

# Temas das aulas

Karl Jan Clinckspoor

01/08/2018

---

## Conteúdo que será explorado

0. Introdução
  1. Objetivo
  2. Motivação
    - Automatizar tarefas repetitivas
    - Análise exploratória de dados
    - Tratamento de dados
  3. Ferramentas específicas para tarefas específicas
  4. Por que Python
    - Simples, alto nível, muito suporte, muitos pacotes
  5. Instalação do Python e testes
    - Uso do console/terminal
    - Variável PATH
    - Início de um servidor
  6. Sobre aprender programação
  7. Recursos para aprendizado
    - Software Carpentry, Stack Overflow, Documentação oficial, tutoriais no Youtube.
  8. Promessas
1. Hello World
  1. `print`
  2. Funções
  3. Argumentos
    - `*args`, sem valores padrão
    - `**kwargs`, opcionais, valores padrão
  4. Obtenção de ajuda com Shift+Tab (1x,2x,4x).
  5. Strings
    1. Aspas simples `'`, duplas `"`, triplas `'''`
    2. Escape Sequences: `\n`, `\t`, `\r`
    3. *raw* strings

6. `dir`, `help`, `?`.
7. Tudo é um objeto em Python
  - Métodos e propriedades internas de objetos
8. Funções retornam valores
2. Operações matemáticas
  1. `+`, `-`, `*`, `/`
  2. `ints` e `floats`.
  3. Tipos de objetos, `type`.
  4. Resto `%`, divisão sem decimal `//`, potenciação `**`.
  5. Operadores matemáticos em objetos não numéricos (*overloading*)
    - `str1 + str2`
  6. Erros e como lidar com eles. Tipos de erros
  7. Variáveis, declaração e atribuição.
  8. Conversão entre tipos de variáveis.
    - `float`, `int`, `str`.
  9. Junção de dois tipos incompatíveis: strings e números
    - `%s%` e `%`
    - `{}` e `.format`
    - `{}` e `f'`
    - Arredondando números com `round` e `{:.2f}`
  10. Comparações entre objetos
    - `==`, `!=`, `>`, `>=`, `<`, `<=`, `a < b < c`.
  11. Valores booleanos `True` e `False`.
3. Estruturas de dados (listas, dicionários, tuplas)
  1. Listas:
    1. Declaração: `[item1, item2]`, `list()`.
    2. Indexação: `lista[índice]`.
      - Começa do **zero**, `0`.
      - Índices negativos
    3. Comparação com strings.
    4. Listas aninhadas, `[] []`
    5. Seccionamento (*slicing*)
      1. `início:fim` (não incluso):`passo`
      2. Valores padrão (`tudo:tudo:1`)
      3. Valores negativos para o passo
    6. `len`, `max`, `min`
    7. Remoção de elementos com `del`
    8. Métodos internos:
      - `append`, `sort`, `pop`, `reverse`, `extend`.
      - `+`
      - Alguns métodos operam diretamente na lista e retornam `None`!
    9. Mutabilidade e Imutabilidade
  2. Dicionários:
    1. Declaração: `{chave1:valor1, chave2:valor2}`, `dict`
    2. Indexação: `dict1[chave1]`

- 3. Métodos internos: `keys`, `values`, `items`
- 3. Tuples:
  - 1. Declaração: `(item1, item2)`, `(item1,)`, `tuple`
  - 2. *unpacking* e expressões com asteriscos.
- 4. Condicionais e loops
  - 1. Condicionais:
    - 1. `if`, `elif`, `else`.
    - 2. Blocos de código precedidos por `:`.
    - 3. Junção de condicionais: `and`, `or`, `not`, `all`, `any`.
    - 4. Aninhamento de condicionais.
    - 5. Outros valores interpretados como verdadeiros e falsos
      - Conjuntos vazios, 0.
    - 6. `in`
  - 2. Loops:
    - 1. `while` e `for`, loops infinitos.
    - 2. Utilizando mais de um valor para um loop `for`.
    - 3. `enumerate`, `range`
    - 4. `break` e `continue`.
    - 5. Aninhamento
  - 3. `List comprehension`
  - 4. Abrindo um arquivo de texto e separando os valores necessários.
    - 1. `open`, `split`, `unpacking`, `continue`, `float`
- 5. Instalando e carregando módulos
  - 1. `pip install`
  - 2. `conda install`
  - 3. `import`
    - `import pacote`
    - `import pacote as apelido`
    - `from pacote import parte`
  - 4. Exemplos do uso de pacotes:
    - 1. `uncertainties` para o cálculo e propagação de incertezas.
    - 2. `sympy` para matemática simbólica.
    - 3. `glob` para criação de listas com nomes de arquivos
    - 4. `os` para funções básicas do sistema operacional.
- 6. Definindo funções
  - 1. Declaração: `def nome(argumentos, opcionais=padrão)`
  - 2. `return`
  - 3. Funções anônimas com `lambda`.
  - 4. Documentação, `docstrings`
  - 5. Lidando com erros e exceções com `try` e `except`
  - 6. Escopo de variáveis
  - 7.
- 7. Numpy:
  - 1. `import numpy as np`
  - 2. `np.lookfor`
  - 3. Numpy *arrays*, *linspace*

- Operações afetam *arrays* por inteiro.
  - 4. Funções trigonométricas `sin`, `cos`, `tan`
  - 5. Valores e índices de mínimos com `min` e `argmin`
  - 6. `logspace`, `log10`
  - 7. Criação de arrays 1D e 2D a partir de listas.
  - 8. Indexação e *slicing* de arrays 2D
    - `[linha][coluna]`, `[linha, coluna]`
    - `[l_i:l_f, col_i:col_f]`
  - 9. Pacote `random`
  - 10. Constantes `pi`, `e`
  - 11. Multiplicação de arrays
    - Termo a termo
    - Matricial, `np.dot`, `@`
    - Transformação de vetores sem dimensão para vetores linha e coluna
8. Pandas:
1. `import pandas as pd`
  2. Carregando arquivos com `read_csv`
    - Objeto criado e chamado de `DataFrame` se tiver mais de duas colunas, ou `Series` se tiver só uma.
    - `sep`
    - `decimal`
    - `names`
    - `engine`
    - `header`
    - `na_values`
    - `encoding`
  3. `head`, `info`
  4. Indexação retorna colunas.
    - `df['col1']` retorna uma coluna
    - `df[['col1, col2']]` retorna duas colunas
  5. `loc` retorna linhas e colunas com base em seus nomes.
    - `df.loc[l_i:l_f, col_i:col_f]`
  6. `iloc` retorna linhas e colunas com base em seus índices.
    - `df.iloc[l_i:l_f, col_i:col_f]`
  7. Máscaras lógicas
    1. Criação de um dataframe seguindo uma condição: `filtro = df['y'] < media`
    2. Aplicação do filtro no dataframe inicial. `filtrado = df[filtro]`
9. Criação de gráficos com `matplotlib` e `pyplot`:
1. `import matplotlib.pyplot as plt`
  2. `plot`, `xlabel`, `ylabel`, `title`, `text`
  3. Customizando linhas
    - `color`
    - `linestyle`

- linewidth
  - marker
  - markersize
  - markerfacecolor
4. plots múltiplos colocam mais linhas
  5. figure, figsize, dpi
  6. ylim, xlim, axhline, axvline
  7. Alterando o separador decimal
  8. label e legend
  9. savefig e formatos.
  10. Método implícito (plt.plot) e método explícito (ax.plot)
  11. Subplots com subplot (imp) e subplots (exp).
  12. twinx
  13. xscale, yscale
  14. errorbar
  15. Junção de pandas e pyplot.
  16. imshow e mapas de cor.
10. Algumas ferramentas para tratamento de dados
    1. Processo de tratamento
    2. scipy
      - find\_peaks\_cwt
      - savgol\_filter
      - curve\_fit
      - uncertainties e plots
      - integrate.quad
    3. scikit-image
    4. lmfit
    5. Expressões regulares com re.
    6. Outras ferramentas de visualização: Seaborn e Altair