



Pró-reitora de Pós-Graduação, Pesquisa e Inovação
Especialização em Ciências de Dados e Analytics

Programação para Ciência de Dados

Parte 1

Agenda Parte 1

- Instalando o ambiente
- Noções básicas de Programação Estruturada em Python
- Conceitos de Orientação a Objetos
- Conceitos de Estruturas de Dados
- Exercícios

Python

- De acordo com Guido van Rossum, Python é:

“high-level programming language, and its core design philosophy is all about code readability and a syntax which allows programmers to express concepts in a few lines of code.”

www.python.org

The screenshot shows the Python.org website's download page. The browser's address bar displays 'https://www.python.org/downloads/'. The page features a dark blue header with the Python logo and a search bar. Below the header, a navigation bar includes links for 'About', 'Downloads', 'Documentation', 'Community', 'Success Stories', 'News', and 'Events'. The main content area is titled 'Download the latest version for Windows' and includes a yellow button for 'Download Python 3.7.2'. It also provides links for other operating systems and pre-releases. A section titled 'Looking for a specific release?' lists recent Python versions in a table.

Release version	Release date	Click for more
Python 3.7.2	2018-12-24	Download Release Notes
Python 3.6.8	2018-12-24	Download Release Notes
Python 3.7.1	2018-10-20	Download Release Notes

Python 3.7.2 documentation

Python Software Foundation [US] | https://docs.python.org/3/

Apps | Bookmarks | Convert PDF to Word | Addons | Ciência da Computaç | App Web Recife | IAPR - International A | DSDocs » Login | DSDI - Indexação | DSDI - Administração | Google Translate | Outros favoritos

Python » English 3.7.2 Documentation »

Quick search Go | modules | index

Download

Download these documents

Docs by version

Python 3.8 (in development)
Python 3.7 (stable)
Python 3.6 (stable)
Python 3.5 (security-fixes)
Python 2.7 (stable)
All versions

Other resources

PEP Index
Beginner's Guide
Book List
Audio/Visual Talks

Python 3.7.2 documentation

Welcome! This is the documentation for Python 3.7.2.

Parts of the documentation:

What's new in Python 3.7?
or all "What's new" documents since 2.0

Tutorial
start here

Library Reference
keep this under your pillow

Language Reference
describes syntax and language elements

Python Setup and Usage
how to use Python on different platforms

Python HOWTOs
in-depth documents on specific topics

Indices and tables:

Global Module Index
quick access to all modules

Installing Python Modules
installing from the Python Package Index & other sources

Distributing Python Modules
publishing modules for installation by others

Extending and Embedding
tutorial for C/C++ programmers

Python/C API
reference for C/C++ programmers

FAQs
frequently asked questions (with answers!)

https://docs.python.org/3/

Programação para
Ciência de Dados

Especialização em
Ciências de Dados e Analytics

Prof. Dr. Byron Leite

5

UPE

The Python Tutorial — Python 3.7

Python Software Foundation [US] | <https://docs.python.org/3/tutorial/index.html>

AppsBookmarksConvert PDF to WordAddonsCiência da ComputaçãoApp Web RecifeIAPR - InternationalDSDocs » LoginDSDI - IndexaçãoDSDI - AdministraçãoGoogle TranslateOutros favoritos

The [Glossary](#) is also worth going through.

- 1. Whetting Your Appetite
- 2. Using the Python Interpreter
 - 2.1. Invoking the Interpreter
 - 2.1.1. Argument Passing
 - 2.1.2. Interactive Mode
 - 2.2. The Interpreter and Its Environment
 - 2.2.1. Source Code Encoding
- 3. An Informal Introduction to Python
 - 3.1. Using Python as a Calculator
 - 3.1.1. Numbers
 - 3.1.2. Strings
 - 3.1.3. Lists
 - 3.2. First Steps Towards Programming
- 4. More Control Flow Tools
 - 4.1. `if` Statements
 - 4.2. `for` Statements
 - 4.3. The `range()` Function
 - 4.4. `break` and `continue` Statements, and `else` Clauses on Loops
 - 4.5. `pass` Statements
 - 4.6. Defining Functions
 - 4.7. More on Defining Functions
 - 4.7.1. Default Argument Values
 - 4.7.2. Keyword Arguments
 - 4.7.3. Arbitrary Argument Lists
 - 4.7.4. Unpacking Argument Lists
 - 4.7.5. Lambda Expressions
 - 4.7.6. Documentation Strings
 - 4.7.7. Function Annotations
 - 4.8. Intermezzo: Coding Style
- 5. Data Structures
 - 5.1. More on Lists
 - 5.1.1. Using Lists as Stacks
 - 5.1.2. Using Lists as Queues
 - 5.1.3. List Comprehensions
 - 5.1.4. Nested List Comprehensions

<https://docs.python.org/3/tutorial/index.html>

Programação para
Ciência de Dados

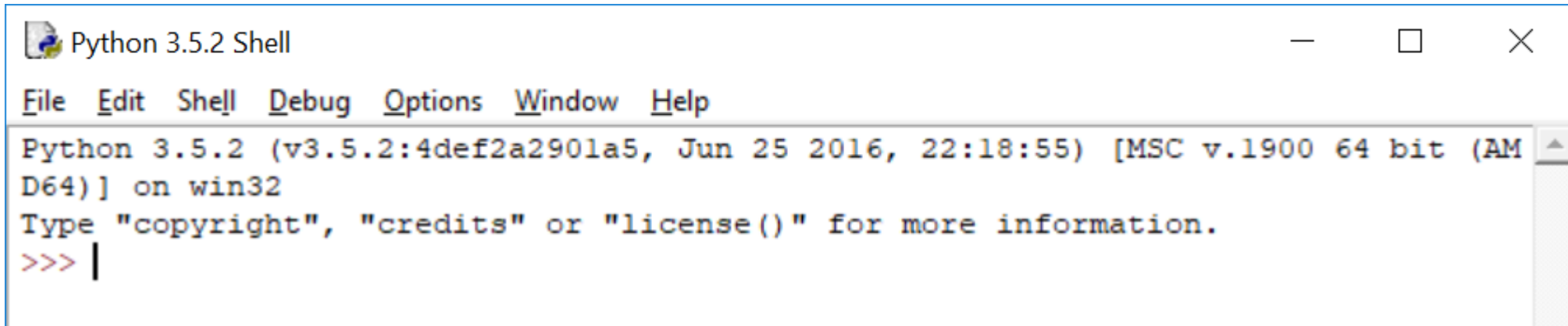
Especialização em
Ciências de Dados e Analytics

Prof. Dr. Byron Leite

6

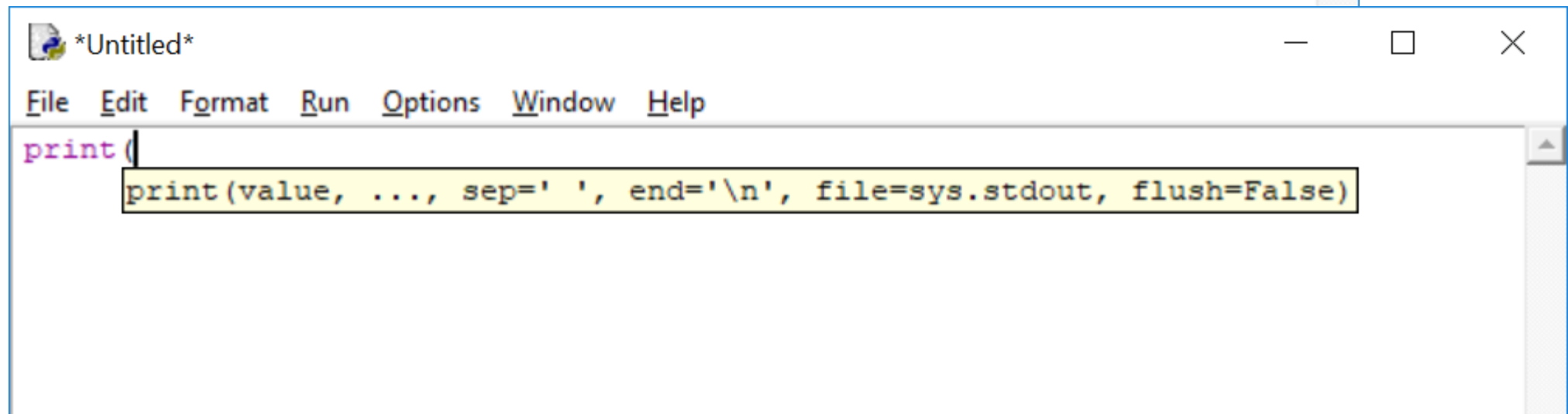
UPE

Python Shell



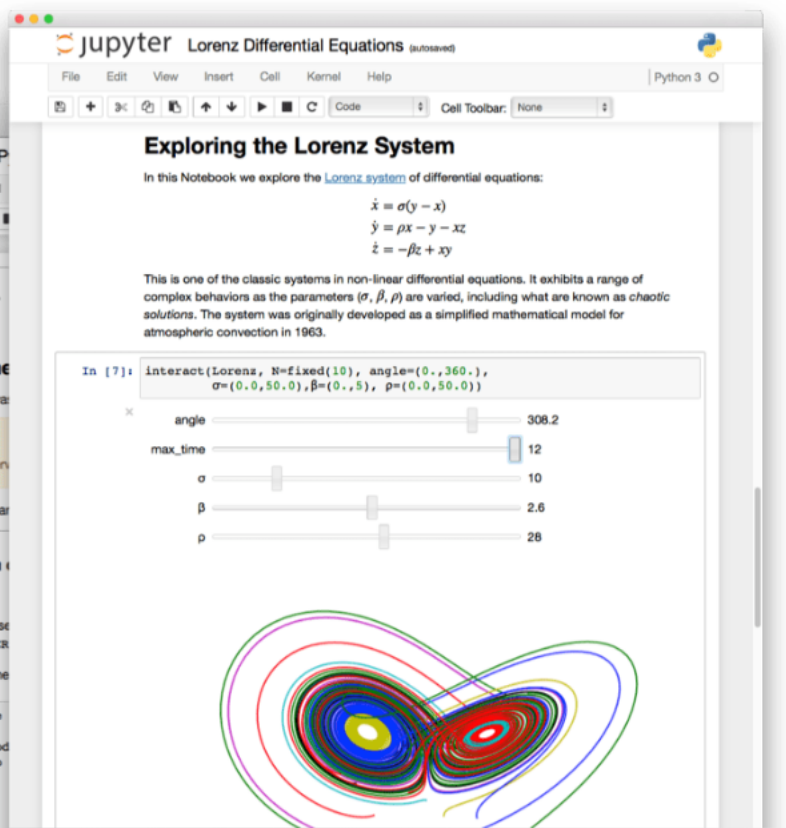
A screenshot of a Windows application window titled "Python 3.5.2 Shell". The window has a standard Windows title bar with minimize, maximize, and close buttons. The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following text: "Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32", "Type 'copyright', 'credits' or 'license()' for more information.", and a prompt ">>> |".

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```



A screenshot of a Windows application window titled "*Untitled*". The window has a standard Windows title bar with minimize, maximize, and close buttons. The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The main text area shows a Python code snippet. The first line is "print(" with the opening parenthesis highlighted. The second line is "print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)" and is fully highlighted.

```
*Untitled*
File Edit Format Run Options Window Help
print(
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```



The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Try it in your browser

Install the Notebook

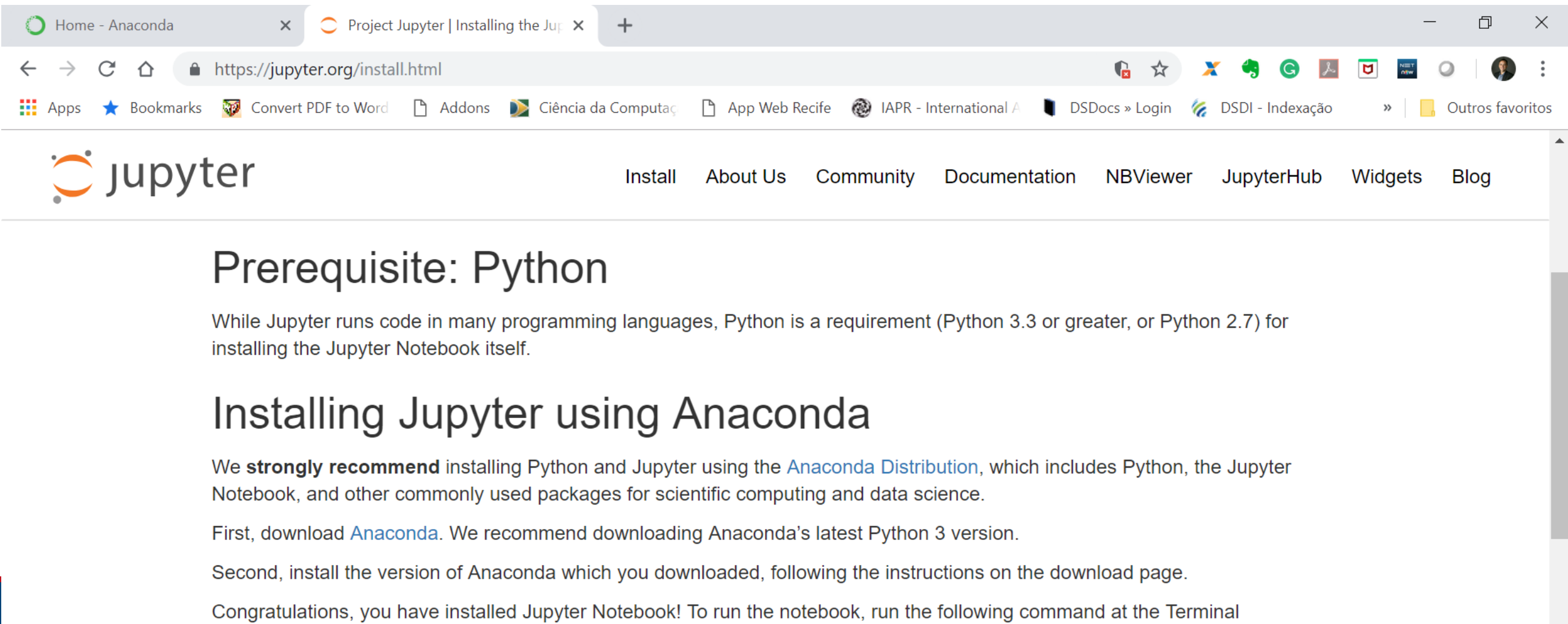
Vamos usar o Jupyter Notebook como IDE neste curso!

Para quem já usou Python

- Pode instalar o Jupyter através do *Python package manager*
 - pip install notebook
- Ou usando o conda:
 - conda install -c conda-forge notebook

Para novos usuários

- Vamos seguir o passo a passo de instalação do site do jupyter usando Anaconda




The screenshot shows a web browser window with two tabs: 'Home - Anaconda' and 'Project Jupyter | Installing the Jupyter Notebook'. The address bar shows the URL 'https://jupyter.org/install.html'. The browser's toolbar includes navigation buttons, a search bar, and various extension icons. Below the browser window, the Jupyter.org website is visible. The header features the Jupyter logo and a navigation menu with links to 'Install', 'About Us', 'Community', 'Documentation', 'NBViewer', 'JupyterHub', 'Widgets', and 'Blog'. The main content area is titled 'Prerequisite: Python' and explains that Python is required for Jupyter. It then provides instructions on how to install Jupyter using Anaconda, including downloading Anaconda, installing it, and running the Jupyter Notebook command in a terminal.

Home - Anaconda x Project Jupyter | Installing the Jupyter Notebook x +

← → ↻ 🏠 🔒 https://jupyter.org/install.html

Apps ★ Bookmarks 📄 Convert PDF to Word 📄 Addons 🇧🇷 Ciência da Computação 📄 App Web Recife 🌐 IAPR - International Association for Pattern Recognition 🖨️ DSDocs » Login 🌐 DSDI - Indexação » 📁 Outros favoritos

 jupyter

Install About Us Community Documentation NBViewer JupyterHub Widgets Blog

Prerequisite: Python

While Jupyter runs code in many programming languages, Python is a requirement (Python 3.3 or greater, or Python 2.7) for installing the Jupyter Notebook itself.

Installing Jupyter using Anaconda

We **strongly recommend** installing Python and Jupyter using the [Anaconda Distribution](#), which includes Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science.

First, download [Anaconda](#). We recommend downloading Anaconda's latest Python 3 version.




Second, install the version of Anaconda which you downloaded, following the instructions on the download page.

Congratulations, you have installed Jupyter Notebook! To run the notebook, run the following command at the Terminal

Programação para Ci x Currículo do Sistema x Anaconda | Individual x Project Jupyter | Insta x Home x ExemplosParte1 x +

anaconda.com/products/individual

Anaconda Installers

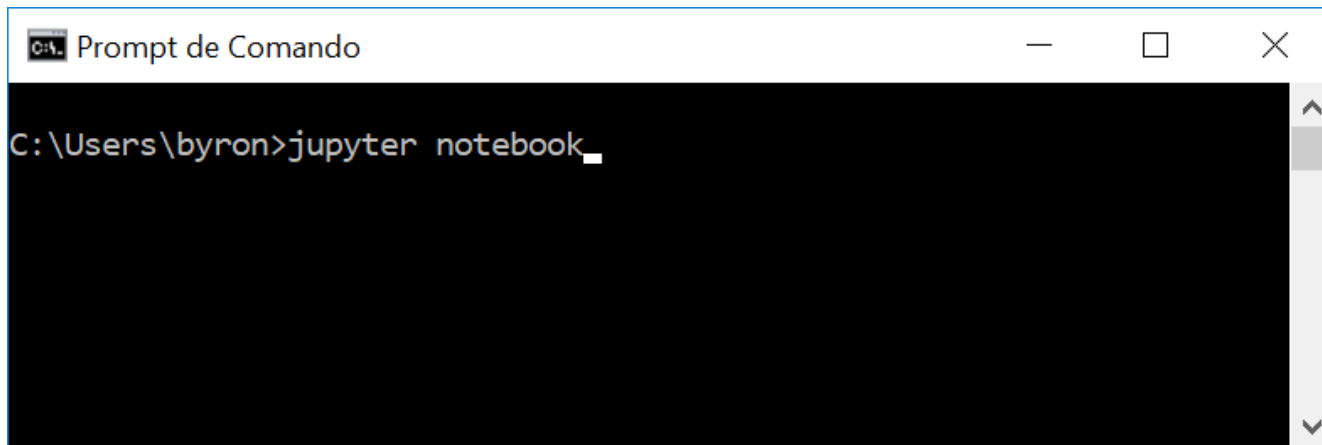
Windows 	MacOS 	Linux 
Python 3.8 64-Bit Graphical Installer (457 MB) 32-Bit Graphical Installer (403 MB)	Python 3.8 64-Bit Graphical Installer (435 MB) 64-Bit Command Line Installer (428 MB)	Python 3.8 64-Bit (x86) Installer (529 MB) 64-Bit (Power8 and Power9) Installer (279 MB)

This website uses cookies to ensure you get the best experience on our website. [Privacy Policy](#) Accept

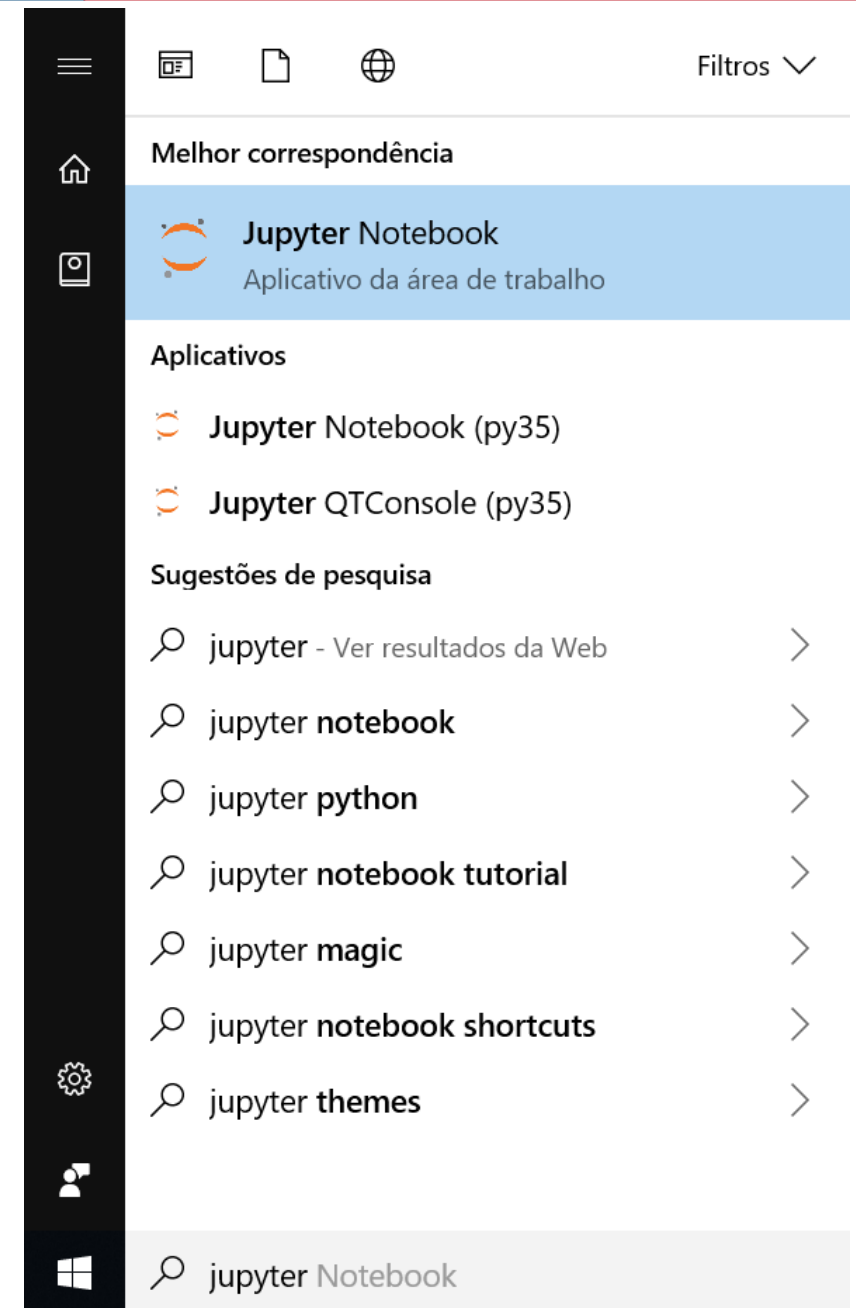
<https://www.anaconda.com/products/individual>

Iniciando o Jupyter

- Após a instalação do Anaconda



```
C:\Users\byron>jupyter notebook_
```



Tela inicial do Jupyter Notebook

The screenshot displays the Jupyter Notebook web interface in a browser. The browser's address bar shows 'localhost:8889/tree'. The Jupyter logo is visible in the top left, and a 'Logout' button is in the top right. Below the logo, there are tabs for 'Files', 'Running', and 'Clusters', with 'Files' being the active tab. A message 'Select items to perform actions on them.' is shown. The file browser lists several folders: '3D Objects', 'Contacts', 'Desktop', 'Documents', and 'Downloads'. A dropdown menu is open, showing options to create a new notebook with 'Python 3' or other file types like 'Text File', 'Folder', and 'Terminal'. The menu also includes a 'Create a new notebook with Python 3' option. The time '43 minutos atrás' is displayed at the bottom right of the file list.

Home - Anaconda x Downloads - Anaconda x Untitled x Home x Untitled x + - □ ×

← → ↻ 🏠 ⓘ localhost:8889/tree ☆

Apps ★ Bookmarks 📄 Convert PDF to Word 📄 Addons 🇧🇷 Ciência da Computaç App Web Recife 🌐 IAPR - International A 📄 DSDocs » Login 🌐 DSDI - Indexação » 📁 Outros favoritos

jupyter Logout

Files Running Clusters

Select items to perform actions on them.

0 ▾ 📁 /

- 📁 3D Objects
- 📁 Contacts
- 📁 Desktop
- 📁 Documents
- 📁 Downloads

Upload New ↕

Notebook: Python 3

Other: Create a new notebook with Python 3

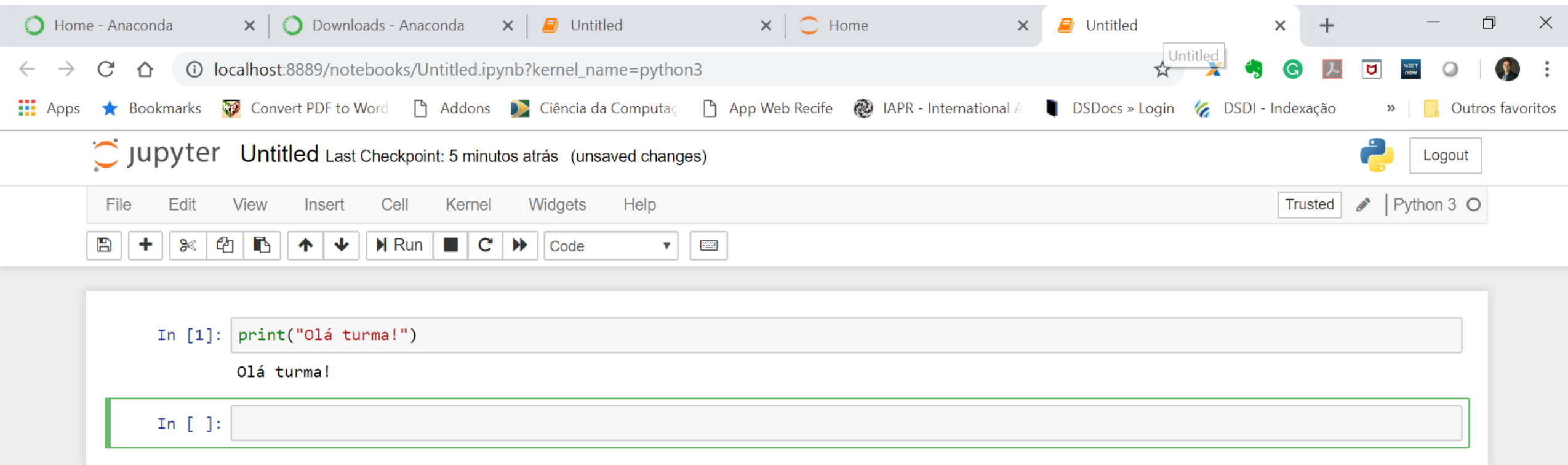
Text File

Folder

Terminal

43 minutos atrás

Primeiros passos no Jupyter Notebook



The screenshot shows a web browser window with multiple tabs. The active tab is titled 'Untitled' and shows the URL `localhost:8889/notebooks/Untitled.ipynb?kernel_name=python3`. The browser's address bar and bookmarks are visible. Below the browser window, the Jupyter Notebook interface is shown. It includes a header with the Jupyter logo, the name 'Untitled', and a status message 'Last Checkpoint: 5 minutos atrás (unsaved changes)'. There is a 'Logout' button. Below the header is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. To the right of the menu bar is a 'Trusted' status indicator and a 'Python 3' kernel selector. Below the menu bar is a toolbar with icons for saving, creating new cells, deleting cells, and running code. The main area of the notebook contains two code cells. The first cell has the code `print("Olá turma!")` and the output `Olá turma!`. The second cell is empty and has the prompt `In []:`.

Atalhos úteis:

- **SHIFT + ENTER** -> executa o bloco de código
- **ALT + ENTER** -> insere uma nova linha no meio do código

Ambientes Virtuais

- Permite instalações e utilizações personalizadas, conforme a necessidade de cada projeto
- Com o Anaconda, podemos configurar os ambientes virtuais usando o seu próprio gerenciador:
 - <https://conda.io/docs/user-guide/tasks/manage-environments.html>
- Podemos usar a biblioteca chamada *virtualenv* para outras distribuições do Python


```
C:\Users\byron>conda env list
```

```
# conda environments:
```

```
#
```

```
ocropus_env          C:\Users\byron\AppData\Local\conda\conda\envs\ocropus_env
```

```
py35                 C:\Users\byron\AppData\Local\conda\conda\envs\py35
```

```
tensorflow           C:\Users\byron\AppData\Local\conda\conda\envs\tensorflow
```

```
ocropus_env          C:\ProgramData\Miniconda2\envs\ocropus_env
```

```
root                 * C:\ProgramData\Miniconda2
```

```
C:\Users\byron>python
```

```
Python 2.7.13 |Continuum Analytics, Inc.| (default, May 11 2017, 13:17:26) [MSC v.1500 64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
Anaconda is brought to you by Continuum Analytics.
```

```
Please check out: http://continuum.io/thanks and https://anaconda.org
```

```
>>> quit()
```

```
C:\Users\byron>activate py35
```

```
(py35) C:\Users\byron>python
```

```
Python 3.5.3 |Anaconda custom (64-bit)| (default, May 15 2017, 10:43:23) [MSC v.1900 64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> quit()
```

```
(py35) C:\Users\byron>deactivate
```

```
C:\Users\byron>_
```



Pró-reitora de Pós-Graduação, Pesquisa e Inovação
Especialização em Ciências de Dados e Analytics

Programação para Ciência de Dados

Parte 1 / Aula 2: Revisão de Programação Estruturada com Python

Agenda

- Overview de Programação com a Linguagem Python
 - Tipos numéricos
 - Strings
 - Prints
 - Listas
 - Dicionários
 - Booleanos
 - Tuplas e Sets
 - Operadores Lógicos
 - if, else, elif
 - for, while
 - range()
 - Funções
 - Expressões lambda
 - Mapas e Filtros

Operações básicas com tipos numéricos

```
In [1]: 2+3
```

```
Out[1]: 5
```

```
In [2]: 3*2
```

```
Out[2]: 6
```

```
In [3]: 2**3
```

```
Out[3]: 8
```

```
In [4]: 5/2
```

```
Out[4]: 2.5
```

```
In [5]: 5%2
```

```
Out[5]: 1
```

```
In [6]: type(5/2)
```

```
Out[6]: float
```

Variáveis

jupyter ExemploAula1 Last Checkpoint: uma hora atrás (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Save Add Split Cell Move Up Move Down Run Interrupt Restart Code Keyboard

```
In [14]: x = 2
```

```
In [15]: 2x = 4
```

File "<ipython-input-15-6f6d766411d7>", line 1
2x = 4
^
SyntaxError: invalid syntax

```
In [16]: x2 = x * 2
```

```
In [17]: X2
```

NameError Traceback (most recent call last)
<ipython-input-17-fc316cca30c5> in <module>()
----> 1 X2

NameError: name 'X2' is not defined

```
In [18]: x2
```

```
Out[18]: 4
```

Convenções

- <https://www.python.org/dev/peps/pep-0008/#naming-conventions>
- <https://docs.python-guide.org/writing/style/>

Strings

```
In [28]: texto = 'exemplo de string'
```

```
In [29]: print(texto)
```

exemplo de string

```
In [30]: texto = texto + ' concatenada '
```

```
In [31]: print(texto)
```

exemplo de string concatenada

```
In [32]: print('{} ao quadrado é {}'.format(x,x2))
```

2 ao quadrado é 4

```
In [33]: print('a raiz quadrada de {p1} é {p2}'.format(p1=x2, p2=x))
```

a raiz quadrada de 4 é 2

Você pode digitar também ***help(str)*** para ver as opções disponíveis!

https://www.w3schools.com/python/python_ref_string.asp

Recebendo Strings do usuário

```
In [52]: nome = input('Qual o seu nome? ')
```

Qual o seu nome? Byron

```
In [53]: ano = input('Em que ano você nasceu? ')
```

Em que ano você nasceu? 1978

```
In [54]: idade = 2019-ano
print('{p1} você tem {p2} anos ou vai completar este ano!'.format(p1=nome, p2=idade))
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-54-fb9b0ac6ce1c> in <module>()
----> 1 idade = 2019-ano
      2 print('{p1} você tem {p2} anos ou vai completar este ano!'.format(p1=nome, p2=idade))

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

```
In [55]: idade = 2019 - int(ano)
print('{p1} você tem {p2} anos ou vai completar este ano!'.format(p1=nome, p2=idade))
```

Byron você tem 41 anos ou vai completar este ano!

Exercício

- Faça um programa que calcule a conversão de dinheiro de dólar para reais. Para isso, solicite do usuário a cotação do dólar para compra e quantos dólares ele deseja comprar.
- Em seguida o programa deve calcular o valor correspondente em reais e exibir na tela com duas casas decimais. Ex.:
 - Informe o valor da cotação do dólar para compra: 4,7905
 - Quantos dólares você deseja comprar? 50
 - O preço de U\$ 50,00 corresponde a R\$ 239,52 na cotação atual.
- *Dica: usar format* (https://www.w3schools.com/python/ref_string_format.asp)