

Programação em Python – UC1

Aula 02 – Parte 1

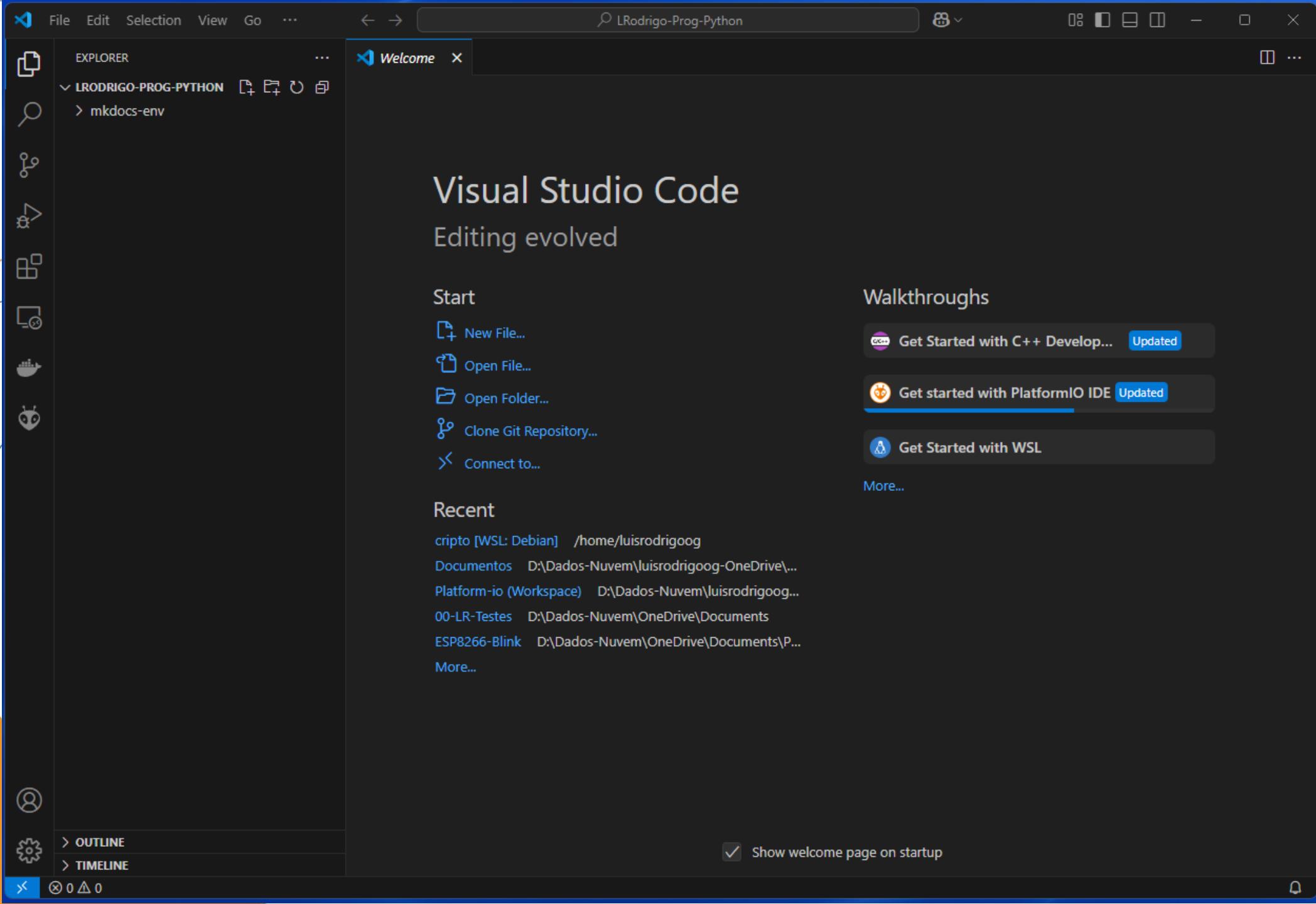


Visual Studio Code (VS Code)

VISUAL STUDIO CODE (VS CODE)



O **Visual Studio Code** (VS Code) é um dos editores de código mais populares entre desenvolvedores de todas as áreas.



VISUAL STUDIO CODE (VS CODE)



Por que usar o VS Code?

- ✓ Leve e rápido.
- ✓ Suporte a múltiplas linguagens
- ✓ Extensões poderosas
- ✓ Depuração integrada
- ✓ Controle de versão Git embutido

VISUAL STUDIO CODE (VS CODE)



Principais funcionalidades para Python

- ✓ Autocompletar e sugestões inteligentes (IntelliSense)
- ✓ Debugging integrado
- ✓ Ambientes virtuais e Jupyter Notebook
- ✓ Formatação automática de código
- ✓ Integração com Git
- ✓ Terminal embutido

VISUAL STUDIO CODE (VS CODE)



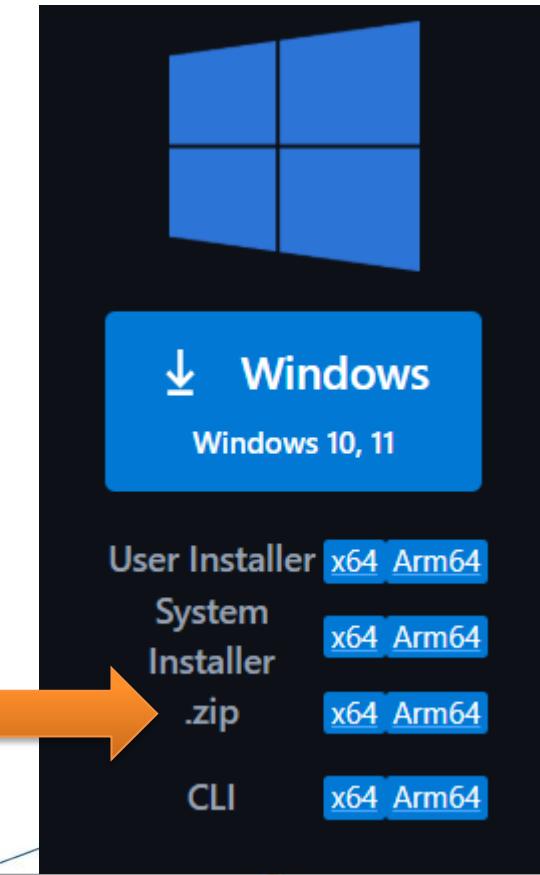
PORTABLE MODE

Dica de instalação em modo portátil

<https://code.visualstudio.com/docs/editor/portable>

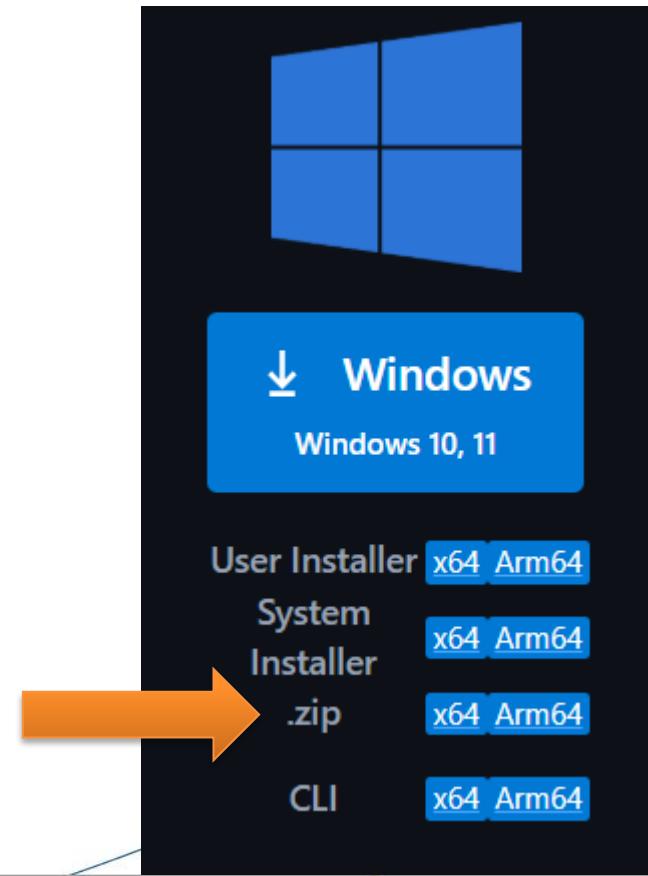
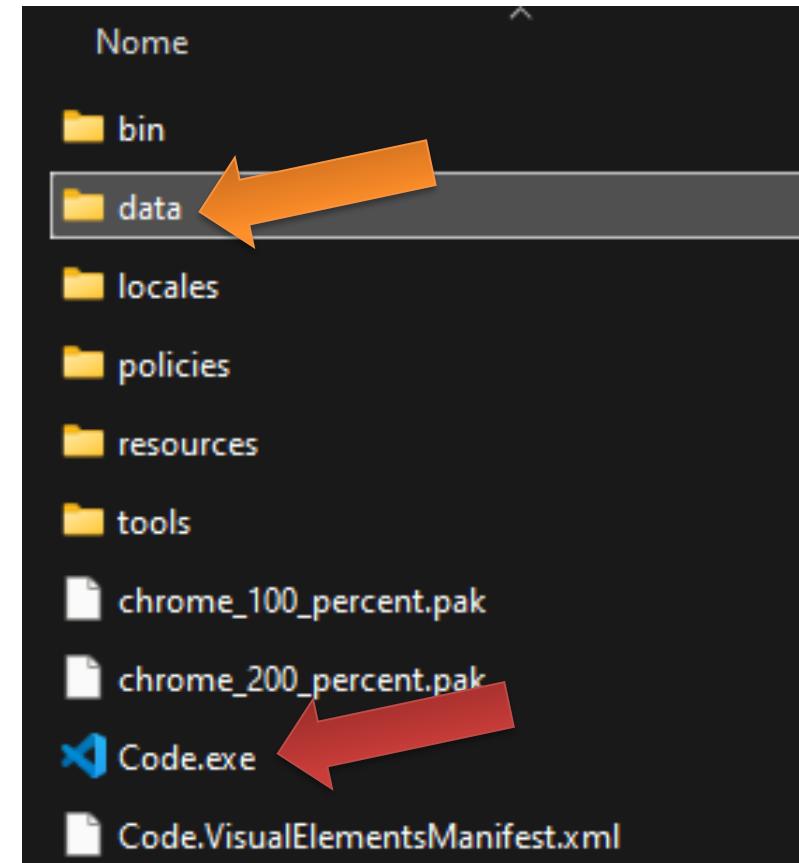
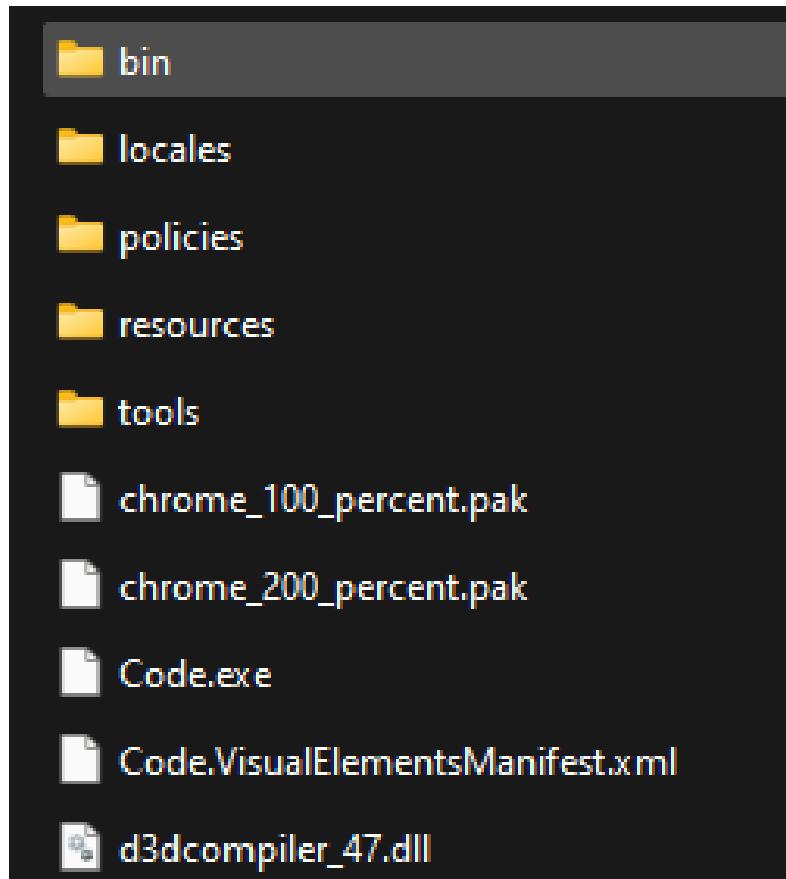
Download:

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



VISUAL STUDIO CODE (VS CODE)

PORTABLE MODE



The image shows the Visual Studio Code (VS Code) application window. The title bar includes standard icons for file operations and a search bar. The left sidebar features a dark theme with icons for file explorer, search, and other tools. The main content area displays the 'Welcome' screen with the title 'Get Started with VS Code'. Below the title, there's a callout box for 'Copilot' and a list of three additional setup steps: 'Choose your theme', 'Rich support for all your languages', and 'Tune your settings'. A note at the bottom states 'Code collects usage data. R'. On the right side, there's a large code editor window showing a file named 'projects.tsx' with some TypeScript code. A floating panel titled 'Copilot' provides AI-generated suggestions for modifying the code. At the bottom right, there's a modal dialog asking if the user wants to install the 'WSL' extension.

Get Started with VS Code

Customize your editor, learn the basics, and start coding

Use AI features with Copilot for free

You can use [Copilot](#) to generate code across multiple files, fix errors, ask questions about your code and much more using natural language.

[Set Up Copilot for Free](#)

Choose your theme

Rich support for all your languages

Tune your settings

Code collects usage data. R

monalisa
Make each project a link and add a new page for each project

GitHub Copilot Generating project.tsx
Modify the Projects component to make each project a link and create the necessary pages for each project.
project.tsx +7 -5
[slug].tsx
Create a dynamic route for each project page
[slug].tsx +10

Working set (2 files)
projects.tsx
[slug].tsx
+ Add Files...
Edit files in your workspace

i You have Windows Subsystem for Linux (WSL) installed on your system. Do you want to install the recommended 'WSL' extension from Microsoft for it?

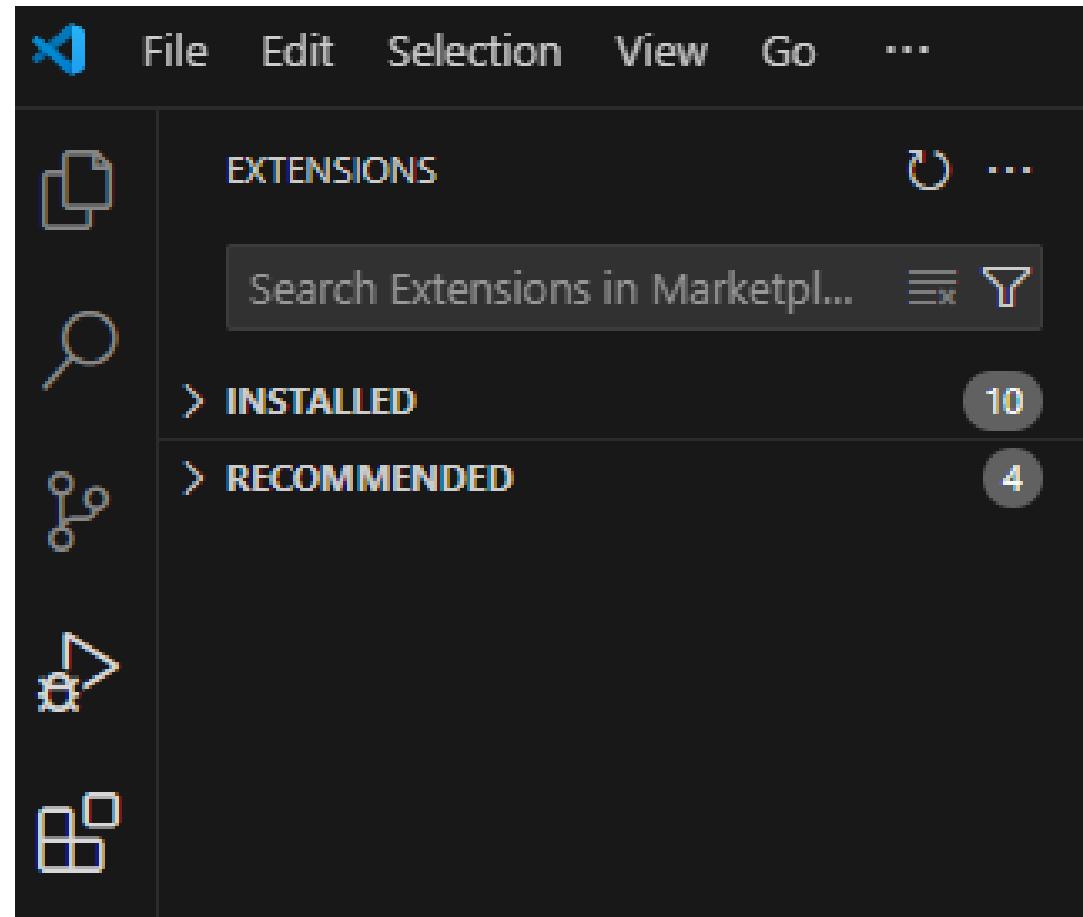
Install Show Recommendations

VISUAL STUDIO CODE (VS CODE)

EXTENSÕES



Extensões essenciais (Ctrl + Shift + X)

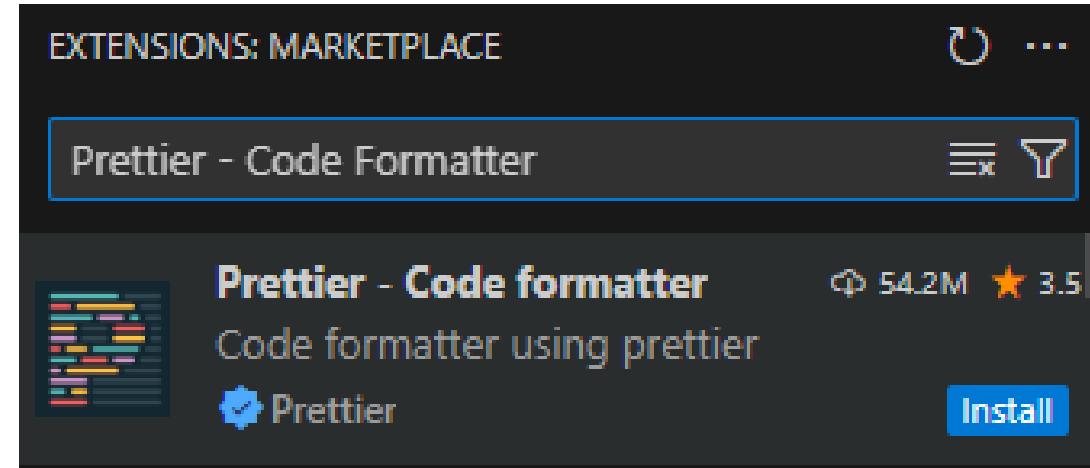
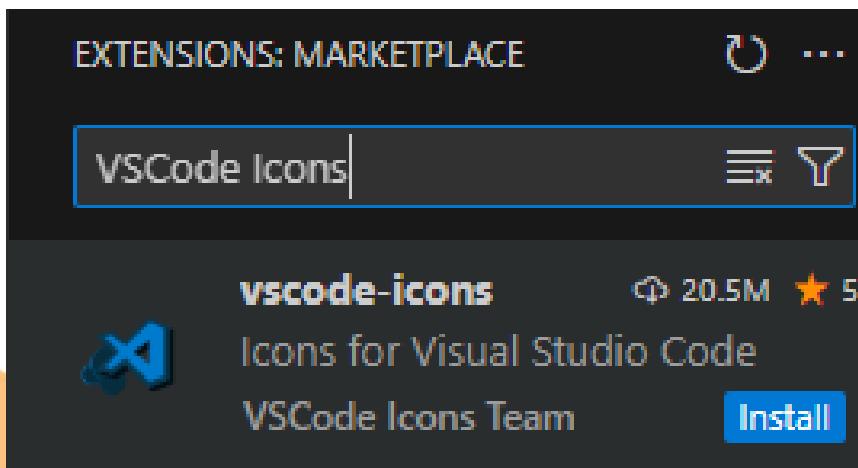


VISUAL STUDIO CODE (VS CODE)



Extensões essenciais (Ctrl + Shift + X)

- **VSCode Icons**: Adiciona ícones personalizados para arquivos e pastas no explorador do VS Code.
- **Prettier - Code Formatter**: Formata automaticamente o código para mantê-lo organizado e padronizado.

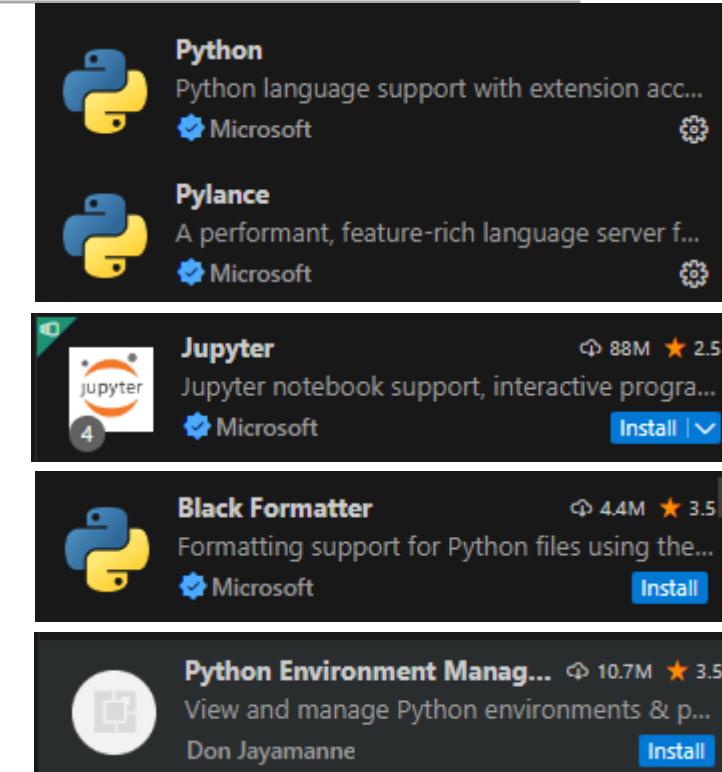


VISUAL STUDIO CODE (VS CODE)



Extensões essenciais para Python

- ✓ **Python (Microsoft)** – suporte completo, incluindo debugging e Jupyter Notebook.
- ✓ **Pylance** – autocompletar e análise de código.
- ✓ **Jupyter** – Permite rodar notebooks Jupyter
- ✓ **Black Formatter** – Formata seguindo boas práticas.
- ✓ **Python Environment Manager** – Gerencia ambientes virtuais.





Algoritmo

Imagine que você quer fazer um bolo. Você segue uma receita que diz passo a passo o que precisa ser feito: pegar os ingredientes, misturar, assar e servir.

Isso é um algoritmo!

Um algoritmo é um **conjunto de instruções bem definidas, organizadas** em uma **sequência lógica**, que resolve um problema ou executa uma tarefa.

Ele pode ser seguido por um **humano** (como em uma receita) ou por um **computador** (como um programa em Python).



ALGORITMO: CARACTERÍSTICAS



Para ser um bom algoritmo, ele precisa ser:

- ✓ **Bem definido:** Cada passo precisa ser claro, sem ambiguidades.
- ✓ **Finito:** Deve ter um começo e um fim.
- ✓ **Executável:** Todas as instruções precisam ser possíveis de realizar.
- ✓ **Preciso:** A sequência de passos deve levar ao objetivo desejado.

ALGORITMO: FAZER UM CAFÉ



ALGORITMO: FAZER UM CAFÉ



- 1** Esquentar a água.
- 2** Colocar o pó de café no filtro.
- 3** Despejar a água quente no filtro.
- 4** Esperar o café coar.
- 5** Servir na xícara.



ALGORITMO: FAZER UM CAFÉ



```
def fazer_cafe():
    print("1. Esquentando a água...")
    print("2. Colocando o pó de café no filtro...")
    print("3. Despejando a água quente no filtro...")
    print("4. Esperando o café coar...")
    print("5. Servindo na xícara. ")
    print("Café pronto! Aproveite!")

fazer_cafe()
```





Algoritmo Constantes e Variáveis





CONSTANTES E VARIÁVEIS

O ARMÁRIO DA PROGRAMAÇÃO!



Imagine que você tem um **armário** em casa.

Dentro dele, você pode **guardar diferentes objetos**, como roupas, livros ou brinquedos.





CONSTANTES E VARIÁVEIS

O ARMÁRIO DA PROGRAMAÇÃO! 🏠📦



Imagine que você tem um **armário** em casa. Dentro dele, você pode **guardar diferentes objetos**, como roupas, livros ou brinquedos.

Agora, pense que **cada prateleira** desse armário pode **armazenar** algo diferente.

Essa ideia nos ajuda a entender **constantes e variáveis** na programação!



CONSTANTES E VARIÁVEIS

O QUE SÃO VARIÁVEIS?



As **variáveis** são como prateleiras do seu armário onde você pode **guardar** algo e **trocar** sempre que precisar.

Por exemplo, imagine que você tem uma **prateleira** chamada "**meu_livro**" e guarda nela um livro chamado "**Python para Iniciantes**".

Se quiser, pode trocar por outro livro depois, como "**Algoritmos Divertidos**".



CONSTANTES E VARIÁVEIS

O QUE SÃO CONSTANTES?



As **constantes** também são como prateleiras no armário, mas com uma diferença importante: depois que colocamos algo nelas, **não podemos trocar**.

Por exemplo, imagine que você tem uma prateleira chamada "**PI**" e nela guarda o número **3.14159**. Esse valor nunca muda, pois é uma constante matemática.

Em Python, por convenção, usamos nomes em **letras maiúsculas** para indicar que um valor não deve ser alterado



CONSTANTES E VARIÁVEIS



TESTE RÁPIDO: VERDADEIRO OU FALSO?



- * Variáveis podem mudar de valor durante a execução do programa.
- * Constantes são valores fixos que não devem ser alterados.
- * Python impede mudanças em constantes automaticamente.



CONSTANTES E VARIÁVEIS



TESTE RÁPIDO: VERDADEIRO OU FALSO?



-
- () Variáveis podem mudar de valor durante a execução do programa.
 - () Constantes são valores fixos que não devem ser alterados.
 - () Python impede mudanças em constantes automaticamente - não bloqueia, mas é uma convenção não mudar!)

VARIÁVEIS E CONSTANTES:

ATIVIDADE 1 - PONDEREM SOBRE



- Todos os armários são do mesmo tamanho e possuem as mesma disposição?
- Um armário é formado apenas por gavetas ou pode conter outro tipo de compartimento?
- Os compartimentos são do mesmo tamanho?
- Por que o armário possui compartimentos de vários tipos?





Uso do Minecraft Education



MINECRAFT EDUCATION



Os alunos entram na plataforma e realizam a
atividade da Unidade 4: Variáveis

Integrated Development Environment



IDE



IDE



Imagine que você precisa **construir uma casa.**

Para isso, você pode usar
ferramentas separadas
(martelo, serra, régua, etc.) **ou**
um **kit completo** que já tem
tudo o que você precisa.



IDE



No mundo da programação, acontece algo parecido!

Podemos escrever código em um **editor de texto simples** e depois rodá-lo manualmente, ou podemos usar uma **IDE (Integrated Development Environment - Ambiente de Desenvolvimento Integrado)**, que é como um kit completo para programadores.



IDE

As IDEs oferecem recursos como:

- ✓ **Editor de código**
- ✓ **Autocompletar e realce de sintaxe**
- ✓ **Depurador (debugger)**
- ✓ **Terminal integrado**
- ✓ **Gerenciamento de pacotes e bibliotecas**

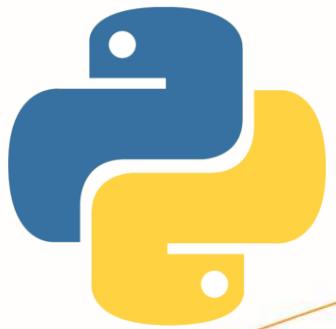
IDE: ATIVIDADE 2

🚀 Agora que sabemos por que é uma **IDE**, realizaremos uma busca na Internet para identificar quais são as mais **indicadas** para programação em **Python** 🐍



IDE: ATIVIDADE 2





Python 

Essa linguagem se tornou uma das mais populares do mundo e é usada desde inteligência artificial até automação de tarefas.

Mas como ela surgiu?
E por que é tão utilizada?

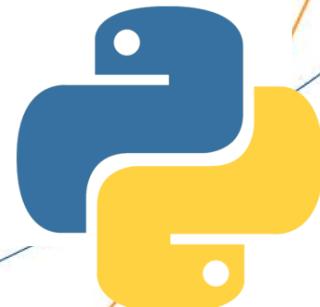




PYTHON: VERSÕES



- Python 1.0 (1991)
- Python 2.0 (2000)
- Python 3.0 (2008) versão mais recente
 - Versão padrão
 - Atualizada
 - Aperfeiçoamento na sintaxe e desempenho
 - Suporte nativo a UTF-8 (codificação)

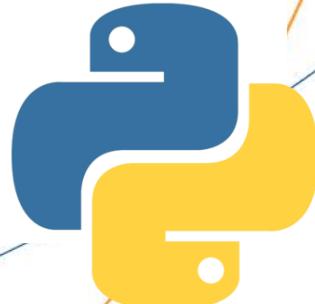




PYTHON: APLICAÇÕES NA INDÚSTRIA



- ✓ Ciência de Dados e Inteligência Artificial (Pandas)
- ✓ Desenvolvimento Web (Django, web2Py e Flask)
- ✓ Automação de Tarefas (Scripting)
- ✓ Desenvolvimento de Jogos (Pygame)
- ✓ Aplicações científicas e engenharia
- ✓ Segurança da informação (Scapy, Paramiko e Nmap)

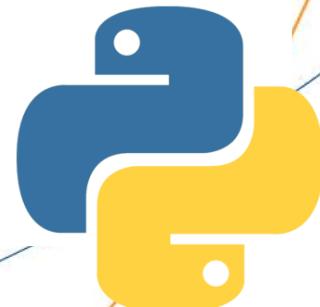




PYTHON: RAZÕES DA POPULARIDADE



- ✓ Sintaxe simples e legível
- ✓ Multiparadigma
- ✓ Bibliotecas poderosas
- ✓ “Já vem com pilhas inclusas”
- ✓ Alta produtividade
- ✓ Comunidade ativa





PYTHON: EM RESUMO



O Python se tornou uma das linguagens mais influentes do mundo, graças à sua simplicidade e versatilidade. Ele está presente na IA, no desenvolvimento web, na automação e muito mais.

Se você quer aprender programação de forma fácil e divertida, Python é o caminho certo!

PYTHON: ATIVIDADE 3 (EXTRA)



🚀 Pesquisem sobre a história da linguagem python, quem a criou, por que foi criada e qual a sua filosofia 🐍



PYTHON: ATIVIDADE 4 (EXTRA)



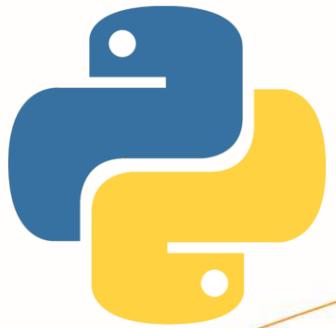
Façam uma pesquisa sobre
a PEP 20 

O que é uma PEP ?

O que é o Zen of Python ?

Cite outra PEP relevante.





Linguagens fortemente e fracamente “tipadas”

LINGUAGENS



Ao discutirmos linguagens de programação, uma das distinções mais significativas é a maneira como elas lidam com os tipos de dados.

Aqui surgem dois conceitos fundamentais: linguagens de programação fortemente tipadas e linguagens de programação fracamente tipadas. Mas, na prática, o que isso realmente implica?



Linguagens Fortemente Tipadas

São aquelas que **não permitem operações entre tipos incompatíveis** sem conversão explícita. Se você tentar somar um número com uma string, por exemplo, a linguagem **vai reclamar e gerar erro!**

Python 

Linguagens Fracamente Tipadas

São linguagens que fazem conversões automáticas de tipos quando necessário. Se você somar um número e uma string, a linguagem tenta resolver sozinha em vez de gerar erro.

JavaScript 

E o Python?

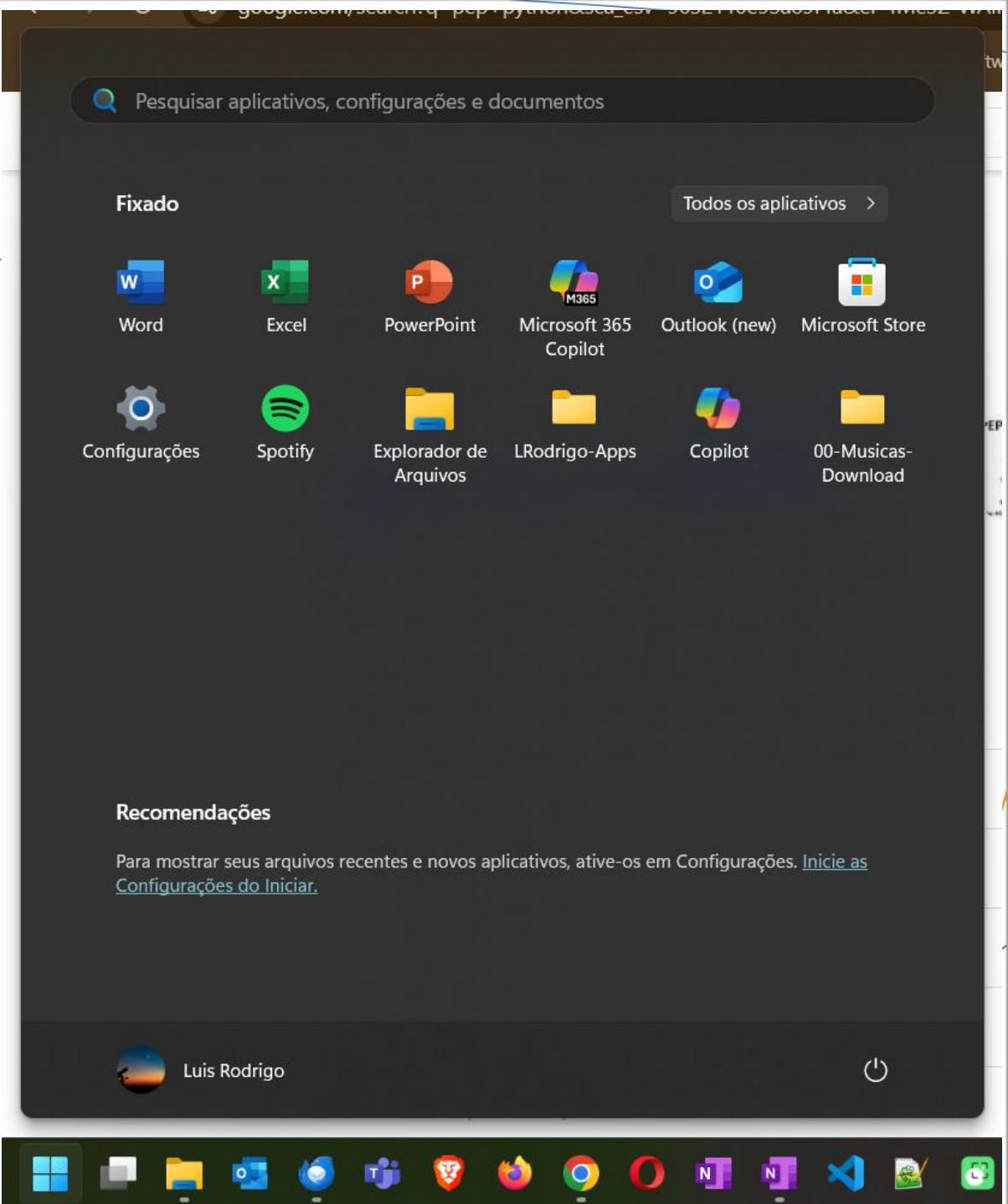
Python é uma linguagem fortemente tipada. Isso significa que não permite operações entre tipos diferentes sem conversão explícita. No entanto, ele é dinamicamente tipado, ou seja, não precisamos declarar o tipo das variáveis, pois ele detecta automaticamente.

Python 

PYTHON: ATIVIDADE 6

Acessando a IDLE pela
primeira vez

CTRL+ESC



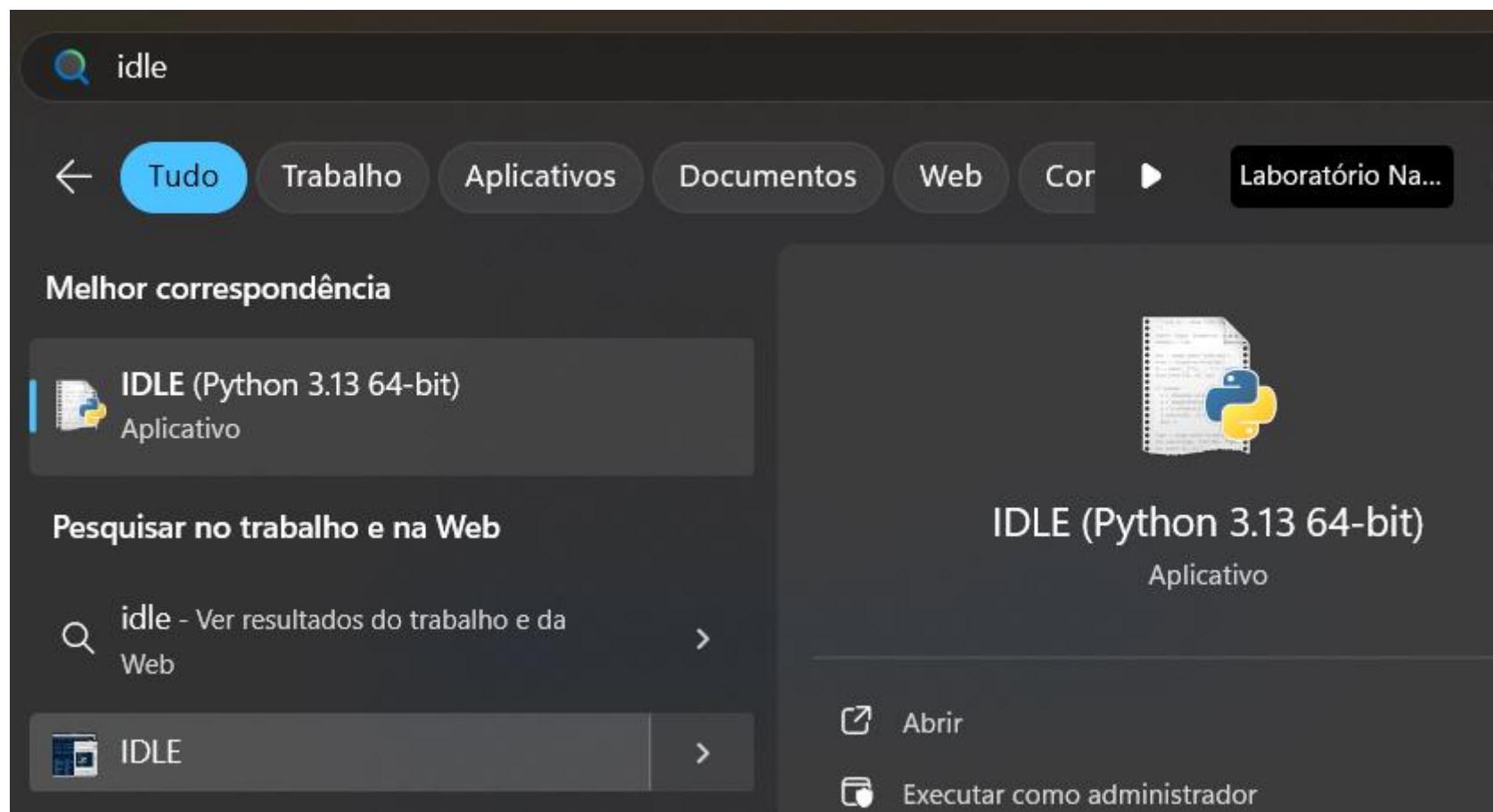
PYTHON: ATIVIDADE 6



Acessando a IDLE pela
primeira vez

CTRL+ESC

idle



PYTHON: ATIVIDADE 6



Acessando a IDLE pela
primeira vez

CTRL+ESC

idle

The screenshot shows the IDLE Shell 3.13.2 window. The title bar reads "IDLE Shell 3.13.2". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the Python version and build information, followed by a command prompt starting with ">>>". The status bar at the bottom right shows "Ln: 3 Col: 0".

```
IDLE Shell 3.13.2
File Edit Shell Debug Options Window Help
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb 4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> |
```

Ln: 3 Col: 0

PYTHON: ATIVIDADE 6



```
numero_inteiro=15
```

```
numero_real=15,15
```

```
texto="cinco"
```

```
type(numero_inteiro)
```

```
type(texto)
```

```
type(numero_real)
```

```
numero_real=15.15
```

```
type(numero_real)
```



PYTHON: ATIVIDADE 6



numero+numero_real

numero+texto

texto*numero

texto2=texto+” ”

texto2*numero



PYTHON: ATIVIDADE 6



```
idade=input("Digite a sua idade ")
type(idade)
help(input)
idade_numero=int(idade)
type(idade_numero)
idade_numero_2=int(input("Qual a sua idade ? "))
type(idade_numero_2)
```



PYTHON: ATIVIDADE 6



```
ANO_ATUAL=2025
```

```
idade_numero_2=int(input("Qual a sua idade ? "))
```

```
ano_nascimento=ANO_ATUAL-idade_numero_2
```

```
print(ano_nascimento)
```



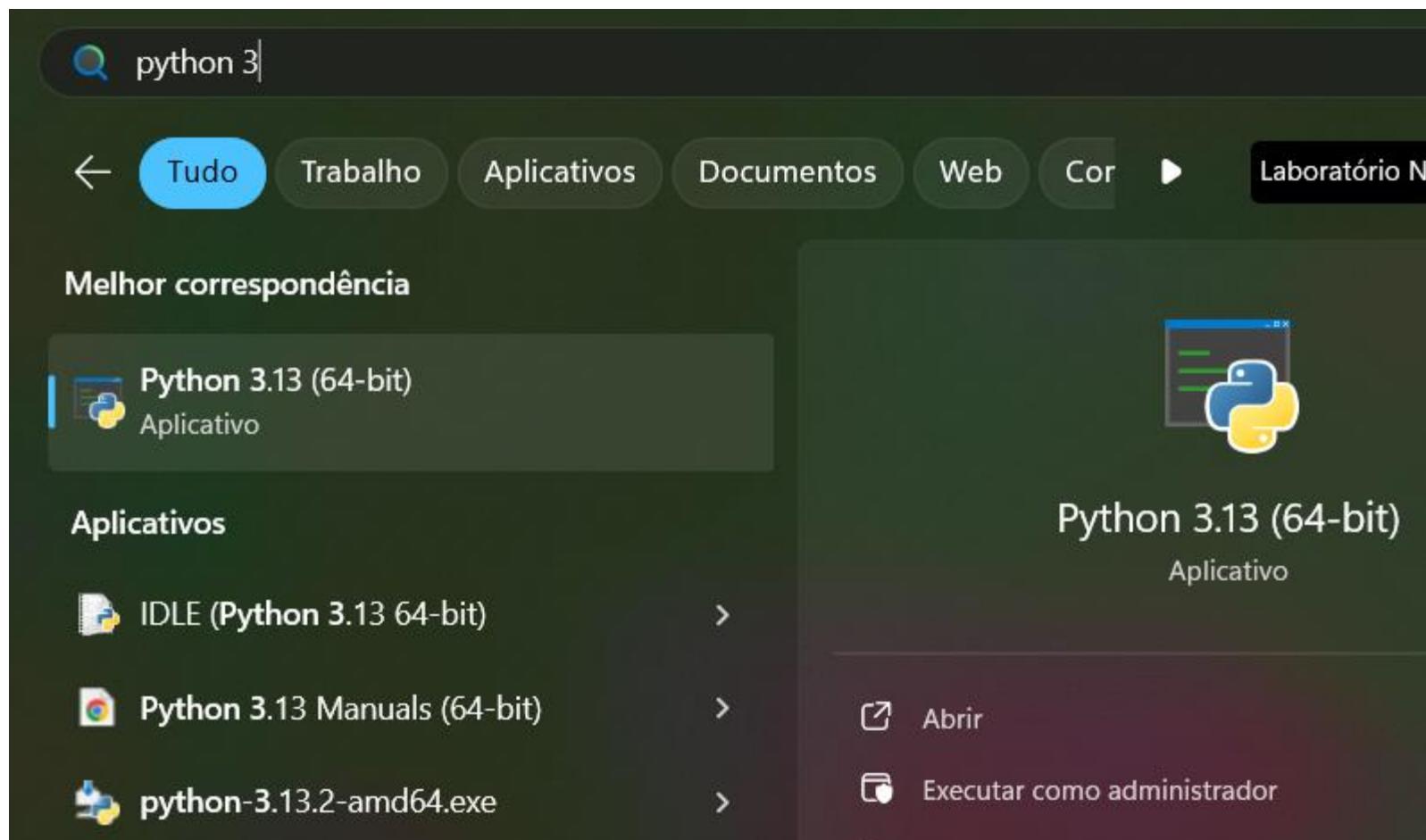
PYTHON: ATIVIDADE 6



Outra forma de acessar a
IDLE

CTRL+ESC

python 3



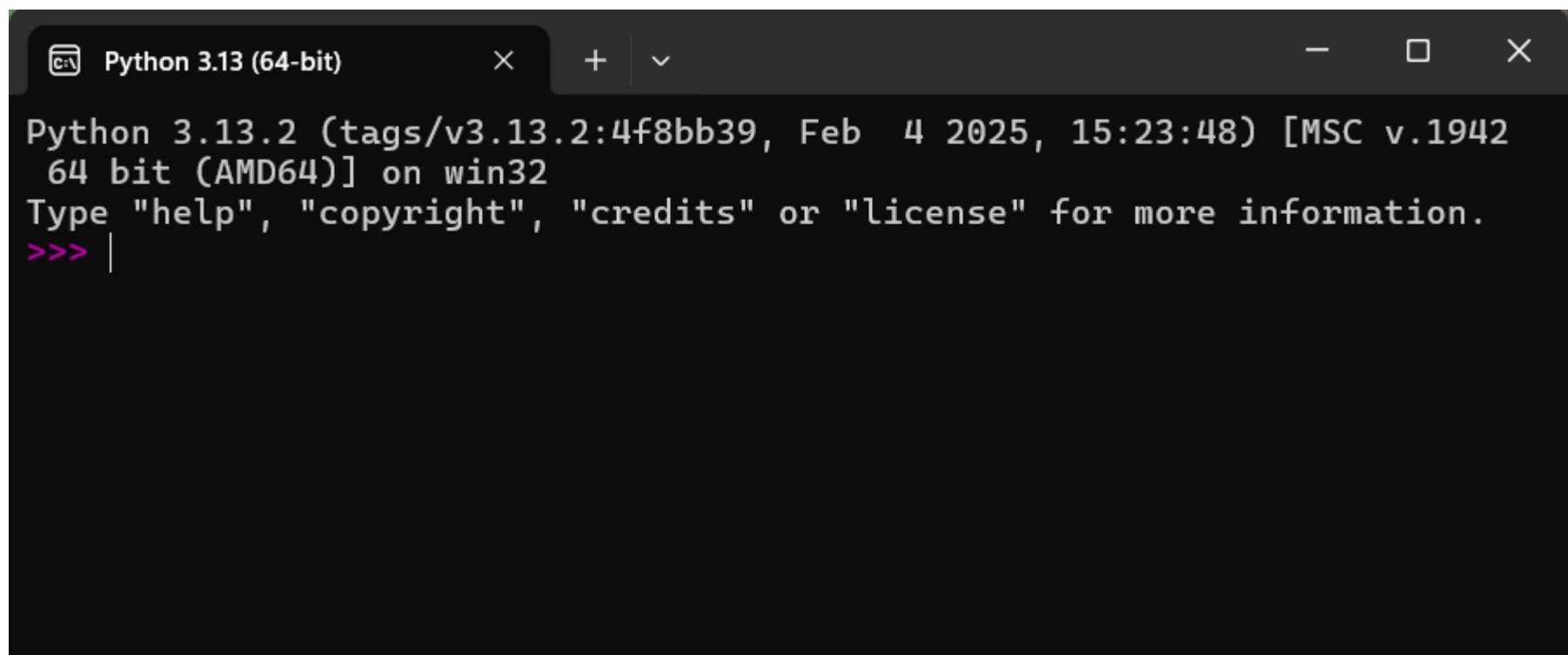
PYTHON: ATIVIDADE 6



Outra forma de acessar a
IDLE

CTRL+ESC

python 3



A screenshot of a terminal window titled "Python 3.13 (64-bit)". The window shows the Python 3.13.2 interpreter running on Windows. The text output is as follows:

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942  
64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>> |
```

PYTHON: ATIVIDADE 7 - VS CODE

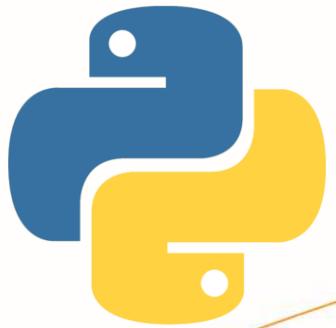


```
meu_livro = "Python para Iniciantes"  
print(meu_livro)
```

```
meu_livro = "Algoritmos Divertidos"  
print(meu_livro)
```

```
PI = 3.14159  
GRAVIDADE = 9.8
```





Linguagens

Compiladas X Interpretadas



LINGUAGENS



Quando falamos sobre linguagens de programação, uma das grandes diferenças está em como o código é executado.

Existem dois grandes grupos: linguagens compiladas e linguagens interpretadas. Mas o que isso significa na prática?





Linguagem Compilada

Uma linguagem compilada necessita passar por um processo de compilação antes da execução do programa. Esse procedimento converte o código-fonte (o que é escrito) em código de máquina (que é compreendido pelo processador).

C e C++

Linguagem Interpretada

Uma linguagem interpretada não precisa de uma etapa de compilação antes da execução. Em vez disso, o código é executado diretamente linha por linha por um interpretador.

Python 



E o Python

Python não é puramente interpretado! Ele funciona de uma forma híbrida



E o Python

- 1** Você escreve o código em Python (.py)
- 2** O interpretador converte para bytecode (.pyc)
- 3** Esse bytecode é executado na PVM

Python é uma linguagem que representa um "equilíbrio" entre linguagens interpretadas e compiladas!

Característica	Compilado 	Interpretado 
Velocidade	 Mais rápido	 Mais lento
Flexibilidade	 Menos flexível	 Mais flexível
Execução	 Código de máquina	 Interpretado na hora
Correção de erros	 Mais difícil	 Mais fácil
Distribuição	 Binário pronto	 Precisa do interpretador



Desafio 2:



DESAFIO 2



Para continuar no processo seletivo da JWC, você deverá fazer seu primeiro código no python com manipulação básica de variáveis.

Luis Rodrigo dot Net
luisrodrigonet

Edit profile

0 followers · 3 following

Petrópolis - RJ
lrodrigo.tec.br

Popular repositories

Python-UC01 (arrow)
Material do Curso de Python - UC1

Python-UC01-Clone
Repositório de teste para clonagem

Public

Public

19 contributions in the last year

Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mon							
Wed							
Fri							

Learn how we count contributions

Contribution activity

<https://github.com/luisrodrigonet/>

luisrodrigonet / Python-UC01

Code Issues Pull requests Actions Projects Wiki Security Insights ...

Python-UC01

Public

Pin Unwatch 1 Fork 0 Starred 1

main Go to file + <> Code

 luisrodrigonet Merge branch 'main' of https://github.c... b63b49a · 3 minutes ago

Aula02 Create Desafio-02.md 3 minutes ago

Aula03 Update Aula03-Tarefa05.md last week

README.md README.md - atualização yesterday

About

Material do Curso de Python - UC1

Readme Activity 1 star 1 watching 0 forks

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

README

Python-UC01

Material de aula utilizado na Unidade 1 curso de Python (2025.1)

luisrodrigonet / Python-UC01

Code Issues Pull requests Actions Projects Wiki Security Insights ...

main · Python-UC01 / Aula02 /

luisrodrigonet Create Desafio-02.md · 97b002f · 4 minutes ago

Name	Last commit message	Last commit date
..		
Desafio-02.md	Create Desafio-02.md	4 minutes ago





Resumindo:



RESUMINDO

-  **Algoritmos: Variáveis e Constantes**
-  **IDE: Conceito e VSCode**
-  **Python: história e aplicação**
-  **Linguagem:**
 -  **fortemente e fracamente tipada**
 -  **compilada e interpretada**