

# Aula 0 - Introdução

Karl Jan Clinckspoor

15 de julho de 2018

## Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivo	1
1.2	Motivação	1
<b>2</b>	<b>Sobre a linguagem e as ferramentas</b>	<b>2</b>
<b>3</b>	<b>Sobre aprender programação</b>	<b>4</b>
<b>4</b>	<b>Estrutura do curso</b>	<b>4</b>
<b>5</b>	<b>Recursos para aprendizado</b>	<b>4</b>
<b>6</b>	<b>Promessas</b>	<b>5</b>
<b>7</b>	<b>Extra: nbextensions</b>	<b>5</b>

## 1 Introdução

### 1.1 Objetivo

O objetivo principal deste curso é ensinar os princípios básicos de programação, especificamente em Python. Uma ferramenta bastante poderosa, que pode acelerar bastante as tarefas mundanas do dia-a-dia acadêmico, e também acelerar o processo de tratamento de dados.

### 1.2 Motivação

Você já deve ter se deparado com uma situação onde você tinha que realizar alguma tarefa repetitiva, resultando num tédio e perda de sanidade. Alguns exemplos:

- Renomear centenas de arquivos
- Plotar dados de dezenas de experimentos
- Realizar ajustes ou tratamentos complexos repetidas vezes. Aí você se depara que havia um erro na forma do seu tratamento, e seria necessário refazer *tudo*.
- Determinar a melhor maneira de se tratar um enorme conjunto de dados, mas você gostaria de comparar um método de tratamento em todos os dados juntos.

Felizmente, esses tipos de tarefa podem ser automatizadas parcial ou totalmente. Porém, é necessário investir tempo para aprender uma linguagem que permita isso, e aplicá-la na resolução de suas tarefas. Algumas vezes o tempo total ganho acaba sendo nulo, ou até negativo. Porém, se a mesma tarefa aparece no futuro, ela pode ser resolvida num piscar de olhos, pois o investimento inicial já foi feito. Meu desejo em aprender uma linguagem de programação veio justamente da necessidade, e crescente falta de paciência, para o tratamento de dados.

O Origin e o Excel conseguem fazer muito do que está escrito aqui. Porém, conhecer e saber como aplicar ferramentas diferentes é uma habilidade essencial para qualquer profissional, de qualquer área, seja acadêmica ou industrial. O objetivo desse curso não é tentar te converter a achar que Python é a melhor coisa já inventada, mas sim mostrar alternativas para o que é comumente utilizado e ensinado no curso de Química. E se você terminar esse curso, espero que consiga ver o poder que saber uma linguagem de programação lhe dá. Além disso, sabendo uma linguagem, aprender outras se torna muito fácil. O ambiente de Jupyter Notebooks, onde este curso foi feito, possibilita utilizar vários *kernels* de várias linguagens. Inclusive, o nome Jupyter vem de *Julia*, *Python*, *R*, que são três linguagem com bastante aplicabilidade científica.

No final do curso, você irá conseguir fazer gráficos deste tipo em apenas algumas linhas. Quem sabe, conseguir tratar seus dados muito rapidamente.

## 2 Sobre a linguagem e as ferramentas

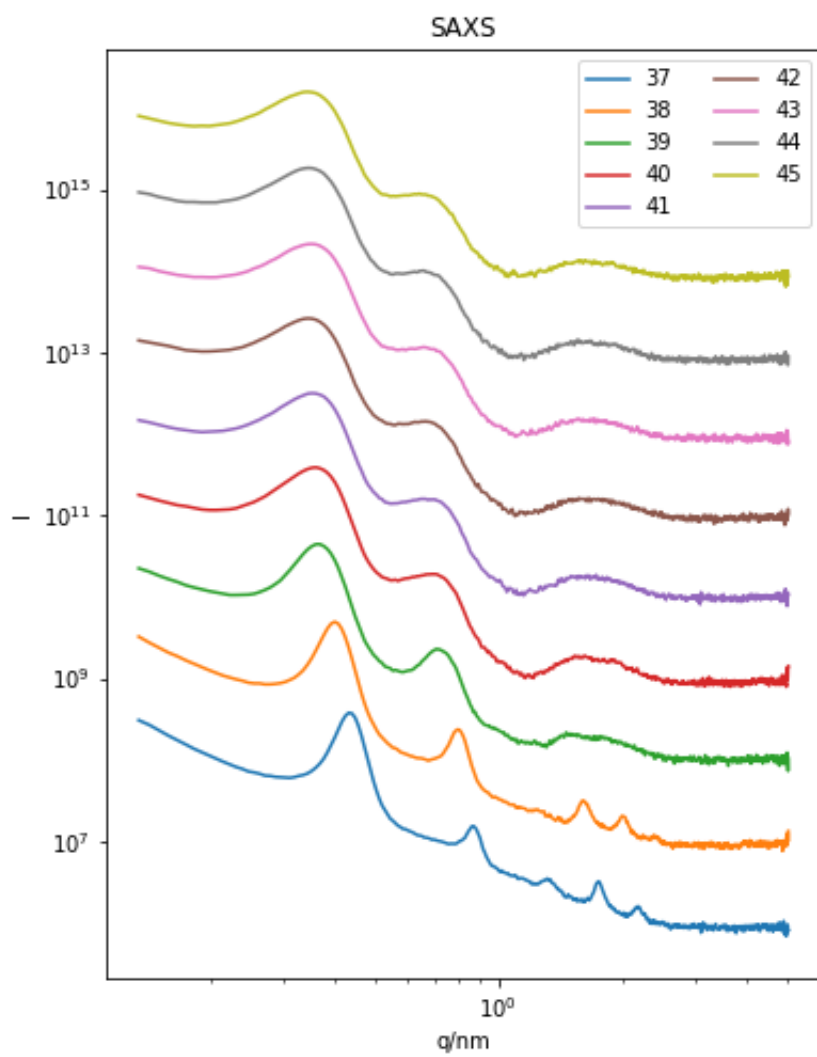
Python é uma linguagem interpretada bastante popular no meio científico devido à sua simplicidade e clareza de sintaxe, comunidade extensa, e, principalmente, pacotes prontos e de fácil uso para a maior parte das tarefas. Além disso, é totalmente gratuita e compatível com Windows, Linux e Mac. Este curso utilizará alguns recursos exclusivos das versões mais recentes do Python, como f-strings, e é recomendado que seja instalada a distribuição [Anaconda](#). Durante a instalação, peça para que sejam adicionados itens na variável PATH de ambiente, pois isso facilita algumas coisas.

A vantagem do pacote Anaconda é que ela vem com muitos pacotes científicos pré-instalados. Este material supõe que o leitor esteja utilizando Windows, mas as diferenças entre Windows e Mac/Linux, quanto a Python, são mínimas. Após a instalação, é possível iniciar uma seção de um Jupyter Notebook no diretório base através do Anaconda Navigator, ou também clicando no ícone do Jupyter. Isso criará uma nova aba no seu navegador (Firefox, Chrome, etc), onde será possível navegar pelo sistema de arquivos, abrir e criar novos arquivos .ipynb. Iniciar um servidor de Jupyter Notebooks pode ser feito também diretamente na linha de comando, uma ferramenta que você se familiarizará um pouco.

Para testar se a instalação ocorreu corretamente:

- Apertar CTRL+R e digitar *cmd* ou *powershell*
- Digitar *cmd* ou *powershell* na barra de endereço de uma janela do windows explorer.
- Clicar com o botão direito numa região vazia do windows explorer, com o Shift apertado, e clicar "Abrir janela de comando/powershell aqui".
- Digite *jupyter notebook*, que inicia o servidor dos notebooks (não feche a janela!) ou *python*, que inicia o interpretador interativo, ou *ipython*, que inicia o interpretador interativo melhorado.

Caso apareça alguma mensagem de erro, mencionando que o arquivo não pode ser encontrado, veja nestes guias como alterar a variável path, para o seu SO. [Windows](#), [Linux](#). Você pode



sempre ir com o console até o diretório onde o Python foi instalado e rodar tudo por lá, ao invés de alterar a variável PATH, mas dá mais trabalho.

É possível utilizar editores de texto ou IDEs alternativas para a criação de scripts, mas Jupyter Notebooks possuem a vantagem de juntarem código e resultados, deixando os dados e a organização do pensamento organizados. Como já dito, é melhor utilizar a ferramenta certa para a tarefa certa, então há casos onde scripts (extensão .py) são mais úteis que notebooks.

É possível rodar arquivos de script diretamente no terminal digitando `python script.py`. Durante as aulas, tente copiar o código que você criou e coloque-os em arquivos .py e tente rodar utilizando o Python.

### 3 Sobre aprender programação

Programação é uma atividade que requer um alto nível de abstração e lógica. Essas habilidades vem somente com a prática e o suor. Algumas pessoas possuem maior aptidão para o tema, mas acredito que todas possuem as habilidades necessárias para aprender a programar, nem que seja o básico. Uma das tarefas principais que um programador deve saber fazer é separar uma tarefa complexa em várias tarefas simples. O quão simples cada passo é depende também da complexidade e das ferramentas de uma linguagem de programação. Python, por ser simples, rica, e de alto nível, facilita bastante o processo. Outro fator essencial para aprender a programar é ter tarefas claras, objetivas, e de complexidade não muito alta. Neste curso, serão fornecidos alguns exemplos reais de problemas sobre tratamento de dados.

Não se sintam mal se tiver que pedir ajuda para resolver um problema, ou se você quiser olhar a resposta. Isso é absolutamente normal, e de vez em quando você aprende melhor olhando a resposta e como alguma coisa foi feita, do que você se frustrar e começar a odiar o que está fazendo.

### 4 Estrutura do curso

1. 'Hello world', strings
2. Operações matemáticas básicas, Variáveis
3. Estruturas de dados. Listas, tuples, dicionários
4. Loops e condicionais
5. Instalação e uso de pacotes externos
6. Funções. Definições e uso. Métodos internos
7. Introdução a numpy
8. Introdução a pandas
9. Introdução a pyplot
10. Ferramentas avançadas

### 5 Recursos para aprendizado

Aprender a programar é uma tarefa árdua e frustrante. Muitas vezes é necessário procurar a resposta para um problema específico. Outras vezes, pode ser que seja necessária uma abordagem diferente de um conteúdo para de fato aprender um tema. Logo, disponho aqui vários links para sites que já utilizei como fonte de informação.

- [Software Carpentry](#), uma organização cujo objetivo é ensinar programação para pesquisadores.

- [Stack Overflow](#), uma comunidade de perguntas e respostas. Grande parte de suas dúvidas já foram perguntadas anteriormente, provavelmente com boas respostas.
- [Documentação oficial do Python](#), onde há informações sobre a linguagem em detalhes profundos. Pode intimidar pessoas novas.
- [Automate the Boring Stuff with Python](#)
- Youtube:
  - [Tutorial por Sendtex](#)
  - [Tutorial por Socratica](#)

## 6 Promessas

Talvez você consiga utilizar essa ferramenta para tratar os seus dados, talvez não. Eu sou só um aluno de doutorado tentando mostrar o que eu aprendi e como eu utilizo essa ferramenta. A chance de sucesso depende de você e de sua situação específica. Não mostrarei como resolver os problemas de todos, mas sim algumas maneiras de resolver alguns problemas, e você depois você altera e aplica esses métodos para seus problemas pessoais.

## 7 Extra: nbextensions

Há algumas opções extras para Jupyter Notebooks que podem ser instaladas, chamadas de nbextensions. Durante as aulas, utilizarei algumas ferramentas extras dessas extensões. Por exemplo, a criação de um sumário automaticamente faz parte desse pacote. Para instalar o nbextensions, rode o seguinte código no console, caso a variável PATH tenha sido configurada corretamente.

```
conda install -c conda-forge jupyter_contrib_nbextensions
```

Depois configure-as pelo browser. Há mais informações sobre o que é *conda* e o que são *pacotes* na aula 5.