

Introdução ao Python para Análise de Dados

André Grégio
Paulo Almeida



JUSTIÇA 4.0: INOVAÇÃO E EFETIVIDADE NA REALIZAÇÃO DA JUSTIÇA PARA TODOS
PROJETO DE EXECUÇÃO NACIONAL BRA/20/015

1. Sobre o Curso





Prof. Dr. André Grégio

- Doutor em Engenharia da Computação.
- Professor Pesquisador no Depto de Informática UFPR.
- Segurança Computacional e Ciência de Dados.
- gregio@ufpr.br



Prof. Dr. Paulo Almeida

- Doutor em Ciência da Computação.
- Professor Pesquisador no Depto de Informática UFPR.
- Machine Learning e Processamento de Imagens.
- paulorla@ufpr.br



Cláudio Torres Junior

- Cursando o 8º período de Informática Biomédica na UFPR.
- Bolsista no C3SL - Centro de Computação Científica e Software Livre
- claudiotorresjunior@gmail.com

Objetivo do Curso

- ▶ Aprender a utilizar a linguagem de programação Python para análise de dados.
- ▶ Entender os elementos e estruturas básicas da programação
- ▶ Criar soluções (programas) para problemas que podem ser resolvidos com o auxílio de um computador

Como serão as aulas

Estudos pré-aula dos conteúdos	
Aula ao vivo - Terças 8h30 - 10h00	
Plantão ao vivo de dúvidas - Quintas 8h30 - 9h30	Estudo individual, com envio de dúvidas ao fórum

Vamos precisar de...

Um computador convencional
Com qualquer sistema
operacional

Python 3

VSCode

Veja o tutorial de como instalar

Você precisa...

Ter conhecimentos básicos

Criar diretórios

Saber o que é um arquivo e
como organizá-los

Instalar programas...

Fazer o solicitado nas aulas

Seguindo o cronograma

Aprender Python é fácil



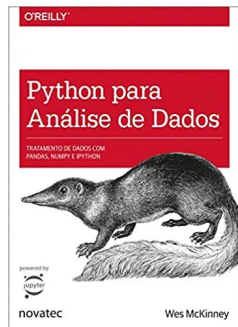
**Programação
Básica**

**Lidando
com dados**

**Bibliotecas
para Análise**

Bibliografia recomendada

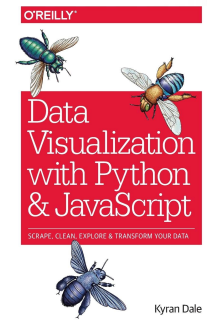
McKinney, W. Python para análise de dados: Tratamento de dados com Pandas, NumPy e IPython. 2019.



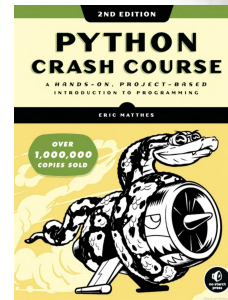
Chen, D. Análise de dados com Python e Pandas. Novatec Editora. 2018.



Dale, K. Data Visualization with Python and JavaScript: Scrape, Clean, Explore & Transform Your Data. O'Reilly Media. 2016.



Matthes, E. Python Crash Course: A Hands-On, Project-Based Introduction to Programming. 2a ed. No Starch Press. 2019.



2. Primeiros Passos



Versão do Python

Será usado Python 3 na disciplina

Versão mais recente do Python

Cuidado: Python 3 não é compatível com Python 2, e vice-versa

Versão do Python

Em um terminal digite

```
python --version
```

```
python3 --version
```

O comando que gerar como resposta Python 3.X é o comando que você deve usar em seu computador para acessar o interpretador Python

Interpretador

- ▶ Para acessar o interpretador Python, digite em um terminal

`python3`

Assumindo que no slide anterior o comando que gerou a saída correta foi `python3`, caso contrário, use `python`.

- ▶ Exemplo de resultado

```
python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Saindo

Para sair do interpretador, insira o comando

```
quit()
```

Interpretador

- ▶ O interpretador aceita comandos Python, e os executa assim que você tecla enter

```
python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- ▶ O comando `print("SEU TEXTO")` imprime na tela o texto solicitado

Interpretador

```
python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("ola mundo")
ola mundo
>>>
```


Realizando Operações

O interpretador é capaz de realizar operações e exibir o resultado

Alguns operadores que podem ser usados:

+	Adição
-	Subtração
/	Divisão
*	Multiplicação
**	Exponenciação
%	Módulo (resto da divisão)

Realizando Operações

```
python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("ola mundo")
ola mundo
>>> 2**3
8
>>> 5%3
2
>>> 5%2
1
>>> 3+4
7
>>>
```

Variáveis

Uma variável **armazena** um valor na memória de trabalho do computador

Para criar uma variável, basta dar um **nome** para ela, seguido de um **=** para atribuir um valor

Variáveis

```
>>> qtde_alunos = 30  
>>> nome = "Maria"  
>>> pi = 3.14
```

Variáveis

```
>>> qtde_alunos = 30  
>>> nome = "Maria"  
>>> pi = 3.14
```

Nome da variável

Valor armazenado

Variáveis

```
>>> qtde_alunos = 30  
>>> nome = "Maria"  
>>> pi = 3.14
```

Na computação, chamamos frases de **strings**. Se a variável deve armazenar uma string, coloque a string entre aspas duplas.

Variáveis

Cada variável no seu programa deve ter um nome **único**

Nomes de variáveis **devem**:

- ▶ Conter somente letras, números e underscores (“_”)
- ▶ Começar com letra ou underscore, nunca números
- ▶ Não possuir espaços em branco

Variáveis

Cada variável no seu programa deve ter um nome **único**

Nomes de variáveis **devem**:

- ▶ Conter somente letras, números e underscores (“_”)
- ▶ Começar com letra ou underscore, nunca números
- ▶ Não possuir espaços em branco

Exemplo de nomes **inválidos**



```
>>> 1frase = "uma string qualquer"
>>> frase 1 = "outra string"
>>> num# = 50
>>> 2 = 'c'
```


Variáveis

Use nomes descritivos e curtos. Por exemplo, uma variável que vai armazenar a quantidade de alunos na sala pode ser

`qtde_alunos`

Alguns nomes **ruins para essa** variável

`quantidade_de_alunos_na_sala`

`x`

`nave`

Variáveis

NÃO usar palavras reservadas (keywords) e nomes de funções de Python

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			

O que é uma variável

Seu computador possui uma memória de trabalho

No seu computador pessoal, geralmente (e imprecisamente)
essa memória é chamada de “Memória RAM”

O que é uma variável

A memória pode ser vista como um vetor

Ou seja, várias “caixinhas”, uma seguida da outra

Podemos assumir que no começo do programa, nenhuma “caixinha” está sendo usada

Memória Principal



O que é uma variável

Quanto é criada uma variável, o interpretador pega uma “caixinha” vazia, e dá esse nome para a caixinha. Agora os valores atribuídos à variável são armazenados nela

```
>>> pi = 3.14
```

Memória Principal

pi

3.14						
------	--	--	--	--	--	--

...

Imprimindo

Você pode imprimir o valor de uma variável usando `print(nome_variavel)`

Não use aspas

Você pode alterar o valor da variável, ou fazer contas com o valor armazenado na variável

Exemplo

```
>>> qtde_alunos = 200  
>>> print(qtde_alunos)  
200
```

Memória Principal

qtde_alunos

200							...
-----	--	--	--	--	--	--	-----

Exemplo

```
>>> qtde_alunos = 200
>>> print(qtde_alunos)
200
>>> qtde_alunos = 220
>>> print(qtde_alunos)
220
```

Memória Principal

qtde_alunos

220						
-----	--	--	--	--	--	--

...

Exemplo

```
>>> qtde_alunos = 200
>>> print(qtde_alunos)
200
>>> qtde_alunos = 220
>>> print(qtde_alunos)
220
>>> alunos_extr = 30
>>> qtde_alunos = qtde_alunos + alunos_extr
>>> print(qtde_alunos)
250
```

Memória Principal

qtde_alunos alunos_extr

250	30					
-----	----	--	--	--	--	--

...

Tipos de variáveis

O tipo de uma variável diz o tipo de dado que ela armazena

Em Python, a **tipagem é dinâmica**

Inferre automaticamente qual o tipo da variável na memória

Tipos de variáveis

Para descobrir o tipo de uma variável no momento, use a função

```
>>> type(nome_variavel)
```

Exemplo

```
>>> peso = 10  
>>> type(peso)  
<class 'int'>
```

A variável peso armazena números **inteiros** no momento.

Exemplo

```
>>> peso = 10
>>> type(peso)
<class 'int'>
>>> peso = 10.3
>>> type(peso)
<class 'float'>
```

A variável peso armazena um número em **ponto flutuante** no momento. Um ponto-flutuante é uma **aproximação** para se representar um número racional.

Exemplo

```
>>> peso = 10
>>> type(peso)
<class 'int'>
>>> peso = 10.3
>>> type(peso)
<class 'float'>
>>> nome = "Maria"
>>> type(nome)
<class 'str'>
```

A variável nome armazena uma **string**.

3. Criando Scripts



Scripts

Um script é uma série de comandos que são executados **em ordem, de cima para baixo**

Em outras palavras, estamos criando um **programa em Python**

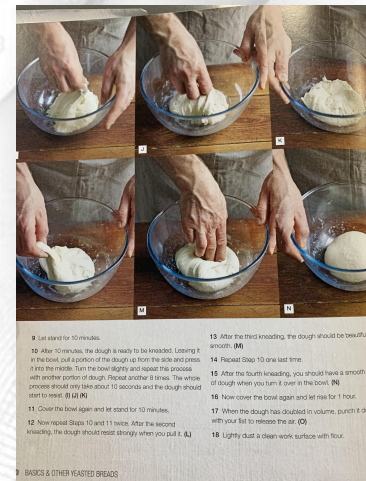
Ou como preferimos na computação, estamos implementando um **algoritmo**

Breve revisão de algoritmos

Procedimento computacional bem definido para:

1. Sistematizar instruções/**atividades repetitivas**
2. Obter uma **solução** para um determinado tipo de **problema**
 - a. Recebe algum valor (ou conjunto de valores) de **ENTRADA**
 - b. Produz algum valor (ou conjunto de valores) de **SAÍDA**

Algoritmo é a **sequência de passos** que transforma a **entrada** na **saída** e deve poder ser **reproduzido!**



Scripts

Um script é um arquivo de texto com comandos

Utilize **qualquer editor de texto simples** (e.g., notepad) para criar o arquivo

IDE

Alguns editores possuem mais “recursos” apropriados para programação. Esse tipo de editor também é chamado de IDE (Integrated Development Environment):

Exemplos com suporte para python

Visual Studio Code (code.visualstudio.com)

MU (codewith.mu)

Sublime (www.sublimetext.com)

Geany (www.geany.org)

IDLE (vem com o Python no Mac OS e Windows, apt no Linux)

PyCharm (www.jetbrains.com/pycharm)

IDE

Alguns editores possuem mais “recursos” apropriados para programação. Esse tipo de editor também é chamado de IDE (Integrated Development Environment):

Exemplos com suporte para python

Visual Studio Code (code.visualstudio.com)

→ Será usado nas aulas, mas você pode trocar por qualquer outro que te agrade.

MU (codewith.mu)

Sublime (www.sublimetext.com)

Geany (www.geany.org)

IDLE (vem com o Python no Mac OS e Windows, apt no Linux)

PyCharm (www.jetbrains.com/pycharm)

IDE

Objetivos do Editor de Texto/IDE:

- Criar programas completos
- Salvar os programas para execução posterior
- *Highlighting* para facilitar leitura (cores para comandos/palavras reservadas)
- Execução e depuração em tempo real
- Automação, atalhos
- Adição de funcionalidades via *plugins*

Primeiro Script

Crie um diretório em seu computador, em um lugar de sua preferência

Esse diretório vai conter o **seu projeto**

Dicas:

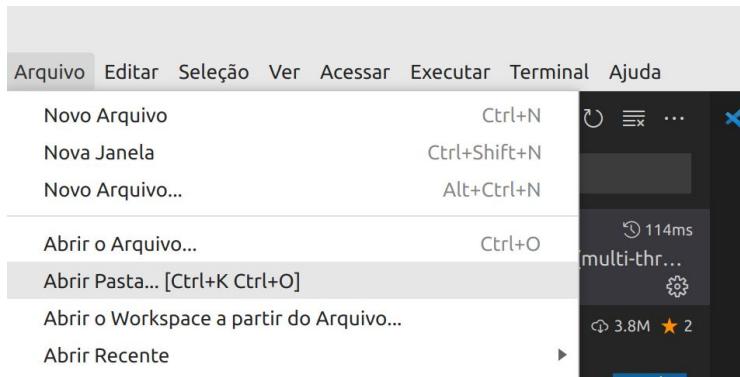
- ▶ Crie em um local fácil para você lembrar e acessar
- ▶ Use o tutorial disponibilizado neste curso

Primeiro Script

Abra o VSCode

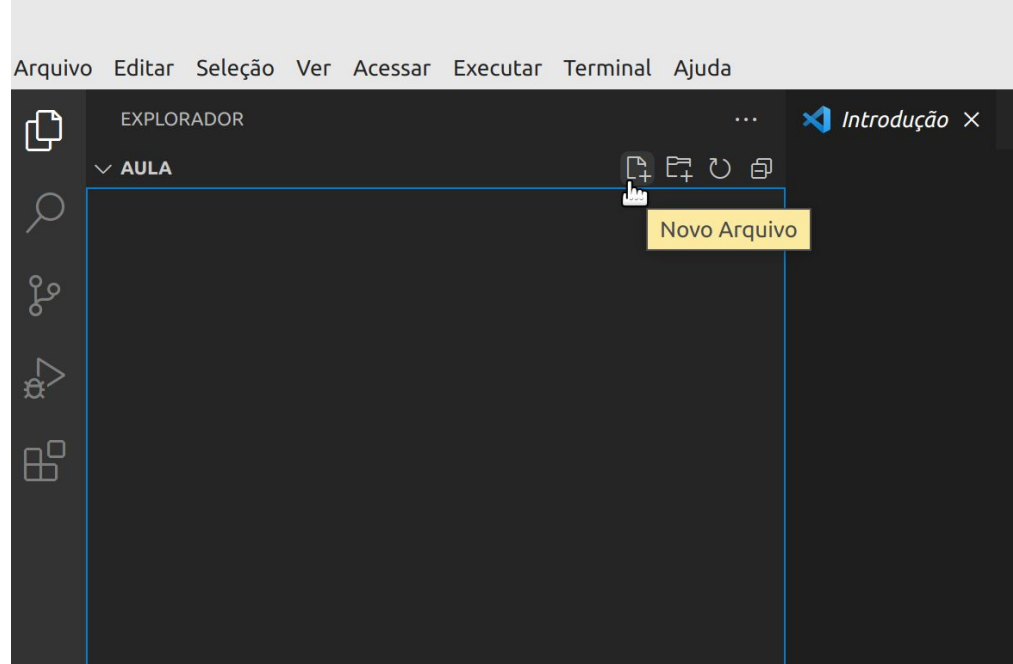
Menu *Arquivo* -> *Abrir Pasta ...*

Abra a pasta recém criada



Primeiro Script

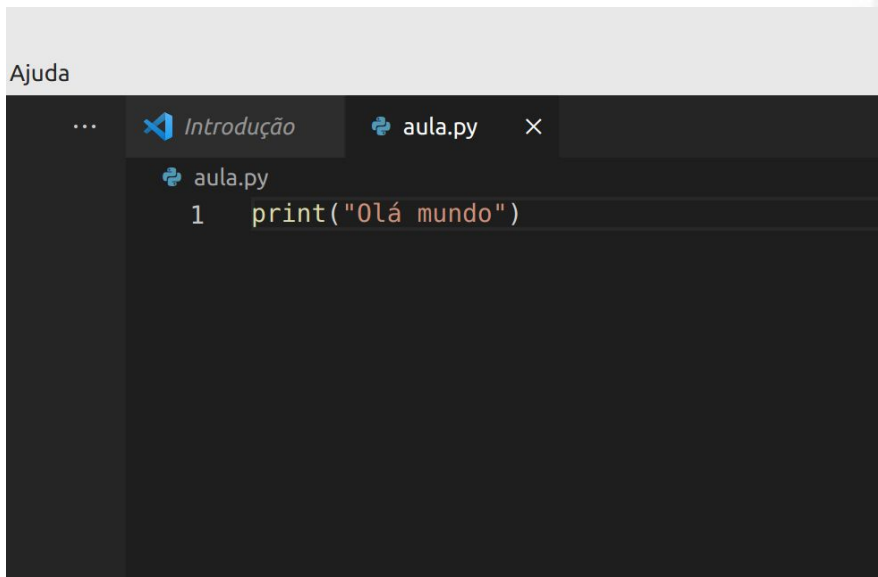
Crie um arquivo chamado aula.py



Em Python, é obrigatório que arquivos de script terminem com a extensão .py

Primeiro Script

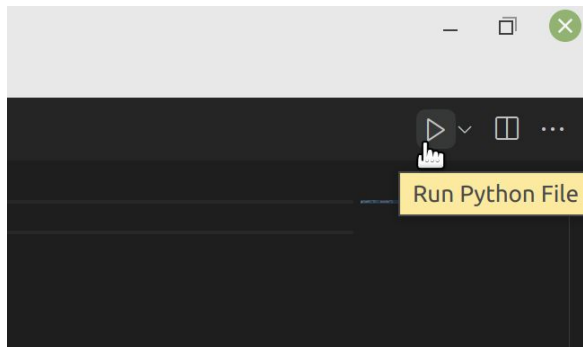
Insira o seguinte comando no arquivo aula.py

A screenshot of a code editor interface. At the top, there is a tab labeled 'aula.py' with a blue icon to its left and a close button to its right. Below the tab, the code editor shows a single line of Python code: `1 print("Olá mundo")`. The text is white on a dark background. Above the editor, there is a grey bar with the word 'Ajuda' in white text. The background of the slide features a faint, stylized graphic of a document with binary code (0s and 1s) and a blue vertical bar on the right side.

```
Ajuda
...
Introdução
aula.py
aula.py
1 print("Olá mundo")
```

Primeiro Script

Execute



```
PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL

Ola mundo
paulo@paulo-desk:~/aula$ /bin/python3 /home/paulo/aula/aula.py
Ola mundo
paulo@paulo-desk:~/aula$ /bin/python3 /home/paulo/aula/aula.py
Ola mundo
paulo@paulo-desk:~/aula$ /bin/python3 /home/paulo/aula/aula.py
Ola mundo
paulo@paulo-desk:~/aula$ /bin/python3 /home/paulo/aula/aula.py
Ola mundo
paulo@paulo-desk:~/aula$ /bin/python3 /home/paulo/aula/aula.py
Olá mundo
paulo@paulo-desk:~/aula$
```

Lendo do Teclado

Para solicitar um dado do usuário e armazenar o que ele digitou em uma variável, usa-se a função *input*

```
>>> variavel_destino = input("Texto opcional para exibir")
```

Lendo do Teclado

```
print("Olá mundo!")  
nome = input("Digite seu nome: ")  
print("Seu nome é", nome)
```

Lendo do Teclado

```
print("Olá mundo!")
nome = input("Digite seu nome: ")
print("Seu nome é", nome)
idade = input("Digite sua idade: ")
idade_apos = input("Com quantos anos você quer aposentar: ")
anos_apos = idade_apos - idade
print("Faltam", anos_apos, " para você aposentar")
```

Lendo do Teclado

```
print("Olá mundo!")
nome = input("Digite seu nome: ")
print("Seu nome é", nome)
idade = input("Digite sua idade: ")
idade_apos = input("Com quantos anos você quer aposentar: ")
anos_apos = idade_apos - idade
print("Faltam", anos_apos, " para você aposentar")
```

Oops!

Traceback (most recent call last):

File "/home/usuario/aula/aula.py", line 6, in <module>

anos_apos = idade_apos - idade

TypeError: unsupported operand type(s) for -: 'str' and 'str'

Python é estranho

Python assume que tudo que é lido do teclado é uma **string**
Isto é, um texto, ou melhor, uma **sequência de caracteres**

```
idade = input("Digite sua idade: ")  
idade_apos = input("Com quantos anos você quer aposentar: ")  
anos_apos = idade_apos - idade
```



Subtrair um texto de outro não faz sentido!

Lendo do tipo certo

Se você quer que a variável armazene o dado como outro tipo que não string, leia da seguinte forma

```
>>> variavel_destino = tipo(input("Texto opcional para exibir"))
```

Substituindo *tipo* pelo tipo correto, como int ou float

Isso fará uma **conversão** de string para o tipo solicitado

Exemplo

```
print("Olá mundo!")
nome = input("Digite seu nome: ")
print("Seu nome é", nome)
idade = int(input("Digite sua idade: "))
idade_apos = int(input("Com quantos anos você quer aposentar: "))
anos_apos = idade_apos - idade
print("Faltam", anos_apos, " para você aposentar")
```

Operações sobre variáveis

Algumas das operações mencionadas anteriormente funcionam em variáveis do tipo *string*

```
>>> frase1 = "Curso de"
>>> frase2 = "Python básico"
>>> frase1 + ' ' + frase2
'Curso de Python básico'
```

```
>>> letra = 'a'
>>> print(10*letra)
aaaaaaaaaa
```

Comentários

Em Python, um comentário de código começa com #

```
#A linha abaixo lê o nome do usuário da entrada padrão  
nome = input("Digite seu nome: ")
```

Documentação

Se você precisar da documentação de uma função por exemplo, pode

Procurar nos livros recomendados

Procurar em na documentação oficial

<https://docs.python.org/pt-br>

Procurar na internet

Mas cuidado, a internet é repleta de pseudoespecialistas!

4.

Você não precisa de uma IDE



Executando o script

O VSCode facilita (muito) a criação e execução do script, mas você não precisa dele

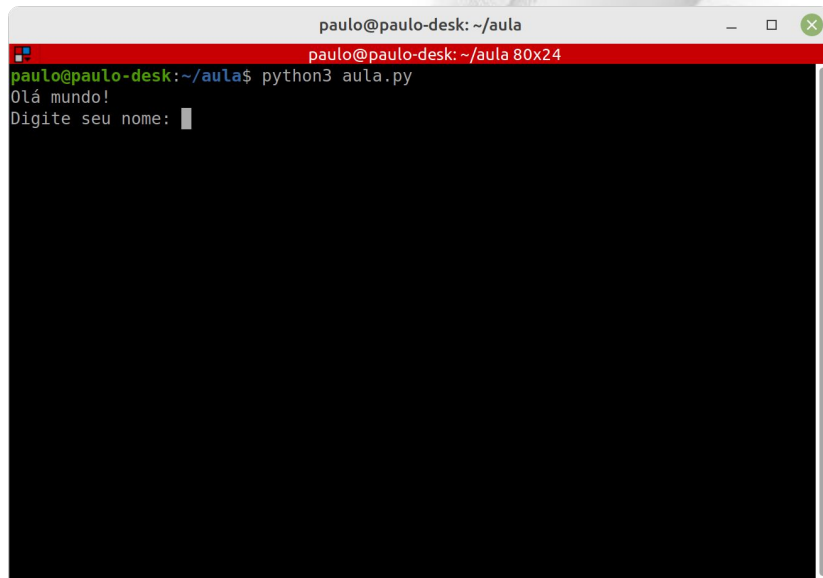
Um *script* nada mais é do que um arquivo de texto

Executando o script no Linux

Abra um terminal onde o arquivo está salvo

Para executar o script sem precisar de nenhuma IDE, basta executar o comando

```
python3 nome_seu_script.py
```

A terminal window titled 'paulo@paulo-desk: ~/aula' with a red title bar. The window shows the command 'python3 aula.py' being executed. The output of the script is 'Olá mundo!' followed by a prompt 'Digite seu nome: ' with a cursor. The terminal background is black, and the text is white. The window has standard Linux window controls (minimize, maximize, close) in the top right corner.

```
paulo@paulo-desk: ~/aula
paulo@paulo-desk: ~/aula 80x24
paulo@paulo-desk: ~/aula$ python3 aula.py
Olá mundo!
Digite seu nome: 
```

Executando o script no Windows

PowerShell

```
Windows PowerShell

PS C:\Users\abedg\UFPR> more .\teste.py.txt
a = 1
b = 2
print(a+b)

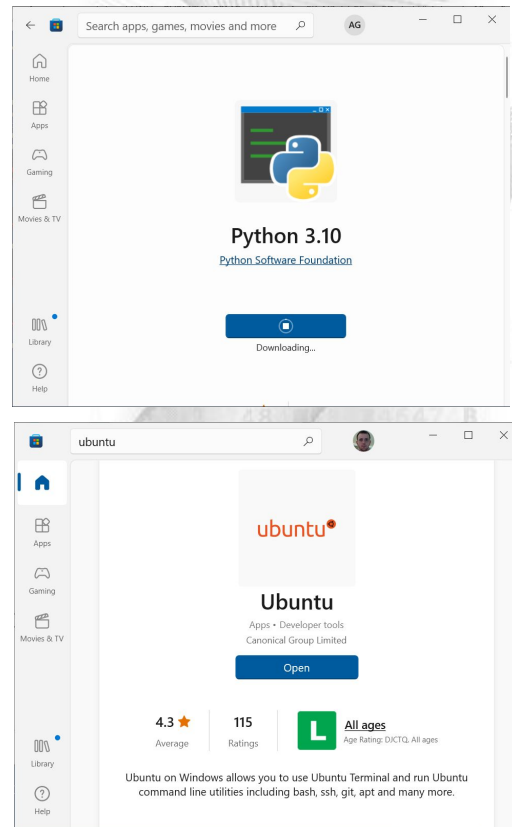
PS C:\Users\abedg\UFPR> python3.exe .\teste.py.txt
3

PS C:\Users\abedg\UFPR>
```

Ubuntu no Windows

```
gregio@DESKTOP-RMOQK12: ~
gregio@DESKTOP-RMOQK12:~$ cat teste.py
#!/usr/bin/python3

print("oi")
gregio@DESKTOP-RMOQK12:~$ python3 teste.py
oi
gregio@DESKTOP-RMOQK12:~$
```



5. Importando



Importando

É possível **importar** bibliotecas com funções já prontas

Existem diversas formas

Dica: a **primeira** coisa presente no seu script devem ser as importações

Formato

```
import nome_arquivo
```

Math

A biblioteca math adiciona diversas funções relacionadas a operações matemáticas

Veja a documentação

docs.python.org/3/library/math.html

Raiz Quadrada

```
import math

valor = 4
raiz = math.sqrt(valor)
print("A raiz de ", valor, "é", raiz)
```

Raiz Quadrada

```
import math

valor = 4
raiz = math.sqrt(valor)
print("A raiz de ", valor, "é", raiz)
```

A função sqrt da biblioteca math vai calcular a raiz quadrada da variável valor, e armazenar o resultado na variável raiz.

6. Exercícios



Executando o script

1. Replique tudo que foi ensinado durante as aulas no seu computador de casa para fixar os conhecimentos.
2. Resolva os exercícios postados no site da disciplina para testar os seus conhecimentos.

7.

Bibliografia adicional



Leitura recomendada

[Think Python 2e – Green Tea Press](#)

(Em português: <https://penseallen.github.io/PensePython2e/>)





Obrigado!

Bons Estudos!!!