



Preditiva.ai

# Python

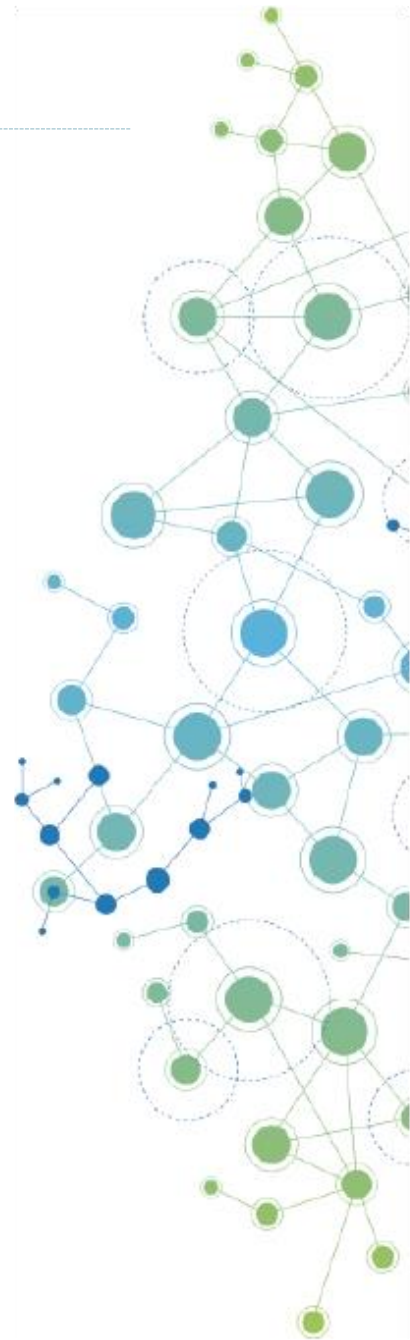
## Configuração de Ambiente

### Windows

# O que você verá nesta aula?

---

- ☐ Instalação do Python 3.8.3
- ☐ Criação de um ambiente virtual
- ☐ Instalação de bibliotecas
- ☐ Inicializando o Jupyter Lab



# Configuração do Ambiente

## Instalação do Python 3.8.3



Para realizar a instalação do Python, o primeiro passo é realizar o download na página oficial:

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

No curso de **Python** utilizaremos a **versão 3.8.3**.

Selecione a versão correspondente ao seu **sistema operacional**.

Clique em **Download Python 3.8.3** e quando finalizar execute o arquivo de instalação.

A screenshot of the Python.org website. The top navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is a secondary navigation bar with links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a large banner with the text "Download the latest version for Windows" and a red arrow pointing to a button labeled "Download Python 3.8.3". Below the button, there are links for "Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other", "Want to help test development versions of Python? Prereleases, Docker images", and "Looking for Python 2.7? See below for specific releases". The banner also includes an illustration of two parachutes carrying boxes. Below the banner, there is a section titled "Active Python Releases" with a table of release information.

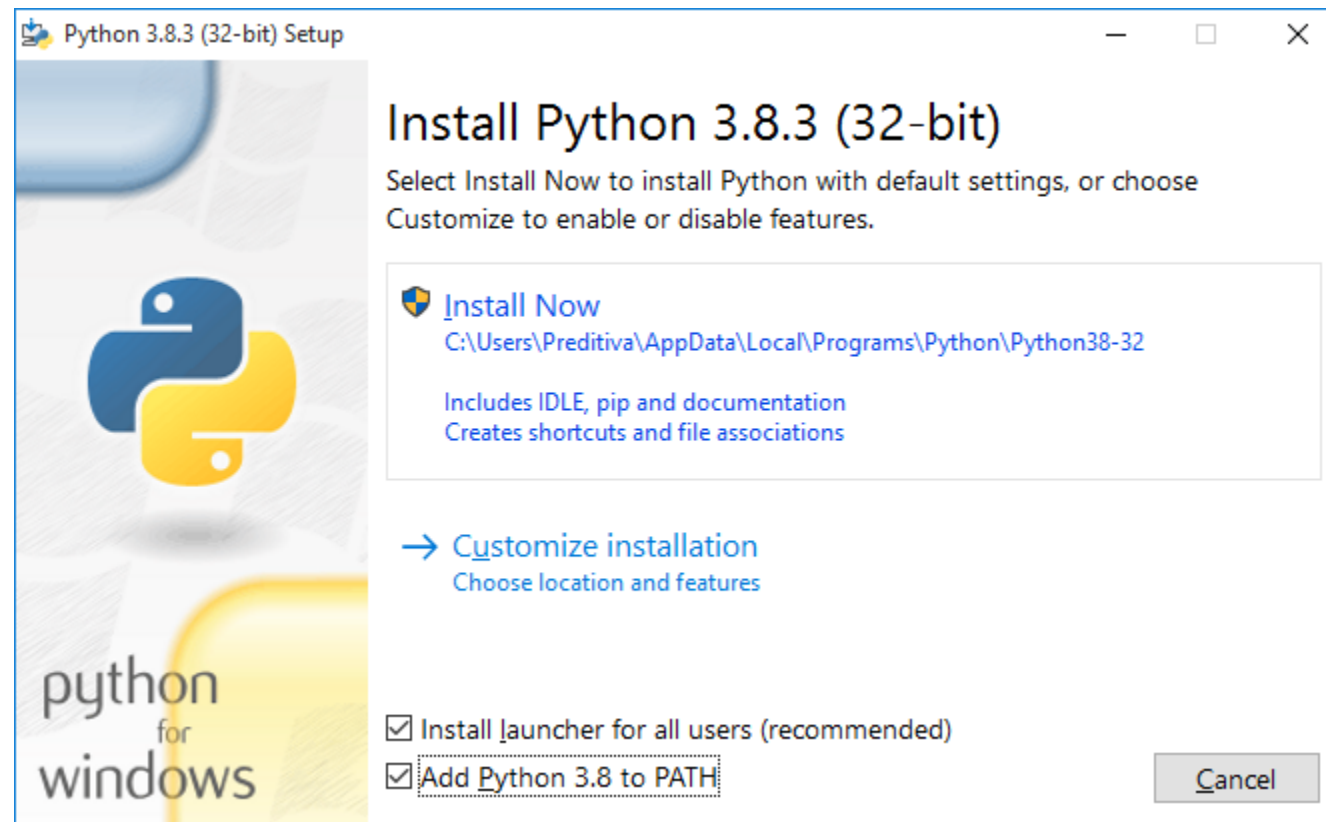
Python version	Maintenance status	First released	End of support	Release schedule
3.8	bugfix	2019-10-14	2024-10	PEP 569
3.7	bugfix	2018-06-27	2023-06-27	PEP 537
3.6	security	2016-12-23	2021-12-23	PEP 494
3.5	security	2015-09-13	2020-09-13	PEP 478
2.7	end-of-life	2010-07-03	2020-01-01	PEP 373

# Configuração do Ambiente

## Instalação do Python 3.8.3 - Windows



Na 1ª tela de instalação, selecione a opção "**Add Python 3.8 to PATH**", isso ajudará na criação do ambiente virtual mais a frente. Em seguida clique em **Install Now**.

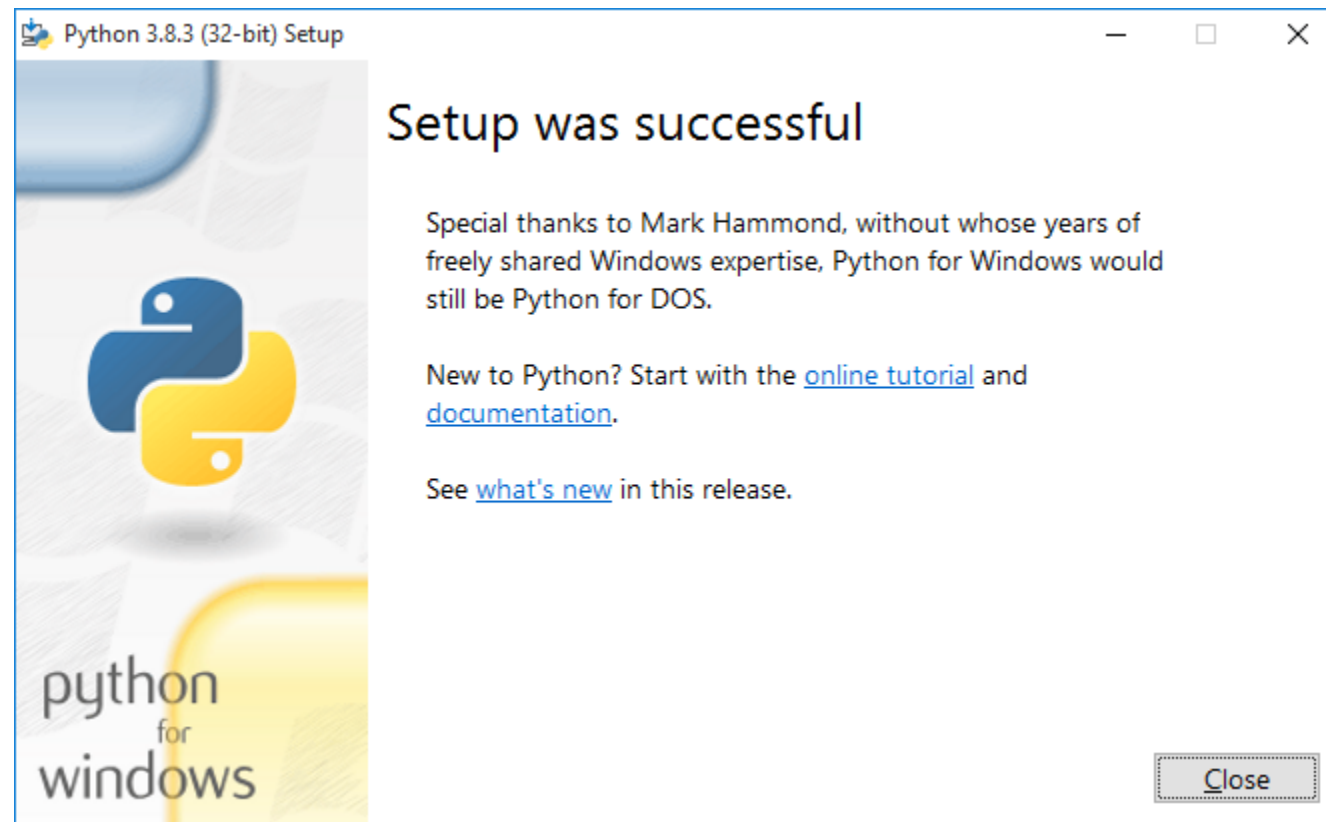


# Configuração do Ambiente

## Instalação do Python 3.8.3 - Windows



Ao final, clique em **Close** para finalizar a instalação.



# Configuração do Ambiente

## Criação de um ambiente virtual

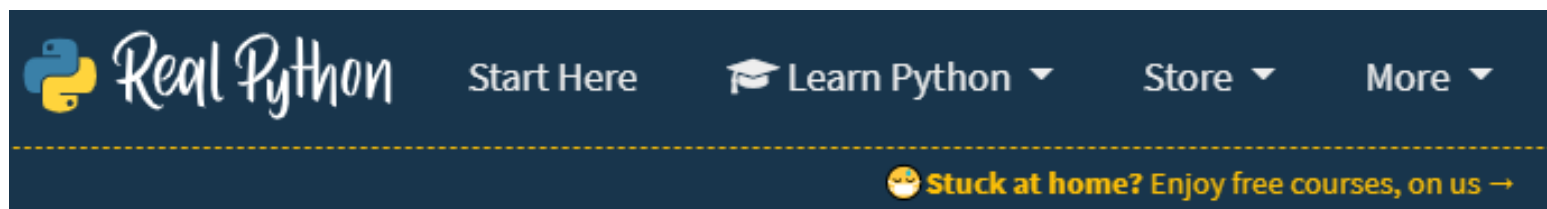


Preditiva.ai

Antes de começar a utilizar a plataforma, vamos criar um **ambiente virtual** especificamente para o curso.

Criar um ambiente virtual Python é uma **boa prática**, fortemente recomendada para que você mantenha as **dependências** de cada projeto **isoladas** umas das outras.

Assim, você instala apenas as **bibliotecas necessárias** para o seu projeto, e **preserva as versões** instaladas evitando que algum update faça com que seu projeto pare de funcionar.



## What Is a Virtual Environment?

At its core, the main purpose of Python virtual environments is to create an isolated environment for Python projects. This means that each project can have its own dependencies, regardless of what dependencies every other project has.

In our little example above, we'd just need to create a separate virtual environment for both *ProjectA* and *ProjectB*, and we'd be good to go. Each environment, in turn, would be able to depend on whatever version of *ProjectC* they choose, independent of the other.

The great thing about this is that there are no limits to the number of environments you can have since they're just directories containing a few scripts. Plus, they're easily created using the `virtualenv` or `pyenv` command line tools.

<https://realpython.com/python-virtual-environments-a-primer/>

# Configuração do Ambiente

## Criação de um ambiente virtual - Windows



**No Windows:** Abra um **Prompt de Comando**, pois nele vamos digitar os seguintes comandos para criar um **ambiente virtual** chamado "**preditiva**" dentro do diretório **Meus Documentos**, e ativá-lo para iniciar a **instalação dos módulos** adicionais.

1º Comando



```
C:\Users\Preditiva>cd Documents
```

2º Comando



```
C:\Users\Preditiva\Documents>python -m venv predictiva
```

3º Comando



```
C:\Users\Preditiva\Documents>predictiva\Scripts\activate
```

```
(predictiva) C:\Users\Preditiva\Documents>_
```

O nome (predictiva) indica que o ambiente virtual foi criado com sucesso e está ativo

# Configuração do Ambiente

## Criação de um ambiente virtual - Windows



Para instalar os módulos necessários para o curso, utilizaremos o comando **pip install** seguido do nome do módulo. Este comando é utilizado para gerenciamento dos módulos Python e será muito útil sempre que você precisar instalar um novo módulo. Por isso, o primeiro passo é atualizá-lo com o comando: `python -m pip install --upgrade pip`

Na sequência, vamos instalar os seguintes módulos utilizando o **pip install**:

- **certifi**
- **jupyterlab==2.2.2**
- **pandas==1.0.5**
- **scikit-learn==0.23.1**
- **wheel==0.34.2**
- **h2o==3.30.0.6**
- **seaborn==0.10.1**
- **nodejs==0.1.1**

Como alguns módulos possuem dependências, muitos outros módulos serão instalados.

Mais adiante veremos uma forma mais fácil de instalar diversos módulos de uma só vez.

```
Prompt de Comando - pip install jupyterlab
Microsoft Windows [versão 10.0.10240]
(c) 2015 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Preditiva>cd Documents

C:\Users\Preditiva\Documents>python -m venv predictiva

C:\Users\Preditiva\Documents>predictiva\Scripts\activate

(predictiva) C:\Users\Preditiva\Documents>pip jupyterlab
ERROR: unknown command "jupyterlab"

(predictiva) C:\Users\Preditiva\Documents>pip install jupyterlab
Collecting jupyterlab
  Downloading https://files.pythonhosted.org/packages/9e/1b/3defee93c6057a50396b2e307db3b6531b09b87a77b2e7b72ca428a963ef/jupyterlab-2.1.5-py3-none-any.whl (7.6MB)
    | 1.9MB 1.1MB/s eta 0:00:06
```

Embora o **Prompt de Comando** não seja exatamente amigável, por não possuir uma Interface Gráfica, ele é a forma mais **rápida** para instalar os módulos e começar a utilizar o Python



# Configuração do Ambiente

## Criação de um ambiente virtual



Para verificar quais módulos estão instalados, vamos utilizar o comando **pip list**. Perceba que são exibidos os módulos e suas respectivas versões. Essa informação é importante para manter a compatibilidade entre os módulos e os códigos desenvolvidos.

```
fernandoitano — zsh — 60x40
[(preditiva) fernandoitano@Mac-mini ~ % pip list]
Package            Version
-----
appnope            0.1.0
attrs              19.3.0
backcall           0.2.0
bleach             3.1.5
certifi            2020.6.20
chardet            3.0.4
colorama           0.4.3
cyclor             0.10.0
decorator          4.4.2
defusedxml         0.6.0
entrypoints        0.3
future             0.18.2
h2o                3.30.0.6
idna               2.10
ipykernel          5.3.4
ipython            7.16.1
ipython-genutils   0.2.0
jedi               0.17.2
Jinja2             2.11.2
joblib             0.16.0
json5              0.9.5
jsonschema         3.2.0
jupyter-client     6.1.6
jupyter-core       4.6.3
jupyterlab         2.2.2
jupyterlab-server  1.2.0
kiwisolver         1.2.0
MarkupSafe         1.1.1
matplotlib         3.3.0
mistune            0.8.4
nbconvert          5.6.1
nbformat           5.0.7
notebook           6.0.3
numpy              1.19.1
packaging          20.4
pandas             1.0.5
pandocfilters      1.4.2
```

```
fernandoitano — zsh — 60x40
mistune            0.8.4
nbconvert          5.6.1
nbformat           5.0.7
notebook           6.0.3
numpy              1.19.1
packaging          20.4
pandas             1.0.5
pandocfilters      1.4.2
parso              0.7.1
pexpect            4.8.0
pickleshare        0.7.5
Pillow             7.2.0
pip                20.2
prometheus-client  0.8.0
prompt-toolkit     3.0.5
ptyprocess         0.6.0
Pygments           2.6.1
pyparsing          2.4.7
pysistent          0.16.0
python-dateutil    2.8.1
pytz               2020.1
pyzmq              19.0.1
requests           2.24.0
scikit-learn       0.23.1
scipy              1.5.2
seaborn            0.10.1
Send2Trash         1.5.0
setuptools         41.2.0
six                1.15.0
tabulate           0.8.7
terminado          0.8.3
testpath           0.4.4
threadpoolctl      2.1.0
tornado            6.0.4
traitlets          4.3.3
urllib3            1.25.10
wcwidth            0.2.5
webencodings       0.5.1
wheel              0.34.2
(preditiva) fernandoitano@Mac-mini ~ %
```

# Configuração do Ambiente

## Criação de um ambiente virtual - Windows



Uma funcionalidade muito utilizada pelos usuários de Python é a de gerar um **arquivo com todos os módulos instalados e suas respectivas versões**.

Dessa forma, as soluções que criam containers para realizar o deployment do modelo já instalarão os módulos nas exatas mesmas versões utilizadas no desenvolvimento.

Vamos treinar utilizar essa funcionalidade! Para isso, acesse o diretório do ambiente virtual e digite:

**pip freeze > requirements.txt**

```

Prompt de Comando

(preditiva) C:\Users\Preditiva\Documents>cd predictiva
(preditiva) C:\Users\Preditiva\Documents\predictiva>pip freeze > requirements.txt
(preditiva) C:\Users\Preditiva\Documents\predictiva>_

```



O arquivo **requirements.txt** gerado possui todas os nomes dos módulos instalados e suas respectivas versões

```

requirements.txt - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda

attrs==19.3.0
backcall==0.2.0
bleach==3.1.5
certifi==2020.6.20
chardet==3.0.4
colorama==0.4.3
cyclr==0.10.0
decorator==4.4.2
defusedxml==0.6.0
entrypoints==0.3
future==0.18.2
h2o==3.30.0.6
idna==2.10
ipykernel==5.3.2
ipython==7.16.1
ipython-genutils==0.2.0
jedi==0.17.1

```

# Configuração do Ambiente

## Inicializando o Jupyter Lab - Windows



No curso de Python utilizaremos o ambiente Jupyter Lab por possuir diversas características interessantes para o desenvolvimento das atividades de análise de dados que exploraremos adiante.

Para iniciar o Jupyter Lab, basta digitar o comando: **jupyter lab**

```
C:\> Prompt de Comando - jupyter lab

(preditiva) C:\Users\Preditiva\Documents\preditiva>jupyter lab
[I 23:50:31.466 LabApp] Writing notebook server cookie secret to C:\Users\Preditiva\AppData\Roaming\jupyter\runtime\notebook_cookie_secret
[W 23:50:32.170 LabApp] Terminals not available (error was No module named 'winpty.cython')
[I 23:50:32.185 LabApp] JupyterLab extension loaded from c:\users\preditiva\documents\preditiva\lib\site-packages\jupyterlab
[I 23:50:32.185 LabApp] JupyterLab application directory is c:\users\preditiva\documents\preditiva\share\jupyter\lab
[I 23:50:32.185 LabApp] Serving notebooks from local directory: C:\Users\Preditiva\Documents\preditiva
[I 23:50:32.201 LabApp] The Jupyter Notebook is running at:
[I 23:50:32.201 LabApp] http://localhost:8888/?token=b5fef4064fe4e40ebe59d3e521bd88d1d3a27bf19fdc3f32
[I 23:50:32.201 LabApp] or http://127.0.0.1:8888/?token=b5fef4064fe4e40ebe59d3e521bd88d1d3a27bf19fdc3f32
[I 23:50:32.201 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 23:50:32.232 LabApp]

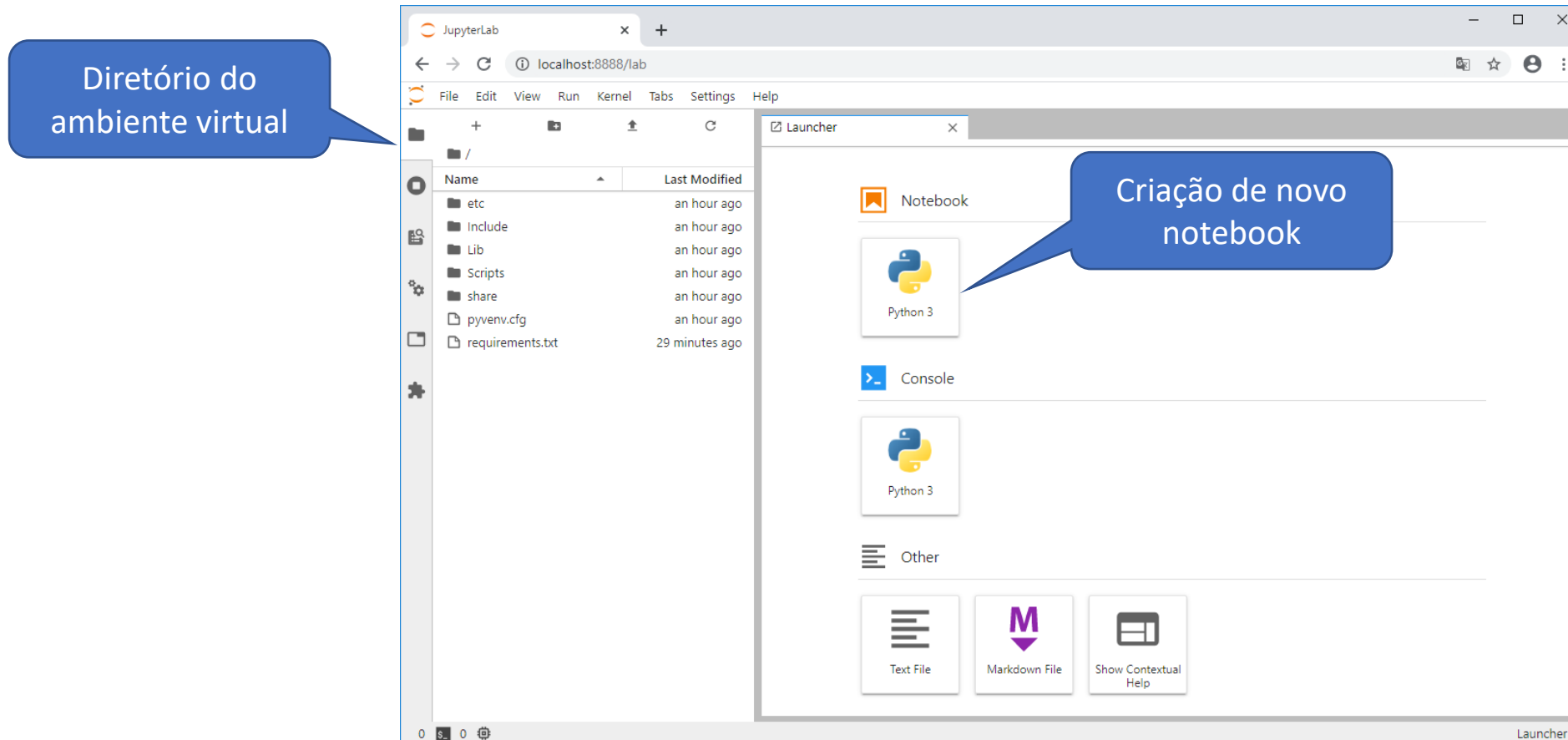
To access the notebook, open this file in a browser:
    file:///C:/Users/Preditiva/AppData/Roaming/jupyter/runtime/nbserver-2600-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=b5fef4064fe4e40ebe59d3e521bd88d1d3a27bf19fdc3f32
    or http://127.0.0.1:8888/?token=b5fef4064fe4e40ebe59d3e521bd88d1d3a27bf19fdc3f32
```

# Configuração do Ambiente

## Primeiros passos no Jupyter Lab



Surgirá então no seu navegador padrão a interface do **Jupyter Lab**. Neste momento vamos focar no diretório do projeto (ambiente virtual) e como podemos criar um novo notebook. Clique em Python 3 na seção Notebook.

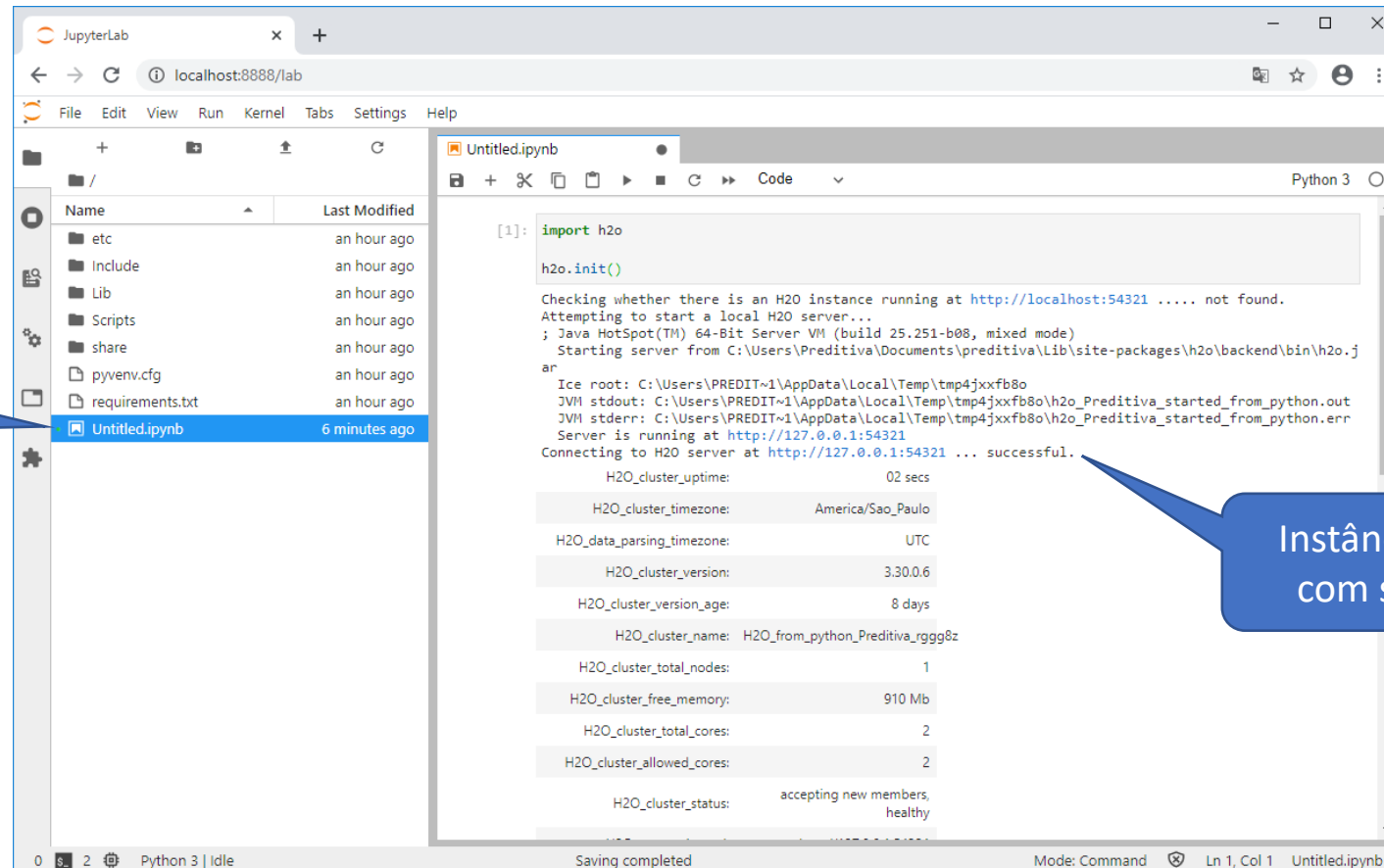
A screenshot of the JupyterLab web interface in a browser window. The browser address bar shows "localhost:8888/lab". The interface has a top menu bar with "File", "Edit", "View", "Run", "Kernel", "Tabs", "Settings", and "Help". On the left is a file browser sidebar showing a directory structure with folders like "etc", "include", "Lib", "Scripts", "share" and files like "pyvenv.cfg" and "requirements.txt". A blue callout bubble points to this sidebar with the text "Diretório do ambiente virtual". The main area is titled "Launcher" and contains several buttons: "Notebook" (with a sub-button for "Python 3"), "Console", and "Other". The "Python 3" button under "Notebook" is highlighted with a blue callout bubble that says "Criação de novo notebook". At the bottom of the Launcher section are buttons for "Text File", "Markdown File", and "Show Contextual Help". The status bar at the very bottom shows "0" and "Launcher".

# Configuração do Ambiente

## Primeiros passos no Jupyter Lab



Note que surgiu um arquivo **Untitled.ipynb** no diretório do projeto e já temos um notebook para começar a trabalhar com os dados. Vamos carregar o módulo **h2o** e criar uma instância do H2O para testar o ambiente:

A screenshot of the JupyterLab web interface. The left sidebar shows a file explorer with a list of files and folders. The main area displays a code editor for a file named 'Untitled.ipynb'. The code in the editor imports the 'h2o' module and initializes it. Below the code, the output shows the process of starting a local H2O server, including log messages and a table of server configuration details. A status bar at the bottom indicates 'Saving completed' and 'Mode: Command'.

Name	Last Modified
etc	an hour ago
include	an hour ago
Lib	an hour ago
Scripts	an hour ago
share	an hour ago
pyvenv.cfg	an hour ago
requirements.txt	an hour ago
Untitled.ipynb	6 minutes ago

```
[1]: import h2o

h2o.init()

Checking whether there is an H2O instance running at http://localhost:54321 ..... not found.
Attempting to start a local H2O server...
; Java HotSpot(TM) 64-Bit Server VM (build 25.251-b08, mixed mode)
Starting server from C:\Users\Preditiva\Documents\preditiva\Lib\site-packages\h2o\backend\bin\h2o.j
ar
Ice root: C:\Users\PREDIT~1\AppData\Local\Temp\tmp4jxxfb8o
JVM stdout: C:\Users\PREDIT~1\AppData\Local\Temp\tmp4jxxfb8o\h2o_Preditiva_started_from_python.out
JVM stderr: C:\Users\PREDIT~1\AppData\Local\Temp\tmp4jxxfb8o\h2o_Preditiva_started_from_python.err
Server is running at http://127.0.0.1:54321
Connecting to H2O server at http://127.0.0.1:54321 ... successful.

H2O_cluster_uptime: 02 secs
H2O_cluster_timezone: America/Sao_Paulo
H2O_data_parsing_timezone: UTC
H2O_cluster_version: 3.30.0.6
H2O_cluster_version_age: 8 days
H2O_cluster_name: H2O_from_python_Preditiva_rggg8z
H2O_cluster_total_nodes: 1
H2O_cluster_free_memory: 910 Mb
H2O_cluster_total_cores: 2
H2O_cluster_allowed_cores: 2
H2O_cluster_status: accepting new members, healthy
```

Arquivo do notebook

Instância criada com sucesso!

# Configuração do Ambiente

## Primeiros passos no Jupyter Lab



Note que surgiu um arquivo **Untitled.ipynb** no diretório do projeto e já temos um notebook para começar a trabalhar com os dados. Vamos carregar o módulo **h2o** e criar uma instância do H2O para testar o ambiente:

```
[1]: import h2o

h2o.init()

Checking whether there is an H2O instance running at http://localhost:54321 ..... not found.
Attempting to start a local H2O server...

H2OConnectionError                                Traceback (most recent call last)
~/preditiva/lib/python3.8/site-packages/h2o/h2o.py in init(url, ip, port, name, https, cacert, insecure, username, password, cookies, proxy, start_h2o, nthreads, ice_root, log_dir, log_level, max_log_file_size, enable_assertions, max_mem_size, min_mem_size, strict_version_check, ignore_config, extra_classpath, jvm_custom_args, bind_to_localhost, **kwargs)
    286     try:
    287         h2oconn = H2OConnection.open(url=url, ip=ip, port=port, name=name, https=https,
    288                                     verify_ssl_certificates=verify_ssl_certificates, cacert=cacert,
~/preditiva/lib/python3.8/site-packages/h2o/backend/connection.py in open(server, url, ip, port, name, https, auth, verify_ssl_certificates, cacert, proxy, cookies, verbose, _msgs)
    383         conn.timeout = 3.0
    384         conn.cluster = conn._test_connection(retries, messages=_msgs)
    385         # If a server is unable to respond within 1s, it should be considered a bug. However we disable this
~/preditiva/lib/python3.8/site-packages/h2o/backend/connection.py in _test_connection(self, max_retries, messages)
    682     else:
    683         raise H2OConnectionError("Could not establish link to the H2O cloud %s after %d retries\n%s"
    684                                   % (self._base_url, max_retries, "\n".join(errors)))

H2OConnectionError: Could not establish link to the H2O cloud http://localhost:54321 after 5 retries
[18:23.78] H2OConnectionError: Unexpected HTTP error: HTTPConnectionPool(host='localhost', port=54321): Max retries exceeded with url: /3/Cloud (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7fd2896354f8>: Failed to establish a new connection: [Errno 61] Connection refused'))
[18:23.99] H2OConnectionError: Unexpected HTTP error: HTTPConnectionPool(host='localhost', port=54321): Max retries exceeded with url: /3/Cloud (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7fd289635d30>: Failed to establish a new connection: [Errno 61] Connection refused'))
[18:24.20] H2OConnectionError: Unexpected HTTP error: HTTPConnectionPool(host='localhost', port=54321): Max retries exceeded with url: /3/Cloud (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7fd289655520>: Failed to establish a new connection: [Errno 61] Connection refused'))
[18:24.41] H2OConnectionError: Unexpected HTTP error: HTTPConnectionPool(host='localhost', port=54321): Max retries exceeded with url: /3/Cloud (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7fd289655cd0>: Failed to establish a new connection: [Errno 61] Connection refused'))
[18:24.62] H2OConnectionError: Unexpected HTTP error: HTTPConnectionPool(host='localhost', port=54321): Max retries exceeded with url: /3/Cloud (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7fd289655970>: Failed to establish a new connection: [Errno 61] Connection refused'))

During handling of the above exception, another exception occurred:

CalledProcessError                                Traceback (most recent call last)
<ipython-input-1-a6acb8ed3bb8> in <module>
    1 import h2o
    2 h2o.init()
```



Caso receba uma mensagem de erro e uma janela informando que é necessária a instalação do Java, faça o download a partir do endereço:

<https://www.oracle.com/java/technologies/javase-jdk13-downloads.html>

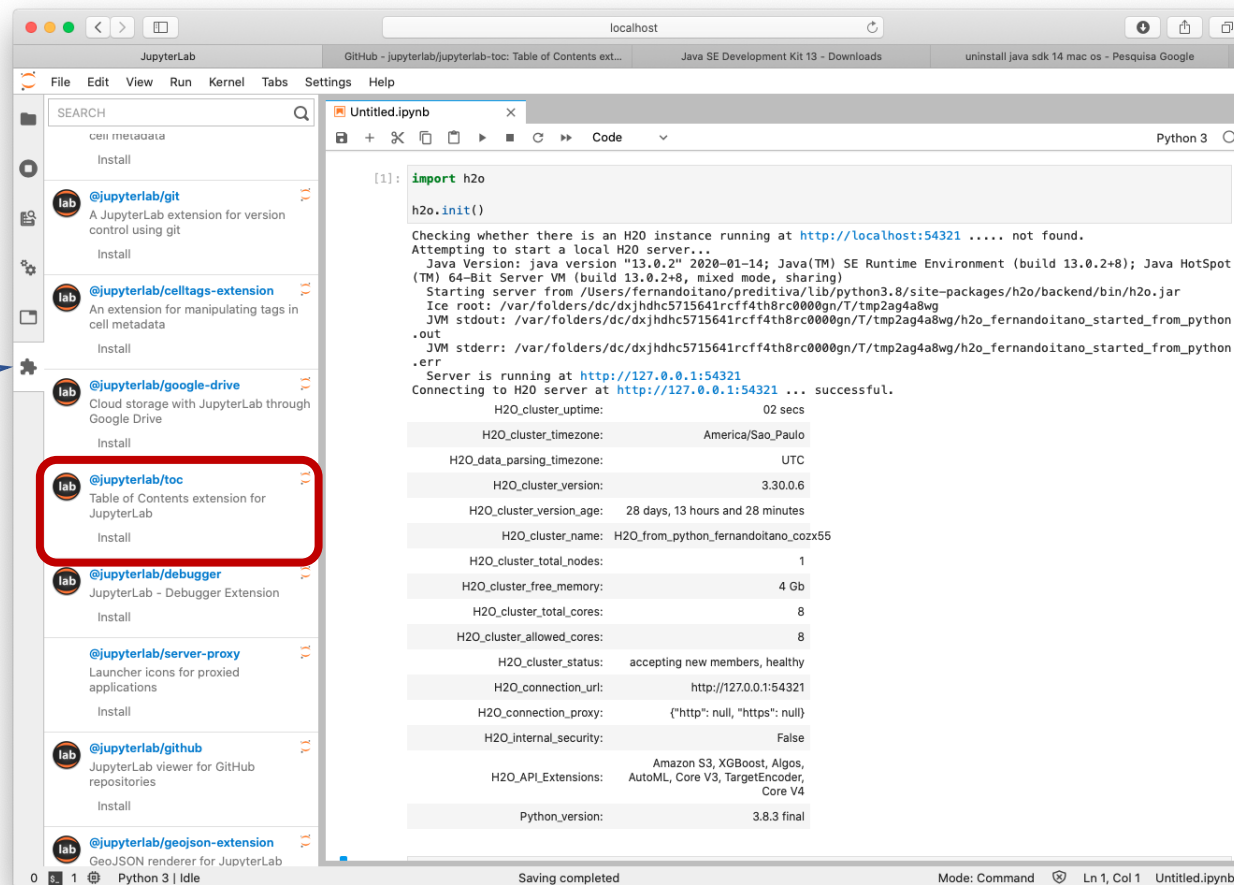
# Configuração do Ambiente

## Primeiros passos no Jupyter Lab



Vamos instalar a extensão **Table of Contents**, para isso habilite a utilização de extensões na barra lateral esquerda e em seguida procure pela extensão [@jupyterlab/toc](https://github.com/jupyterlab/jupyterlab-toc).

Extensões do  
Jupyter Lab



The screenshot shows the JupyterLab interface. On the left, the 'Extensions' sidebar is open, displaying a list of available extensions. The extension '@jupyterlab/toc' is highlighted with a red box. The main area shows a terminal output for a JupyterLab instance running on localhost. The output indicates that the H2O server is running successfully at http://127.0.0.1:54321. The terminal output includes the following details:

```
[1]: import h2o
h2o.init()

Checking whether there is an H2O instance running at http://localhost:54321 ..... not found.
Attempting to start a local H2O server...
Java Version: java version "13.0.2" 2020-01-14; Java(TM) SE Runtime Environment (build 13.0.2+8); Java HotSpot
(TM) 64-Bit Server VM (build 13.0.2+8, mixed mode, sharing)
Starting server from /Users/fernandoitano/preditiva/lib/python3.8/site-packages/h2o/backend/bin/h2o.jar
Ice root: /var/folders/dc/dxjdhc5715641rcff4th8rc0000gn/T/tmp2ag4a8wg
JVM stdout: /var/folders/dc/dxjdhc5715641rcff4th8rc0000gn/T/tmp2ag4a8wg/h2o_fernandoitano_started_from_python
JVM stderr: /var/folders/dc/dxjdhc5715641rcff4th8rc0000gn/T/tmp2ag4a8wg/h2o_fernandoitano_started_from_python
Server is running at http://127.0.0.1:54321 ... successful.
Connecting to H2O server at http://127.0.0.1:54321 ... successful.
H2O_cluster_uptime: 02 secs
H2O_cluster_timezone: America/Sao_Paulo
H2O_data_parsing_timezone: UTC
H2O_cluster_version: 3.30.0.6
H2O_cluster_version_age: 28 days, 13 hours and 28 minutes
H2O_cluster_name: H2O_from_python_fernandoitano_cozx55
H2O_cluster_total_nodes: 1
H2O_cluster_free_memory: 4 Gb
H2O_cluster_total_cores: 8
H2O_cluster_allowed_cores: 8
H2O_cluster_status: accepting new members, healthy
H2O_connection_url: http://127.0.0.1:54321
H2O_connection_proxy: ("http": null, "https": null)
H2O_internal_security: False
H2O_APIExtensions: Amazon S3, XGBoost, Algos, AutoML, Core V3, TargetEncoder, Core V4
Python_version: 3.8.3 final
```

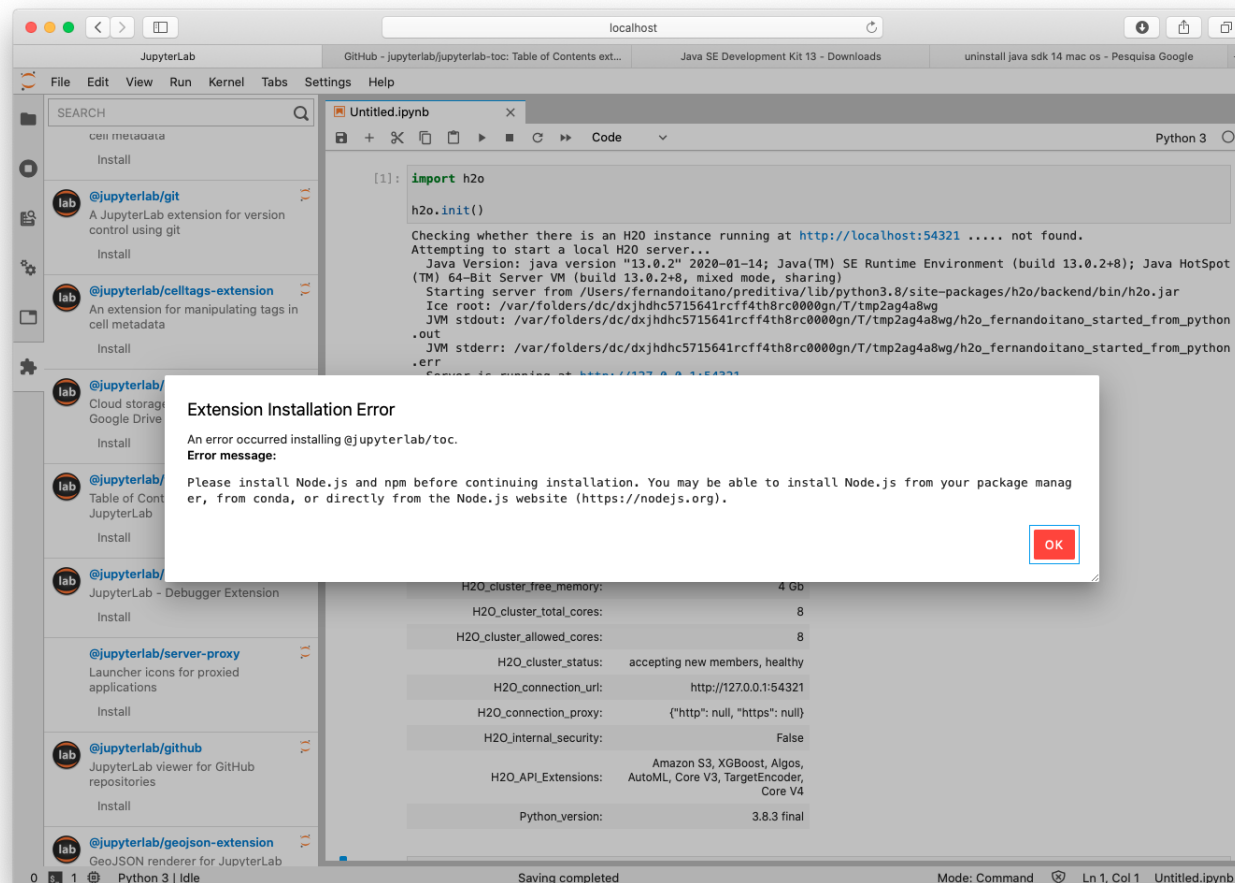


# Configuração do Ambiente

## Primeiros passos no Jupyter Lab



Caso receba essa mensagem de erro, é necessário instalar o NodeJS 12+. Utilize os links ao lado par fazer o download e na sequência instale o NodeJS, encerre o Jupyter Lab (próximos slides) e ative-o novamente.



## Windows

<https://nodejs.org/dist/v12.18.3/node-v12.18.3-x86.msi>



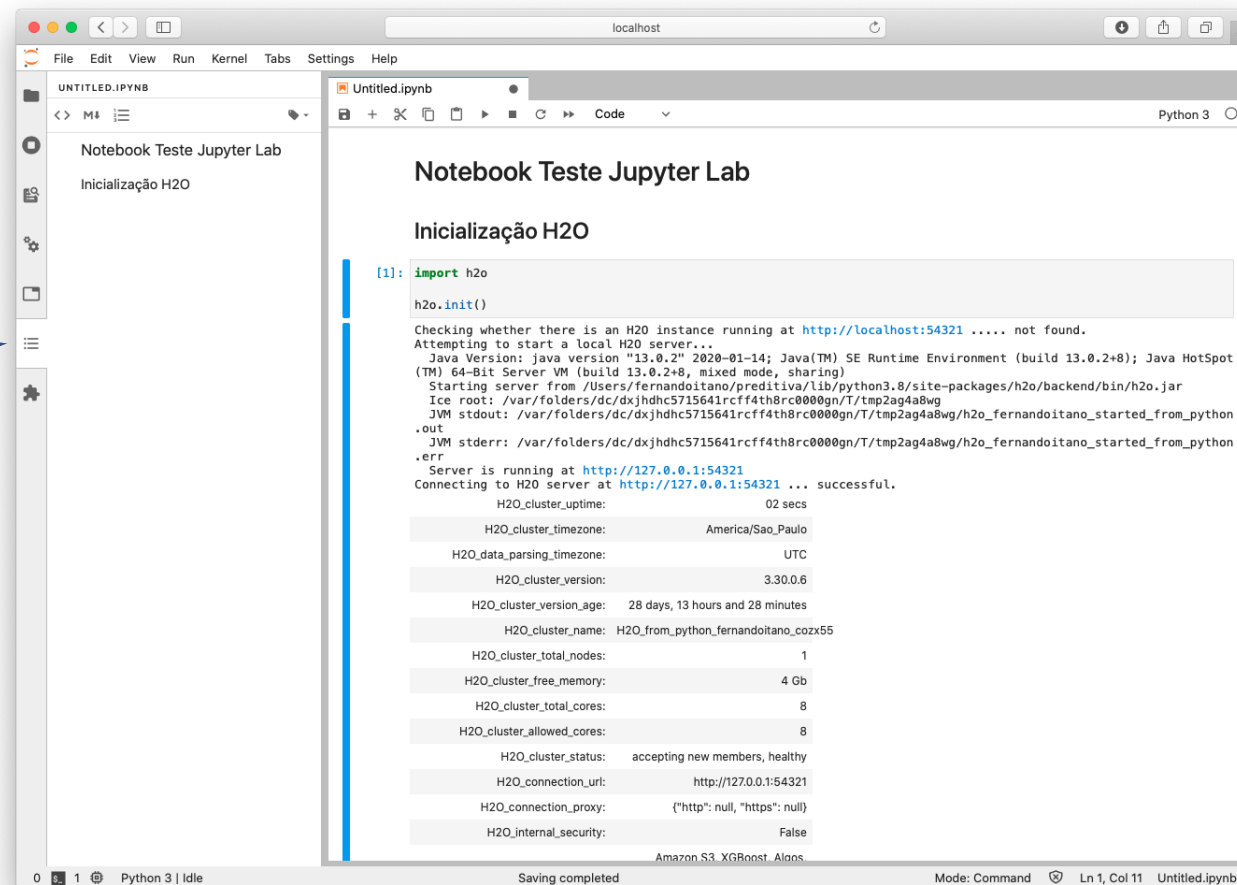
# Configuração do Ambiente

## Primeiros passos no Jupyter Lab



Com a extensão **Table of Contents** instalada, ficará muito mais fácil organizar e navegar nos notebooks.

Nova Opção  
Table of Contents

A screenshot of the Jupyter Lab web interface in a browser window. The left sidebar shows a file explorer with a folder named "Notebook Teste Jupyter Lab" containing a file "Inicialização H2O". The main area displays the content of this file, which includes a code cell with the following code:

```
[1]: import h2o
h2o.init()
```

Below the code cell, the output is displayed, showing the H2O server startup process and a detailed status report. The status report includes the following information:

H2O_cluster_uptime:	02 secs
H2O_cluster_timezone:	America/Sao_Paulo
H2O_data_parsing_timezone:	UTC
H2O_cluster_version:	3.30.0.6
H2O_cluster_version_age:	28 days, 13 hours and 28 minutes
H2O_cluster_name:	H2O_from_python_fernandoitano_cox55
H2O_cluster_total_nodes:	1
H2O_cluster_free_memory:	4 Gb
H2O_cluster_total_cores:	8
H2O_cluster_allowed_cores:	8
H2O_cluster_status:	accepting new members, healthy
H2O_connection_url:	http://127.0.0.1:54321
H2O_connection_proxy:	("http": null, "https": null)
H2O_internal_security:	False

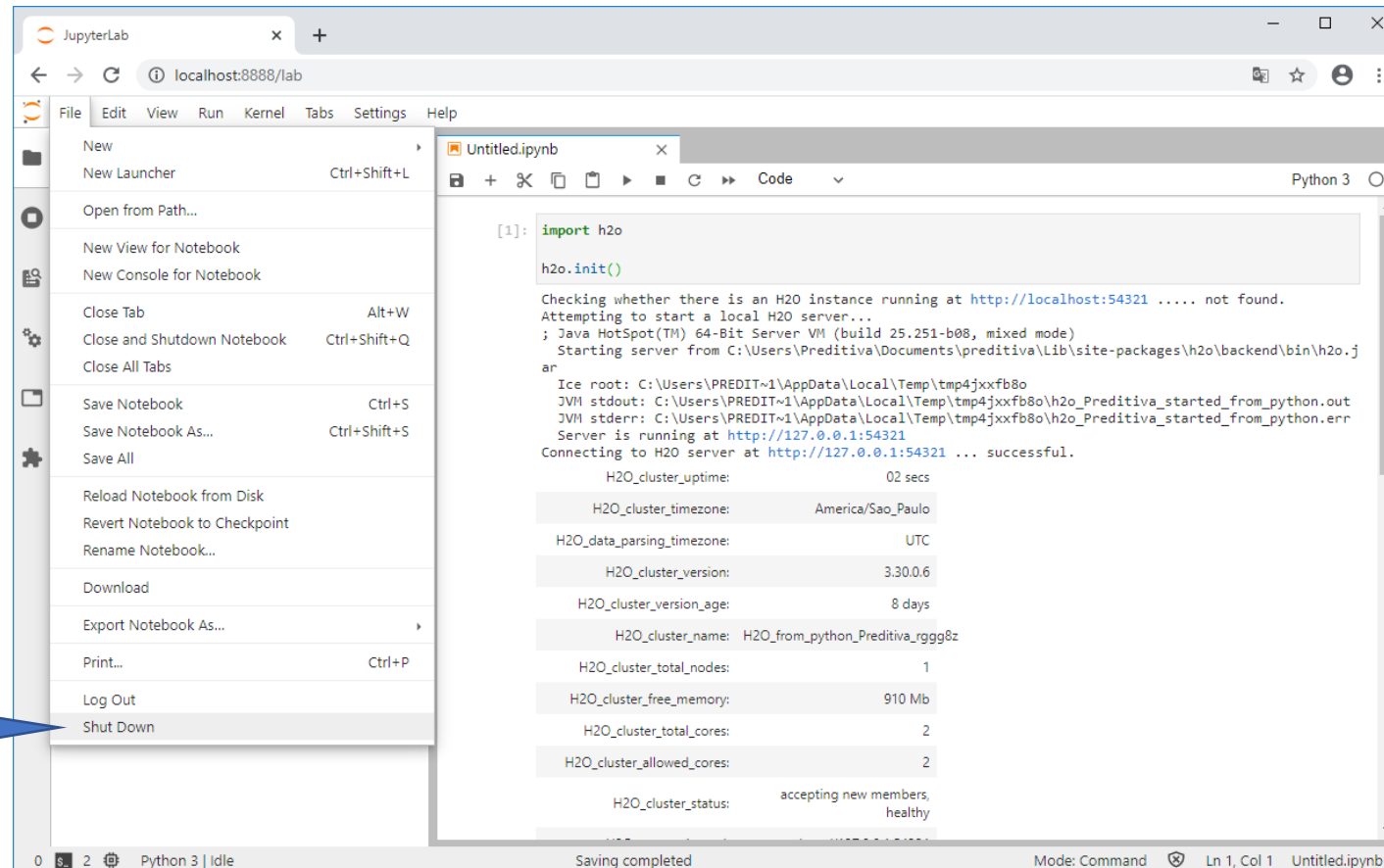
The bottom status bar of the Jupyter Lab interface shows "Saving completed", "Mode: Command", and "Ln 1, Col 11 | Untitled.ipynb".

# Configuração do Ambiente

## Primeiros passos no Jupyter Lab



Para encerrar o **Jupyter Lab** ao final do trabalho, selecione no menu a opção **File -> Shut Down**. Com isso o **Jupyter Lab** será encerrado e qualquer sessão **Python** existente também será encerrada.



Desative o Jupyter Lab ao final do trabalho

# Próximos passos

---

Com o ambiente devidamente configurado, você pode iniciar o curso de **Python**.

Nele apresentaremos os principais conceitos e técnicas utilizados em **Análise de Dados**.

Nos vemos lá!





Preditiva.ai