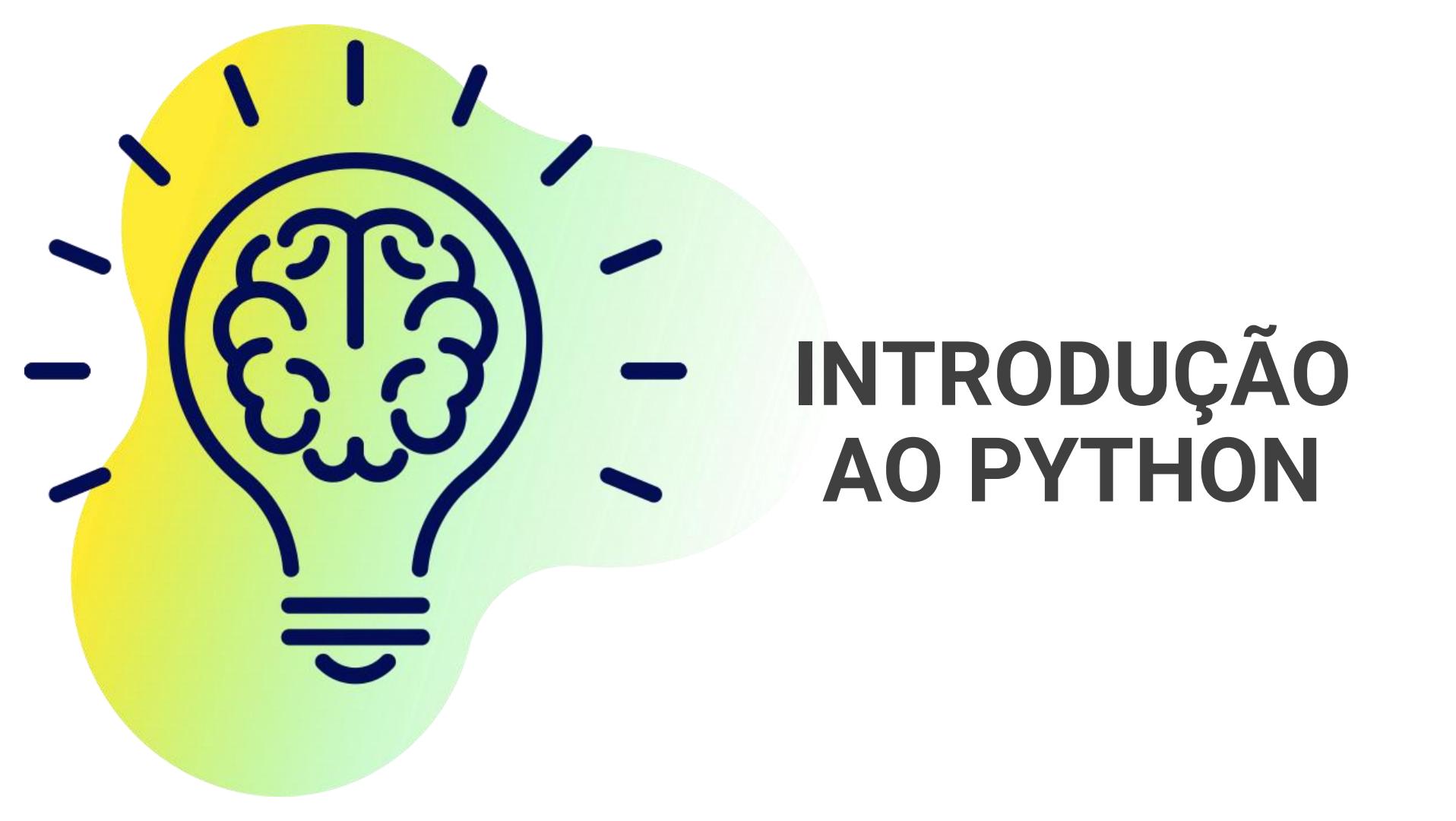


# Escola Livre de IA

Inteligência Artificial ao alcance de todos





# INTRODUÇÃO AO PYTHON

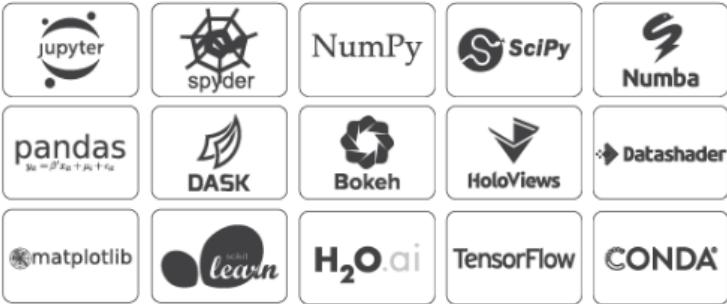


# Motivação e Objetivos da Aula



# ANACONDA®

<https://www.anaconda.com/distribution/>



Windows | macOS | Linux

O Anaconda é uma distribuição gratuita e de código aberto das linguagens de programação Python e R para Computação Científica e Ciência de Dados, que visa simplificar o gerenciamento e a implantação de pacotes.



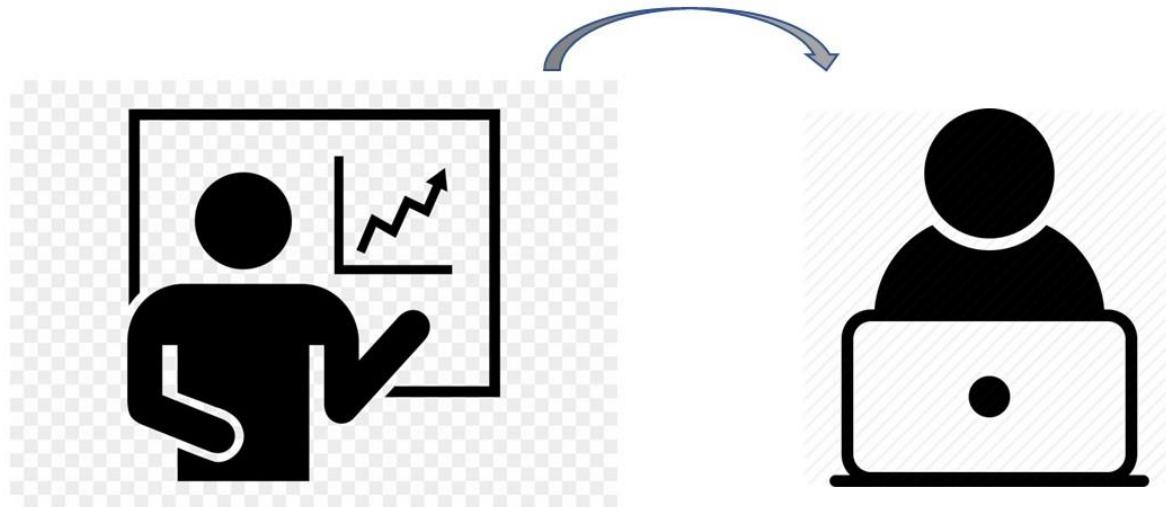
# Quais perguntas vamos responder na aula de hoje?



- Por que devemos aprender **Programação**?
- Quais são os conceitos fundamentais de **Ciência da Computação** para aprender a programar?
- Como iniciar a programação em **Python**?
- Que **Ferramentas** podemos utilizar para programar em Python?
- O que é uma “**biblioteca**” em programação?
- Por que utilizar bibliotecas em Python?
- Quais bilbiotecas seriam recomendadas para se trabalhar com **Inteligência Artificial**?



# DINÂMICA AULA



**Exposição dos  
Fundamentos Teóricos de  
Programação e Ciência da  
Computação**

**Prática através do Jupyter  
Notebook disponibilizado**



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

**Python é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte.**

**Foi lançada por Guido van Rossum em 1991.**



<https://youtu.be/ghwaliE3Nd8>





# POR QUE APRENDER A PROGRAMAR?



<https://www.youtube.com/watch?v=CLf7fxqltgg>



# POR QUE CIÊNCIA DA COMPUTAÇÃO?

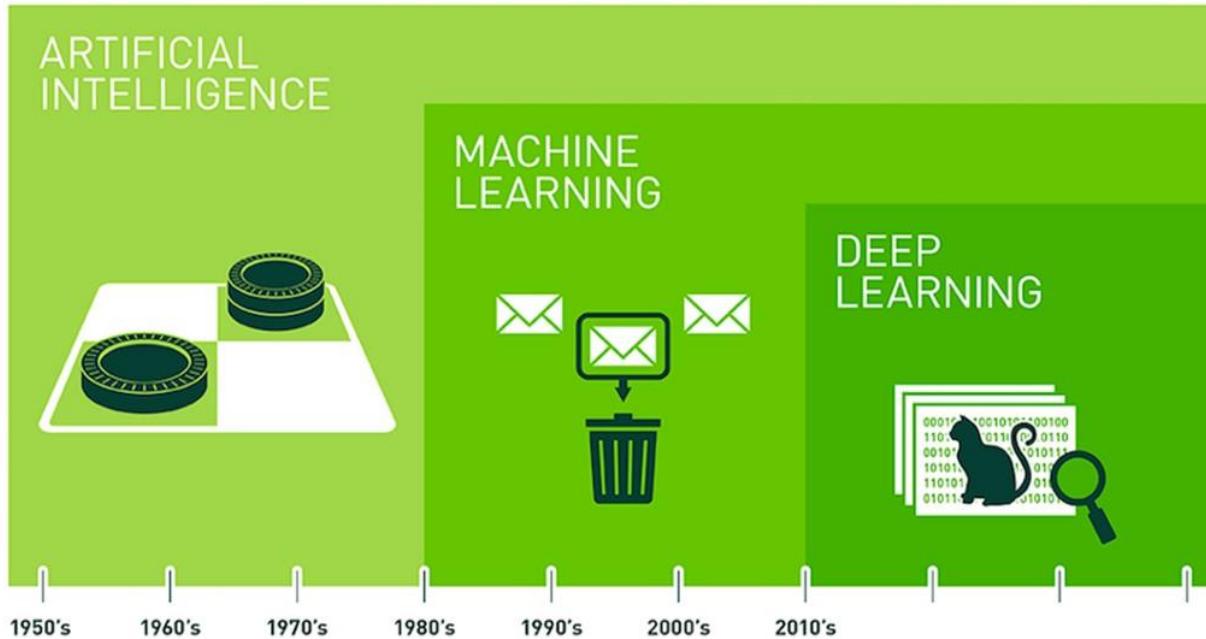


<https://youtu.be/IY7EsTnUSxY>





# MACHINE LEARNING x AI x DEEP LEARNING



<https://medium.com/data-science-brigade/a-diferen%C3%A7a-entre-intelig%C3%A1ncia-artificial-machine-learning-e-deep-learning-930b5cc2aa42>



# CARRO AUTÔNOMO – PYTHON + GTA V



<https://www.youtube.com/watch?v=ks4MPfMq8aQ>



# CONCEITOS CIÊNCIA DA COMPUTAÇÃO EM PYTHON COM JUPYTER NOTEBOOKS



# ALGORITMOS

## Algoritmos - Exemplo

### Algoritmo: Sacar dinheiro

#### INÍCIO

1. Ir até o caixa eletrônico.
2. Colocar o cartão.
3. Digitar a senha.
4. Solicitar o saldo.
5. Se o saldo for maior ou igual à quantia desejada, sacar a quantia desejada; caso contrário sacar o valor do saldo.
6. Retirar dinheiro e cartão.

#### FIM.

5

<https://pt.slideshare.net/josecintra/algoritmos-e-logicade-programao-57345155>

### Algoritmo "Trabalhar pela manhã"

1. Acordar
2. Tomar banho
3. Vestir-se
4. Tomar café
5. Tirar o carro da garagem
6. Ir para o trabalho

<https://c2ti.com.br/blog/entenda-o-que-e-algoritmo-e-como-ele-determinar-o-que-voce-ve-na-internet-tecnologia>



# TIPOS DE DADOS EM PYTHON



Os **Tipos** em Python  
definem  
**Comportamentos** e  
**Funcionalidades**  
**Específicos** para cada  
tipo de dados

## Tipos – Primitivos\*

- Inteiros: 1, 10, 235, 10.000, etc
- Ponto flutuante: 10,51 – 23.000,5
- Booleanos: True, False
- Strings
- Números Complexos
- ...

## Tipos - Estruturas de Dados\*

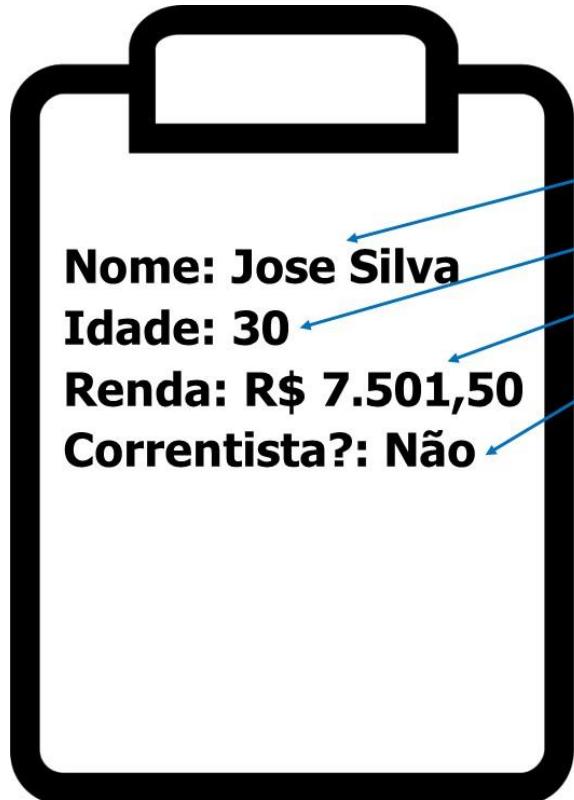
- Listas
- Dicionários
- Tuplas
- Conjuntos
- ...

▣ *Tipos que serão abordados neste material*

\* *Lista não exaustiva dos tipos disponíveis em Python*

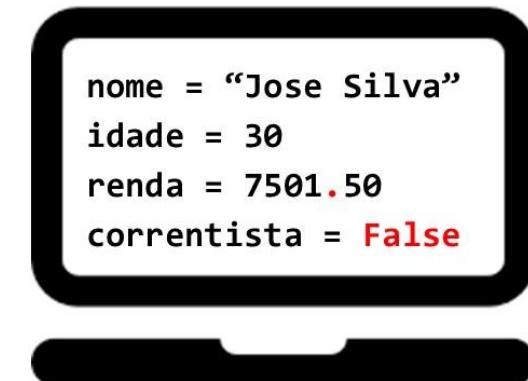


# VARIÁVEIS - TIPOS PRIMITIVOS

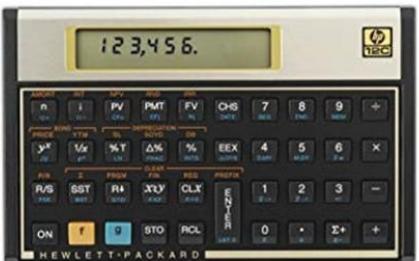


## Tipos Primitivos\*

- **Strings:** "Avenida Angélica, 2318"
- **Inteiros:** 1, 10, 235, 10.000
- **Ponto flutuante:** 10,51 , 23.000,50
- **Booleanos:** True, False



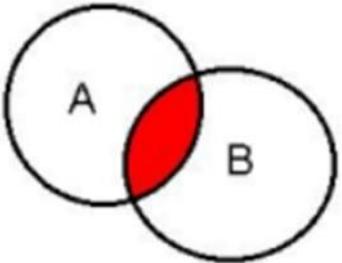
# OPERADORES MATEMÁTICOS



- + soma
- subtração
- / divisão
- \* multiplicação
- < menor
- > maior
- <= menor ou igual
- >= maior ou igual
- == igual
- % resto divisão
- \*\* potência

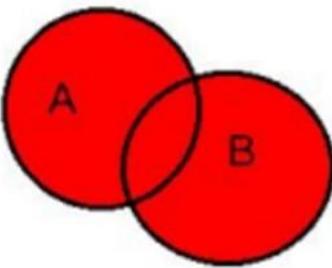


# OPERADORES BOOLEANOS



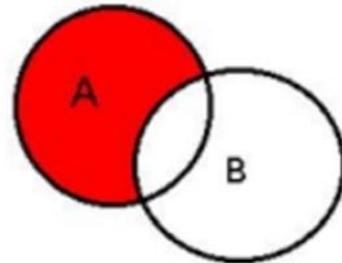
A **AND** B

**“o Cliente deve apresentar comprovantes de renda (A) E (AND) residência (B) para aprovação da proposta”**



A **OR** B

**“Necessário apresentar RG (A) OU (OR) CNH (B) para entrar no prédio”**

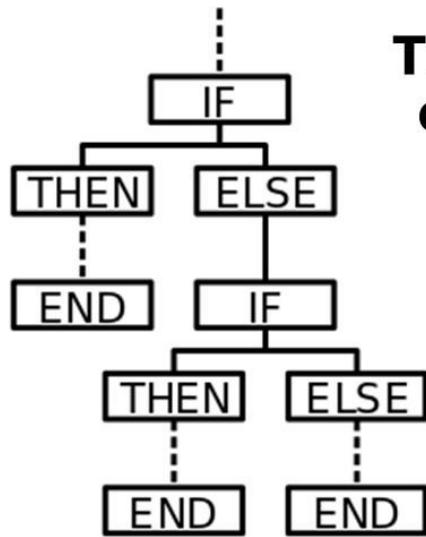


A **NOT** B

**“Operação permitida para correntistas do banco (A) que NÃO (NOT) possuem um imóvel próprio (B)”**



# CONDICIONAIS



**Tradicionalmente\*** - utilizam-se os comandos condicionais - **if, then, else** para escrever as regras de negócio e comportamentos desejados no programa.

```
if idade < 18:  
    print("não é permitida a entrada")  
else:  
    print("acesso liberado")
```

\* Em programas de Machine Learning / IA podem ser utilizadas outras técnicas para implementação das regras. - Tabulação (TAB), equivalente a 4 espaços



# JUPYTER NOTEBOOKS



The screenshot shows a Jupyter Notebook interface running in a web browser on localhost. The title bar indicates it's a 'DataFrames' notebook, last checked for autosave a day ago. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar includes icons for New, Open, Save, Cell, CellToolbar, and other notebook-specific functions. The Python kernel is selected.

In [6]:

```
import pandas as pd
import numpy as np
```

In [7]:

```
df = pd.DataFrame(data=np.array([[1,2,3], [4,5,6]]), dtype=int, columns=['A', 'B', 'C'])
df
```

Out[7]:

	A	B	C
0	1	2	3
1	4	5	6





# IDEs E EDITORES TEXTO



Sublime Text



IDE, do inglês *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo. É normalmente onde o programador realiza a codificação.



Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share Sign in

+ Code + Text Copy to Drive

Connect Editing

## What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

### Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

<https://colab.research.google.com/>

colab

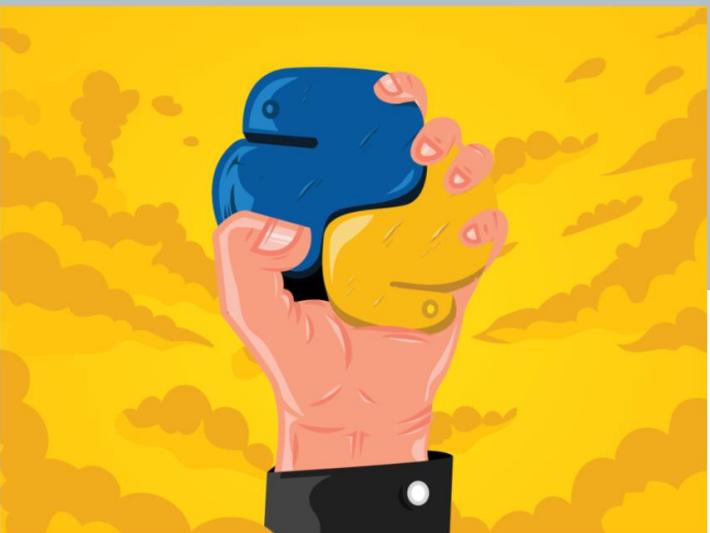


CÓMO INICIAR COM DATA SCIENCE ?

## Como instalar Python com o Anaconda Passo a Passo.



Escrito por Rodrigo Santana  
em abril 22, 2019



<https://minerandodados.com.br/installar-python-anaconda/>



alura

O que você quer aprender?



FORMAÇÕES

PARA EMPRESAS

DEV EM <T>

ENTRAR

MATRICULE-SE

MOBILE

PROGRAMAÇÃO

FRONT-END

DEVOPS

DESIGN & UX

MARKETING DIGITAL

DATA SCIENCE

INOVAÇÃO & GESTÃO

ARTIGOS > DATA SCIENCE

## Google Colab: o que é e como usar?



Thiago G Santos  
19/04/2020

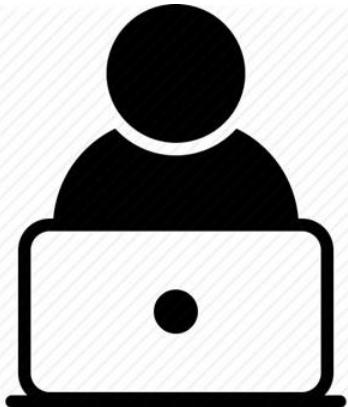
Nesse artigo vamos entender o que é o **Google Colab**, como ele funciona, como abrir arquivos externos e como escrever códigos e textos em markdown. O **Google Colaboratory**, carinhosamente chamado de **Colab**, é um serviço de **nuvem** gratuito hospedado pelo **Google** para incentivar a pesquisa de **Aprendizado de Máquina** e **Inteligência Artificial**.

Agora que já temos uma introdução do que é o colab, vamos ver como usá-lo. Similar ao famoso **jupyter notebook**, o Colab é uma lista de células que podem conter textos explicativos ou códigos executáveis e suas respectivas saídas.

<https://www.alura.com.br/artigos/google-colab-o-que-e-e-como-usar>



# EXERCÍCIOS PRÁTICOS - 1



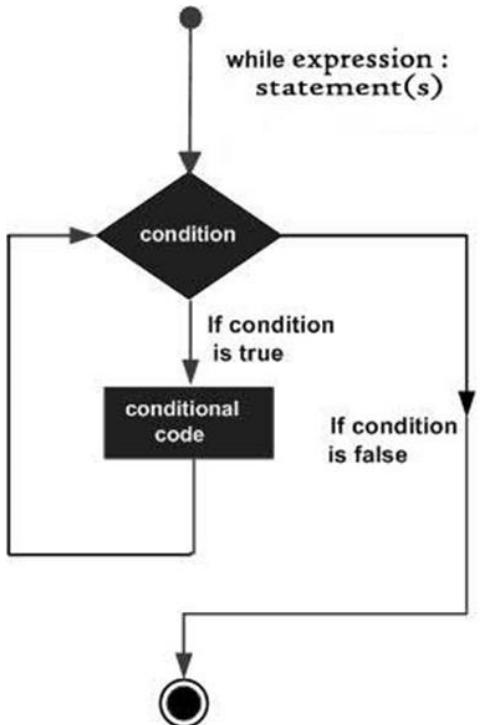
**Agora nós vamos:**

1. Escrever nosso primeiro programa – **HELLO WORLD!**
2. Aprender a **Comentar o código do programa**
3. Aprender a utilizar **Variáveis**
4. Aprender a capturar ***Input*** do Teclado
5. Utilizar **Condicionais e Operadores para determinar se a pessoa nasceu em um ano bissexto**

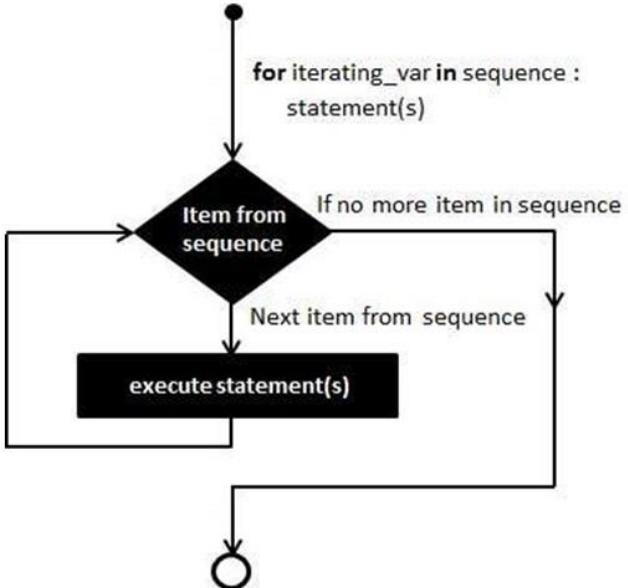
**Algumas funções que usaremos nestes exercícios**

- Função `print()` – apresenta na tela o texto
- Função `input()` – apresenta uma janela para a captura de informações
- Função `int()` – converte tipos distintos para INTEIRO
- Função `str()` – converte tipos distintos para STRING

# COMANDOS DE REPETIÇÕES (LOOPS)



**WHILE** – a repetição se mantém enquanto a condição é verdadeira



**FOR** – a repetição percorre todos os itens de uma lista ou intervalo de valores



# TIPOS ENUMERADOS - LISTAS



90s

ÍNDICE

0	1	2	3	4	5
"Metallica"	"Iron Maiden"	"AC/DC"	"Guns and Roses"	"Megadeth"	"Van Halen"

**Exemplos de como selecionar elementos em Listas**

```
> rock90[0]  
output:"Metallica"  
> rock90[1:3]  
output:[“Iron Maiden”,  
“AC/DC”]  
> rock90 [-1]  
output:“Van Halen”
```

**Alguns métodos para realizar operações em listas**

*append()* – Adiciona um elemento ao final da lista  
*insert()* – Insere um elemento à lista, em um índice  
*remove()* – Remove um elemento da lista  
*count()* – Retorna a quantidade de elemetos na lista  
*sort()* – Ordena elementos na lista  
*reverse()* - Reverte a ordem dos elementos na lista  
...

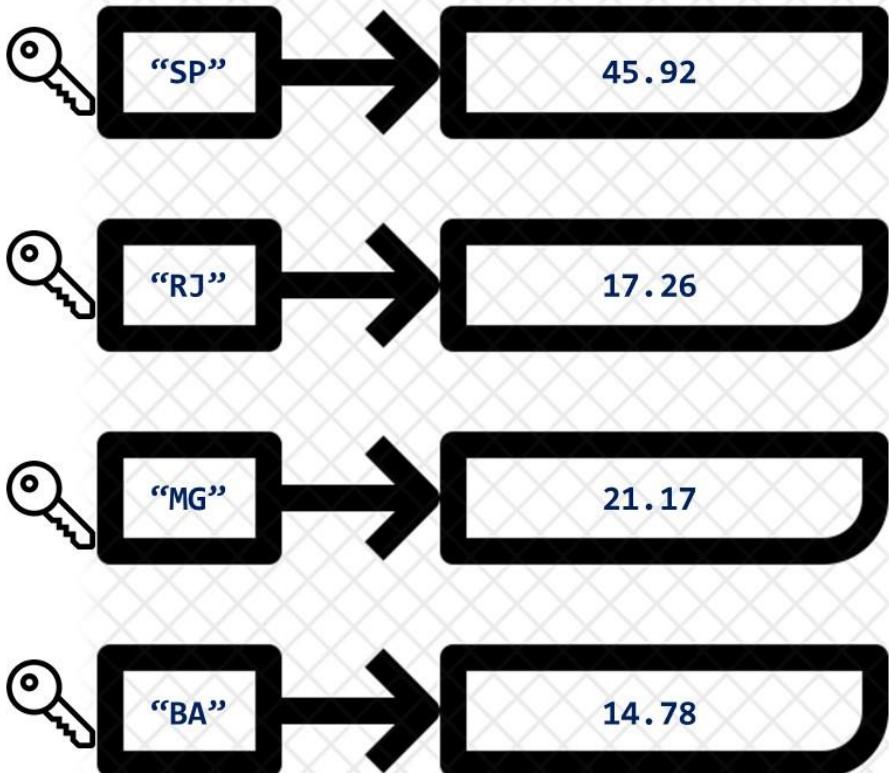
**Listas podem ser compostas de tipos primitivos ou estruturados, inclusive de outras listas!**

# TIPOS ENUMERADOS - DICIONÁRIOS



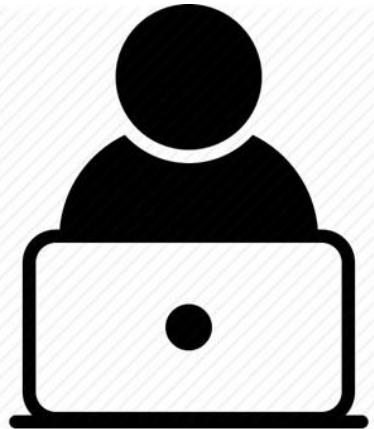
**Tipo estruturado  
composto de  
Chave:Valor**

```
dict = {"SP":45.92,  
        "RJ":17.26,  
        "MG":21.17,  
        "BA":14.87}
```





# EXERCÍCIOS PRÁTICOS - 2



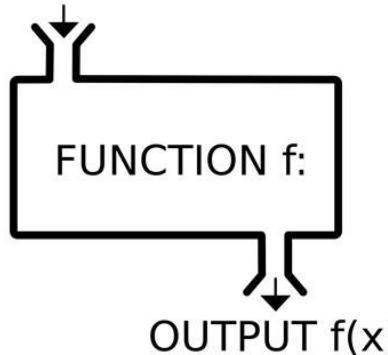
## Agora nós vamos:

- 1. Calcular a somatória dos número de 1 a N de um valor informado**
- 2. Criar uma Lista**
- 3. Recuperar elementos e trechos de uma Lista**
- 4. Manipular elementos de uma Lista**
- 5. Criar um Dicionário**
- 6. Recuperar e manipular elementos de um Dicionário**



# FUNÇÕES - DEFINIÇÃO

INPUT x



- Podem ser divididas em **Built-in** (que já existem no Python) e **User-defined** (definidas pelo usuário)
- É um trecho de código que realiza uma tarefa específica
- Ajuda a tornar o programa mais modular e organizado
- Torna o código mais **enxuto** - evitando repetições desnecessárias de linhas
- Pode retornar um valor através do comando **return**

## Exemplo – utilizando a função max()

	0	1	2
notasProva =	3,0	2,5	4,5

```
> max(notasProva)  
> 4,5
```



# FUNÇÕES

## Built-in (que já existem no Python)

type() - retorno o tipo do objeto

max() – retorno o valor máximo

min() – retorna o valor mínimo

list() – cria uma lista

dict() – cria um dicionário

str() – converte um valor para string

int() – converte um valor para inteiro

help() – mostra ajuda para funções

...

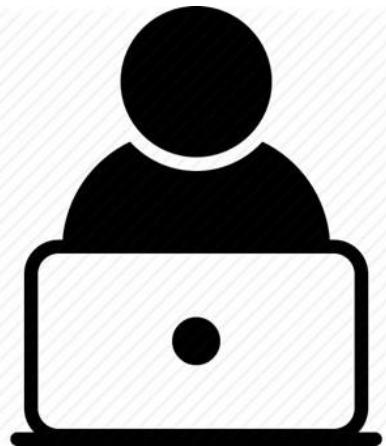
## Escrevendo suas próprias funções

```
def olaMundoNome(nome):  
    print("Olá Mundo, " + nome)  
    contadorUso += 1
```

```
>olaMundoNome("Wagner")  
Olá Mundo, Wagner
```



# EXERCÍCIOS PRÁTICOS - 3



**Agora nós vamos:**

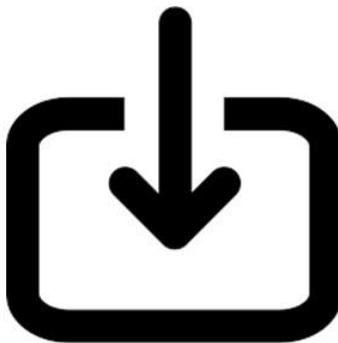
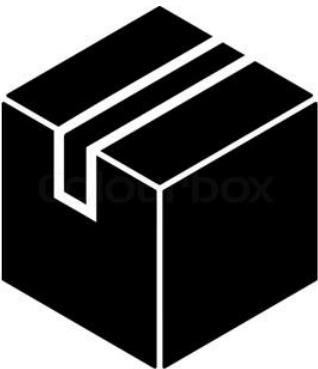
- Utilizar algumas das funções pré-existentes mais conhecidas
- Escrever uma função que verifica se um texto (string) é um palíndromo



# BIBLIOTECAS DE DATA SCIENCE / IA EM PYTHON



# BIBLIOTECAS / PACOTES



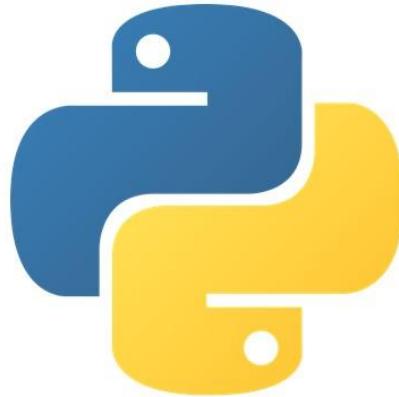
- Bibliotecas, também conhecidas como pacotes (tradução de *packages*), são trechos de código que realizam operações específicas de uma maneira otimizada.
- As bibliotecas são inseridas automaticamente no Código através de um comando simples:

Ex: *import numpy as np*

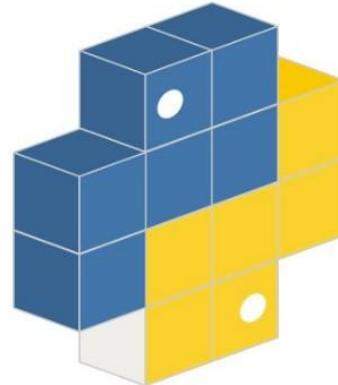


# BIBLIOTECAS / PACOTES

- Python possui uma riquíssima coleção de bibliotecas, que cresce a cada dia, para resolver inúmeros problemas.
- A maior parte dos projetos e programas feitos em Python se utilizam de uma série de bibliotecas.
- As distribuições mais populares de Python, como o Anaconda, já trazem as bibliotecas mais utilizadas.



Logo Python



Logo do “Python Package Index”  
<https://pypi.org/>



# PYPI – PYTHON PACKAGE INDEX

The screenshot shows the PyPI homepage with a blue header bar. The header includes the PyPI logo, navigation links for Help, Donate, Log in, and Register, and a search bar. Below the header, a large text area reads "Find, install and publish Python packages with the Python Package Index". It features a search bar with the placeholder "Search projects" and a link "Or [browse projects](#)". A summary bar at the bottom displays statistics: 192,292 projects, 1,427,926 releases, 2,089,727 files, and 357,930 users. The main content area contains a paragraph about PyPI being a repository for Python software, followed by links for learning about installing packages and packaging code for PyPI.

Find, install and publish Python packages with the Python Package Index

Search projects

Or [browse projects](#)

192,292 projects 1,427,926 releases 2,089,727 files 357,930 users

**python™ Package Index**

The Python Package Index (PyPI) is a repository of software for the Python programming language.

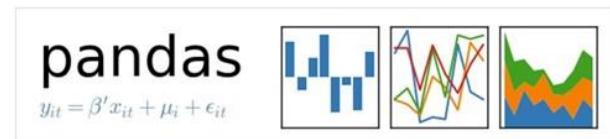
PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages](#).

Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI](#).

<https://pypi.org>



Quais são algumas das principais bibliotecas utilizadas em Inteligência Artificial?



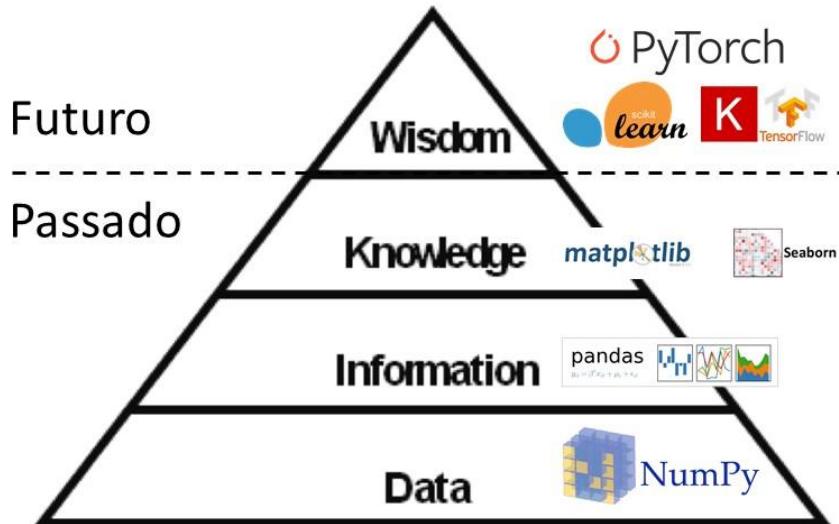


# PIRÂMIDE MASLOW





# PIRÂMIDE DIKW



**Sabedoria:** diminuir a velocidade e parar o carro

**Conhecimento:** o próximo farol na minha rota ficou vermelho

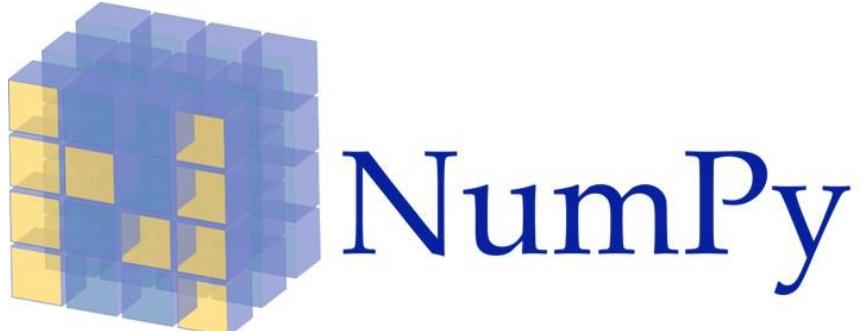
**Informação:** Farol Vermelho – Esquina Av. Paulista x Av. Angélica, sentido centro

**Dados:** C451 - Vermelho - 41°24'12.2"N 2°10'26.5"L

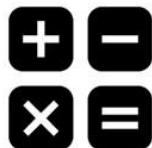
Com Python e as bibliotecas a serem apresentadas  
poderemos percorrer os quatro estágios da DIKW



# NumPy – NUMERICAL PYTHON



<https://numpy.org>



*import numpy as np*

- Poderosa biblioteca de computação científica que realiza de maneira muito eficiente **operações matemáticas** em grande escala. Destaque para: **Álgebra Linear**, Transformações de Fourier e geração de números aleatórios.
- Implementa o objeto **NumPy Array** – que permite fazer em N-dimensões – operações matemáticas em cada elemento do vetor. No entanto todos elementos do vetor devem ser do **mesmo tipo** e são **referenciados** pelos seus **índices**.



# DIFERENÇAS NumPy ARRAY x LISTAS

notasProva1 =

0	1	2
3,0	2,5	4,5

notasProva2 =

0	1	2
3,5	2,0	5,0

Lista

notasProva1 + notasProva2 →

0	1	2	3	4	5
3,5	2,0	4,5	3,5	2,0	5,0

NumPyArray

notasProva1 + notasProva2 →

0	1	2
6,5	4,5	9,5

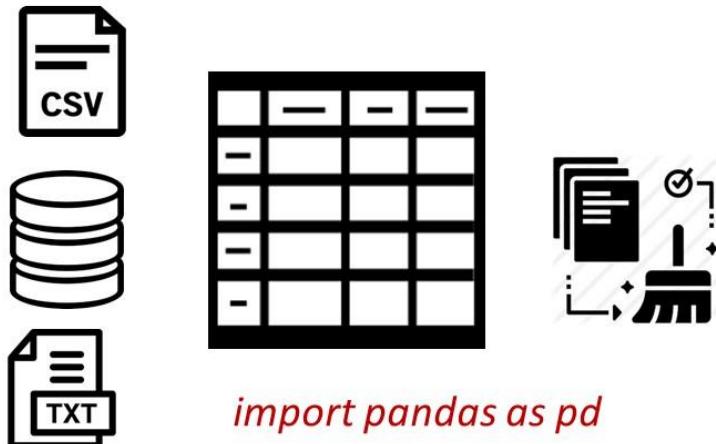
Obs: Arrays NumPy devem ter todos elementos do mesmo tipo e estes podem ser referenciados somente pelo índice.



# PANDAS – PANel DATA



<https://pandas.pydata.org>



*import pandas as pd*

- Biblioteca de **leitura, análise e manipulação de dados** – construída a partir do NumPy.
- Implementa os objetos **Series e Dataframe** – que permitem a realização de operações com **dados heterogêneos** e referências por nomes de colunas.
- Permite a carga de dados através de diferentes fontes e formatos: **CSV**, JSON, bases de dados, etc.
- Realiza diversas operações que facilitam o tratamento e preparação dos dados: retirada de nulos, *join* de bases, achar valores únicos, etc.



# PANDAS – Series e DataFrames

EIXO 0 LINHAS

EIXO 1 COLUNAS

	0	1	2	3
0	Nome	Idade	Série	Responsável
1	Ivone	11	5	Antônio
2	Maria	10	S	Davi
3	João		5	Dariana
4	Paulo	11	5	Zélia

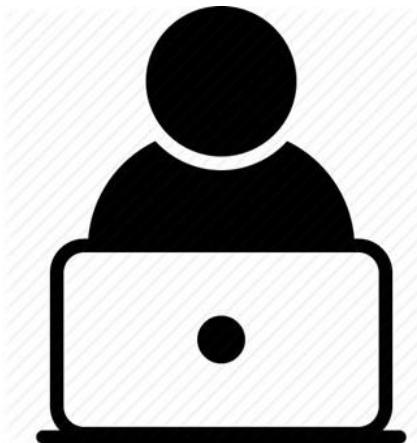
DATAFRAME

series series series series

Elementos podem ter tipos distintos e as Series (colunas) podem ser referenciadas pelo nome ou pelo índice!



# EXERCÍCIOS PRÁTICOS - 4



Agora nós vamos:

1. Fazer operação elemento a elemento num vetor NumPy Array)
2. Ler um arquivo CSV e criar um Dataframe
3. Recuperar informações de um Dataframe



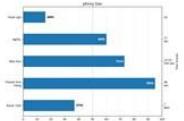
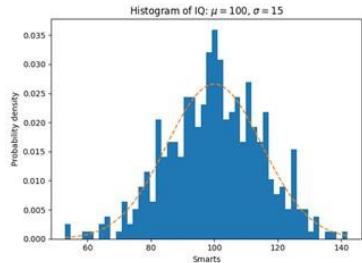
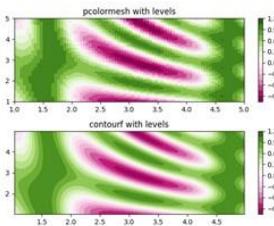
# matplotlib



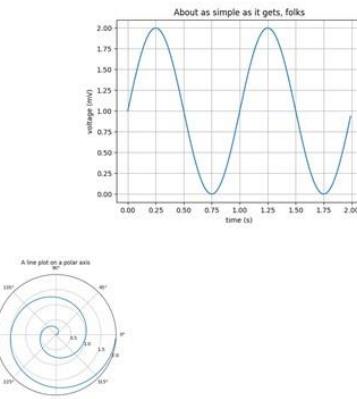
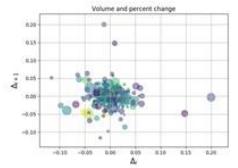
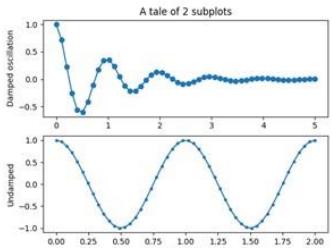
Version 3.1.1

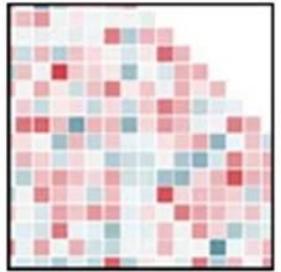
<https://matplotlib.org>

*import matplotlib as plt*



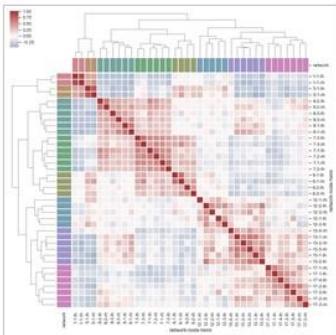
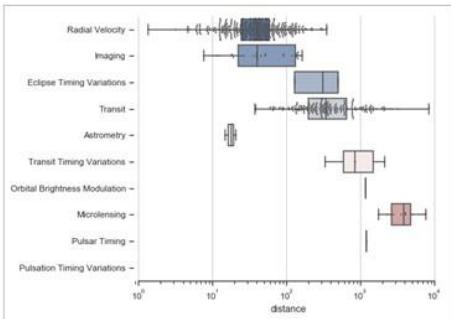
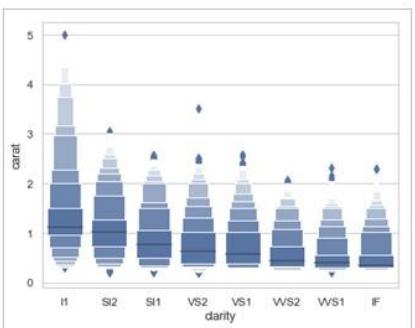
- Biblioteca para geração de diversos tipos de **gráficos**, em geral **matemáticos**, a partir de dados.
- Capaz de gerar a maioria dos gráficos mais utilizados: histogramas, dispersão, barras, linhas, etc





# Seaborn

<https://seaborn.pydata.org>



*import seaborn as sns*



# SCIKIT-LEARN



<https://scikit-learn.org>

- Biblioteca de **aprendizado de máquina** feita a partir do ScyPy e NumPy.
- Permite implementar com facilidade diversos algoritmos de:
  - **Pré-processamento**
  - **Classificação**
  - **Regressão**
  - **Agrupamento**
  - **Redução de dimensionalidade**
  - **Comparação entre modelos**

*from sklearn import \_\_\_\_\_*

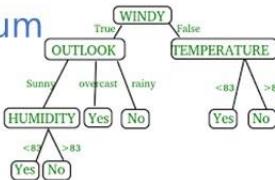


# SCIKIT-LEARN

## Classificação

Ex: Identificar a que categoria um objeto pertence

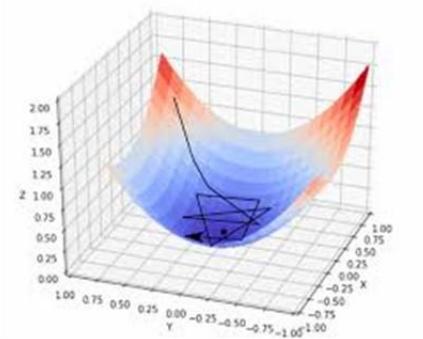
- *Support Vector Machines*
- *Decision Trees*
- ...



## Regressão

Ex: Prever um valor contínuo – preços ações

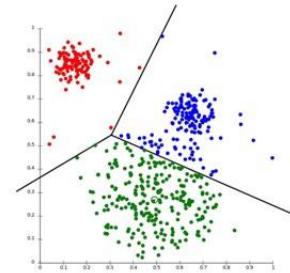
- *Logistic Regression*
- *Stochastic Gradient Descent*
- ...



## Agrupamento

Ex: Identificar clientes com mesmo perfil

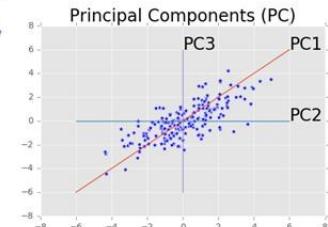
- K-means
- KNN
- ...



## Red. Dimensionalidade

Ex: Simplificar o modelo, retirando variáveis de menor impacto

- *Principal Component Analysis*
- *Single Value Decomposition*
- ...

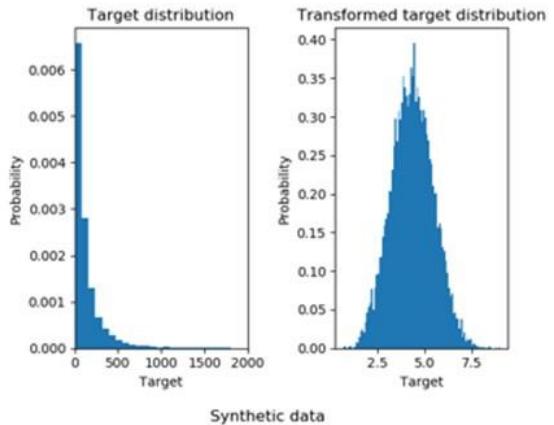
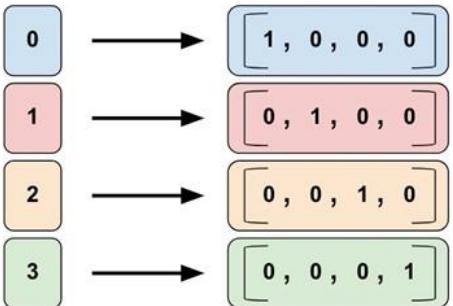




# SCIKIT-LEARN

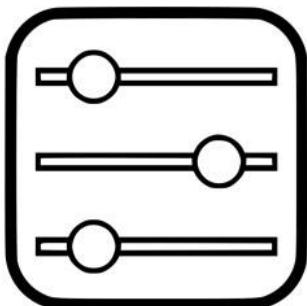
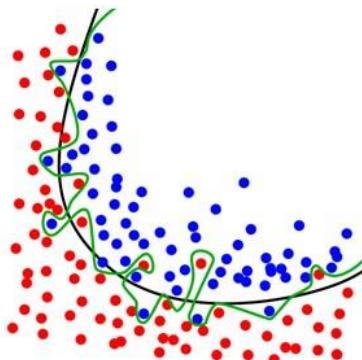
## Pré-processamento

- Normalização
- Eliminação *outliers*
- *One hot encoding*
- ...



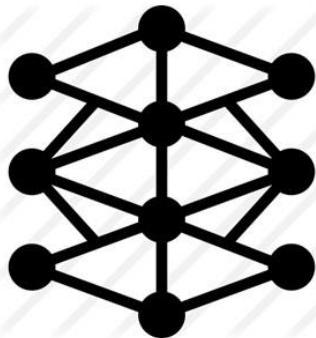
## Comparação entre Modelos

- Acurácia
- Hiperparâmetros
- ...



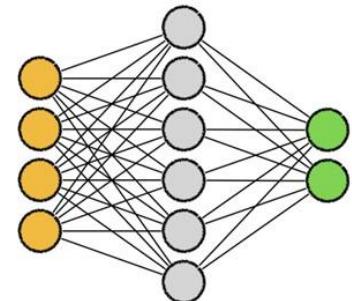


# FRAMEWORKS – DEEP LEARNING



PyTorch

K TensorFlow

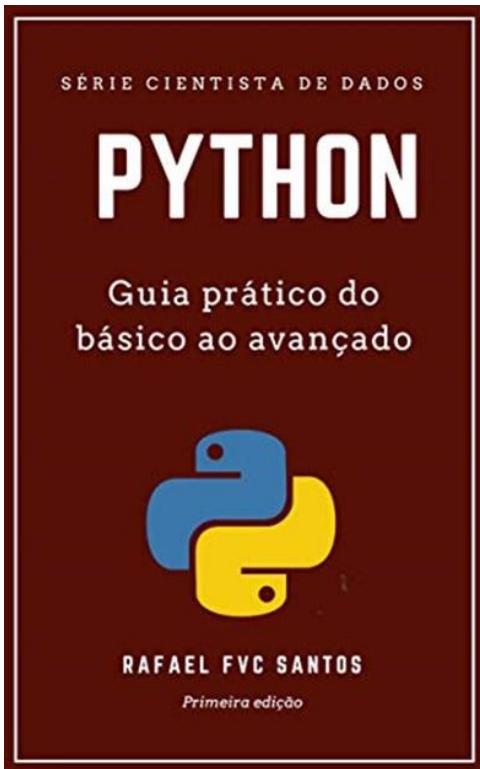


# REFERÊNCIAS

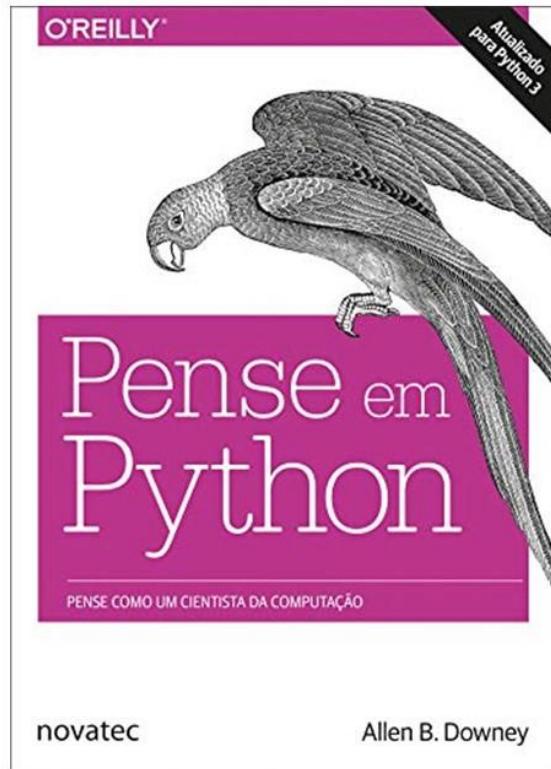




# REFERÊNCIAS - LIVROS



<https://www.amazon.com.br/Python-pr%C3%A1tico-b%C3%A1sico-avan%C3%A7ado-Cientista-ebook/dp/B07KML8M9L/>



<https://www.amazon.com.br/Pense-Python-como-cientista-computa%C3%A7%C3%A3o-ebook/dp/B07QL2LKBG/>

# REF. - CRASH COURSE IN COMPUTER SCIENCE



[https://www.youtube.com/playlist?list=PLME-KWdxI8dcaHSzzRsNuOLXtM2Ep\\_C7a](https://www.youtube.com/playlist?list=PLME-KWdxI8dcaHSzzRsNuOLXtM2Ep_C7a)



# YOUTUBE – APRENDA PROG. - PYTHON



[https://www.youtube.com/playlist?list=PLHz\\_AreHm4dIKP6QQCekulPky1Ciwmdl6](https://www.youtube.com/playlist?list=PLHz_AreHm4dIKP6QQCekulPky1Ciwmdl6)



# AS PERGUNTAS FORAM RESPONDIDAS?



- Por que devemos aprender programação?
- Quais são os conceitos fundamentais de Ciência da Computação para aprender a programar?
- Como iniciar a programação em Python?
- Que ferramentas podemos utilizar para programar em Python?
- O que é uma “biblioteca” em programação?
- Por que utilizar bibliotecas em Python?
- Quais bibliotecas seriam recomendadas para se trabalhar com Inteligência Artificial?

# **Escola Livre de IA**

Obrigado!!

