

Pós-Graduação em Ciência dos Dados

Prof. Dr. Tiago Eugenio de Melo
tmelo@uea.edu.br

Ementa

Ementa

- Linguagem de programação Python: tipos básicos de dados, comandos de entrada e saída de dados, desvio condicional, comandos de repetição, manipulação de arquivos;

Ementa

- Linguagem de programação Python: tipos básicos de dados, comandos de entrada e saída de dados, desvio condicional, comandos de repetição, manipulação de arquivos;
- Biblioteca Pandas e operações com DataFrames em conjuntos de dados;

Ementa

- Linguagem de programação Python: tipos básicos de dados, comandos de entrada e saída de dados, desvio condicional, comandos de repetição, manipulação de arquivos;
- Biblioteca Pandas e operações com DataFrames em conjuntos de dados;
- Instalação e utilização do ambiente de desenvolvimento Jupyter Notebooks;

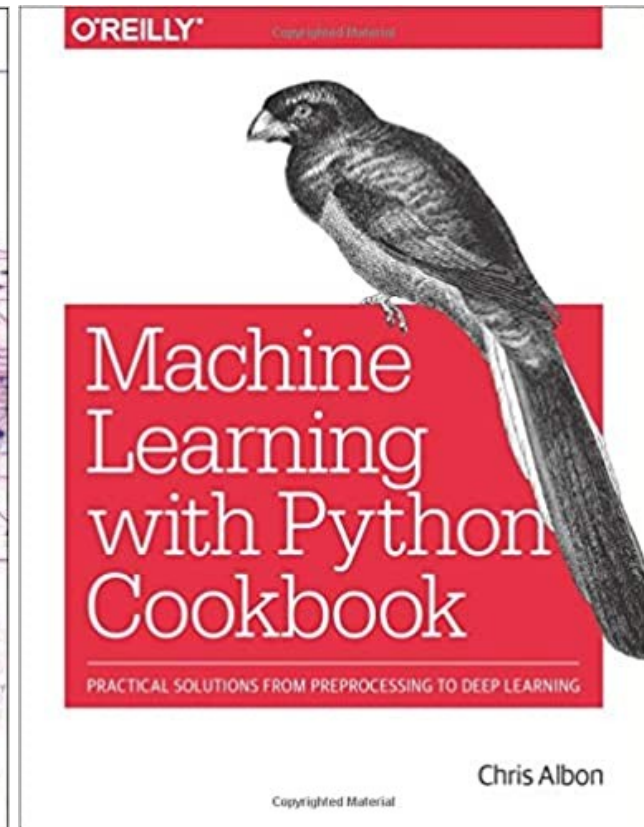
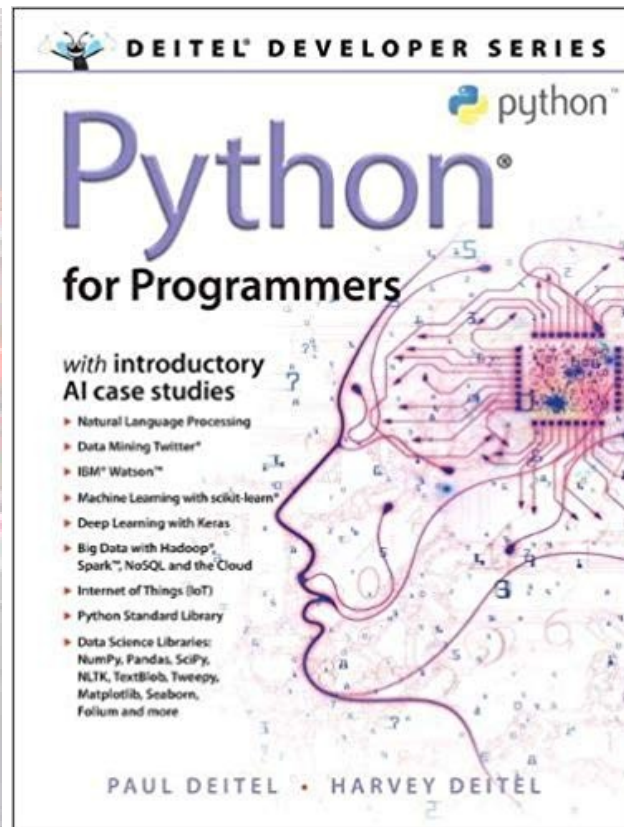
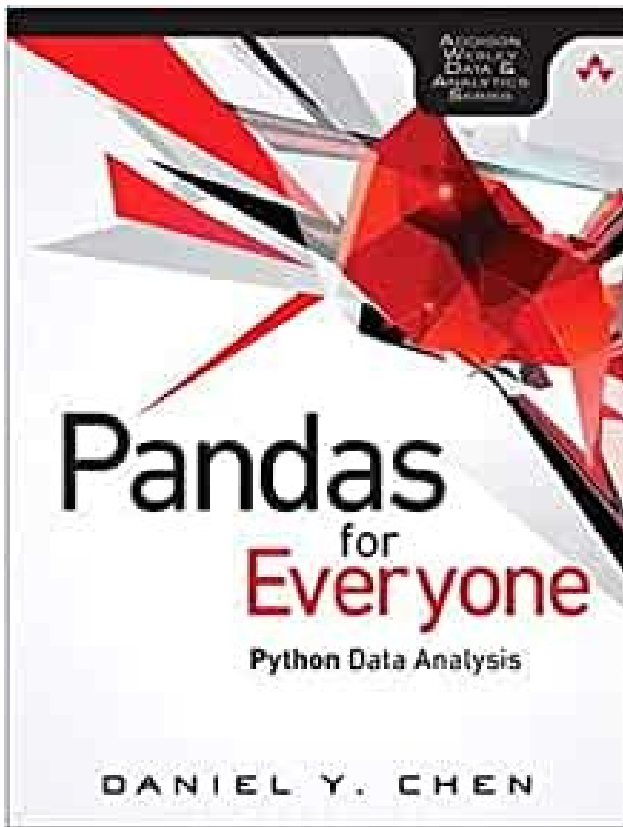
Ementa

- Linguagem de programação Python: tipos básicos de dados, comandos de entrada e saída de dados, desvio condicional, comandos de repetição, manipulação de arquivos;
- Biblioteca Pandas e operações com DataFrames em conjuntos de dados;
- Instalação e utilização do ambiente de desenvolvimento Jupyter Notebooks;
- Noções de utilização da biblioteca Numpy.

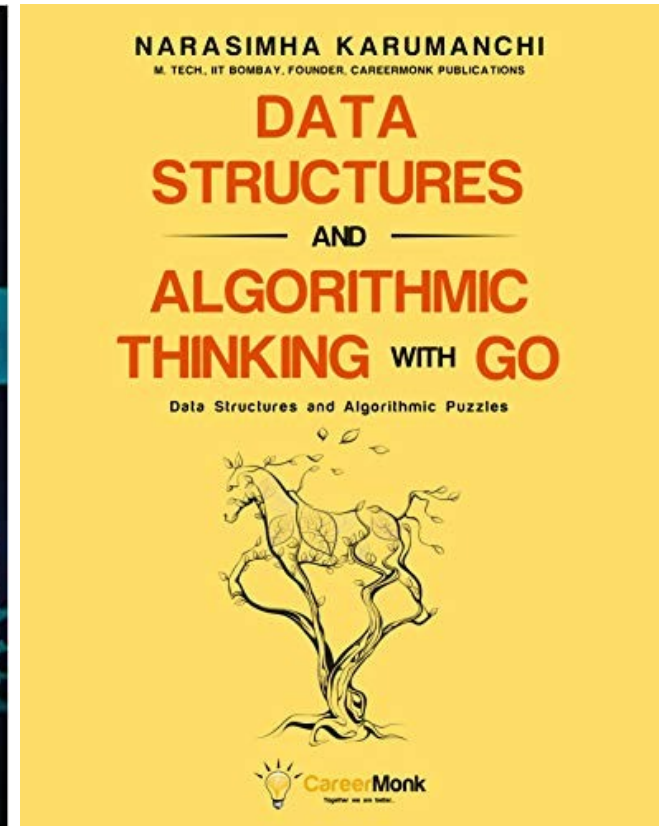
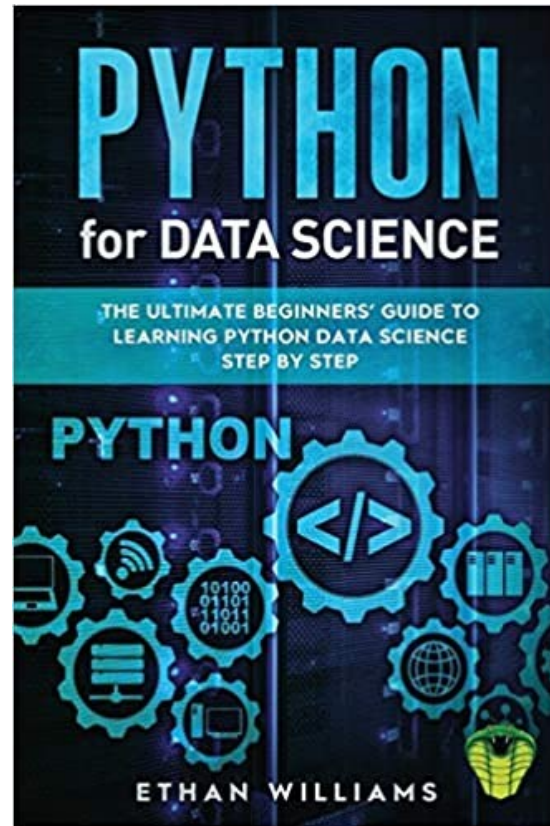
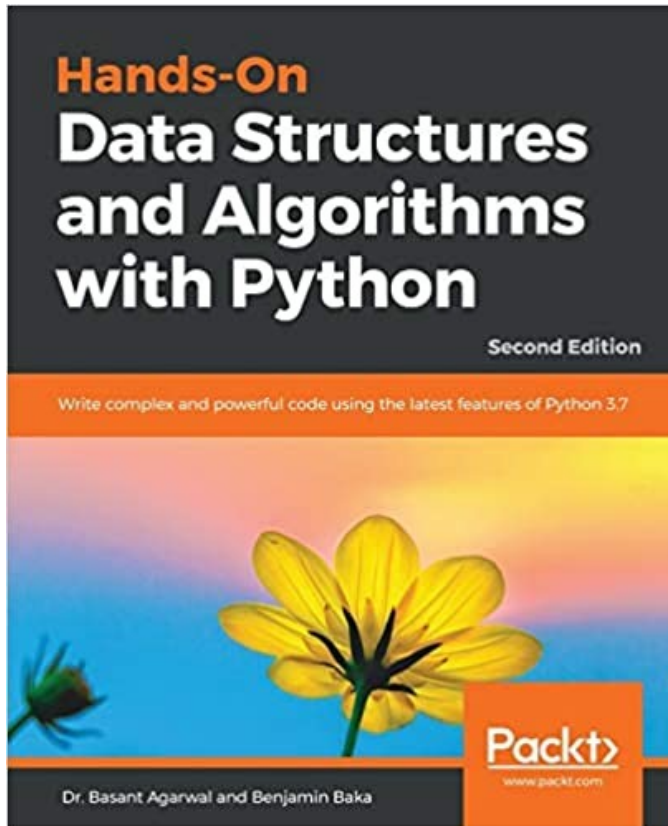
Plano de Ensino



Referências Bibliográficas



Referências Bibliográficas



Avaliação

- Nós iremos discutir a avaliação ao final da aula de hoje.

Roda de Conversa (apresentações)





Python

Python

- Linguagem de *script* orientada a objetos (1991) e desenvolvida por Guido van Rossum.

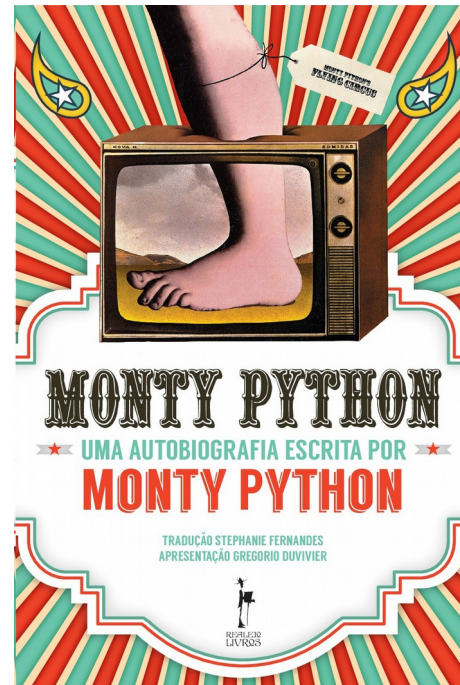
Python

- Linguagem de *script* orientada a objetos (1991) e desenvolvida por Guido van Rossum.



Python

- Linguagem de *script* orientada a objetos (1991) e desenvolvida por Guido van Rossum.



Características

Características

- Código-aberto.

Características

- Código-aberto.
- Linguagem interpretada.

Características

- Código-aberto.
- Linguagem interpretada.
- Facilidade de aprendizagem.

Características

- Código-aberto.
- Linguagem interpretada.
- Facilidade de aprendizagem.
- Comunidade ativa.

Características

- Código-aberto.
- Linguagem interpretada.
- Facilidade de aprendizagem.
- Comunidade ativa.
- Versões para diversas plataformas (Linux, MacOS e Windows).

Características

- Código-aberto.
- Linguagem interpretada.
- Facilidade de aprendizagem.
- Comunidade ativa.
- Versões para diversas plataformas (Linux, MacOS e Windows).
- Suporte aos principais paradigmas de programação: procedural, funcional e orientada a objetos.

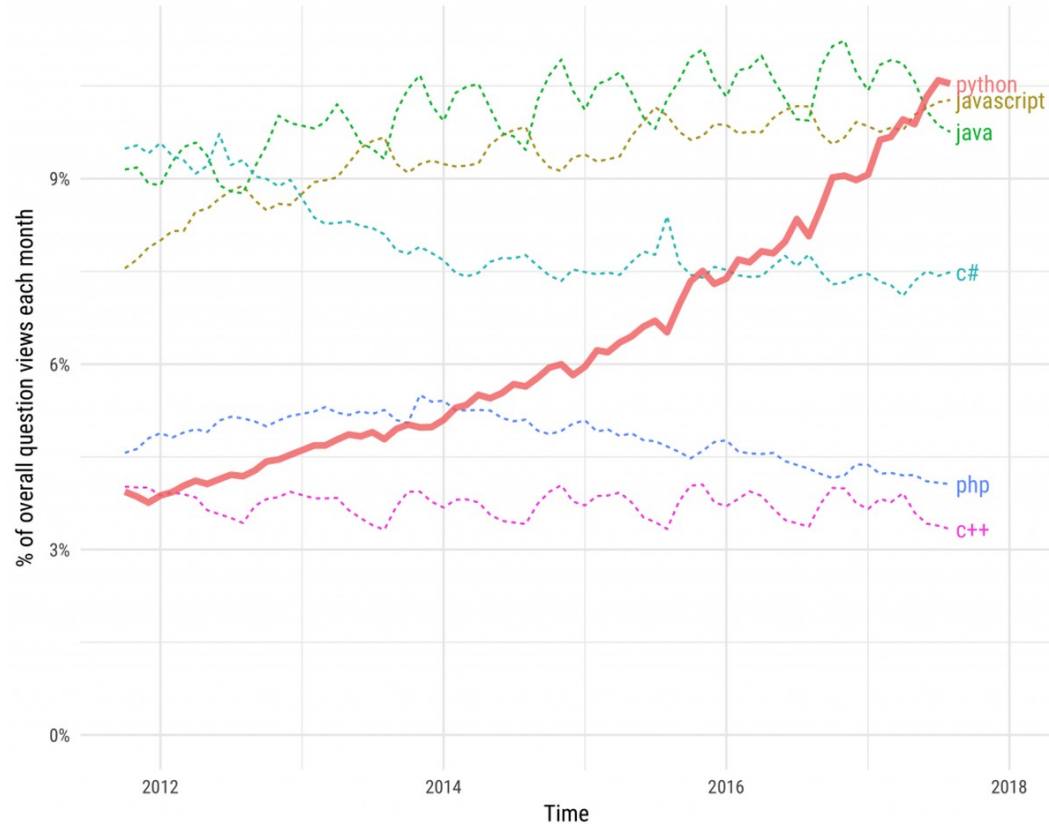
Características

- Código-aberto.
- Linguagem interpretada.
- Facilidade de aprendizagem.
- Comunidade ativa.
- Versões para diversas plataformas (Linux, MacOS e Windows).
- Suporte aos principais paradigmas de programação: procedural, funcional e orientada a objetos.
- Muito popular em aplicações nas áreas de educação, Web e científica.
















Popularidade

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



Popularidade

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	96.3
3	C	  	94.4
4	C++	  	87.5
5	R		81.5
6	JavaScript		79.4
7	C#	   	74.5
8	Matlab		70.6
9	Swift	 	69.1
10	Go	 	68.0

1. <https://www.zdnet.com>

Python 2 versus Python 3

Python 2 versus Python 3

- Python 2.0 foi lançada em 2000.

Python 2 versus Python 3

- Python 2.0 foi lançada em 2000.
- Python 3.0 foi lançada em 2008.

Python 2 versus Python 3

- Python 2.0 foi lançada em 2000.
- Python 3.0 foi lançada em 2008.
- Python 2.7 foi atualizada em 2010.

Python 2 versus Python 3

- Python 2.0 foi lançada em 2000.
- Python 3.0 foi lançada em 2008.
- Python 2.7 foi atualizada em 2010.
- Python 3.7 foi lançada em 2016.

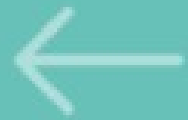
Python 2 versus Python 3

- Python 2.0 foi lançada em 2000.
- Python 3.0 foi lançada em 2008.
- Python 2.7 foi atualizada em 2010.
- Python 3.7 foi lançada em 2016.
- Python 3.8 foi lançada em 2019.

Python 2 versus Python 3

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

Python 2 versus Python 3



Legacy



Future



It is still entrenched in the software at certain companies

It will take over Python 2 by 2020

Python 2 versus Python 3



Library

Many older libraries built for Python 2 are not forwards-compatible



Library



Many of today's developers are creating libraries strictly for use with Python 3

Python 2 versus Python 3



Python 2 versus Python 3



$$5/2=2$$

It rounds your calculation down to the nearest whole number



$$5/2=2.5$$



The expression `5 / 2` will return the expected result

Python 2 versus Python 3

```
print "hello"
```



```
print ("hello")
```

Python 2 print statement

The print statement has been replaced with a print () function

Python 2 versus Python 3

- Python 2 deve se descontinuada.



Active Python Releases

For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.8	bugfix	2019-10-14	2024-10	PEP 569
3.7	security	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

Posso escolher sempre a versão
mais atual?

Posso escolher sempre a versão mais atual?

- Não.

Posso escolher sempre a versão mais atual?

- Não.
- É comum utilizarmos diferentes versões de uma mesma biblioteca entre diferentes projetos.

Posso escolher sempre a versão mais atual?

- Não.
- É comum utilizarmos diferentes versões de uma mesma biblioteca entre diferentes projetos.
- Isto pode gerar muitos conflitos entre as versões e causar muita dor de cabeça.

Qual alternativa para lidar com diferentes versões de Python?

Qual alternativa para lidar com diferentes versões de Python?

- O mais comum é utilizarmos diferentes ambientes virtuais.

Qual alternativa para lidar com diferentes versões de Python?

- O mais comum é utilizarmos diferentes ambientes virtuais.
- Estes ambientes são chamados de **virtualenvs**.

Qual alternativa para lidar com diferentes versões de Python?

- O mais comum é utilizarmos diferentes ambientes virtuais.
- Estes ambientes são chamados de **virtualenvs**.
- Um ambiente para cada projeto.

VirtualEnv

VirtualEnv

- Um ambiente virtual empacota todas as dependências que um projeto precisa e armazena em um diretório.

VirtualEnv

- Um ambiente virtual empacota todas as dependências que um projeto precisa e armazena em um diretório.
- Portanto, nenhum pacote será instalado diretamente no sistema operacional.

VirtualEnv

- Um ambiente virtual empacota todas as dependências que um projeto precisa e armazena em um diretório.
- Portanto, nenhum pacote será instalado diretamente no sistema operacional.
- Cada projeto terá o seu próprio ambiente e suas bibliotecas em versões específicas.

Como funciona VirtualEnv?

Como funciona VirtualEnv?

- Funcionamento simples.

Como funciona VirtualEnv?

- Funcionamento simples.
- Cópia de diretórios necessários para que um programa Python seja executado.

Como funciona VirtualEnv?

- Funcionamento simples.
- Cópia de diretórios necessários para que um programa Python seja executado.
- Além de outros recursos:

Como funciona VirtualEnv?

- Funcionamento simples.
- Cópia de diretórios necessários para que um programa Python seja executado.
- Além de outros recursos:
 - PIP (gerenciador de pacotes).

Como funciona VirtualEnv?

- Funcionamento simples.
- Cópia de diretórios necessários para que um programa Python seja executado.
- Além de outros recursos:
 - PIP (gerenciador de pacotes).
 - Versão do Python (2.x ou 3.x).

Como funciona VirtualEnv?

- Funcionamento simples.
- Cópia de diretórios necessários para que um programa Python seja executado.
- Além de outros recursos:
 - PIP (gerenciador de pacotes).
 - Versão do Python (2.x ou 3.x).
 - Dependências instaladas com PIP.

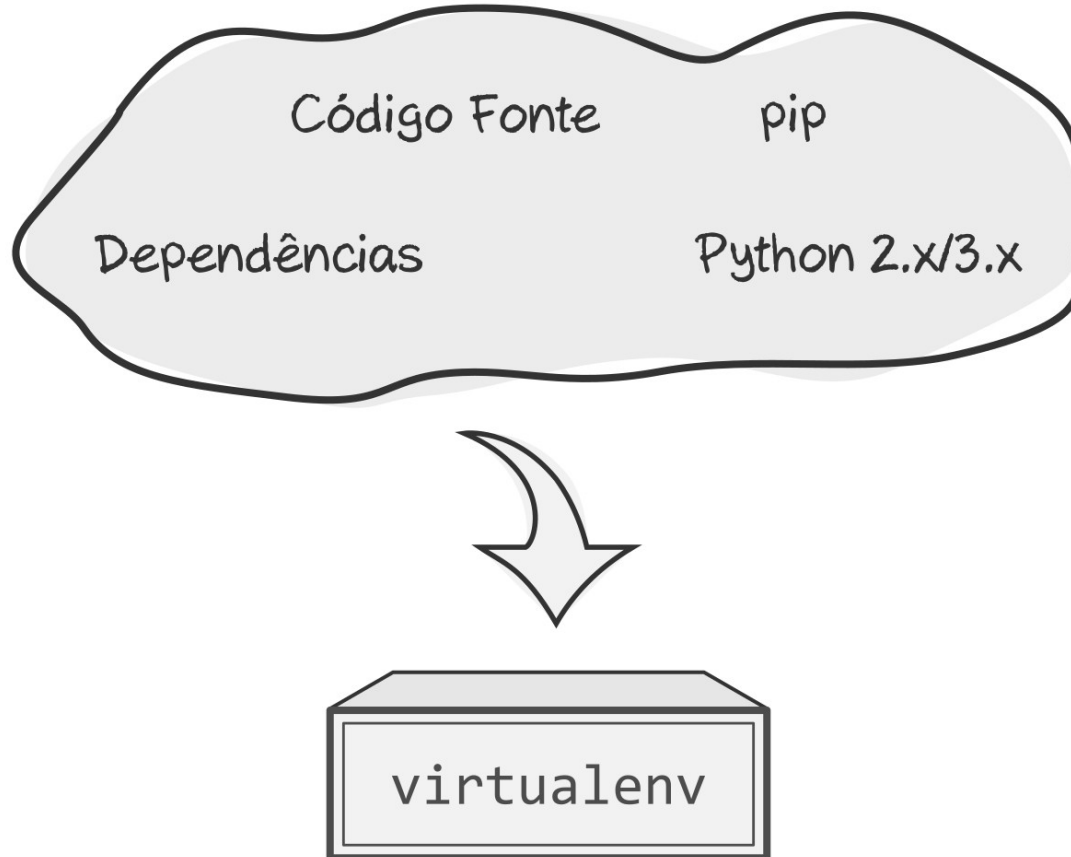
Como funciona VirtualEnv?

- Funcionamento simples.
- Cópia de diretórios necessários para que um programa Python seja executado.
- Além de outros recursos:
 - PIP (gerenciador de pacotes).
 - Versão do Python (2.x ou 3.x).
 - Dependências instaladas com PIP.
 - Código-fonte.

Como funciona VirtualEnv?

- Funcionamento simples.
- Cópia de diretórios necessários para que um programa Python seja executado.
- Além de outros recursos:
 - PIP (gerenciador de pacotes).
 - Versão do Python (2.x ou 3.x).
 - Dependências instaladas com PIP.
 - Código-fonte.
 - Bibliotecas comuns de Python

VirtualEnv



Bibliotecas de Ciência de Dados

Bibliotecas de Ciência de Dados

- Enorme conjunto de desenvolvedores

Bibliotecas de Ciência de Dados

- Enorme conjunto de desenvolvedores
- Muitas bibliotecas de código-aberto (*open source*)

Bibliotecas de Ciência de Dados

- Enorme conjunto de desenvolvedores
- Muitas bibliotecas de código-aberto (*open source*)
 - Computação científica.

Bibliotecas de Ciência de Dados

- Enorme conjunto de desenvolvedores
- Muitas bibliotecas de código-aberto (*open source*)
 - Computação científica.
 - Análise e processamento de dados.

Bibliotecas de Ciência de Dados

- Enorme conjunto de desenvolvedores
- Muitas bibliotecas de código-aberto (*open source*)
 - Computação científica.
 - Análise e processamento de dados.
 - Visualização de dados.

Bibliotecas de Ciência de Dados

- Enorme conjunto de desenvolvedores
- Muitas bibliotecas de código-aberto (*open source*)
 - Computação científica.
 - Análise e processamento de dados.
 - Visualização de dados.
 - Aprendizagem de máquinas e redes neurais.

Bibliotecas de Ciência de Dados

- Enorme conjunto de desenvolvedores
- Muitas bibliotecas de código-aberto (*open source*)
 - Computação científica.
 - Análise e processamento de dados.
 - Visualização de dados.
 - Aprendizagem de máquinas e redes neurais.
 - Processamento de texto.

Anaconda

Anaconda

- Pacote com muitas ferramentas usadas para Ciência de Dados com aprendizagem de máquina.

Anaconda

- Pacote com muitas ferramentas usadas para Ciência de Dados com aprendizagem de máquina.
- Funciona com Python e R.

Anaconda

- Pacote com muitas ferramentas usadas para Ciência de Dados com aprendizagem de máquina.
- Funciona com Python e R.
- Roda em Linux, MacOS e Windows.

Anaconda

- Pacote com muitas ferramentas usadas para Ciência de Dados com aprendizagem de máquina.
- Funciona com Python e R.
- Roda em Linux, MacOS e Windows.
- COM versus ORG.

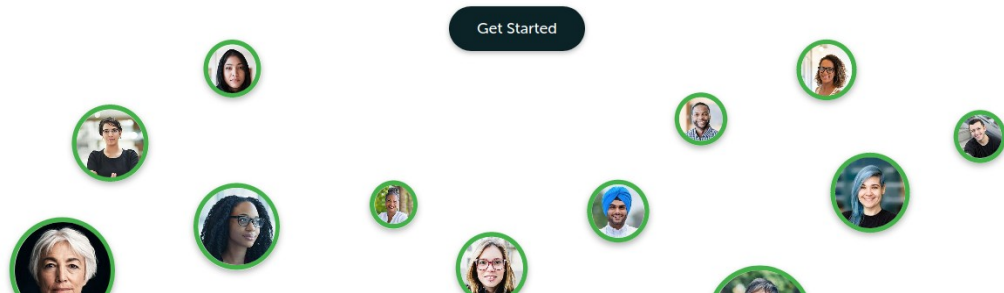
Anaconda

- anaconda.com



Data science technology for
human sensemaking.

A movement that brings together millions of data science practitioners,
data-driven enterprises, and the open source community.



Anaconda

- anaconda.org

The screenshot displays the Anaconda Cloud website interface. At the top, the navigation bar includes the Anaconda Cloud logo and links for Gallery, About, Anaconda, Help, Download Anaconda, and Sign In. A search bar is positioned below the navigation. The main content area features the Anaconda Cloud logo and the text: "Where packages, notebooks, projects and environments are shared. Your place for free public conda package hosting." On the right side, there is a sign-up form with fields for "Pick a username", "Your email", "Create a password" (with a note: "Use at least one lowercase letter, one numeral, and seven characters."), and "Confirm password". Below the form are checkboxes for "I accept the Terms & Conditions" and "Não sou um robô" (reCAPTCHA). A prominent green "Sign up!" button is located at the bottom of the form. A footer bar at the bottom contains the text "Anaconda Enterprise".

Computação Científica

Computação Científica

- NumPy

Computação Científica

- NumPy
 - Fornece uma representação de alta performance na manipulação de listas e matrizes.

Computação Científica

- NumPy
 - Fornece uma representação de alta performance na manipulação de listas e matrizes.
 - Além de rotinas para processamento de tais estruturas.

Computação Científica

- NumPy
 - Fornece uma representação de alta performance na manipulação de listas e matrizes.
 - Além de rotinas para processamento de tais estruturas.
 - <https://numpy.org>

Computação Científica

Computação Científica

- SciPy

Computação Científica

- SciPy
 - Construída sobre NumPy.

Computação Científica

- SciPy
 - Construída sobre NumPy.
 - Rotinas de processamento científico, tais como integrais, equações diferenciais e processamento de matrizes.

Computação Científica

- SciPy
 - Construída sobre NumPy.
 - Rotinas de processamento científico, tais como integrais, equações diferenciais e processamento de matrizes.
 - <https://www.scipy.org>

Análise e Processamento de Dados

Análise e Processamento de Dados

- Pandas

Análise e Processamento de Dados

- Pandas
 - Popular biblioteca para processamento de dados.

Análise e Processamento de Dados

- Pandas
 - Popular biblioteca para processamento de dados.
 - Faz uso de NumPy.

Análise e Processamento de Dados

- Pandas
 - Popular biblioteca para processamento de dados.
 - Faz uso de NumPy.
 - <https://pandas.pydata.org>

Visualização de Dados

Visualização de Dados

- Matplotlib

Visualização de Dados

- MatPlotLib
 - Biblioteca com muitos recursos de visualização e plotagem de dados.

Visualização de Dados

- MatPlotLib
 - Biblioteca com muitos recursos de visualização e plotagem de dados.
 - Possui um vasto conjunto de tipos de gráficos (barra, pizza, 3D, texto, etc).

Visualização de Dados

- MatPlotLib
 - Biblioteca com muitos recursos de visualização e plotagem de dados.
 - Possui um vasto conjunto de tipos de gráficos (barra, pizza, 3D, texto, etc).
 - <https://matplotlib.org>

Visualização de Dados

Visualização de Dados

- Seaborn

Visualização de Dados

- Seaborn
 - Biblioteca de alto nível construída sobre Matplotlib.

Visualização de Dados

- Seaborn
 - Biblioteca de alto nível construída sobre Matplotlib.
 - Recurso de *look-and-feel*.

Visualização de Dados

- Seaborn
 - Biblioteca de alto nível construída sobre Matplotlib.
 - Recurso de *look-and-feel*.
 - Visualizações adicionais.

Visualização de Dados

- Seaborn
 - Biblioteca de alto nível construída sobre Matplotlib.
 - Recurso de *look-and-feel*.
 - Visualizações adicionais.
 - Geração de gráficos com menos código.

Visualização de Dados

- Seaborn
 - Biblioteca de alto nível construída sobre Matplotlib.
 - Recurso de *look-and-feel*.
 - Visualizações adicionais.
 - Geração de gráficos com menos código.
 - <https://seaborn.pydata.org>

Aprendizagem de Máquina



Aprendizagem de Máquina



- SciKit Learn

Aprendizagem de Máquina

- SciKit Learn
 - Considerada como a principal biblioteca de aprendizagem de máquina (*machine-learning*).

Redes Neurais



Redes Neurais



- Keras

Redes Neurais



- Keras
 - Uma das ferramentas mais fáceis de usar bibliotecas de redes neurais (*deep learning*).

Redes Neurais



- Keras
 - Uma das ferramentas mais fáceis de usar bibliotecas de redes neurais (*deep learning*).
 - Roda sobre o TensorFlow (Google), CNTK (Microsoft) e Theano (Université de Montréal).

Redes Neurais



- Keras
 - Uma das ferramentas mais fáceis de usar bibliotecas de redes neurais (*deep learning*).
 - Roda sobre o TensorFlow (Google), CNTK (Microsoft) e Theano (Université de Montréal).
 - <https://keras.io>

Redes Neurais



Redes Neurais



- TensorFlow

Redes Neurais



- TensorFlow
 - Biblioteca desenvolvida pelo Google.

Redes Neurais



- TensorFlow
 - Biblioteca desenvolvida pelo Google.
 - É a biblioteca de *deep learning* mais popular.

Redes Neurais

- TensorFlow
 - Biblioteca desenvolvida pelo Google.
 - É a biblioteca de *deep learning* mais popular.
 - Roda com GPUs (*graphics processing units*) e TPUs (*tensor processing units*).

Redes Neurais

- TensorFlow
 - Biblioteca desenvolvida pelo Google.
 - É a biblioteca de *deep learning* mais popular.
 - Roda com GPUs (*graphics processing units*) e TPUs (*tensor processing units*).
 - Importante plataforma na área de *big data*.

Redes Neurais

- TensorFlow
 - Biblioteca desenvolvida pelo Google.
 - É a biblioteca de *deep learning* mais popular.
 - Roda com GPUs (*graphics processing units*) e TPUs (*tensor processing units*).
 - Importante plataforma na área de *big data*.
 - <https://www.tensorflow.org>

Processamento de Texto

Processamento de Texto

- NLTK

Processamento de Texto

- NLTK
 - *Natural Language Toolkit.*

Processamento de Texto

- NLTK
 - *Natural Language Toolkit.*
 - Usada em problemas de linguagem natural (*Natural Language Processing* - NLP).

Processamento de Texto

- NLTK
 - *Natural Language Toolkit.*
 - Usada em problemas de linguagem natural (*Natural Language Processing* - NLP).
 - Principal ferramenta para processamento de texto.

Processamento de Texto

- NLTK
 - *Natural Language Toolkit*.
 - Usada em problemas de linguagem natural (*Natural Language Processing* - NLP).
 - Principal ferramenta para processamento de texto.
 - <https://www.nltk.org>

Processamento de Texto

- NLTK
 - *Natural Language Toolkit.*
 - Usada em problemas de linguagem natural (*Natural Language Processing* - NLP).
 - Principal ferramenta para processamento de texto.
 - <https://www.nltk.org>
 - Um problema é a falta de ferramentas para PT-BR.

Processamento de Texto

Processamento de Texto

- TextBlob

Processamento de Texto

- TextBlob
 - Ferramenta com vários recursos de processamento de texto.

Processamento de Texto

- TextBlob
 - Ferramenta com vários recursos de processamento de texto.
 - Construída sobre a NLTK.

Processamento de Texto

- TextBlob
 - Ferramenta com vários recursos de processamento de texto.
 - Construída sobre a NLTK.
 - Fácil de usar.

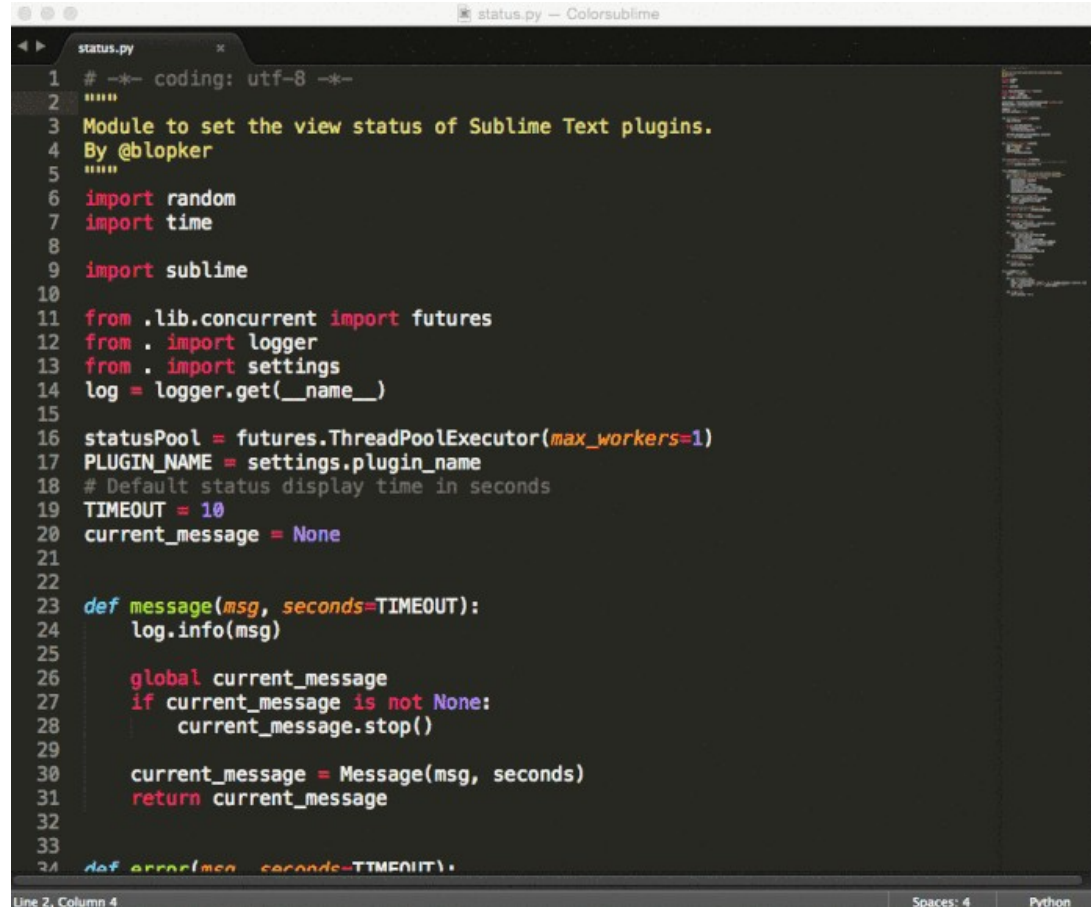
Processamento de Texto

- TextBlob
 - Ferramenta com vários recursos de processamento de texto.
 - Construída sobre a NLTK.
 - Fácil de usar.
 - <https://textblob.readthedocs.io>

Ambiente de Desenvolvimento

IDE

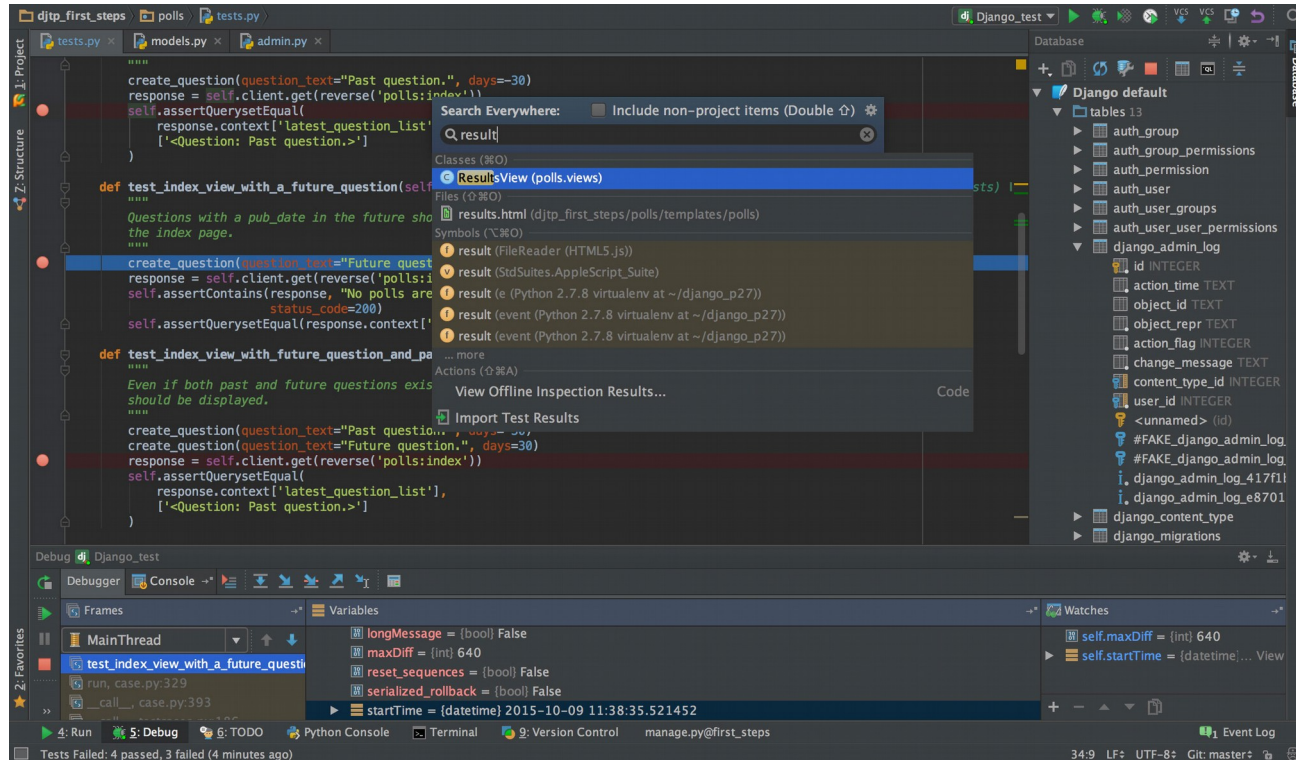
- Sublime Text



```
1 # -*- coding: utf-8 -*-
2 """
3 Module to set the view status of Sublime Text plugins.
4 By @bloper
5 """
6 import random
7 import time
8
9 import sublime
10
11 from .lib.concurrent import futures
12 from . import logger
13 from . import settings
14 log = logger.get(__name__)
15
16 statusPool = futures.ThreadPoolExecutor(max_workers=1)
17 PLUGIN_NAME = settings.plugin_name
18 # Default status display time in seconds
19 TIMEOUT = 10
20 current_message = None
21
22
23 def message(msg, seconds=TIMEOUT):
24     log.info(msg)
25
26     global current_message
27     if current_message is not None:
28         current_message.stop()
29
30     current_message = Message(msg, seconds)
31     return current_message
32
33
34 def error(msg, seconds=TIMEOUT):
```

IDE

- PyCharm



IDE

- Rodeo

The screenshot displays the Rodeo IDE interface. The main editor window shows a Python script with the following code:

```
1 import time
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5
6 d = []
7 d.
8 d.append
9 d.count
10 d.extend
11 d.index
12 d.insert
13 d.pop
14 d.remove
15 d.reverse
16
17
18
19 for i in range(4):
20     d.append(np.random.exponential(i+1,(1,500))[0].tolist())
21     sns.distplot(d[i],hist=False)
22
23 from yhat import Yhat, YhatModel, preprocess
24
```

A tooltip for `d.append` is visible, showing the signature: `L.append(object) -- append object to end`.

The Environment panel on the right shows the following variables:

Variable	Value	Action
d	List of length 4	View
preprocess	def preprocess(func)	View
i	<type 'int'>	View
x	<type 'int'>	View
Sleep	<type 'type'>	View
Yhat	<type 'type'>	View
YhatModel	<type 'type'>	View

The Console panel shows the following output:

```
>>> pd.DataFrame
```

The Plots panel shows a histogram of the data generated by the script, with four overlapping density curves in blue, green, red, and purple. The x-axis ranges from -5 to 30, and the y-axis ranges from 0.0 to 0.7. A vertical stack of smaller plot thumbnails is visible on the right side of the Plots panel.

IDE



SPYDER

The Scientific Python Development Environment

The screenshot displays the Spyder IDE interface with the following components:

- Code Editor:** Contains Python code for generating data, performing calculations, and plotting a 3D surface. The code includes imports for `numpy` and `matplotlib`, and uses `splprep` and `splev` for spline fitting.
- Variable Explorer:** Shows a table of variables in the current environment, including `bars`, `df`, `filename`, `list_test`, `nrows`, `r`, `radius`, `region`, `rgb`, `series`, and `test_size`.
- Python Console:** Displays the execution of the code, showing the output of `plt.show()` and the generation of a 3D plot.
- 3D Plot:** A 3D surface plot showing a landscape with a color gradient from blue to red. The axes are labeled with numerical values.

NOTEBOOKS

Jupyter

Jupyter

- É uma ferramenta *open source* baseada em navegadores Web que integra linguagens de programação, bibliotecas e ferramentas de visualização de dados.

Jupyter

- É uma ferramenta *open source* baseada em navegadores Web que integra linguagens de programação, bibliotecas e ferramentas de visualização de dados.
- Jupyter notebook pode ser utilizada localmente ou em nuvem (*cloud*).

Jupyter

- É uma ferramenta *open source* baseada em navegadores Web que integra linguagens de programação, bibliotecas e ferramentas de visualização de dados.
- Jupyter notebook pode ser utilizada localmente ou em nuvem (*cloud*).
- Cada documento é composto de múltiplas células, onde cada célula contém um código de programação ou uma linguagem de marcação.

Jupyter

- É uma ferramenta *open source* baseada em navegadores Web que integra linguagens de programação, bibliotecas e ferramentas de visualização de dados.
- Jupyter notebook pode ser utilizada localmente ou em nuvem (*cloud*).
- Cada documento é composto de múltiplas células, onde cada célula contém um código de programação ou uma linguagem de marcação.
- Saídas típicas incluem texto, tabelas e gráficos.

Jupyter

- É uma ferramenta *open source* baseada em navegadores Web que integra linguagens de programação, bibliotecas e ferramentas de visualização de dados.
- Jupyter notebook pode ser utilizada localmente ou em nuvem (*cloud*).
- Cada documento é composto de múltiplas células, onde cada célula contém um código de programação ou uma linguagem de marcação.
- Saídas típicas incluem texto, tabelas e gráficos.
- Essa tecnologia torna mais fácil o compartilhamento de dados e a reprodução de experimentação científica.

Jupyter

- É uma ferramenta *open source* baseada em navegadores Web que integra linguagens de programação, bibliotecas e ferramentas de visualização de dados.
- Jupyter notebook pode ser utilizada localmente ou em nuvem (*cloud*).
- Cada documento é composto de múltiplas células, onde cada célula contém um código de programação ou uma linguagem de marcação.
- Saídas típicas incluem texto, tabelas e gráficos.
- Essa tecnologia torna mais fácil o compartilhamento de dados e a reprodução de experimentação científica.
- <https://jupyter.readthedocs.io>

Google Colaboratory



Google Colaboratory

- É um projeto que tem o objetivo de disseminar a pesquisa em *machine learning*.

Google Colaboratory

- É um projeto que tem o objetivo de disseminar a pesquisa em *machine learning*.
- Colaboratory notebooks são baseados em Jupyter e funcionam como um objeto do Google Docs.

Google Colaboratory

- É um projeto que tem o objetivo de disseminar a pesquisa em *machine learning*.
- Colaboratory notebooks são baseados em Jupyter e funcionam como um objeto do Google Docs.
- Roda em Python 2 e 3.

Google Colaboratory

- É um projeto que tem o objetivo de disseminar a pesquisa em *machine learning*.
- Colaboratory notebooks são baseados em Jupyter e funcionam como um objeto do Google Docs.
- Roda em Python 2 e 3.
- Possui um ambiente pré-configurado com as principais bibliotecas de *machine learning*, tais como TensorFlow, Matplotlib e Keras.

Google Colaboratory

- É um projeto que tem o objetivo de disseminar a pesquisa em *machine learning*.
- Colaboratory notebooks são baseados em Jupyter e funcionam como um objeto do Google Docs.
- Roda em Python 2 e 3.
- Possui um ambiente pré-configurado com as principais bibliotecas de *machine learning*, tais como TensorFlow, Matplotlib e Keras.
- A máquina virtual (VM) é desativada depois de um período de tempo e todos os dados e configurações do usuário são perdidas.

Google Colaboratory

- É um projeto que tem o objetivo de disseminar a pesquisa em *machine learning*.
- Colaboratory notebooks são baseados em Jupyter e funcionam como um objeto do Google Docs.
- Roda em Python 2 e 3.
- Possui um ambiente pré-configurado com as principais bibliotecas de *machine learning*, tais como TensorFlow, Matplotlib e Keras.
- A máquina virtual (VM) é desativada depois de um período de tempo e todos os dados e configurações do usuário são perdidas.
- Porém, o notebook é mantido (persistido).

Google Colaboratory

- É um projeto que tem o objetivo de disseminar a pesquisa em *machine learning*.
- Colaboratory notebooks são baseados em Jupyter e funcionam como um objeto do Google Docs.
- Roda em Python 2 e 3.
- Possui um ambiente pré-configurado com as principais bibliotecas de *machine learning*, tais como TensorFlow, Matplotlib e Keras.
- A máquina virtual (VM) é desativada depois de um período de tempo e todos os dados e configurações do usuário são perdidas.
- Porém, o notebook é mantido (persistido).
- Possui ainda recursos de GPU.

Google Colaboratory

- É um projeto que tem o objetivo de disseminar a pesquisa em *machine learning*.
- Colaboratory notebooks são baseados em Jupyter e funcionam como um objeto do Google Docs.
- Roda em Python 2 e 3.
- Possui um ambiente pré-configurado com as principais bibliotecas de *machine learning*, tais como TensorFlow, Matplotlib e Keras.
- A máquina virtual (VM) é desativada depois de um período de tempo e todos os dados e configurações do usuário são perdidas.
- Porém, o notebook é mantido (persistido).
- Possui ainda recursos de GPU.
- <https://colab.research.google.com>

OUTROS AMBIENTES

DashBoards

DashBoards

- É um painel visual que apresenta, de maneira centralizada, um conjunto informações.

DashBoards

- É um painel visual que apresenta, de maneira centralizada, um conjunto informações.
- Ferramenta usada para tomada de decisões.

DashBoards

- É um painel visual que apresenta, de maneira centralizada, um conjunto informações.
- Ferramenta usada para tomada de decisões.
- Aplicações:

DashBoards

- É um painel visual que apresenta, de maneira centralizada, um conjunto informações.
- Ferramenta usada para tomada de decisões.
- Aplicações:
 - Processos da empresa.

DashBoards

- É um painel visual que apresenta, de maneira centralizada, um conjunto informações.
- Ferramenta usada para tomada de decisões.
- Aplicações:
 - Processos da empresa.
 - Indicadores em tempo real.

DashBoards

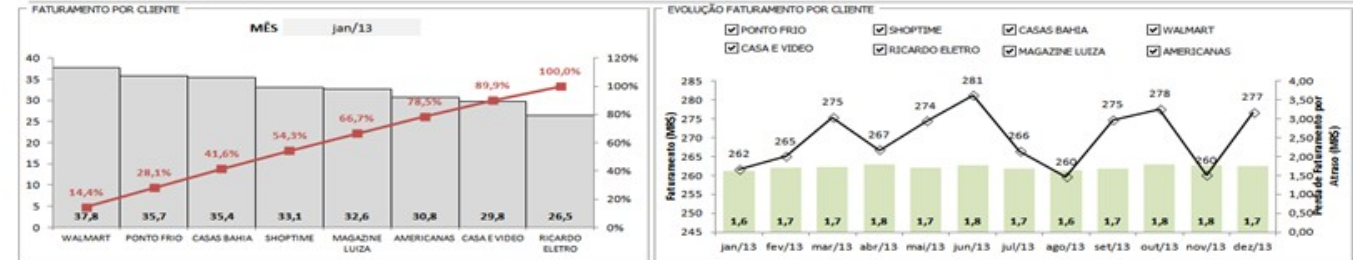
PAINEL GERENCIAL DA OPERAÇÃO

Atualização: 24/02/2014

1 VISÃO GERAL



2 FATURAMENTO DETALHADO



3 INDICADORES OPERACIONAIS



Exemplo

Casos recuperados

3.453.336

Em acompanhamento
616.014

CASOS CONFIRMADOS

4.197.889

Acumulado

1997,6

Incidência*

35.816

Casos novos

ÓBITOS CONFIRMADOS

128.539

Óbitos acumulados

3,1 %

Letalidade

1.075

Casos novos

61,2

Mortalidade*

Painel Interativo

Navegue pelos dados do Sistema Único de Saúde - SUS, com informações estratégicas e conheça tudo sobre a COVID-19 de forma transparente e analítica.

Acesse



Síntese de casos, óbitos, incidência e mortalidade

Pesquise uma localidade

Brasil

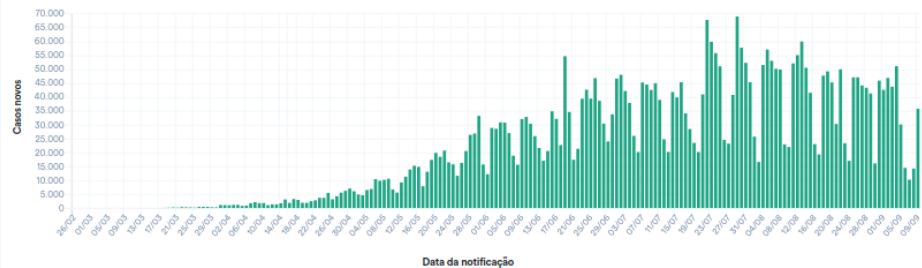
	Casos	Óbitos	Incidência/100mil hab.	Mortalidade/100mil hab	Atualização
> RO	59.196	1.207	3330,8	67,9	09/09 18:30
> AC	25.668	631	2910,4	71,5	09/09 18:30
> AP	45.251	675	5350,5	79,8	09/09 18:30
> TO	56.684	762	3603,9	48,4	09/09 18:30
> AM	124.959	3.862	3015,0	93,2	09/09 18:30
> Sul	486.891	10.002	1624,3	33,4	09/09 18:30

Fonte: Secretarias Estaduais de Saúde. Brasil, 2020

Exemplo

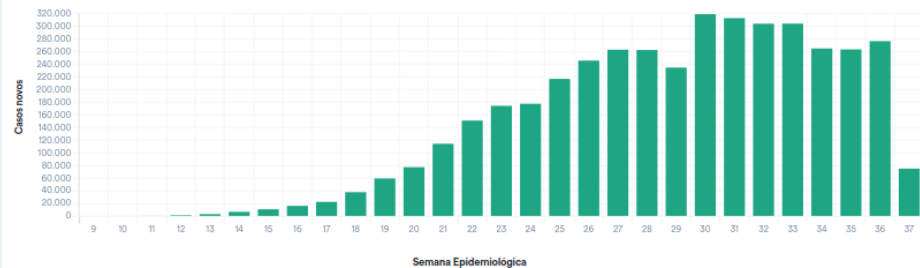
Casos Confirmados

Casos novos de COVID-19 por data de notificação



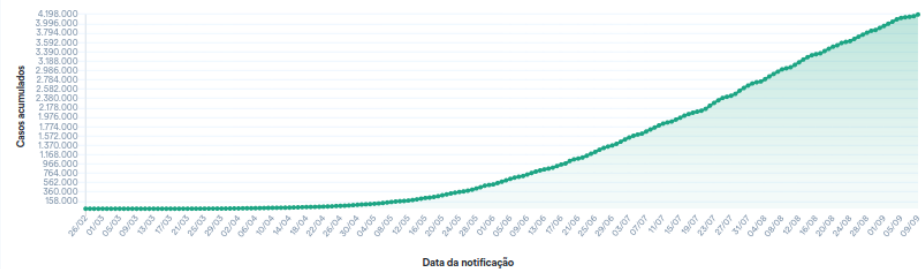
Fonte: Secretarias Estaduais de Saúde, Brasil, 2020

Casos novos de COVID-19 por Semana Epidemiológica de notificação



Fonte: Secretarias Estaduais de Saúde, Brasil, 2020

Casos acumulados de COVID-19 por data de notificação



Fonte: Secretarias Estaduais de Saúde, Brasil, 2020

Casos acumulados de COVID-19 por Semana Epidemiológica de notificação





Fonte: Secretarias Estaduais de Saúde, Brasil, 2020


SITES RELACIONADOS

Stack Overflow em Português é um site de perguntas e respostas para programadores profissionais e entusiastas. Leva apenas um minuto para se inscrever.

[Inscreva-se para participar desta comunidade](#)

 Qualquer pessoa pode fazer uma pergunta ×

 Qualquer um pode responder

 As melhores respostas recebem votos positivos e sobem para os primeiros lugares

- [Home](#)
- [Perguntas](#)
- [Tags](#)
- [Usuários](#)
- [Sem resposta](#)

Explore nossas perguntas

[Faça uma pergunta](#)

- [php](#)
[javascript](#)
[java](#)
[html](#)
[c#](#)
[jquery](#)
[mysql](#)

[android](#)
[css](#)
[python](#)
[mais tags](#)

- [Ativos](#)
[1 Com recompensa](#)
[Destaques](#)
[Semana](#)
[Mês](#)

0	0	4	Tempo de execução do código	
votos	respostas	visitas	python-3.x	perguntada 24 mins atrás user158657 97
1	1	220	Tempo de limite excedido tentando enviar email c# windows forms	
voto	resposta	visitas	c# winforms smtp backgroundworker	modificada 54 mins atrás Comunidade 1
-1	0	7	Ajuda com erro '>' not supported between instances of 'NoneType' and 'int'	
votos	respostas	visitas	python	modificada 57 mins atrás Naissinger 1

Tags recentes

- [javascript](#) × 157
- [python](#) × 95
- [html](#) × 67
- [r](#) × 50
- [css](#) × 40
- [php](#) × 38
- [java](#) × 32
- [c#](#) × 31
- [python-3.x](#) × 31
- [mysql](#) × 29
- [node.js](#) × 28

Repositories

New

Find a repository...

- 📁 [tmelo-uea/covid19](#)
- 🔒 [ThomazMatos/monitoropinion](#)
- 📁 [tmelo-uea/opinion-mining](#)
- 🔒 [elloa/lsi-covid19](#)

Working with a team?



GitHub is built for collaboration. Set up an organization to improve the way your team works together, and get access to more features.

Create an organization


Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.


Read the guide
Start a project

-  **migvanderlei** started following **LuizHenrique-BS** and **dudemaiaf** 9 days ago
-  **Henrique Barros** LuizHenrique-BS Follow

Estudante de Engenharia da Computação na Universidade do Estado do Amazonas (UEA-EST)

10 repositories 9 followers
-  **migvanderlei** starred 2 repositories 9 days ago
- realsirjoe/instagram-scraper** ☆ Star

scrapes medias, likes, followers, tags and all metadata. Inspired by instagram-php-scraper,bot



Python 1.6k Updated Sep 9
-  **migvanderlei** starred **gautham20/pytorch-ts** 11 days ago
- gautham20/pytorch-ts** ☆ Star

Tutorials on using encoder decoder architecture for time series forecasting

Jupyter Notebook 21 Updated Sep 3

GitHub for mobile

Projects, Ideas, & code to go. Available natively for Android and IOS.

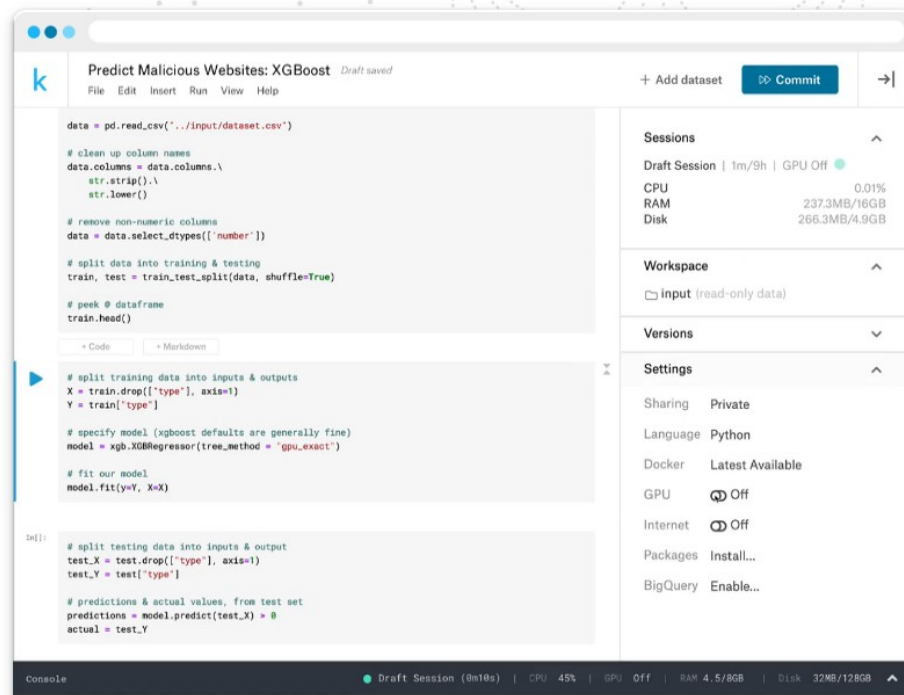
- Explore repositories**
- realsirjoe/instagram-scraper**
scrapes medias, likes, followers, tags and all metadata. Inspired by instagram-php-scraper,bot
Python 1.6k
-
- OpenGenus/cosmos**
Algorithms that run our universe | Your personal library of every algorithm and data structure code that you will ever encounter | Ask us anything at our forum |
C++ 12.3k
-
- tensorflow/tensorflow**
An Open Source Machine Learning Framework for Everyone
C++ 148k
- Explore more →

Start with more than a blinking cursor

Kaggle offers a no-setup, customizable, Jupyter Notebooks environment. Access free GPUs and a huge repository of community published data & code.

 REGISTER WITH GOOGLE

Register with Email



The screenshot shows a Kaggle notebook interface. The title is "Predict Malicious Websites: XGBoost" and it is marked as "Draft saved". The notebook contains Python code for data preprocessing, model training, and testing. The code includes steps for reading a CSV file, cleaning column names, removing non-numeric columns, splitting data into training and testing sets, training an XGBoost model, and evaluating it on the test set. The right sidebar shows the notebook's status: "Draft Session | 1m/9h | GPU Off", system resources (CPU 0.01%, RAM 237.3MB/16GB, Disk 266.3MB/4.9GB), workspace (input read-only data), versions, and settings (Sharing: Private, Language: Python, Docker: Latest Available, GPU: Off, Internet: Off, Packages: Install..., BigQuery: Enable...).

```
data = pd.read_csv('../input/dataset.csv')

# clean up column names
data.columns = data.columns.\
    str.strip().\
    str.lower()

# remove non-numeric columns
data = data.select_dtypes(['number'])

# split data into training & testing
train, test = train_test_split(data, shuffle=True)

# peek @ dataframe
train.head()

# split training data into inputs & outputs
X = train.drop(['type'], axis=1)
Y = train['type']

# specify model (xgboost defaults are generally fine)
model = xgb.XGBRegressor(tree_method = "gpu_exact")

# fit our model
model.fit(y=Y, X=X)

# split testing data into inputs & output
test_X = test.drop(['type'], axis=1)
test_Y = test['type']

# predictions & actual values, from test set
predictions = model.predict(test_X) > 0
actual = test_Y
```

Inside Kaggle you'll find all the code & data you need to do your data science work. Use over 50,000 public [datasets](#) and 400,000 public [notebooks](#) to conquer any analysis in no time.



FILTER BY

All posts

Citation updates

Article suggestions

Recently published articles

Share updates and links with your followers

Post

Download Mendeley

Access your library on mobile, tablet and desktop

Get Reference Manager



Tiago de Melo has new publications
Jun 20

Contender: Leveraging user opinions for purchase decision-making

de Melo T, da Silva A, de Moura E et al. [See more](#)

Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2019)

[View in library](#) [Get full text](#)



Tiago de Melo has new publications
Jun 20

An aspect-driven method for enriching product catalogs with user opinions

de Melo T, da Silva A, de Moura E

Journal of the Brazilian Computer Society (2018)



© 2020 Mendeley Ltd.

[Discover Mendeley](#) • [About](#) • [Work at Mendeley](#)

[Copyright](#) • [Terms of use](#) • [Privacy policy](#)

Mendeley supports responsible sharing

RELX Group™

EXERCÍCIOS

Questionário

- Explique quais são as características de Python que a tornam uma linguagem de programação apropriada para Ciência de Dados.
- Explique as diferenças entre Python 2.X e 3.X?
- O que são sistemas legados? Qual é a importância na escolha da versão da linguagem de programação em um projeto?
- Explique, com as suas próprias palavras, o benefício do uso de ambientes virtuais.
- Comenta uma vantagem no uso do Anaconda.
- Quais são os critérios na escolha de uma IDE?
- Qual é a diferença entre Jupyter e COLAB?
- Mencione dois sites onde podemos encontrar conjuntos de dados (datasets) para uso em problemas relacionados à Ciência de Dados.