

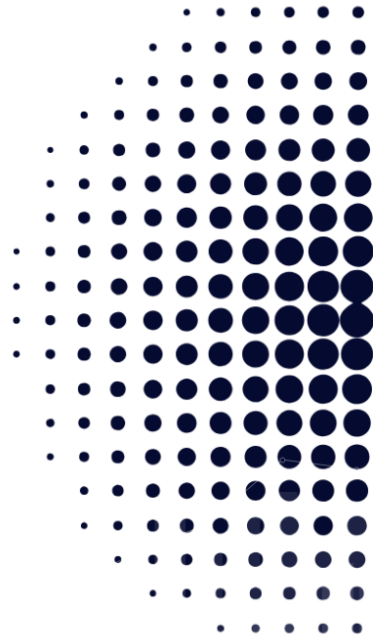
Python Essencial

Treinamento in-company para

genial
investimentos

powered by

Trading com Dados



Comandos Básicos do Google Colab

- 1) Ctrl + enter = Executa a célula e permanece nela
- 2) Shift + enter = Executa a célula e vai para célula seguinte
- 3) # = Cria títulos e subtítulos ou níveis (estrutura de tópicos)
- 4) Esc + A = Cria uma célula de códigos logo acima
- 5) Ctrl + M (juntos) + M = Transforma célula de código em célula de texto
- 6) Ctrl + M (juntos) + D = Elimina (remove) a célula
- 7) Link com teclas de atalho:
<https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/01.02-Shell-Keyboard-Shortcuts.ipynb>
- 8) Link com orientações gerais sobre como usar o Google Colab e seus recursos:
https://colab.research.google.com/?utm_source=scs-index

Exibindo Mensagens

- 1) Print = Função (comando que recebe um input e solta uma saída) usada para exibir mensagens no python
- 2) Link com sintaxe (regras que definem como o programa será escrito) do Python:

https://pt.wikipedia.org/wiki/Sintaxe_e_sem%C3%A2ntica_de_Python

Tipos de Variáveis

- 1) Variáveis = Aponta/indica para algo está na memória
- 2) Objetos = É o objeto em si que está na memória
- 3) Função type = Demonstra qual o tipo de variável
- 4) Tipos de Variáveis:
 - a) Integer (inteiro) = Números inteiros
 - b) Float = Números fracionados
 - I. O separador decimal no python é o ponto (.) e não a vírgula (,)
 - c) String (str) = Textos, nomes, caracteres alfanuméricos
 - d) Boolean (bool) = Condições lógicas (V ou F, Sim ou Não, ...)
- 5) Dentro do markdown, se quisermos transformar o texto em fórmula matemática, devemos usar o cifrão (\$)
 - a) Se quisermos manter o texto devemos inserir uma barra (/) antes do cifrão (\$)
 - b) R/\$ 2226,00 em ações da PETR4
- 6) == é um operador lógico de igualdade que pode levar a um booleano
 - a) x = 9
 - b) y = 10
 - c) x == y ➔ False

Operadores Aritméticos

- 1) Sempre que for possível devemos utilizar operadores do python que deixem o nosso código mais simples
- 2) Soma +
- 3) Subtração -
- 4) Multiplicação *
- 5) Divisão /
- 6) Divisão // (parte inteira)
- 7) Resto das divisões %
- 8) Potência **

Operadores Lógicos

- 1) Operadores Comparativos
 - a) Igualdade ==
 - b) Diferença !=
 - c) Menor <
 - d) Maior >
 - e) Menor ou igual <=
 - f) Maior ou igual >=
- 2) Operadores Condicionais
 - a) and (e)
 - b) or (ou)
 - c) not
- 3) Operador and pode ser substituído por &
- 4) Operador or pode ser substituído por | (shift + \)
- 5) Operador de Identidade
 - a) is
 - b) is not

Strings

- 1) Str somente pode concatenar (relacionar) com Str, já com Int ou Float não é possível
- 2) Dentro da Str o sinal de + realiza a concatenação e não a soma
- 3) Str deve ser demonstrada entre aspas
- 4) Mesmo que a Str seja um número entre aspas, o python lê a Str como texto
- 5) Função len = Comprimento da manchete (quantidade de caracteres)
- 6) [] = O colchete demonstra o(s) caractere(s) indexado(s) ou fatiado(s)
- 7) [:] = O sinal de dois pontos dentro do colchete de início ou fim

Propriedade das Strings

- 1) Imutabilidade
- 2) Adição
 - a) Para Str somente podemos usar o sinal de +; já o sinal de menos (-) não é suportado
- 3) Modificação de upper e lower case
 - a) Método = Algo que vem depois de um objeto ou variável com ponto
- 4) Operador Split = Separa caracteres ou palavras

Listas (Estruturas de Dados)

- 1) Para criar listas abrimos colchetes ([])
- 2) Para criar dicionários abrimos chaves ({ })
- 3) Para criar tuplas abrimos parênteses ()
- 4) Métodos para manipulação de listas
 - a) .append() = Adiciona novo elemento à lista
 - b) .pop() = Remove o último item da lista
 - c) .sort() = Classifica elementos de uma lista em ordem crescente
 - d) .reverse () = Classifica elementos de uma lista em ordem decrescente
- 5) Operador lógico in = Dentro de alguma coisa

Dicionários

- 1) Usamos chaves { } no Dicionário
- 2) O que está à direita é a chave; o que está a esquerda é o valor da chave
- 3) O uso de colchetes [] no Dicionário tem significado diferente da Lista
 - a) Na lista o uso do colchetes [] significa posição do objeto
 - b) No dicionário o uso do colchetes [] demonstra o valor da chave que quero extrair o resultado

Tuplas

- 1) Usamos parênteses () nas Tuplas
- 2) len = Mede o comprimento da Tupla
- 3) .index = Indica a posição de um papel na Tupla

Conjuntos

- 1) set é sinônimo de conjunto
- 2) É criado utilizando chaves { } ou a função set()
- 3) .add = Adiciona elementos ao conjunto
- 4) .update = Atualiza/redefine o conjunto todo
- 5) .remove ou .discard = Remove elemento do conjunto
- 6) .pop = Remove elementos aleatórios do conjunto
- 7) .intersection = Interseção entre conjuntos
- 8) .difference = Diferença entre conjuntos
- 9) .symmetric_difference = Exclui a interseção entre conjuntos

Declarações Condicionais

- 1) if = Se
- 2) : = Então
- 3) else: =
 - a) Complementa a condição Se
 - b) Significa qualquer coisa que não está nas condições acima
- 4) elif = É uma condicional intermediária que fica entre o se e o else

Estruturas de Repetição (for e while)

- 1) Loop = Estrutura de Repetição
- 2) For = Para cada elemento dentro da lista eu quero que você faça alguma coisa
- 3) While = Enquanto a condição for verdadeira ele vai continuar executando a ação
- 4) Break = Interrompe o loop assim que uma condição for satisfeita
- 5) Round = Função arredondar
- 6) Função input = Interação com o usuário

Outros Operadores

- 1) Range = Indica um intervalo entre 2 números. Segue o padrão de indexação (primeiro argumento inclusivo e segundo argumento não inclusivo)
- 2) Enumerate = Função que transforma uma coleção de dados (lista, tupla, string) em um objeto enumerado de cada item da coleção
- 3) Zip & In = A função zip consegue executar interações entre listas. O operador in (dentro/pertencente em português) permite a leitura sequencial de objetos zip (assim como de listas), fundamental nas estruturas 'for'
- 4) Random = Este operador pode ser utilizado para obter números aleatórios de acordo com alguma condição pré-determinada

List Comprehension

- 1) List Comprehension = Forma de criar listas a partir de outras listas, baseadas em loops com uma estrutura de código mais resumida do que os "for" tradicionais

Funções e Métodos

- 1) Função = Recebe uma entrada e emite uma saída
- 2) Função = Comando capaz de realizar uma tarefa de acordo com critérios que determinamos
- 3) A Função só é executada de fato quando é chamada
- 4) Exemplos de Funções = print, type,
- 5) def = Define/cria uma função
- 6) Para a Função ser executada ela precisa dos parênteses no final ()
- 7) “Docstring” é a documentação da função, ou seja, a mensagem que aparece quando o usuário consulta a ajuda da função.
- 8) Métodos = Métodos são funções aplicáveis apenas a determinados tipos de objetos. Todos os métodos são funções, mas nem todas as funções são métodos. Os métodos são executados ao fim dos objetos e separados com um ponto.
- 9) Escopo das Variáveis = Área de atuação de uma variável
- 10) Função f = Serve para que você consiga colocar uma variável dentro de um texto

Map e Filter

- 1) Funções embutidas = Funções nativas do python, não precisa de biblioteca para executar
- 2) Aplicam-se a vários elementos de uma única vez
- 3) map() = Aplica um comando (função) a vários elementos de uma lista, de uma vez
- 4) filter() = Aplica um filtro a vários elementos de uma lista de uma só vez

Funções Lambda

- 1) Funções Lambda = Também chamadas de Funções Anônimas
- 2) São funções que não precisam de definição forma (explícita) podendo ser chamadas em apenas uma linha

Args e Kwargs

- 1) Args = Argumentos de palavras-chave
- 2) Kwargs = Argumentos que tem palavras-chave associadas

Trabalhando com Bibliotecas

- 1) Duas principais bibliotecas
 - a) pandas - para se trabalhar com dados (tabela, dataframe)
 - b) numpy - para se trabalhar com matemática
- 2) Explore o repositório de bibliotecas – PyPi
 - a) <https://pypi.org/>
- 3) Wes McKinney
 - a) <https://wesmckinney.com/>
- 4) Duas etapas
 - a) Instalá-las
 - b) Importá-las
- 5) !pip install = Comando para instalar a biblioteca no google colab, jupyter,
...
 - a) Obs: As principais bibliotecas já vêm instaladas

Biblioteca Numpy

- 1) Criamos o array com o Numpy
- 2) Array pode ser entendido como um vetor
- 3) Atributos = Características do objeto