Introdução

Olá! Seja bem-vindo ao curso de lógica de programação com linguagem Python! Vamos repassar algumas orientações iniciais para otimizar o seu uso deste material.

1. Onde programar em Python?

O Python é uma linguagem interpretada. Isso significa que é necessário termos um **interpretador** Python instalado em nosso computador para poder executar nossos programas.

Além disso, é importante termos um programa que permita a edição dos códigos. Tecnicamente, qualquer editor de textos puro serviria, incluindo o Bloco de Notas do Windows. Mas programadores normalmente preferem utilizar uma **IDE** (*Integrated Development Environment*), que oferecerá uma série de recursos para facilitar ainda mais o nosso trabalho, como utilizar código de cores e formatação para deixar o código mais legível, ferramentas para autocompletar o código, ferramentas para procurar erros (*debuggers* e *linters*), entre outros.

Vamos passar brevemente por algumas diferentes opções que poderiam ser adotadas. Mas antes de iniciar a instalação de qualquer uma delas, verifique se o seu professor planejou o curso considerando alguma ferramenta específica.

> Caso queira entender melhor como funciona o processo de interpretação do Python, aqui está uma ótima referência!

1.1. IDLE

Ao baixar o interpretador Python no site oficial, ele irá trazer junto consigo um editor bastante simples, o IDLE.

Ele possui poucos recursos comparado com as próximas IDEs que mencionaremos, mas já é um bom *quebra-galho* caso você precise editar rapidamente um código.

Caso seu professor tenha optado por trabalhar com a distribuição padrão do Python disponível no site oficial, procure baixar a versão mais recente disponível no site e lembre-se de marcar a opção "*Add Python to PATH*" durante a instalação. Isso evitará uma série de problemas na hora de instalar outros recursos e bibliotecas.

1.2. Thonny

O Thonny não é exatamente uma IDE profissional, utilizada no mercado. Seu foco é no aprendizado de Python. Ela oferece recursos especialmente úteis para quem ainda está aprendendo, como acompanhar o valor de variáveis e chamadas de função. Ela também é bastante leve. Ela pode ser obtida gratuitamente no site do projeto.

1.3. Visual Studio Code

O Visual Studio Code é uma IDE gratuita da Microsoft. Seu grande charme é a possibilidade de ter seus recursos expandidos através de extensões. Sendo assim, ele é, na prática, compatível com praticamente qualquer linguagem em uso atualmente, e é bastante fácil acrescentar novas ferramentas a ele. Ele também é bastante leve e fácil de ser configurado.

Você pode fazer o download no site oficial e configurá-lo para suportar Python.

1.4. PyCharm

O PyCharm é uma IDE profissional fornecida pela JetBrains, responsável por diversas IDEs, como o IntelliJ, bastante popular entre desenvolvedores Java e Kotlin. Suas ferramentas são populares de maneira geral entre desenvolvedores web e mobile. O PyCharm possui alguns recursos para facilitar a integração com outros sistemas (em particular com linguagem JavaScript e SQL), e para gerenciar a instalação de bibliotecas, evitando que uma atualização de biblioteca em um projeto possa acidentalmente "quebrar" outro projeto. Em contrapartida, ela é mais pesada, o que pode atrapalhar a experiência de quem estiver utilizando computadores mais lentos.

Sua versão *Community* pode ser baixada gratuitamente no site da JetBrains.

1.5. Anaconda e Jupyter Notebook

O Anaconda é um pacote completo para análise de dados que já vem com uma versão especial do Python (o IPython, ou Interactive Python) pré-instalada, bem como diversas bibliotecas prontas extremamente populares na comunidade científica. Essas bibliotecas incluem uma infinidade de funções prontas para conexão com internet, computação matemática, computação científica, aprendizado de máquina, visualização de dados, conexão com banco de dados, entre outras.

Uma das ferramentas inclusas no Anaconda é o Jupyter Notebook. O grande diferencial dele em relação a uma IDE convencional é o seu formato de "caderno", onde é possível intercalar trechos de códigos com anotações formatadas utilizando a linguagem Markdown.

Ele é extremamente popular na análise de dados, pois é possível ir escrevendo pequenos trechos de código e já visualizando o resultado na forma de gráficos, tabelas e anotações.

O Anaconda pode ser obtido gratuitamente em seu site oficial e já virá com o Jupyter instalado. Alternativamente, você pode seguir o tutorial no blog da Let's Code para instalar apenas o Jupyter e dar os seus primeiros passos.

> Tanto o Visual Studio Code quanto o PyCharm podem ser utilizados para abrir os *notebooks*. Inicie a instalação pelo Anaconda, e através do aplicativo Anaconda Navigator, busque a opção de abrir a IDE desejada e ele fará a mágica por você.

1.6. Google Colab

O Google possui o seu próprio *notebook*, compatível com Jupyter. Ele é 100% online e integrado com Google Drive. Você precisa estar logado com a sua conta Google e o seu trabalho será salvo em seu Drive.

Ele pode ser acessado aqui.

1.7. Programando online

Caso algum dia você se encontre sem o Python instalado em sua máquina - por exemplo, o seu computador quebrou e você conseguiu outro emprestado em cima da hora da aula - não se preocupe. Não faltam opções para programar direto no navegador. Vejamos algumas opções:

> Repl.it: uma IDE online bastante simples, com visualização da saída do código logo ao lado do editor.
>
> OnlineGDB: uma IDE online com suporte a múltiplas linguagens de programação diferentes. Não se esqueça de selecionar "Python3" no menu antes de começar os seus trabalhos.
>
> Jupyter: O Jupyter oferece uma área em seu site para você experimentá-lo sem precisar instalar. Ele será executado direto no seu navegador e possui acesso às bibliotecas padrão de ciência de dados.
>
> Google Colab: Já mencionado anteriormente, vale reforçar que ele é online e não exige qualquer instalação na sua máquina.

2. Como estudar Python?

Gostamos de dizer que o nome da nossa escola não é *Let's Talk*, e sim *Let's Code*. Participar das aulas é bastante importante, mas mais importante do que escutar o professor falar é programar.

Sempre que você ver um código de exemplo nesse material, execute o código na sua IDE. Experimente com o código também: modifique, altere detalhes e veja o que acontece. E, claro, não deixe de fazer os exercícios propostos pelo professor. Eles foram planejados para sedimentar os novos conceitos recém-introduzidos e ajudar a consolidar e integrar os conceitos anteriores.

> **DICA:** Quando for executar os exemplos, evite utilizar o famoso “copia e cola”. Ao invés disso, digite o código e aproveite para se familiarizar com a sintaxe da linguagem.

Quando o seu código não funcionar, não desista, nem simplesmente copie de alguém que funcionou. Busque entender a mensagem de erro (caso haja alguma) e tente ler e entender o que o código está fazendo em cada passo. Caso não consiga resolver, busque o professor, um monitor ou mesmo um colega de turma com quem você se sinta à vontade e mostre seu código para que eles possam ajudar você a entender o seu erro.

E quando você tiver uma ideia, não tenha medo de tentar implementá-la e ver o que acontece!

3. Material extra e referências

Caso você queira complementar seus estudos, há diversas opções baratas ou mesmo gratuitas. Citaremos alguns aqui que são bastante populares e serviram de base para o nosso material aqui.

Livros

> **Pense em Python (2ª edição)** - Allen B. Downey: esse livro é bastante introdutório, e possui uma linguagem extremamente acessível. É ideal para quem está começando e usa

vários exemplos em código. Ele é disponibilizado **gratuitamente** e pode ser lido online. A versão impressa pode ser comprada no site da Editora Novatec.

Think Python (2nd edition) - Allen B. Downey: é a versão original do livro acima. Quem tem facilidade com a língua inglesa pode preferir ler o original. Assim como na tradução, é possível ler gratuitamente online ou adquirir a versão impressa no mesmo link.

Python Basics (4th edition) - RealPython.com: disponível apenas em inglês, é um livro escrito pela equipe que mantém o site RealPython.com. Esse livro é bastante introdutório, mirando estudantes iniciantes. Mas ele percorre uma variedade muito maior de tópicos que o Think Python, e acaba oferecendo mais aplicações.

Python Tricks - Dan Bader: disponível apenas em inglês, escrito por um dos editores do RealPython.com. Esse livro é um pouco mais avançado, e seria mais interessante para quem já chegou aqui sabendo Python ou então após a conclusão do módulo. Ele é um livro de aprofundamento, que irá ajudar quem já sabe o básico de Python a explorar mais a fundo os diferentes recursos que a linguagem oferece.

Python Fluente - Luciano Ramalho: assim como o livro anterior, ele é recomendável para quem já está confortável com a linguagem e gostaria de se aprofundar. Foi escrito por um brasileiro e possui traduções para diversas línguas. É uma leitura obrigatória para programadores Python experientes.

Toda a coleção do Al Sweigart: esse autor escreve diversos livros totalmente baseados em projetos e atividades. Eles são bem legais para complementar os estudos justamente pela oportunidade de ver tudo funcionando na prática. Seus livros são em inglês, mas estão **todos** disponíveis gratuitamente no próprio site do autor, e assim como em outros casos, também podem ser comprados em versões impressas.

Sites

Documentação Oficial do Python: disponível em português, é ponto de parada obrigatório para todos os programadores. Ela explica detalhadamente todos os recursos da linguagem e possui uma referência bastante completa de todas as bibliotecas incluídas na linguagem.

RealPython.com: um site riquíssimo em cursos, artigos e tutoriais. Alguns são gratuitos, outros são pagos. Eles possuem um newsletter que envia dicas de Python regularmente por e-mail.

StackOverflow: talvez você não tenha utilidade imediata para ele. Mas conforme você for programando e pesquisando erros, frequentemente cairá neste site. Ele é um site de perguntas e respostas sobre programação com filtros bastante eficientes para garantir a prevalência de perguntas relevantes e respostas corretas.

Ademais, há uma infinidade de blogs, tutoriais e vídeos, muitas vezes em sites com temáticas específicas (ex: análise de dados ou machine learning). Não deixe de pesquisar quando você tiver dúvidas específicas, você ficará surpreso com quanto material bom de Python existe gratuitamente por aí! :)

Cursos digitais Ada

Além dos materiais acima, temos também diversos cursos digitais da Ada, que podem ser muito úteis em sua jornada de aprendizagem de lógica de programação e Python! Seguem as principais recomendações:

Funcionamento do computador: https://cursos.letscode.com.br/curso-digital/65757051-1636-4c95-8cb3-fd14198d741b

Lógica Pura: https://cursos.letscode.com.br/curso-digital/2120c9f0-02ba-45c1-a81d-3ed26232cc0c

Introdução a Python: https://cursos.letscode.com.br/curso-digital/ad77ffa3-6dde-4efb-9e0d-3b14e60b097b

Como melhorar o seu aprendizado: https://cursos.letscode.com.br/curso-digital/956710f8-d7cf-4bff-93d7-b9d544a82e77

História da Ciência da Computação: https://cursos.letscode.com.br/curso-digital/4114870c-c8e7-4701-aa55-3b6277a8ae67