**Aula 03 – 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>

**Aula 04 – Exibindo Mensagens**

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

[https://pt.wikipedia.org/wiki/Sintaxe\_e\_sem%C3%A2ntica\_de\_Python](https://colab.research.google.com/corgiredirector?site=https%3A%2F%2Fpt.wikipedia.org%2Fwiki%2FSintaxe_e_sem%25C3%25A2ntica_de_Python)

**Aula 05 – 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:
5. Integer (inteiro) = Números inteiros
6. Float = Números fracionados
7. O separador decimal no phyton é o ponto (.) e não a vírgula (,)
8. String (str) = Textos, nomes, caracteres alfanuméricos
9. Boolean (bool) = Condições lógicas (V ou F, Sim ou Não, ...)
10. Dentro do markdown, se quisermos transformar o texto em fórmula matemática, devemos usar o cifrão ($)
11. Se quisermos manter o texto devemos inserir uma barra (/) antes do cifrão ($)
12. R/$ 2226,00 em ações da PETR4
13. == é um operador lógico de igualdade que pode levar a um booleano
14. x = 9
15. y = 10
16. x == y 🡺 False

**Aula 06 – Operadores Aritméticos**

1. Sempre que for possível devemos utilizar operadores do phyton 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 \*\*

**Aula 07 – Operadores Lógicos**

1. Operadores Comparativos
2. Igualdade ==
3. Diferença !=
4. Menor <
5. Maior >
6. Menor ou igual <=
7. Maior ou igual >=
8. Operadores Condicionais
9. and (e)
10. or (ou)
11. not
12. Operador and pode ser substituído por &
13. Operador or pode ser substituído por | (shift + \)
14. Operador de Identidade
15. is
16. is not

**Aula 8 – 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 phyton lê a Str como texto
5. Função len = Comprimento da manchete (quantidade de catacteres)
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

**Aula 09 – Propriedade das Strings**

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

**Aula 10 – 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
5. .append( ) = Adiciona novo elemento à lista
6. .pop( ) = Remove o último item da lista
7. .sort( ) = Classifica elementos de uma lista em ordem crescente
8. .reverse ( ) = Classifica elementos de uma lista em ordem decrescente
9. Operador lógico in = Dentro de alguma coisa

**Aula 11 = 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
4. Na lista o uso do colchetes [ ] significa posição do objeto
5. No dicionário o uso do colchetes [ ] demonstra o valor da chave que quero extrair o resultado

**Aula 12 – Tuplas**

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

**Aula 13 – 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

**Aula 14 – Declarações Condicionais**

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

**Aula 15 – 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 = Interage com o usuário

**Aula 16 – 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

**Aula 17 – 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

**Aula 18 – 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

**Aula 19 – Map e Filter**

1. Funções embutidas = Funções nativas do phyton, 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

**Aula 20 – Funções Lambda**

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

**Aula 21 – Args e Kwargs**

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

**Aula 22 – Trabalhando com Bibliotecas**

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

**Aula 23 – Biblioteca Numpy**

1. Criamos o array com o Nunpy
2. Array pode ser entendido como um vetor
3. Atributos = Características do objeto