

## Python: uma Linguagem de Programação Objetivos

### Algoritmos e lógica de programação

Este livro trata de programação de computadores, assunto que atrai muitos interessados, seja porque gostam de computadores, porque querem criar aquele aplicativo revolucionário, porque querem desenvolver websites ou porque ouviram dizer que trabalhar com isso resulta em bom salário. Seja qual for a motivação, muitos chegam a esse mundo da programação de computadores e logo se deparam com dois desafios:

- aprender lógica e, com ela, criar em algoritmo;
- aprender uma linguagem de programação e, com ela, fazer o algoritmo funcionar em um computador.

Esses dois desafios representam duas coisas que precisam ser aprendidas simultaneamente, uma vez que, se faltar uma das duas, não haverá programa de computador.

Um **algoritmo** é uma sequência bem definida e ordenada de passos necessários à solução de algum problema. É comum que professores da área, normalmente na primeira aula, apresentem a seus alunos a ideia de que um algoritmo é como uma receita de bolo ou de ovo frito. É um bom ponto de partida. Porém, assim como algumas receitas exigem um bom chefe de cozinha para serem executadas, alguns algoritmos exigem bons programadores para serem escritos.

Essa sequência bem definida e ordenada supramencionada é aquilo a que se dá o nome de **lógica**. E não é só de passos em sequência que a lógica é feita. Há também a necessidade de escolher o caminho que será tomado, a depender de certa condição ser falsa ou verdadeira; há certos passos que devem ser repetidos um número de vezes ou até que algo ocorra. Um algoritmo, no entanto, não precisa ser um programa de computador, mas, sim, os passos necessários para a solução de um problema.

Para que o algoritmo se torne um programa de computador, é preciso escolher e utilizar uma linguagem de programação. Cada linguagem de programação existente foi criada com algum objetivo, tem características próprias, seu paradigma, um



conjunto de comandos com determinada sintaxe etc. Houve um tempo em que se contava nos dedos o número de linguagens existentes, e são dessa época alguns nomes clássicos como *Assembly*, *Cobol*, *Fortran*, *PL/I*, *C* e *Pascal*. Atualmente, a quantidade de linguagens disponível é tão grande que há um enorme potencial para deixar perdido o iniciante no mundo da computação. Feita a escolha de uma linguagem e iniciados os estudos, começam as dúvidas referentes aos detalhes da linguagem, surgindo questões como: “Onde usar ponto e vírgula (;)?”, “É preciso pular a linha?”, “Tem parênteses ou não?”, e por aí vai.

Assim, considerando que para escrever um programa o estudante precisa desenvolver o algoritmo e utilizar a linguagem de programação para implementá-lo, não há alternativa: é preciso aprender as duas coisas, e deve ser ao mesmo tempo. Existem registros de que a linguagem Python tem sido adotada em muitos cursos, no mundo todo, como linguagem de programação introdutória (GUO, 2014), em um esforço por parte das instituições de ensino superior em adotar uma linguagem que possa ser mais facilmente assimilada pelo estudante, que poderá, assim, concentrar-se melhor no aprendizado dos algoritmos. Por exemplo, informações disponíveis à época da redação deste livro indicam que Python é usada no Massachusetts Institute of Technology (MIT) em Boston, nas universidades paulistas USP e Unicamp, na Universidade Federal de São Carlos, entre outras. Recentemente, o corpo docente da Faculdade de Tecnologia de São Paulo (Fatec-SP), onde o autor deste livro atua como docente na área de programação, adotou também a linguagem Python como ferramenta na disciplina de Algoritmos do primeiro semestre do curso de Análise e Desenvolvimento de Sistemas.

Por outro lado, para o programador que já domina a lógica de programação e conhece pelo menos uma linguagem, aprender uma nova linguagem é um caminho mais rápido e suave. Nestes casos, aprender Python será muito prazeroso, pois é uma linguagem que conta com recursos poderosos que garantem uma produtividade e uma qualidade de software muito significativas.

## **A linguagem Python**

### **Breve histórico e características**



A linguagem Python foi concebida entre o fim de 1989 e o início dos anos 1990 como projeto pessoal de Guido van Rossum, que até hoje continua liderando seu desenvolvimento, contando com a colaboração de muitos desenvolvedores ao redor do mundo. Em uma contínua trajetória evolutiva, ela reúne características relevantes, tais como:

- **Portabilidade:** seu código-fonte é escrito em linguagem ANSI C, e o interpretador Python, bem como suas bibliotecas-padrão, está disponível para um extenso leque de plataformas, que incluem Unix, Linux, Windows (todas as versões), macOS, BeOS, VMS, entre outras. Isso significa que um programa escrito em Python e que utilize apenas as bibliotecas-padrão será executado da mesma maneira em qualquer uma dessas plataformas.
- **Código livre (opensource):** o fato de Python ser *opensource* significa que pode ser utilizado e distribuído livremente. Ele pode ser utilizado por um programador para desenvolver e distribuir um software, assim como seu código-fonte pode ser baixado, adaptado e utilizado sem qualquer restrição.
- **Simplicidade com robustez:** Python é simples como as linguagens de programação de scripts como Perl e Scheme. Por outro lado, conta com recursos que a equiparam a linguagens como C, C++ e Java, permitindo o desenvolvimento de grandes projetos que podem ser constituídos por diversos módulos, que acessem bancos de dados, que enviem e recebam dados por meio de redes, trabalhem com recursos multimídia, entre outros. Python também dispõe de mecanismos que permitem a integração com softwares escritos em outras linguagens, como C.
- **Fácil de aprender:** dizer que Python é fácil de aprender implica entender como seria aprender uma linguagem considerada difícil. Ao iniciante que ainda não conhece linguagem alguma, talvez a primeira seja algo difícil de conseguir entender. Nesse caso, surge a dúvida: será que é mesmo fácil de aprendê-la? Para responder a isso, cabe olhar para o artigo já mencionado (GUO, 2014), que mostra a grande adesão das universidades norte-americanas à Python como linguagem de programação introdutória.
- **Grande aplicabilidade:** o Python pode ser utilizado em um grande número de áreas do desenvolvimento de software, das quais se destacam:



ferramentas para administração e interface com sistemas operacionais; aplicações que trabalhem com grandes volumes de dados armazenados em sistemas gerenciadores de bancos de dados, como Oracle, SQL Server, MySQL e outros; aplicações gráficas e multimídia; desenvolvimento de jogos digitais; programação para internet; desenvolvimento de software para engenharia; aplicações científicas.

### **Versões da linguagem Python**

A linguagem Python conta com duas versões que coexistem. No período em que este livro foi redigido, as versões disponíveis eram: 2.7.14 e 3.6.3. As duas versões apresentam diferenças importantes, a ponto de a versão 3.x representar uma importante quebra de compatibilidade em relação à versão 2.x. Sobre isso, Guido van Rossum (2007) declarou:

Por um bom tempo não havia muito mais do que uma lista de arrependimentos e defeitos estruturais que eram impossíveis de corrigir sem quebrar a compatibilidade retroativa. A ideia era que Python 3000 seria o primeiro release do Python a desistir desta compatibilidade em favor de tornar-se uma linguagem melhor e evoluir.

Uma das características mais marcantes da comunidade de desenvolvedores Python é seu conservadorismo com relação a mudanças que possam causar incompatibilidade retroativa. O lançamento do Python 3.0, em 2008, foi um evento singular e cercado de grande cuidado, atenção e muitas e muitas horas de testes. A quantidade de aplicações e bibliotecas desenvolvidas em Python 2.x é tão grande que, passados quase dez anos do lançamento da versão 3.0, a versão anterior ainda é muito utilizada e permanece viva. Desse modo, tanto ao estudante que inicia seu aprendizado de Python quanto ao programador experiente que deseja aprender a linguagem vem a dúvida: a qual versão se dedicar?

No website oficial de Python (<[www.python.org](http://www.python.org)>) pode-se encontrar a seguinte afirmação: “Python 2.x é legado, Python 3.x é o presente e o futuro da linguagem” (PYTHON SOFTWARE FOUNDATION, 2017). Dessa afirmação o leitor pode tirar as próprias conclusões e decidir a qual versão dedicará seus esforços. Na mesma página onde se lê tal afirmação são descritas as diferenças entre as versões. Se desejar, pode consultar o site para obter mais informações.



## Instalação do Python

A instalação do Python é um processo rápido, simples e seguro. No endereço <[www.python.org/downloads](http://www.python.org/downloads)> estão disponíveis os binários para diversas plataformas. As distribuições GNU/Linux costumam ter alguma versão de Python instalada por padrão. O usuário só precisará verificar qual a versão, 2.x ou 3.x. Para comprovar se o Python está instalado no Linux, pode-se usar o seguinte comando:

```
$ which python
```

Esse comando retornará à pasta onde o Python está instalado ou à mensagem “no python in...”, em caso contrário. Se a distribuição contiver a versão 2.x e desejar a versão 3.x, não há problema. Basta baixá-la e instalá-la em outra pasta.

Quanto ao sistema operacional Windows, é preciso baixar e instalar. O processo é rápido e fácil, sendo apenas necessário que se tenha acesso ao modo administrador do Windows para poder fazê-lo. Acesse a página de downloads do website oficial (Figura 1.1), baixe o instalador da versão desejada e execute-o. É possível executar a instalação-padrão, que incluirá o interpretador, o IDLE, as bibliotecas, a documentação e o gerenciador de bibliotecas Pip. Alternativamente, é possível usar a opção de instalação personalizada, por meio da qual poderá selecionar o que deseja e o que não deseja instalar.

Acesse o endereço <https://www.python.org/downloads/>



## Instalação do Python

A versão mais recente disponível quando este livro foi escrito é a 3.6.3. Desse





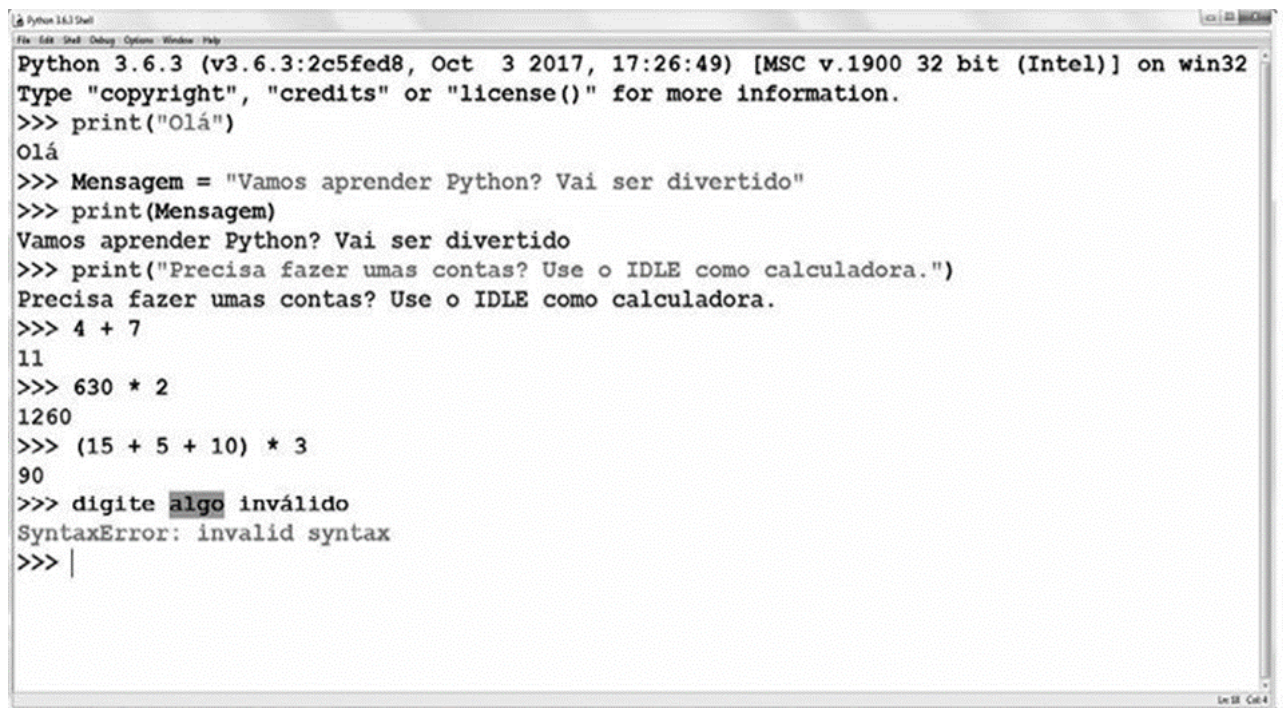
modo, todo o material aqui disponível, incluindo exemplos e exercícios resolvidos, foi elaborado com base nessa versão, para o sistema operacional Windows. Por ser multiplataforma, todo esse material poderá ser utilizado e testado em qualquer outra plataforma para a qual esteja disponível o interpretador Python 3.6.

## Iniciando com o Python

### O ambiente IDLE

Uma vez instalado, os primeiros passos podem ser dados com o IDLE. Esse nome refere-se a uma interface que o programador pode utilizar para interagir com o interpretador Python de uma maneira muito dinâmica.

Procure por ele em seu computador e abra-o. Será aberta uma janela, como mostrado na Figura 1.2, na qual há um *prompt* indicado pelo sinal ">>>". No *prompt* é possível digitar comandos Python, e o interpretador imediatamente os executará. Faça alguns testes repetindo os comandos mostrados na figura.



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Olá")
Olá
>>> Mensagem = "Vamos aprender Python? Vai ser divertido"
>>> print(Mensagem)
Vamos aprender Python? Vai ser divertido
>>> print("Precisa fazer umas contas? Use o IDLE como calculadora.")
Precisa fazer umas contas? Use o IDLE como calculadora.
>>> 4 + 7
11
>>> 630 * 2
1260
>>> (15 + 5 + 10) * 3
90
>>> digite algo inválido
SyntaxError: invalid syntax
>>> |
```

IDLE, a básica interface de usuário de Python.

Sempre que estiver em dúvida sobre o funcionamento de um comando,



método, função ou qualquer outro recurso da linguagem, poderá recorrer ao IDLE para testá-lo e entender como funciona e como utilizar o recurso. Se algo inválido, incorreto ou inexistente for digitado no IDLE, o interpretador avisará com uma mensagem. As mensagens emitidas são, em geral, autoexplicativas, contendo um grau de detalhe nas informações que ajudam bastante no entendimento e na solução do erro. Além disso, na internet há uma profusão de informações disponíveis, uma vez que a comunidade Python é grande e muito atuante.

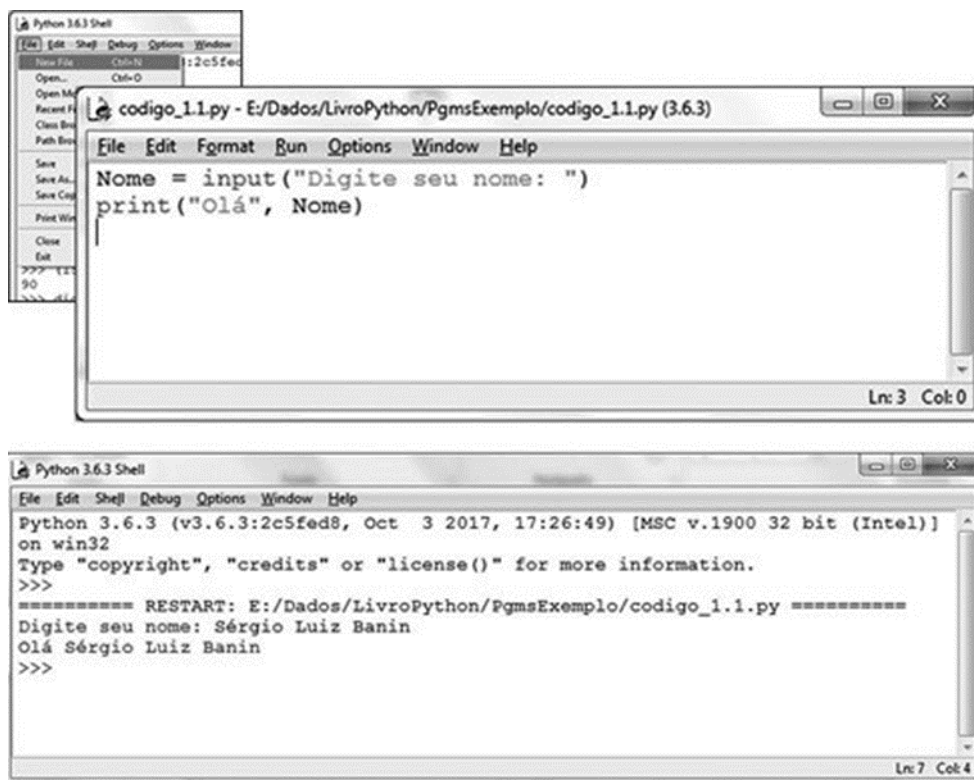
Em todos os capítulos deste livro o IDLE é utilizado para exemplificar os comandos, conceitos e recursos abordados.

### **O editor de scripts**

O IDLE é muito útil para testes e aprendizado, haja vista sua interatividade, porém, chegará o momento em que será necessário escrever um programa completo, salvá-lo em disco e posto a executar com começo, meio e fim, sem a interatividade.

Para isso, pode-se utilizar qualquer editor de textos simples como o Bloco de Notas ou o Notepad++. Porém, a opção mais simples é utilizar o editor de scripts que pode ser aberto a partir do IDLE, a partir do menu *File* → *New File*. Esse editor é integrado ao IDLE, de modo que é possível digitar seu programa, salvá-lo na pasta de sua preferência e, em seguida, pressionar a tecla de atalho F5 para executar o programa. Ao fazer isso, o programa é posto em execução no IDLE, e será possível testá-lo.





Editor de scripts do Python.

### Outras IDEs

Os programadores já experientes e acostumados a trabalhar com outras linguagens talvez estranhem a simplicidade do ambiente IDLE e seu editor de scripts. Bem, o Python é assim mesmo. A filosofia promovida por Guido van Rossum e o time de desenvolvedores é uma filosofia que contém a simplicidade, a elegância, a coerência e a consistência entre seus atributos.

No entanto, com o passar do tempo e o aumento da utilização do Python ao redor do mundo, surgiram diversas opções de **ambientes integrados de desenvolvimento**, ou no termo pelo qual é de fato conhecido, mesmo no Brasil, IDE (*Integrated Development Environment*).

Um IDE é um programa de computador que contém recursos e funcionalidades direcionados ao desenvolvimento de programas de computador.

É possível encontrar na internet uma grande variedade de IDEs que podem ser utilizadas para escrever programas em linguagem Python. Uma rápida busca o levará a websites intitulados “Os top 10 IDEs para Python” ou “os 3 melhores IDEs para Python”. Listas assim refletem as preferências e necessidades de quem as elabora.





Cada um desses ambientes terá recursos, características, vantagens e desvantagens próprios que não são objeto do texto deste livro. E, embora haja boas discussões técnicas acerca de benefícios e problemas de cada IDE, há também certa disputa de torcidas, mais ou menos aos moldes de “é biscoito ou bolacha?”. Há IDEs que são gratuitas e *opensource*, ao passo que outras são proprietárias e requerem o pagamento de licenças.

No Quadro 1.1 é oferecida uma lista parcial dos ambientes disponíveis e mais utilizados à época de redação deste livro, sem entrar no mérito de qual é melhor, mais poderoso, mais bonito ou qualquer outra adjetivação que envolva as palavras “mais” ou “menos”.

•Atom	•PyCharm
•Eclipse com PyDev	•Spyder
•Eric Python	•VIM
•Komodo IDE	•Visual Code Studio
•Ninja IDE	•Wing IDE

**Quadro** - Lista parcial de IDEs para escrever programas em Python em ordem alfabética (para uma lista mais completa, consulte<<https://wiki.python.org.br/IdesPython>>).

### Documentação e suporte

A comunidade Python, além de grande, é muito engajada, ativa e dinâmica. Com isso, não só a linguagem vem sofrendo constantes melhorias e acréscimos, como muito material de apoio tem sido produzido.

A documentação básica é o *Python Docs*, que está disponível no endereço <<https://docs.python.org/3/>> e que também é instalada na máquina do usuário, junto



com o interpretador. No IDLE, aperte F1 para ter acesso a ele. O *Python Docs* é a fonte de referência primária para todo programador que trabalhe com a linguagem.

Somando-se ao *Python Docs*, há um grande número de artigos, fóruns, listas de discussões, livros, blogs, e vídeos no *YouTube*, tudo isso acessível on-line em vários idiomas, inclusive português. Para os brasileiros, a comunidade Python Brasil (<<http://python.org.br>>) é um excelente recurso em português.

Outra referência importante e avançada é o conjunto de PEPs (*Python Enhancement Proposals*). PEP é um documento padronizado utilizado para formalizar a divulgação de informações à comunidade Python, para descrever uma nova funcionalidade, para que sejam apresentadas propostas de novos recursos, para a coleta de informações sobre problemas e para documentar as decisões de projeto que foram adotadas no Python.

### **A quem se destina este livro**

O conteúdo foi pensado para atender desde o aluno iniciante até o programador experiente que necessita ou deseja aprender a linguagem Python.

Ao iniciante a linguagem Python propicia uma ferramenta simples, intuitiva e de fácil compreensão. Isso permite que o estudante não gaste seu tempo e energia com detalhes e especificidades da linguagem e dedique-os à resolução dos problemas de lógica, construindo algoritmos que representem a solução para tais problemas.

Ao programador que precisa aprender Python este livro oferece o modelo, os conceitos, elementos e detalhes da linguagem, com bom grau de profundidade, não encontrado em livros básicos. Tais elementos levam a uma compreensão de como o interpretador foi pensado para que todos os recursos existentes sejam coerentes, robustos e confiáveis, garantindo a qualidade do produto final, que é o software escrito em Python.

### **Requisitos mínimos**

Trabalhar com Python não requer um equipamento caro e cheio de recursos. Ao contrário, uma máquina mediana será capaz de executar o Python com folga. Um computador com processador i3 das primeiras gerações e com 2 GB de memória já é suficiente para isso. Quanto a software, todos os exemplos deste livro foram escritos



e testados com uso de Python versão 3.6.3 para o sistema operacional Windows. No entanto, não foi utilizado nenhum recurso específico dessa plataforma, de modo que se utilizar outro sistema operacional não encontrará dificuldades.

### **Hardware**

- Processador i3 de primeira geração 1220 MHz, ou compatível.
- 2 gigabytes (GB) de memória do sistema.
- Disco rígido com 1 GB livre.
- Acesso à internet para baixar o instalador, no caso do ambiente Windows e eventuais atualizações, no caso de Linux e macOS.
- 

### **Software**

No caso do sistema operacional Windows, é exigida a versão Vista ou superior para o Python 3.6 em diante. Caso ainda opte pelo Windows XP, é possível instalar e utilizar o Python 3.4.

Quanto ao sistema operacional Linux, a maioria das distribuições Linux existentes atualmente já disponibilizam o interpretador Python pré-instalado ou pacotes binários que podem ser facilmente instalados. Para verificar, abra seu terminal e digite: `python -v`.

Os computadores da Apple com o macOS também já acompanham um interpretador Python pré-instalado que pode ser atualizado baixando a última versão disponibilizada no site oficial da linguagem Python na versão 3.6.

