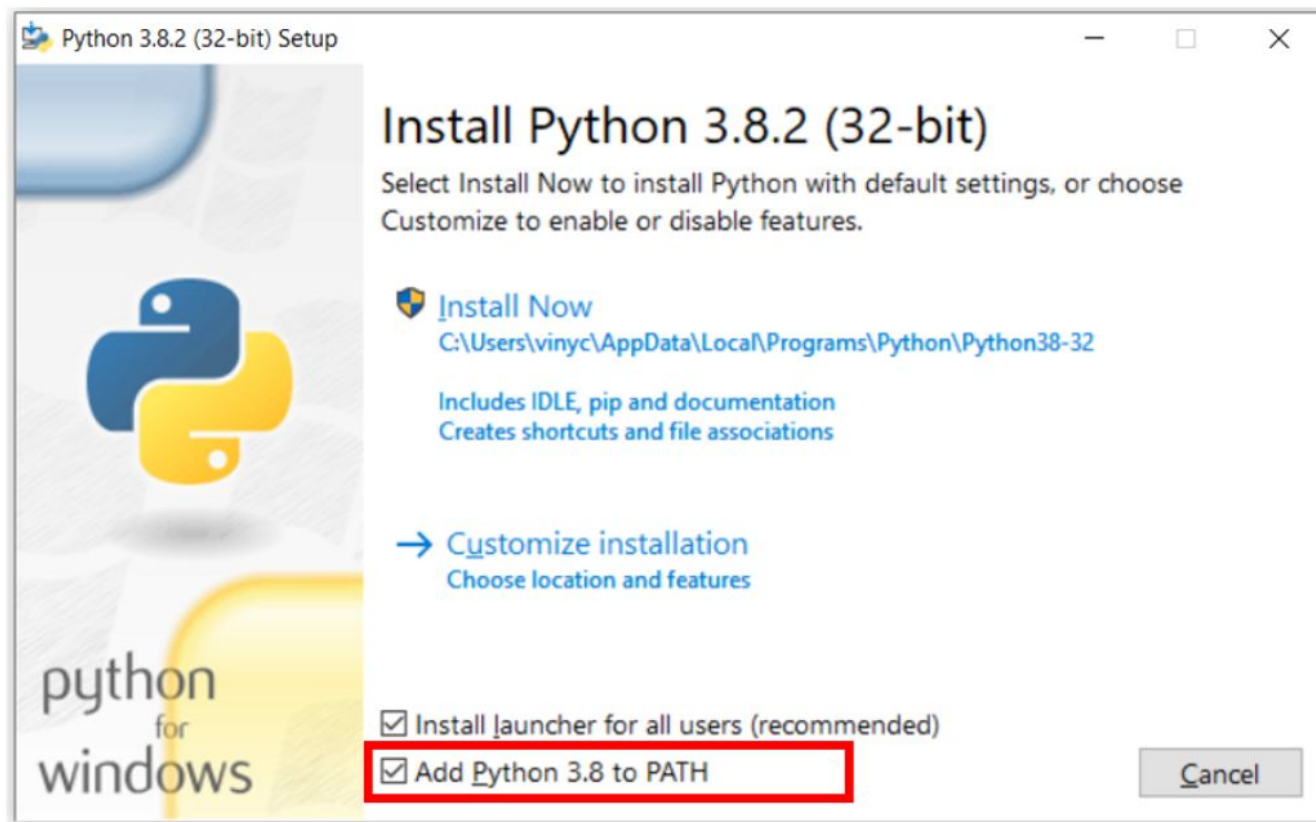
Ambiente de Desenvolvimento

- Python
 - Baixar o instalador da versão mais recente do python.
 - <https://www.python.org/downloads/>



Ambiente de Desenvolvimento

- Executar o instalador e marcar a opção **Add Python to PATH**



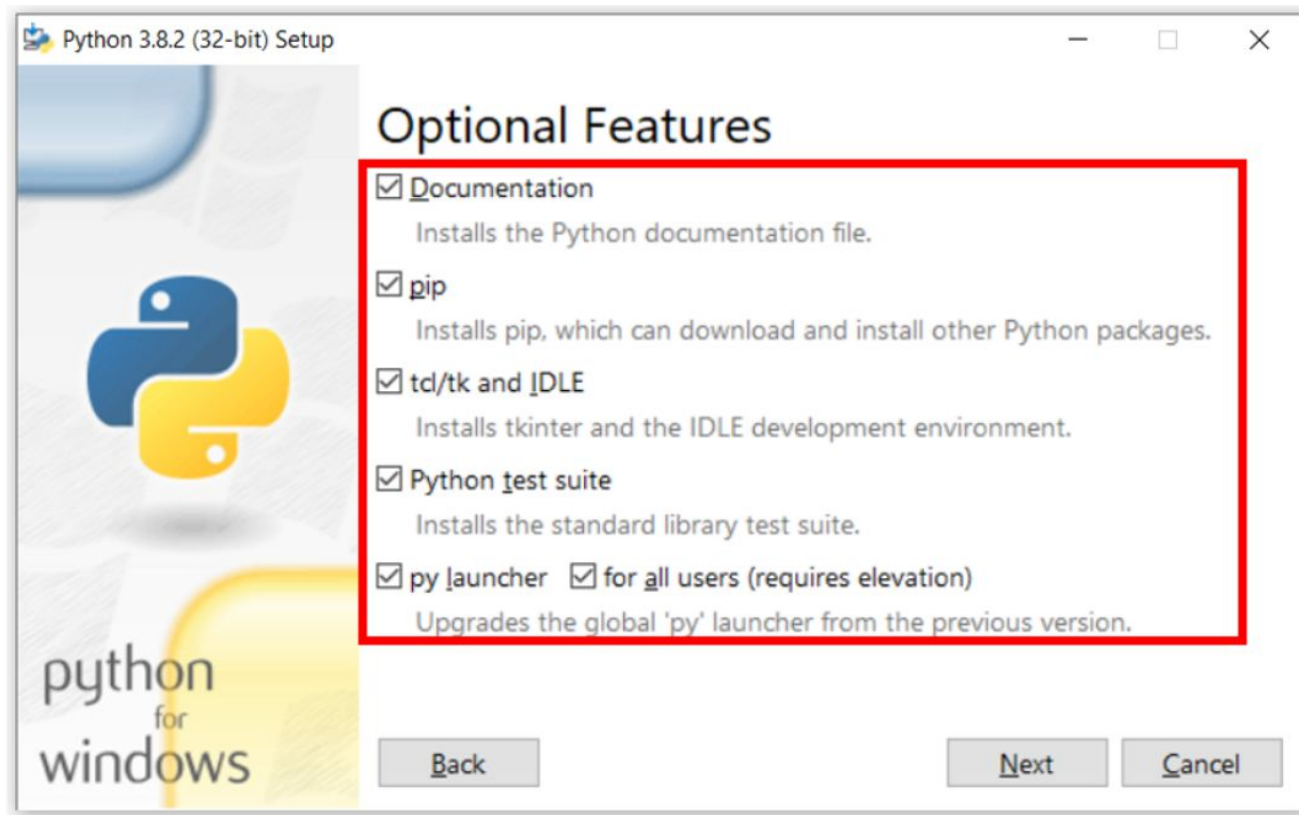
Ambiente de Desenvolvimento

- Escolher a opção de instalação **Customize installation**



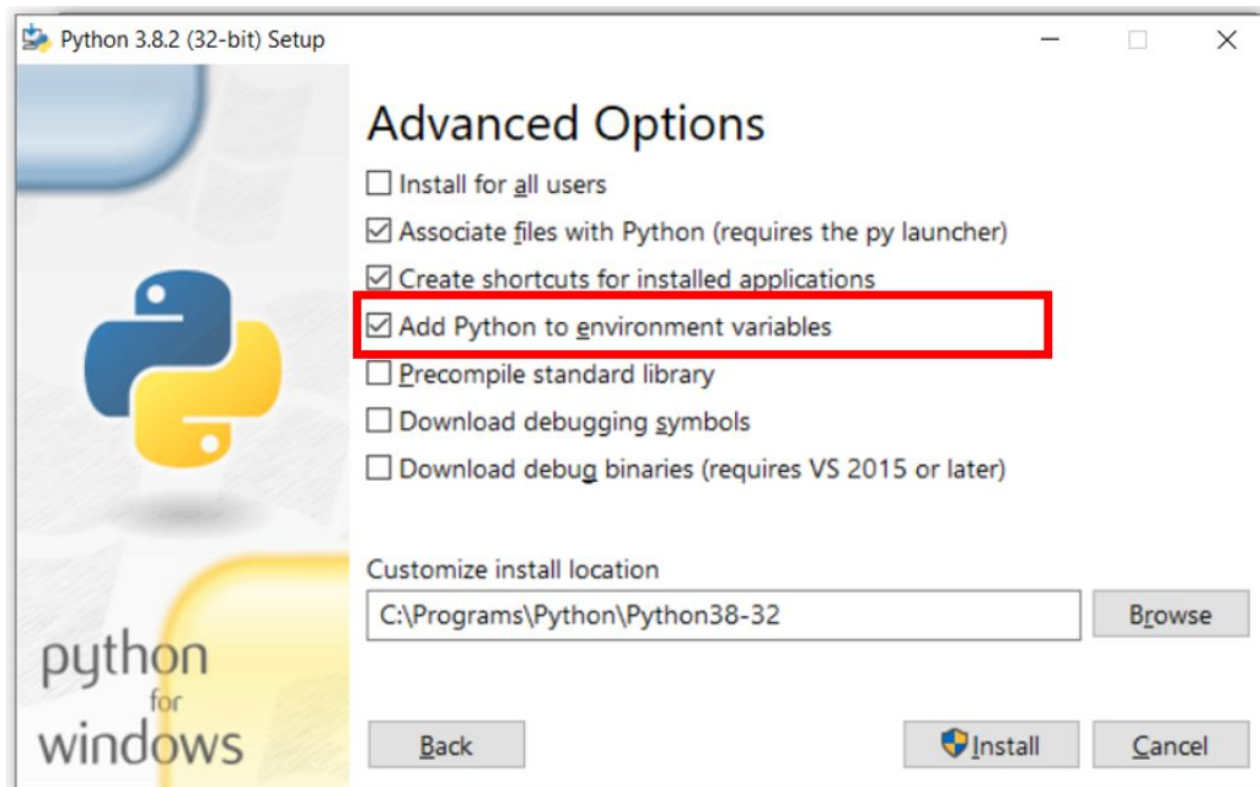
Ambiente de Desenvolvimento

- Certifique-se que todas as opções estão selecionadas e clique em Next



Ambiente de Desenvolvimento

- Na tela seguinte, certifique-se que a opção **Add Python to environment variables** está selecionada e clique em **Install**

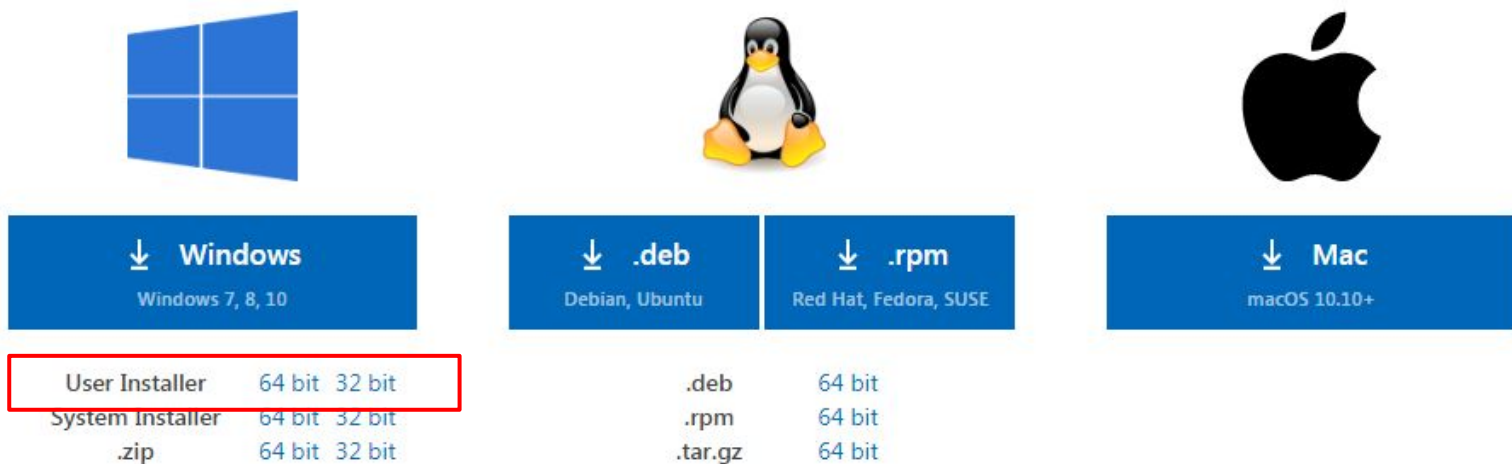


Ambiente de Desenvolvimento

Visual Studio Code

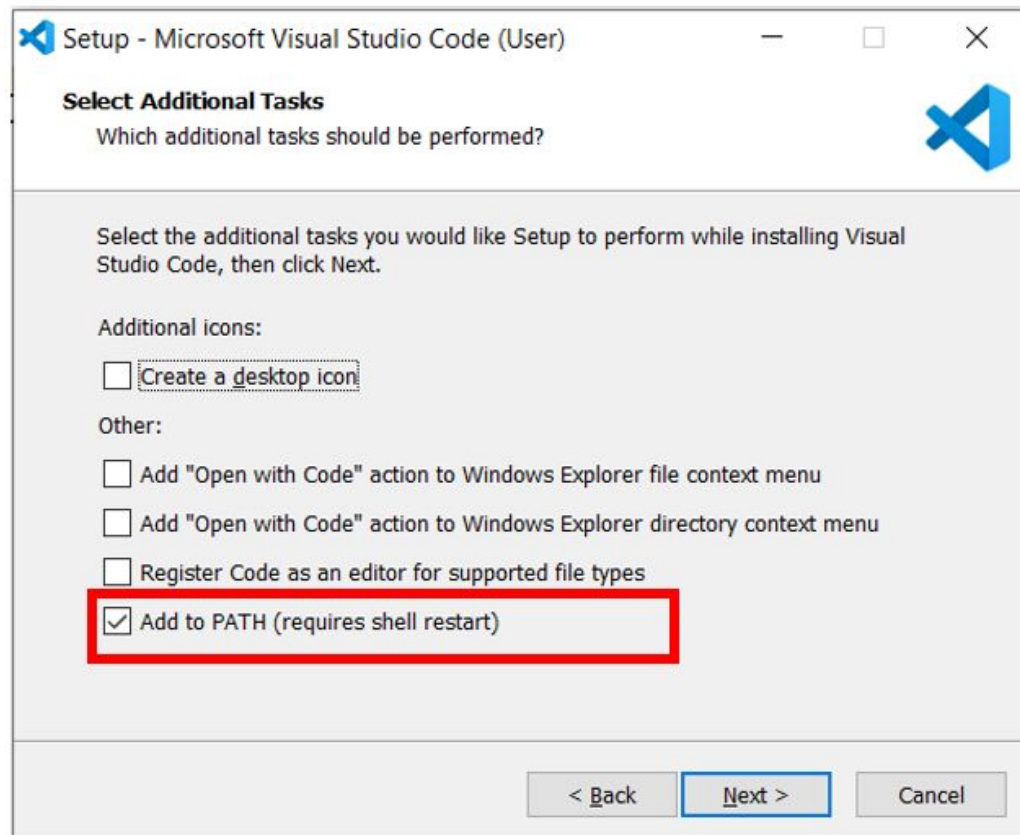
Ambiente de Desenvolvimento

- Após a instalação do Python, baixe a versão mais recente do Visual Studio Code (VS Code)
 - Selecione a opção correspondente ao seu sistema operacional
 - <https://code.visualstudio.com/download>



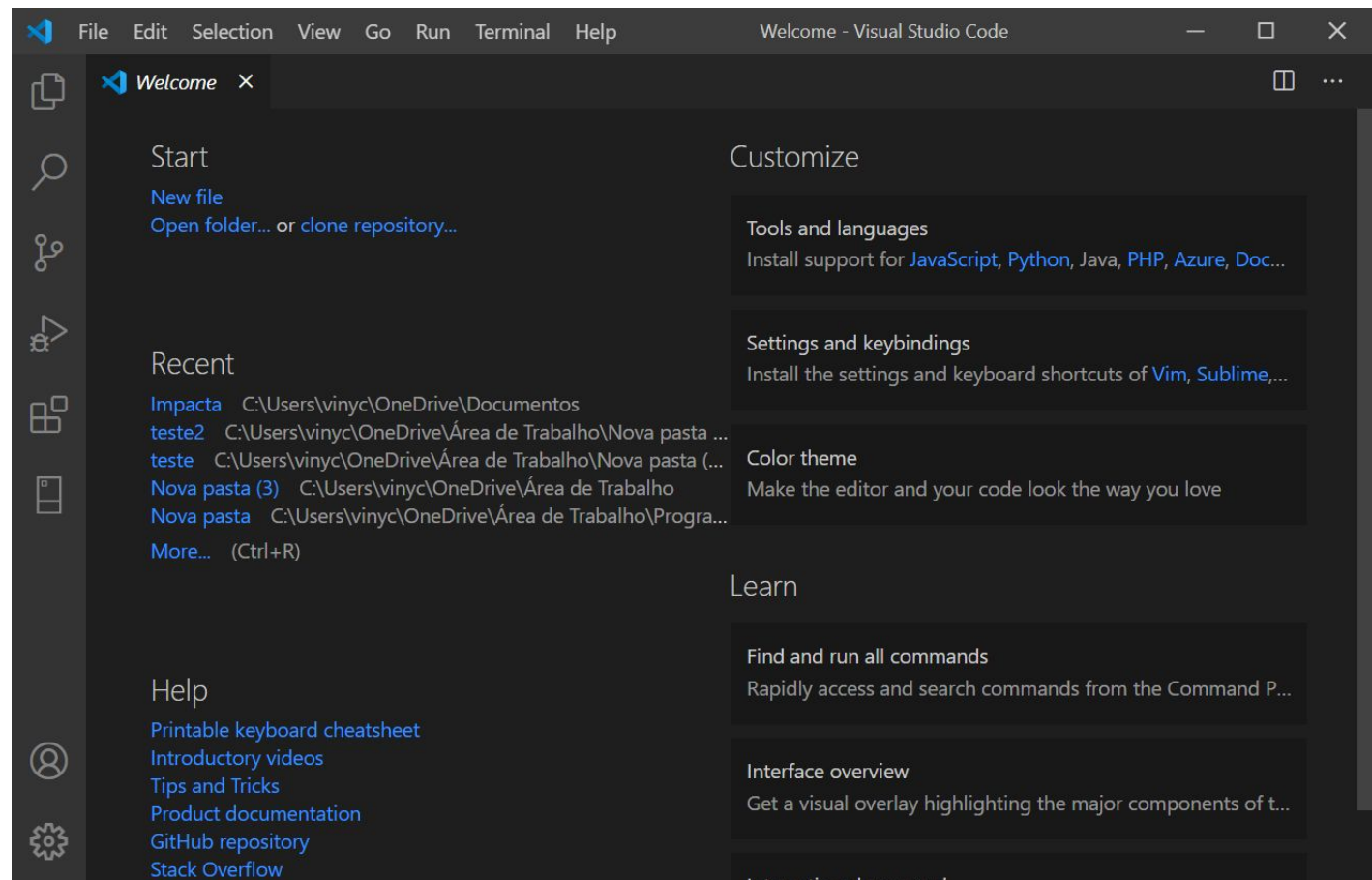
Ambiente de Desenvolvimento

- Execute o instalador e durante a instalação, certifique-se que a opção **Add to PATH** está selecionada.



Ambiente de Desenvolvimento

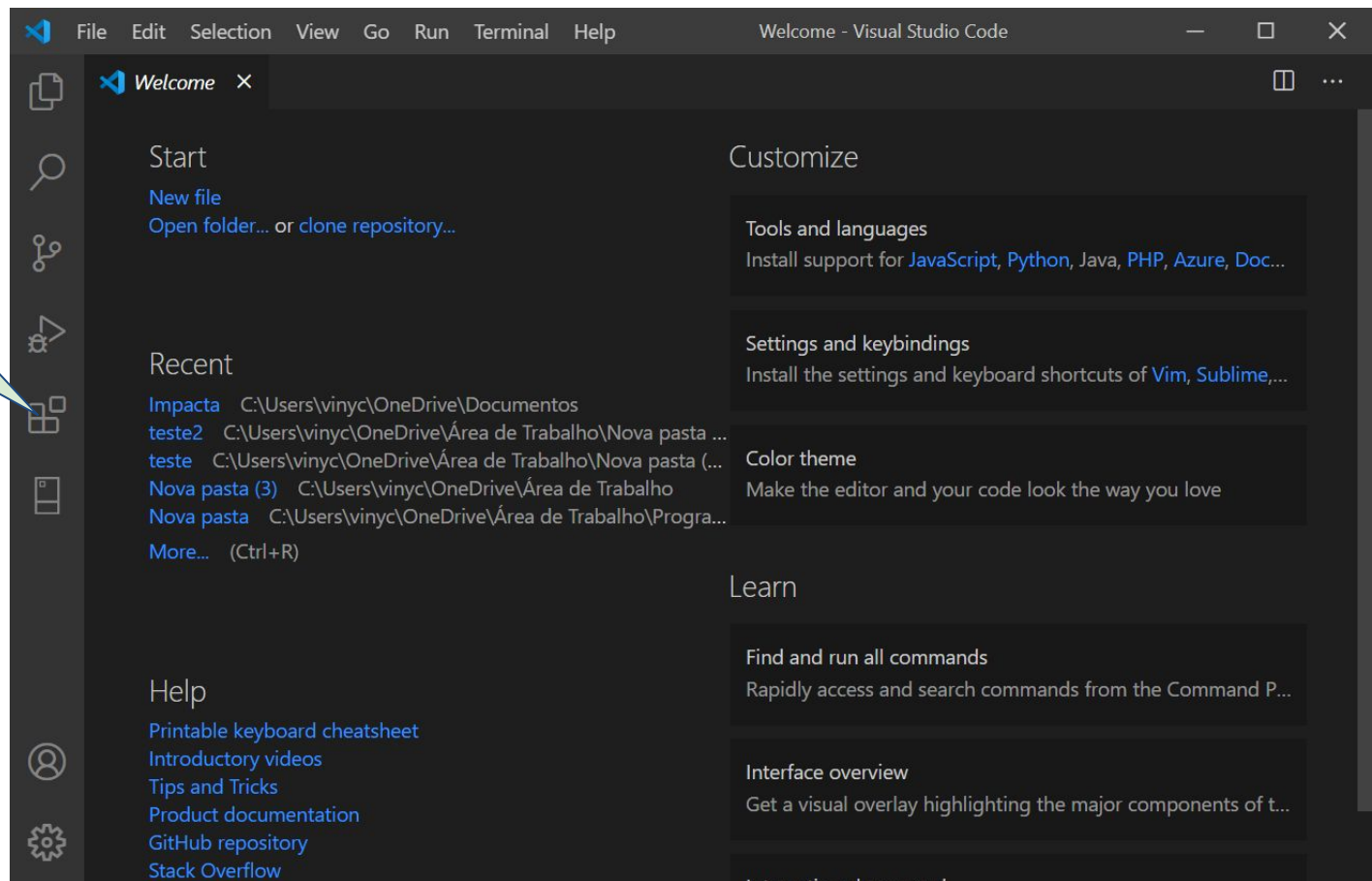
- Após a instalação, execute o VSCode



Ambiente de Desenvolvimento

- Instale a extensão do Python

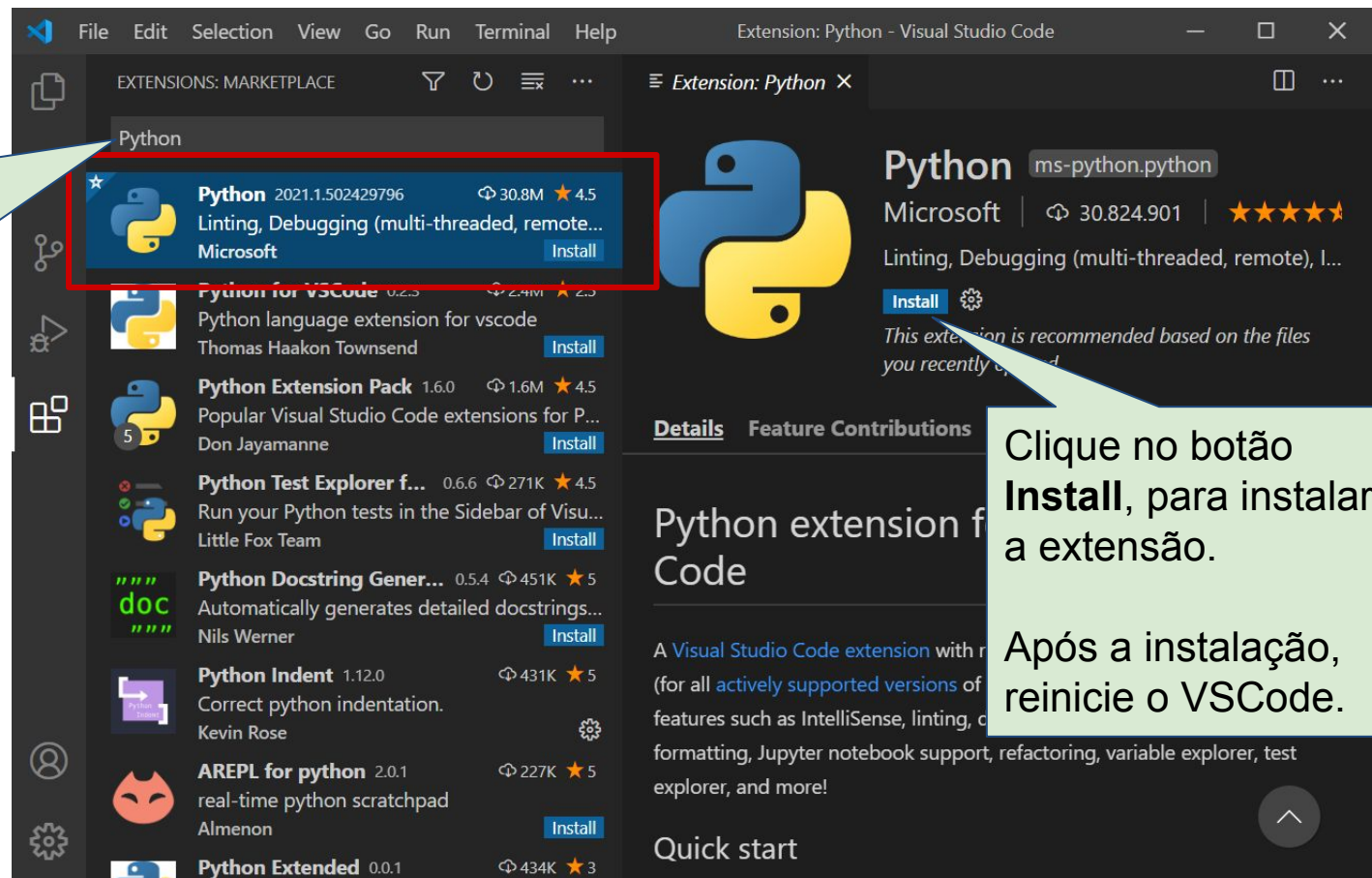
Clique no ícone
Extension



Ambiente de Desenvolvimento

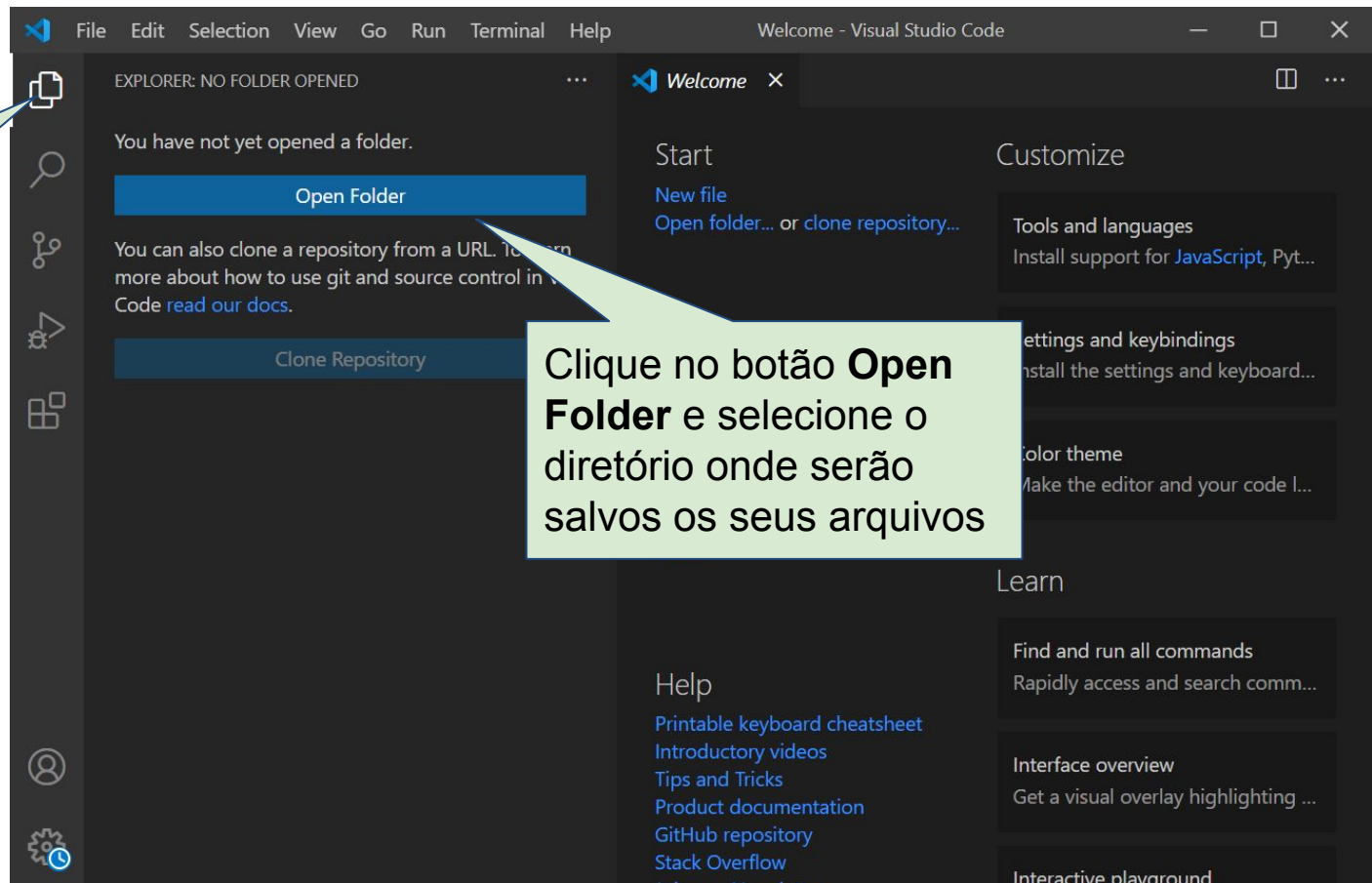
- Instale a extensão do Python

No campo de busca, procure pela extensão **Python**, distribuída pela Microsoft



Ambiente de Desenvolvimento

- Criando o primeiro programa

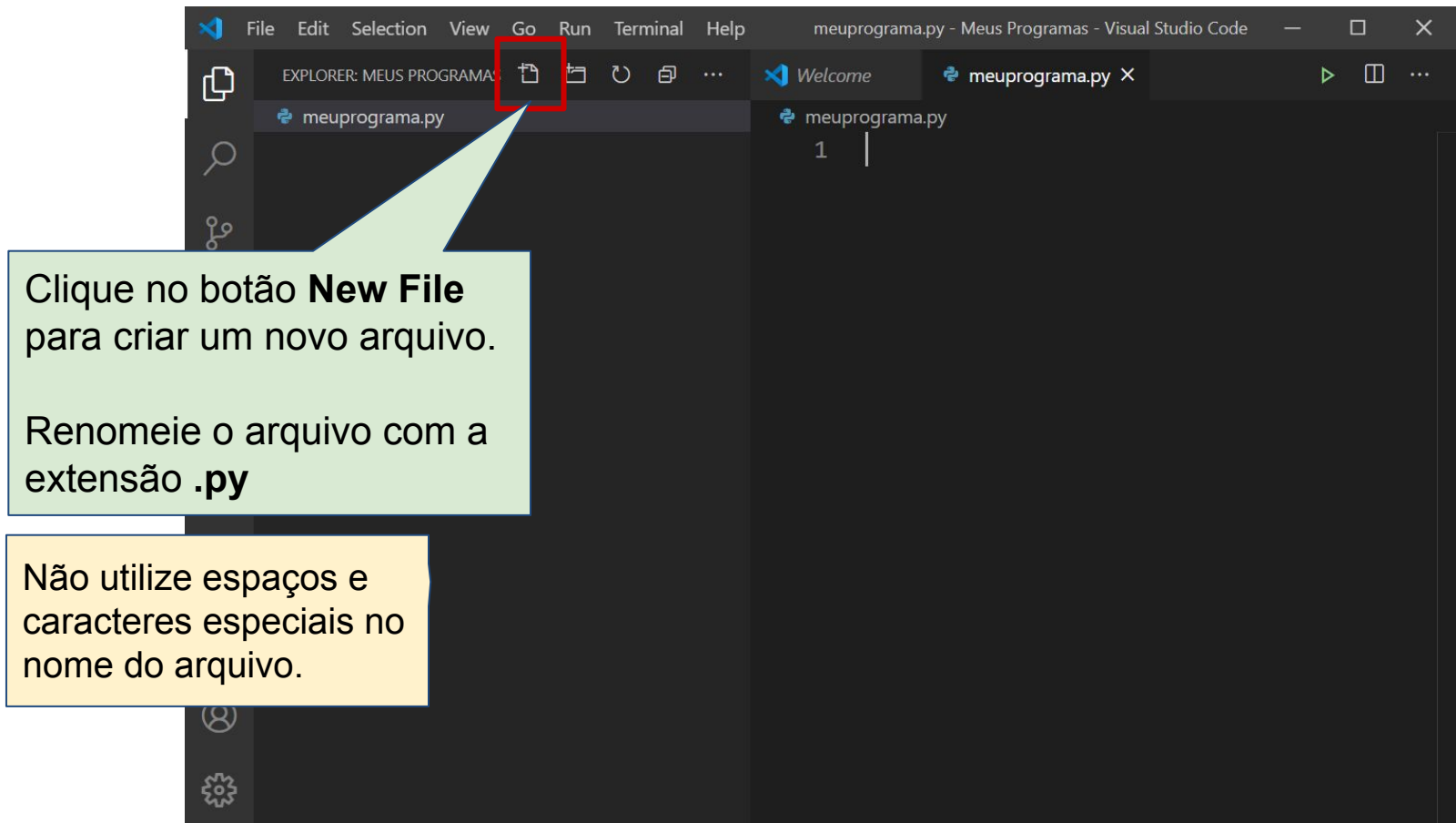


Clique no ícone
Explorer

Clique no botão **Open Folder** e selecione o diretório onde serão salvos os seus arquivos

Ambiente de Desenvolvimento

- Criando o primeiro programa



Ambiente de Desenvolvimento

- Criando o primeiro programa

The screenshot shows the Visual Studio Code interface with a file named 'meuprograma.py' open. The code is as follows:

```

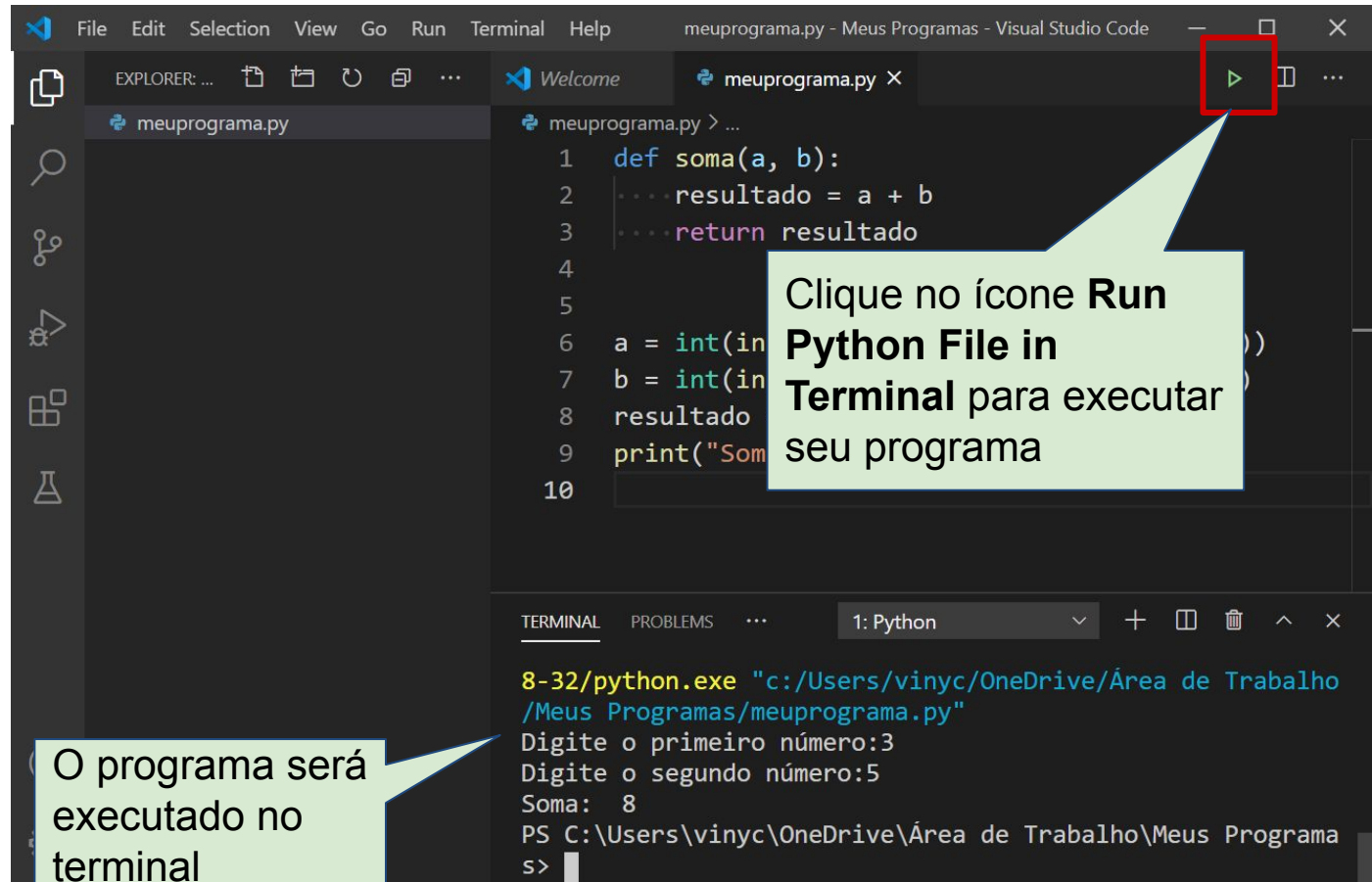
1  def soma(a, b):
2      ...resultado = a + b
3      ...return resultado
4
5
6  a = int(input("Digite o primeiro número:"))
7  b = int(input("Digite o segundo número:"))
8  resultado = soma(a, b)
9  print("Soma: ", resultado)
10

```

A green callout box with a pointer to the editor area contains the text: "Escreva o código do seu programa aqui".

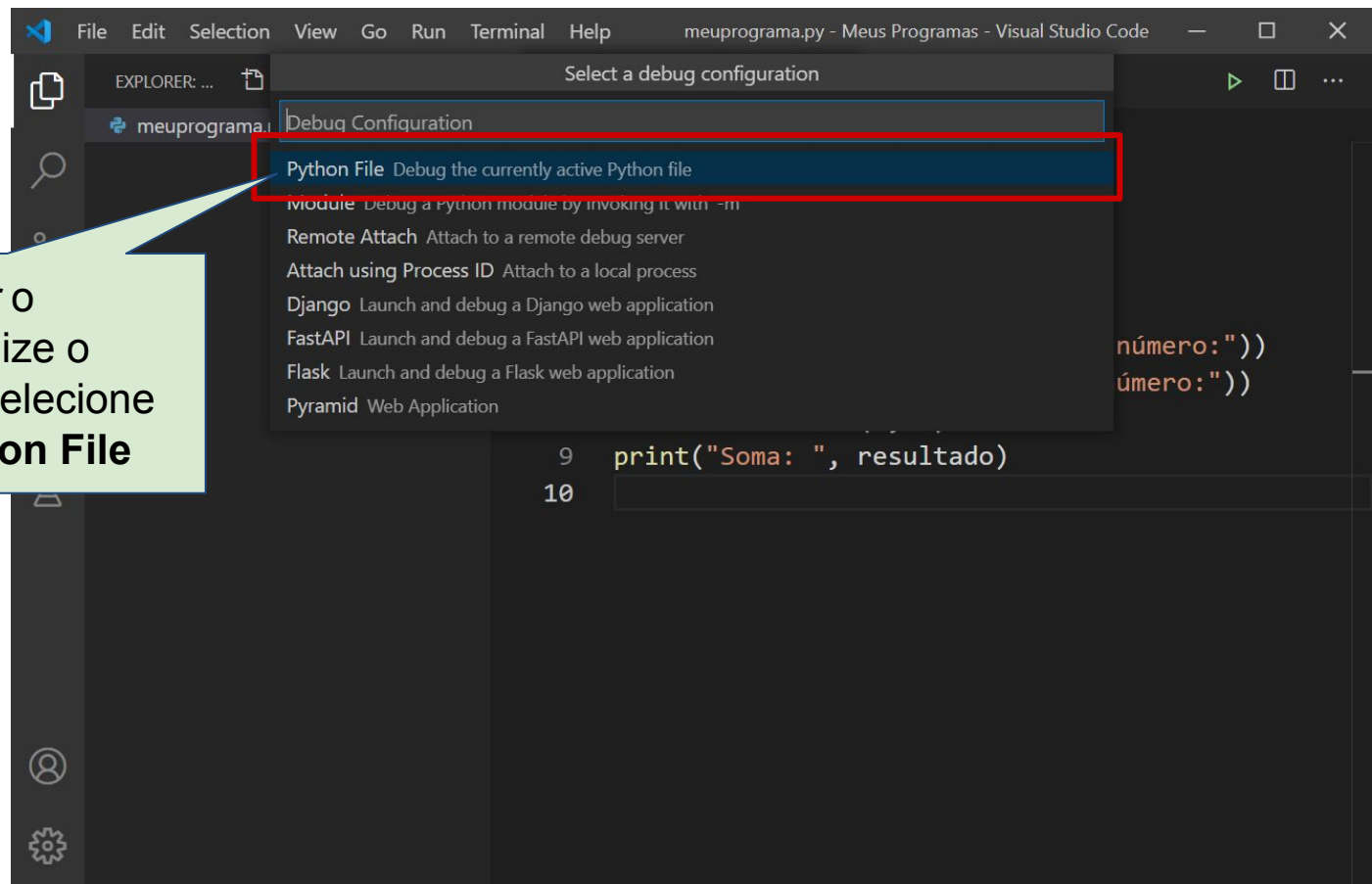
Ambiente de Desenvolvimento

- Executando o programa



Ambiente de Desenvolvimento

- Executando o programa



Para *debugar* o programa, utilize o atalho **F5**, e selecione a opção **Python File**

Ambiente de Desenvolvimento

- Executando o programa

