

July 23, 2021

1 Conhecendo e Preparando o Ambiente

1.1 Introdução

Esse curso busca capacitar seus estudantes no tocante à biblioteca Sympy, do Python. Portanto, há a necessidade do conhecimento do próprio Python base. Com isso, o curso foi dividido em uma ordem lógica de aprendizagem para que o aluno consiga aprender continuamente, sem grandes “degraus” entre os assuntos. O capítulo 1 é uma introdução ao Python base, o que é suficiente para esse curso.

Neste capítulo, temos como objetivo fazer a instalação e configuração do nosso ambiente de desenvolvimento. Se você já é desenvolvedor Python, não acredito que seja necessária sua mudança ao ambiente que será indicado a esse curso. Mas, caso seja iniciante, recomendo que siga as instruções.

1.2 Escolhendo um Ambiente

Há duas opções para seguir esse curso:

1. Na forma de Notebooks
2. Na forma de Scripts

No curso, serão utilizado notebooks. Mais especificamente, Jupyter Notebooks, através do ambiente JupyterHub. Mas, não se preocupe, ambas as opções serão eficientes. Embora a utilização do Sympy seja regularmente associada ao desenvolvimento em Jupyter.

Na primeira opção, você só terá que instalar um pacote que virá com tudo incluso (IDE, Pacotes, etc.), e sua programação será mais parecida como um caderno. Onde você pode escrever, colocar imagens, matemática, e separar o código em blocos. Contudo, ele é mais pesado e menos flexível.

Na segunda opção, você terá um ambiente mais semelhante à programação tradicional. Linha após linha e comentários. Você terá que instalar o Python puro, versionar/atualizar seus módulos manualmente (o que é mais trabalhoso, mas dá mais autonomia) e utilizar um Editor de Código/IDE.

Caso não tenha entendido o que foi explicado acima, acredito que, nesse momento inicial, a primeira opção é a melhor para você. Contudo, ela não te ajudará a migrar para outra linguagem de programação no futuro, caso queira ou seja necessário. (Isso não é um problema, ao meu ver).

1.3 Instalando o Ambiente (1ª Opção)

1.3.1 Windows

No Windows, a instalação é bem simples, utilizaremos o Anaconda. Para isso, baixe o [instalador do Anaconda](#).

Prossiga a instalação normalmente. Contudo, preste atenção às duas caixas de marcação (checkboxes) que perguntam sobre adicionar o Anaconda ao PATH e torná-lo padrão. Ambas as opções devem estar marcadas.

Se ele oferecer a instalação de algum outro IDE (como o PyCharm), você pode negar.

1.3.2 Linux

A instalação no Linux é um pouco mais complicada. Mas, se você utiliza Linux, possivelmente consegue instalar com certa facilidade.

Verifique as dependências da sua distribuição nesse link: <https://docs.anaconda.com/anaconda/install/linux/>

E depois siga a instalação com o [instalador para Linux](#). Ele é um instalador em Bash, então recomendo que utilize os comandos prescritos no site do Anaconda.

1.4 Instalando o Ambiente (2ª Opção)

Como dito na seção anterior, para essa opção temos que instalar 2 programas, além dos pacotes separadamente.

1.4.1 Windows

No Windows, você terá que baixar e instalar o Python que está disponível nesse link: <https://www.python.org/downloads/>

A instalação é bem simples, você só deve se certificar que ele será adicionado ao PATH e que o pip será instalado. (São duas caixinhas de marcar que devem aparecer)

Para testar se a instalação deu certo, abra um prompt de comando (basta inserir ou cmd ou powershell no menu de pesquisa) e digite:

```
py --version
```

Deve aparecer a versão do Python que você instalou. Caso isso não ocorra, repita o procedimento e veja se fez tudo corretamente.

Também confirme se o pip foi instalado corretamente:

```
pip --version
```

Após isso, você deve escolher um editor. A minha recomendação é o [Visual Studio Code](#).

Mas você pode utilizar outros como o [Atom](#) ou [PyCharm](#).

A instalação de todos é bem simples. Ao abri-los, certifique-se se é necessário instalar uma extensão para suporte ao Python. Caso for, instale-a.

1.4.2 Linux

No Linux, a imensa maioria das distribuições já vêm com o Python instalado e disponível. A única diferença é que algumas adotam o nome `python3` e outras somente `python`.

Para isso, teste no terminal:

```
python --version
python3 --version
```

E comece a usar o que tem a versão mais atual.

Também certifique-se se você tem o `pip` instalado

```
pip --version
pip3 --version
```

Caso não tenha, veja como instalá-lo em sua distribuição.

Assim como no Windows, você deve instalar um editor. As minhas recomendações são as mesmas para o Windows. Ou seja, primeiramente o [Visual Studio Code](#). E depois ou o [Atom](#) ou o [PyCharm](#).

Todos têm versão para Linux e a instalação deve ser até mais simples que para windows. No caso do Visual Studio Code, recomendo que utilize a versão flatpak ou snap.

1.5 Como acompanhar as explicações do curso

O nosso curso será baseado em Jupyter Notebooks, uma forma de programar em blocos. Esses blocos podem ser tanto de código como de texto.

No caso, a primeira opção de ambiente lhe entregará um ambiente muito parecido com o que eu utilizei para escrever os meus Notebooks. Para a segunda, acredito que será mais difícil. Principalmente, pela ausência de alguns pacotes e softwares que vêm com o Anaconda. Mas, decidi incluí-la, pois há pessoas que precisam somente de uma consulta rápida, e possivelmente já têm seu ambiente de desenvolvimento, sendo em Anaconda ou não.

Então, para a primeira opção, você deverá abrir o “Anaconda Navigator”. Esse software servirá como gerenciador de pacotes e dos serviços.

Como dito anteriormente, utilizaremos Jupyter Notebooks. Então basta clicar em “Launch” no card do Jupyter Notebook.

Ele abrirá um navegador de arquivos no navegador, semelhante a um site. Basta escolher uma pasta onde gostaria de criar seus notebooks, e criá-los a partir do botão “New” e então escolha o *Python 3*.

O curso não tem como foco específico ensinar o uso desse ambiente, uma vez que ele é bem simples. Basicamente, você deverá criar blocos (que costumamos chamar de *chunks*) de forma a organizar o entendimento e a saída do seu código. Ao escolher a opção “Code” para seu bloco, você deverá escrever códigos Python nele. No caso de “Markdown”, você deverá escrever texto de acordo com a sintaxe da linguagem de marcação Markdown.

Essa sintaxe é bem simples, dê uma olhada nessa “tabela de cola”: https://github.com/luong-komorebi/Markdown-Tutorial/blob/master/README_pt-BR.md

Para qualquer tipo de bloco, basta segurar a tecla Ctrl e apertar Enter para compilá-lo.

1.6 Exercícios

1. Crie um chunk de markdown e escreva um título com o texto “Olá mundo em Python”.
2. Abaixo desse chunk, crie outro com o código `print('Olá mundo')`.

1.7 Próximos passos

Embora esse capítulo não tenha tido um bloco sequer de código, acredito que agora estão preparados para aprender os básicos de Python