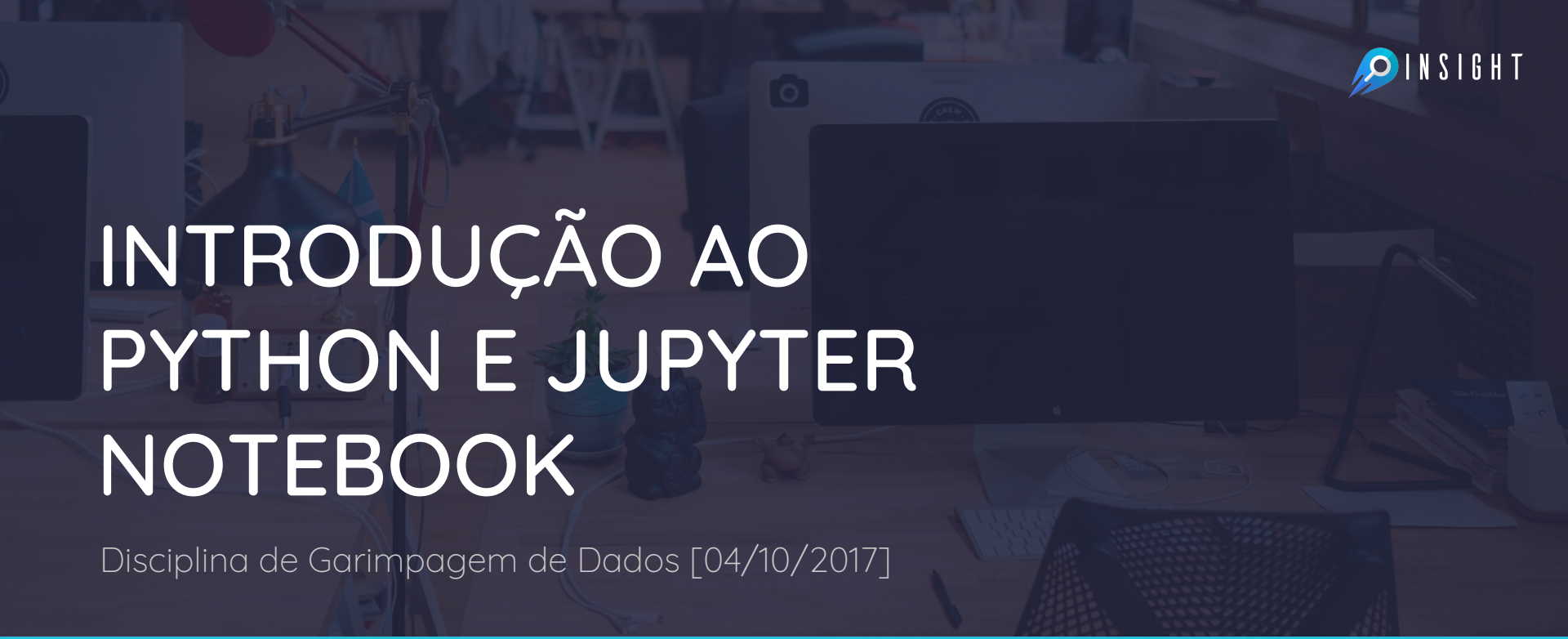

INSIGHT



INSIGHT

INTRODUÇÃO AO PYTHON E JUPYTER NOTEBOOK

Disciplina de Garimpagem de Dados [04/10/2017]

DEPARTAMENTO DE COMPUTAÇÃO

UNIVERSIDADE FEDERAL DO CEARÁ

AGENDA

1. Ambiente
2. Por quê Python?
3. Miniconda
4. Jupyter Notebook
5. Manipulação de Arquivo

AMBIENTE

O que usaremos?

Python 3

Miniconda

Scientific Python Stack

Jupyter Notebook

Apache Spark

GPU

POR QUÊ PYTHON?

- 5th Largest **StackOverflow** Community
- 4th Most-Used Language at **GitHub**
- Google, Dropbox, Spotify, Netflix...
- **DevOps** tool for configuration management and deployment automation
- Large Internet Applications (YouTube, Instagram, Dropbox, ...)
- **HUGE** community
- Several solutions for **Machine Learning** and **Big Data** tools

[1] Stack Overflow. **Developer Survey Results**, 2016. [http://stackoverflow.com/research/developer-survey-2016]

1. MINICONDA

Gerenciador de pacotes Python

MINICONDA

- **Miniconda is a small, bootstrap version of Anaconda**
- **Gerenciador de pacotes**
- **Environments**
- **Instalação**
  - Entrar no site e baixar a versão compatível
  - $ chmd +x [arquivo].sh
  - $ ./[arquivo].sh
- **Criando primeiro ambiente**
  - $ conda create -n [nome] python=3.6

Miniconda. **Installation Guide**, 2017. [https://conda.io/miniconda.html]

2. JUPYTER NOTEBOOK

Open source web application

JUPYTER NOTEBOOK

- Open source project was born out of the **IPython Project**
- Ciência de Dados de forma interativa
- Diversas linguagens de programação e frameworks
- 100% open source software
- E é WEB!!
- Arquivos interpretados por diversas plataformas
- **Instalação**
  - $ source activate [nome do ambiente]
  - $ conda install jupyter
  - $ jupyter notebook

Jupyter Notebook. **About**, 2017. [http://jupyter.org/about.html]

IPython. **Interactive computing**, 2017. [https://ipython.org/]

10

VAMOS PRATICAR PYTHON EM UM JUPYTER NOTEBOOK!!

3. MANIPULAÇÃO DE ARQUIVOS

ABRINDO UM ARQUIVO

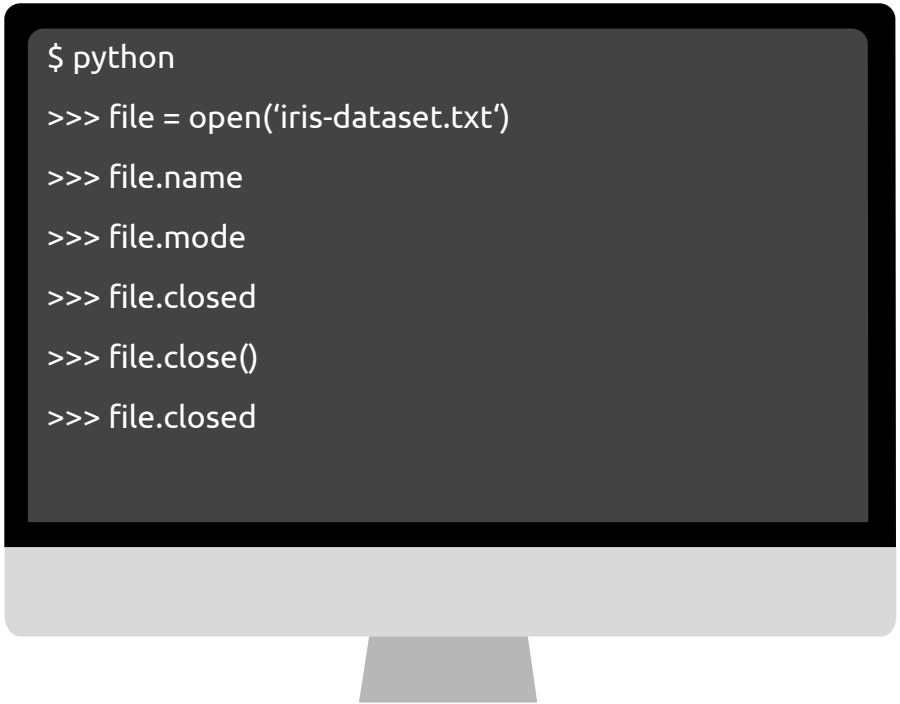
Utilizaremos a função **open()**

file = **open**(*file_name* [, *access_mode*][, *buffering*])

file_name: um valor string que contém o nome do arquivo que você deseja acessar.

access_mode: determina o modo em que o arquivo será aberto, isto é, **read**, **write**, **append**, etc. É um parâmetro opcional e o valor padrão é ***r***.

buffering: argumento opcional que especifica o tamanho do buffer desejado do arquivo: **0** significa sem buffer, **1** significa *line buffered*, e qualquer outro valor positivo significa usar um buffer de (aproximadamente) o valor informado (em bytes). Um buffer negativo significa usar o padrão do sistema (comportamento padrão).



LENDO TEXTO DE UM ARQUIVO

Assumindo que temos um **File Object**

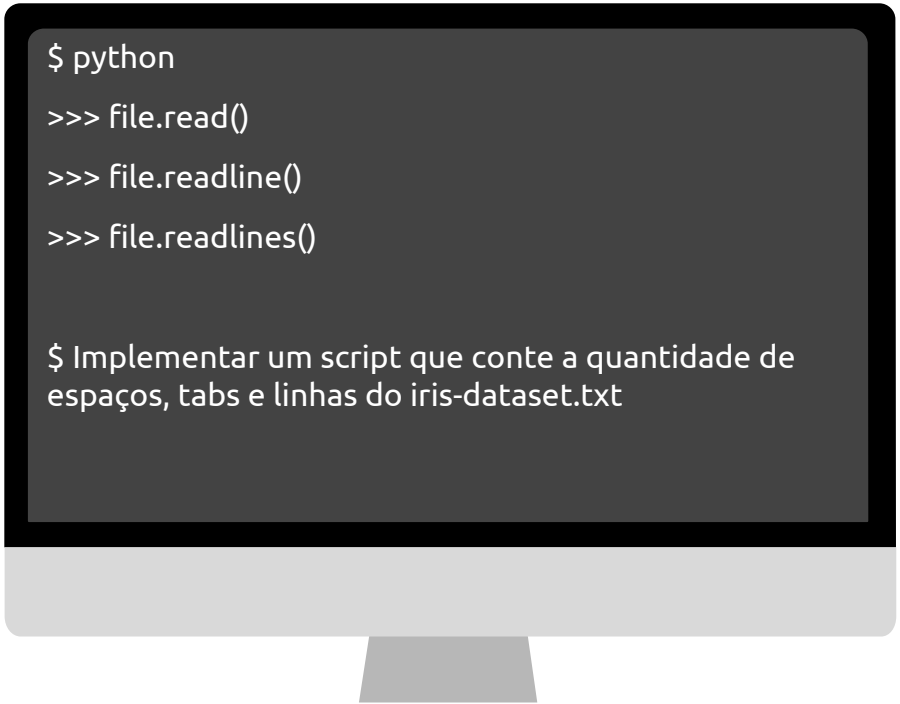
file.read() - ler todo o arquivo de uma vez.

file.readline() - ler uma linha por vez.

file.readlines() - ler todas as linhas e retorna uma lista.

file.tell() - informar qual a posição atual do cursor no arquivo.

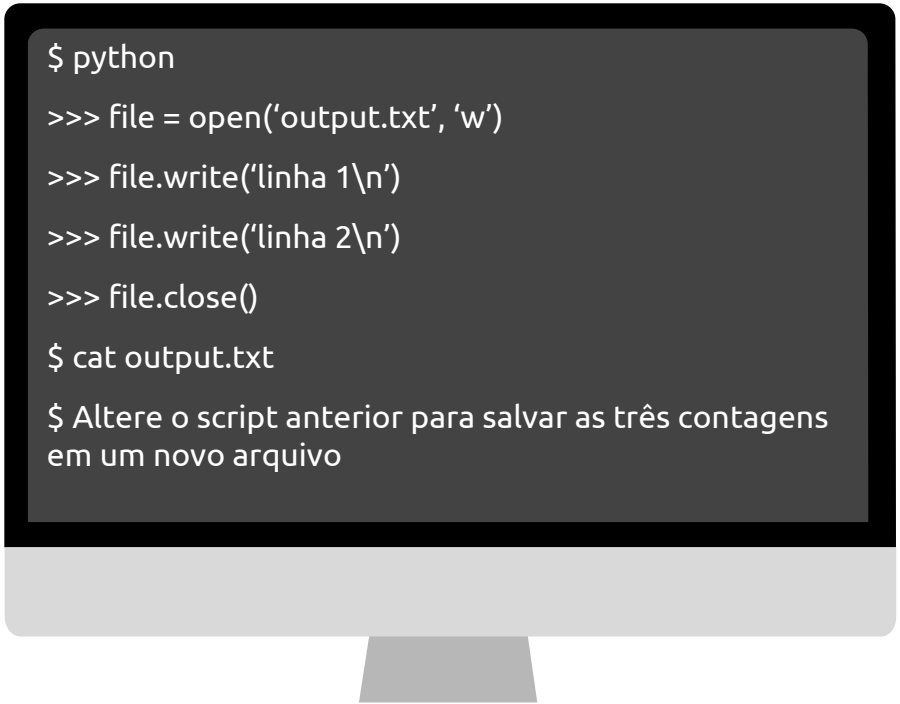
file.seek(offset) - alterar a posição atual do curso do arquivo. Se for definido como **0**, significa usar o inicio do arquivo como a posição de referência.



ESCREVENDO TEXTO NO ARQUIVO

Utilizaremos a função **write()**

file.write(*str*) - escreve uma string no arquivo. Não há nenhum valor de retorno. Devido ao buffer, a nova string pode não aparecer no arquivo até que o ***flush()*** ou ***close()*** sejam chamados.



4.
MÓDULOS E PACOTES

MÓDULOS E PACOTES

São conceitos básicos para se trabalhar com Python!

- **Módulos**
  - Um módulo é um arquivo Python contendo definições e sentenças.
  - Módulos são arquivos de código Python cuja interface pode ser importada por outros módulos
  - Todos os arquivos com código Python são módulos, mesmo que não sejam importados
- **Pacotes**
  - Módulos são estruturados em arquivos, enquanto que, pacotes são estruturados em pastas.
  - Todos os pacotes devem conter um arquivo ***__init__.py***
  - Executar este exemplo.

TIPS AND TRICKS

UTILIDADES

- **A collection of design patterns in Python**
  https://github.com/faif/python-patterns
- **A curated list of awesome Python frameworks, libraries, software and resources**
  https://github.com/vinta/awesome-python
- **PyPI - The Python Package Index is a repository of software for the Python programming language**
  https://pypi.python.org/pypi
- **30 ESSENTIAL PYTHON TIPS AND TRICKS FOR PROGRAMMERS**
  http://www.techbeamers.com/essential-python-tips-tricks-programmers/
- **Hidden features of Python**
  http://stackoverflow.com/questions/101268/hidden-features-of-python