

Começando a Programar em Python Para Leigos

Folha de Cola

De **Começando a Programar em Python Para Leigos**
por John Paul Mueller

Python é uma linguagem de programação incrível e flexível que tem um significativo apoio de terceiros e é usada em uma farta área de aplicações. As aplicações que você construir rodarão em qualquer plataforma que dê suporte ao Python sem qualquer modificação desde que você crie uma solução usando somente o Python. É claro que você gostaria de garantir que as suas aplicações tivessem a melhor chance possível de funcionar exatamente como você esperaria onde quer que elas rodem e é por isso que você precisa destas informações nesta Folha de Cola.

8 Erros de Programação Comuns no Python

Todo desenvolvedor no planeta comete algum erro. Entretanto, conhecer alguns erros comuns poderá economizar seu tempo e esforço posteriormente. A lista seguinte apresenta os erros mais comuns que os desenvolvedores vivenciam ao trabalhar com o Python.

- **Usando a indentação incorreta:** Muitas funcionalidades do Python baseiam-se em indentação. Por exemplo, quando você cria uma nova classe, tudo naquela classe é indentado em baixo da declaração da classe. O mesmo ocorre para a decisão, o loop e outras declarações de estruturas. Se você achar que seu código está executando uma tarefa quando não deveria, comece a rever a indentação que você está usando.
- **Baseando-se no operador de atribuição, em vez do operador de igualdade:** Ao executar uma comparação entre dois objetos ou valores, você somente deve usar o operador de igualdade (`==`), não o operador de atribuição (`=`). O operador de atribuição coloca um objeto ou um valor em uma variável e não compara com nada.
- **Colocando a função chamadora na ordem errada ao criar declarações complexas:** o Python sempre executa funções da esquerda para a direita. Então, a declaração `MyString.strip().center(21, "*")`, produz um resultado diferente do que `MyString.center(21, "*").strip()`. Quando você encontrar uma situação em que a saída da série de funções concatenadas é diferente daquela que estava esperando, você precisa verificar a ordem das funções para garantir que cada função esteja no lugar certo.
- **Pontuação posicionada incorretamente:** É possível colocar a pontuação no local errado e criar um resultado completamente diferente. Lembre-se de que você deve incluir dois pontos no final de cada declaração de estrutura. Além disso, a colocação dos parênteses é crítica. Por exemplo,

```
(1 + 2) * (3 + 4), 1 + ((2 * 3) + 4), e  
1 + (2 * (3 + 4)) produzem resultados diferentes.
```

Para Leigos: A série de livros para iniciantes que mais vende no mundo.

Começando a Programar em Python Para Leigos

Folha de Cola

- **Usando o operador lógico incorreto:** A maioria dos operadores não apresenta problemas, mas os operadores lógicos podem apresentar. Lembre-se de usar `and` para determinar quando ambos os operandos devem ser `True` e `or` quando um dos operandos pode ser `True`.
- **Criando erros, com incremento de um, em loops:** Lembre-se de que um loop não conta o último número que você especificou em uma faixa de números. Então, se você especificar a faixa `[1:11]`, na verdade, você obtém saída com valores entre 1 e 10.
- **Usando a capitalização errada:** Python distingue entre caixas alta e baixa, então `MyVar` é diferente de `myvar` e de `MYVAR`. Sempre verifique a capitalização quando você perceber que não consegue acessar um valor que você esperava acessar.
- **Cometendo um erro de ortografia:** Mesmo os desenvolvedores experientes sofrem com erros de ortografia algumas vezes. Garantir que você use uma mesma abordagem para nomear as variáveis, classes e funções ajuda bastante. No entanto, mesmo um esquema de nomenclatura bem consistente nem sempre vai prevenir que você digite `MyVer` quando você gostaria de ter digitado `MyVar`.

Operadores utilizados regularmente com o Python

É importante saber quais operadores o Python suporta, e lembrar de todos eles não é sempre fácil. A tabela seguinte fornece um rápido resumo dos operadores regularmente usados com o Python.

Operador	Tipo	Descrição	Exemplo
-	Aritmético	Subtrai o operando da direita do operando da esquerda.	$5 - 2 = 3$
-	Unário	Nega o valor original, de modo que positivo se torna negativo e vice-versa.	$-(-4)$ resulta em 4 enquanto -4 resulta em -4
-=	Atribuição	Subtrai o valor encontrado no operando da direita do valor encontrado no operando da esquerda e coloca o resultado no operando da esquerda.	<code>MyVar -= 2</code> resulta em <code>MyVar</code> contendo 3
!=	Relacional	Determina se dois valores não são iguais. Algumas versões mais antigas do Python permitiriam que você usasse o operador <code><></code> no lugar do operador <code>!=</code> . Usar o operador <code><></code> resulta em um erro nas versões atuais do Python.	$1 != 2$ é verdadeiro

Para Leigos: A série de livros para iniciantes que mais vende no mundo.

Começando a Programar em Python

PARA
LEIGOS®

por John Paul Mueller



ALTA BOOKS
E D I T O R A
Rio de Janeiro, 2015

Sobre o Autor

John Mueller é autor freelancer e editor técnico. Tendo a escrita em seu sangue, já produziu 95 livros e mais de 300 artigos até a presente data. Os assuntos variam de *networking* a inteligência artificial e de gerenciamento de banco de dados a programação avançada. Alguns de seus livros mais recentes incluem uma referência de linha de comando do Windows, livros sobre VBA e Visio 2007, um manual de design e desenvolvimento em C#, e um guia para programadores de IronPython. A habilidade em edições técnicas de John ajudou mais de 63 autores no refinamento de conteúdo de seus manuscritos. Ele forneceu serviços de edição técnica para as revistas *Data Based Advisor* e *Coast Compute*. Ele também contribuiu com artigos para revistas como *Software*, *Quality Connection*, *DevSource*, *InformIT*, *SQL Server Professional*, *Visual C++ Developer*, *Hard Core Visual Basic*, *asp.netPRO*, *Software Test and Performance*, e *Visual Basic Developer*. Leia o blog do John em <http://blog.johnmuellerbooks.com/> — conteúdo em inglês.

Quando John não está trabalhando no computador, você pode encontrá-lo ao ar livre: no jardim, cortando madeira, ou geralmente curtindo a natureza. John também gosta de produzir vinho, cozinhar biscoitos e fazer tricô. Quando não está ocupado com nada, ele faz sabão de glicerina e velas, que caem bem em cestas de presentes. Você pode achar John na internet em John@JohnMuellerBooks.com. John também está configurando um website em <http://www.johnmuellerbooks.com/> — conteúdo em inglês. Sinta-se à vontade para dar uma olhada no website e fazer sugestões em como ele pode melhorá-lo.

Dedicatória

Algumas pessoas simplesmente estão na sua vida — tão confiáveis quanto o dia é longo. Scott e Pegg Conderman são duas destas pessoas — simplesmente sendo eles mesmos e sabendo o que fazer para tornar o dia um pouco melhor, eles me ajudaram a passar por um momento extremamente difícil.

Agradecimentos do Autor

Obrigado à minha esposa, Rebecca. Mesmo que ela tenha falecido, o seu espírito está em cada livro que eu escrevo, em cada palavra que aparece na página. Ela acreditava em mim quando ninguém mais acreditava.

Russ Mullen merece um agradecimento por sua edição técnica deste livro. Ele somou grandemente à precisão e à profundidade do material que você vê aqui. Russ está sempre me mostrando importantes URLs para novos produtos e ideias. No entanto, é a verificação que Russ faz que mais me ajuda. Ele faz o controle de qualidade do meu trabalho. Russ também tem equipamentos de computação diferentes dos meus, então ele pode apontar falhas que eu provavelmente não observaria.

Matt Wagner, meu agente, merece créditos primeiramente por me ajudar a conseguir o contrato, e por cuidar de todos os detalhes que a maioria dos autores não levam muito em consideração. Eu sempre gostei da sua assistência. É bom saber que alguém quer me ajudar.

Várias pessoas leram tudo ou parte deste livro para ajudar-me a refinar a abordagem, testar os exemplos de código, e, no geral, fornecer informações que todos os leitores gostariam de ter. Estes voluntários, não pagos, ajudaram de tantas formas que nem poderia mencionar todas aqui. Eu aprecio especialmente os esforços de Eva Beattie, Glenn A. Russell, Emanuel Jonas e Michael Sasseen, que forneceram informações diversas, leram o livro inteiro, e generosamente dedicaram-se a este projeto.

Finalmente, eu gostaria de agradecer Kyle Looper, Susan Christophersen, e o restante da equipe editorial e de produção.

Sumário Resumido

Introdução	1
Parte I: Iniciando com Python	5
Capítulo 1: Falando com Seu Computador.....	7
Capítulo 2: Obtendo Sua Própria Cópia do Python.....	21
Capítulo 3: Interagindo com Python.....	39
Capítulo 4: Escrevendo Sua Primeira Aplicação.....	57
Parte II: Falando a Língua.....	81
Capítulo 5: Armazenando e Modificando Informações.....	83
Capítulo 6: Gerenciando Informações.....	93
Capítulo 7: Tomando Decisões.....	117
Capítulo 8: Executando Tarefas Repetitivas.....	133
Capítulo 9: Lidando com Erros.....	149
Parte III: Executando Tarefas Comuns.....	181
Capítulo 10: Interagindo com Módulos.....	183
Capítulo 11: Trabalhando com Strings.....	205
Capítulo 12: Gerenciando Listas.....	223
Capítulo 13: Coletando Vários Tipos de Dados.....	243
Capítulo 14: Criando e Usando Classes.....	267
Parte IV: Executando Tarefas Avançadas	291
Capítulo 15: Armazenando Dados em Arquivos.....	293
Capítulo 16: Enviando um E-mail.....	309
Parte V: A Parte dos Dez	327
Capítulo 17: Dez Recursos de Incríveis de Programação.....	329
Capítulo 18: Dez Meios de Sustentar-se com Python.....	339
Capítulo 19: Dez Ferramentas Interessantes.....	347
Capítulo 20: Dez Bibliotecas que Você Precisa Conhecer.....	357
Índice	365

Sumário

Introdução	1
Sobre Este Livro	1
Penso Que...	2
Ícones Utilizados Neste Livro	3
Além Deste Livro.....	3
De Lá para Cá, Daqui para Lá	4
Parte I: Começando com Python	5
Capítulo 1: Falando com Seu Computador	7
Entendendo Por Que Você Quer Falar com Seu Computador	7
Sabendo que uma Aplicação é uma Forma de Comunicação	9
Pensando sobre procedimentos que você usa diariamente.....	9
Escrevendo procedimentos.....	10
Vendo aplicações como se fossem qualquer outro procedimento....	11
Entendendo que os computadores entendem coisas literalmente	11
Definindo o Que É uma Aplicação.....	12
Entendendo que os computadores usam uma linguagem especial...	12
Ajudando os humanos a falarem com o computador	13
Entendendo Por Que Python É Tão Legal	14
Descobrimo motivos para usar Python	15
Decidindo como você pode ter benefícios pessoais com Python.....	16
Descobrimo quais empresas usam Python.....	17
Encontrando aplicações Python úteis	18
Comparando Python com outras linguagens	18
Capítulo 2: Obtendo sua Própria Cópia do Python.	21
Baixando a Versão que Você Precisa	21
Instalando Python.....	24
Trabalhando com o Windows.....	25
Trabalhando com o Mac.....	27
Trabalhando com o Linux	29
Acessando Python na Sua Máquina	32
Usando o Windows.....	32
Usando o Mac.....	35
Usando o Linux	36
Testando Sua Instalação	36

Capítulo 3: Interagindo com o Python 39

Abrindo a Linha de Comando	40
Iniciando Python.....	41
Usando a linha de comando em sua vantagem.....	42
Usando variáveis de ambiente do Python em sua vantagem.....	44
Digitando um comando.....	45
Dizendo ao computador o que fazer.....	46
Dizendo ao computador que você terminou	46
Vendo o resultado	46
Usando a Ajuda.....	48
Entrando em modo ajuda	48
Pedindo ajuda	49
Saindo do modo ajuda.....	52
Obtendo ajuda diretamente	52
Fechando a linha de comando.....	54

Capítulo 4: Escrevendo Sua Primeira Aplicação. 57

Entendendo o Ambiente Integrado de Desenvolvimento (IDLE)	58
Iniciando IDLE	59
Usando comandos padrões.....	60
Entendendo a codificação por cores	61
Conseguindo ajuda da GUI.....	62
Configurando IDLE	63
Criando a Aplicação.....	67
Abrindo uma nova janela	67
Digitando o comando	68
Salvando o arquivo.....	69
Rodando a Aplicação	71
Entendendo o Uso da Indentação.....	72
Adicionando Comentários	74
Entendendo os comentários.....	74
Usando comentários para deixar lembretes para si mesmo.....	75
Usando comentários para impedir a execução de um código.....	75
Carregando e Rodando Aplicações Existentes	78
Usando a linha de comando ou janela terminal.....	78
Usando a janela Edit.....	79
Usando a janela Shell ou linha de comando do Python.....	79
Fechando o IDLE	80

Parte II: Falando a Língua 81**Capítulo 5: Armazenando e Modificando Informações 83**

Armazenando Informações.....	83
Vendo variáveis como caixas de armazenamento	84
Usando a caixa certa para armazenar os dados	84

Definindo Tipos de Dados Essenciais do Python	85
Colocando informações dentro de variáveis.....	85
Entendendo os tipos numéricos	85
Entendendo valores Booleanos	89
Entendendo strings	90
Trabalhando com Datas & Horas.....	91
Capítulo 6: Gerenciando Informações	93
Controlando como Python Vê os Dados	94
Fazendo comparações.....	94
Entendendo como os computadores fazem comparações.....	95
Trabalhando com Operadores.....	95
Definindo os operadores.....	96
Entendendo a precedência do operador.....	103
Criando e Usando Funções	104
Visualizando funções como pacotes de código	104
Entendendo a reutilização de códigos.....	104
Definindo uma função	105
Acessando funções.....	107
Enviando informações às funções.....	108
Retornando informações das funções.....	112
Comparando funções de saída	114
Obtendo entrada do usuário.....	114
Capítulo 7: Tomando Decisões	117
Tomando Decisões Simples	
Utilizando a Instrução if.....	118
Entendendo a instrução if.....	118
Usando a instrução if em uma aplicação	119
Escolhendo Alternativas Usando a Instrução if...else.....	124
Entendendo a instrução if...else.....	124
Usando a instrução if...else em uma aplicação.....	124
Usando a instrução if...elif em uma aplicação	125
Usando Instruções de Decisões Aninhadas.....	129
Usando múltiplas instruções if ou if...else.....	129
Combinando outros tipos de decisões.....	130
Capítulo 8: Executando Tarefas Repetitivas	133
Processando Dados Usando a Instrução for.....	134
Entendendo a instrução for.....	134
Criando um loop for básico	135
Controlando execuções com a instrução break.....	136
Controlando execuções com a instrução continue	138
Controlando execuções com a cláusula pass	140
Controlando execuções com a instrução else	141

Processando Dados Usando a Instrução while	143
Entendendo a instrução while	143
Usando a instrução while em uma aplicação	144
Aninhando Instruções de Loop	145

Capítulo 9: Lidando com Erros. 149

Sabendo Porque Python Não Entende Você	150
Examinando as Fontes dos Erros.....	151
Classificando quando os erros ocorrem	152
Distinguindo tipos de erros.....	153
Capturando exceções	155
Manipulando exceções básicas	156
Manipulando exceções mais específicas para menos específicas ...	167
Manipulação de exceções aninhadas.....	170
Levantando Exceções	174
Levantando exceções durante condições excepcionais.....	174
Passando informação de erro ao chamador	175
Criando e Usando Exceções Customizadas	176
Usando a Cláusula finally	178

Parte III: Executando Tarefas Comuns 181

Capítulo 10: Interagindo com Módulos 183

Criando Agrupamentos de Código	184
Importando Módulos.....	185
Usando a instrução import	187
Usando a instrução from...import	188
Encontrando Módulos no Disco	191
Visualizando o Conteúdo do Módulo	193
Usando o Módulo de Documentação do Python	198
Abrindo a aplicação pydoc	198
Usando links de acesso rápido.....	200
Digitando um termo de pesquisa.....	202
Visualizando os resultados	203

Capítulo 11: Trabalhando com Strings. 205

Entendendo Que Strings São Diferentes	206
Definindo um caractere usando números	206
Usando caracteres para criar strings.....	207
Criando Strings com Caracteres Especiais.....	208
Selecionando Caracteres Individuais.....	211
Detalhando Strings	213
Localizando um Valor em uma String.....	217
Formatando Strings	219

Capítulo 12: Gerenciando Listas	223
Organizando Informações em uma Aplicação	224
Definindo organizações usando listas	224
Entendendo como computadores veem as listas	225
Criando Listas	226
Acessando Listas	228
Fazendo Loop através de Listas	231
Modificando Listas	232
Pesquisando em Listas	236
Ordenando Listas	238
Trabalhando com o Objeto Counter	240
Capítulo 13: Coletando Vários Tipos de Dados	243
Entendendo Coleções	243
Trabalhando com Tuplas	245
Trabalhando com Dicionários	248
Criando e usando um dicionário	249
Substituindo a instrução switch por um dictionary	253
Criando Pilhas Usando Listas	256
Trabalhando com queue (filas)	260
Trabalhando com deque	263
Capítulo 14: Criando e Usando Classes.	267
Entendendo a Classe como um Método de Empacotamento	268
Analisando as Partes de uma Classe	269
Criando a definição de classe	269
Considerando os atributos de classe embutidos	271
Trabalhando com métodos	273
Trabalhando com construtores	275
Trabalhando com variáveis	277
Usando métodos com listas de argumentos variáveis	281
Sobrecarregando operadores	282
Criando uma Classe	284
Usando a Classe em uma Aplicação	285
Estendendo Classes para Fazer Novas Classes	287
Construindo a classe filha	287
Testando a classe em uma aplicação	289
Parte IV: Executando Tarefas Avançadas	291
Capítulo 15: Armazenando Dados em Arquivos.	293
Entendendo Como o Armazenamento Permanente Funciona	294
Criando Conteúdo para Armazenamento Permanente	295
Criando um Arquivo	298

Lendo o Conteúdo do Arquivo	301
Atualizando o Conteúdo do Arquivo	303
Apagando um Arquivo.....	308
Capítulo 16: Enviando um E-mail	309
Entendendo O Que Acontece Quando Você Envia Um E-mail	310
Vendo o e-mail como você vê uma carta	311
Definindo as partes do envelope	312
Definindo as partes da carta.....	318
Criando a Mensagem de E-mail	322
Trabalhando com uma mensagem de texto.....	323
Trabalhando com uma mensagem em HTML.....	324
Vendo a Saída do E-mail.....	325
Parte V: A Parte dos Dez.....	327
Capítulo 17: Dez Recursos Incríveis de Programação	329
Trabalhando com a Documentação Online do Python.....	330
Usando o Tutorial LearnPython.org	331
Executando Programação Web Usando Python.....	332
Obtendo Bibliotecas Adicionais.....	332
Criando Aplicações Rapidamente Usando um IDE.....	334
Checando Sua Sintaxe com Maior Facilidade	334
Usando XML a Seu Favor	335
Superando Erros Comuns de Novatos do Python.....	336
Entendendo Unicode	337
Tornando Sua Aplicação Python Mais Rápida.....	338
Capítulo 18: Dez Meios de Sustentar-se com Python	339
Trabalhando em Qualidade de Software (QA).....	340
Tornando-se Staff de T.I. de uma Pequena Empresa.....	341
Executando Scripts Especiais para Aplicações.....	342
Administrando uma Rede.....	343
Ensinando Técnicas de Programação.....	343
Ajudando as Pessoas a Decidirem sobre Localização	344
Executando Mineração de Dados	344
Interagindo com Sistemas Embarcados.....	345
Executando Tarefas Científicas.....	345
Executando Análises de Dados em Tempo Real	346

Capítulo 19: Dez Ferramentas Interessantes	347
Procurando Erros com Roundup Issue Tracker.....	348
Criando um Ambiente Virtual Usando VirtualEnv.....	349
Instalando sua Aplicação Usando PyInstaller.....	350
Construindo a Documentação de Desenvolvedor Usando o pdoc.....	351
Desenvolvendo o Código de Aplicação Usando o Editor Komodo.....	352
Depurando Sua Aplicação Usando o pydbg.....	353
Entrando em um Ambiente Interativo Usando o IPython.....	354
Testando Aplicações Python Usando o PyUnit.....	354
Arrumando Seu Código Usando o Isort.....	355
Criando Controle de Versão Usando Mercurial.....	355
Capítulo 20: Dez Bibliotecas que Você Precisa Conhecer	357
Desenvolvendo um Ambiente Seguro Usando o PyCrypto.....	358
Interagindo com Bancos de Dados Usando o SQLAlchemy.....	358
Vendo o Mundo Usando o Google Maps.....	359
Adicionando uma Interface de Usuário Gráfico Usando o TkInter.....	359
Fazendo uma Boa Apresentação de Dados Tabulares Usando o PrettyTable...	360
Otimizando Sua Aplicação com Som Usando o PyAudio.....	360
Manipulando Imagens usando o PyQtGraph.....	361
Localizando Informações Usando o IRLib.....	362
Criando um Ambiente Java Interoperacional Usando o Jpype.....	363
Acessando Recursos de Rede Local Usando o Twisted Matrix.....	364
Acessando Recursos de Internet Usando Bibliotecas.....	364
<i>Índice</i>.....	365

Introdução

Rápido! Qual linguagem de programação vai permitir a você escrever e executar aplicações em todas as plataformas populares por aí? Desiste? Sim, é o Python. O que é incrível sobre Python é que você realmente consegue escrever uma aplicação em uma plataforma e usá-la em qualquer outra que você precise. Ao contrário de outras linguagens de programação que prometeram fornecer independência da plataforma, Python realmente traz esta possibilidade. Neste caso, a promessa é tão boa quanto o resultado que você tem.

Python destaca a legibilidade do código e uma sintaxe concisa, que permite que você escreva aplicações usando menos linhas de código que outras linguagens de programação. E mais, devido à forma como Python trabalha, você o encontra sendo usado em várias áreas que estão cheias de não-programadores. Algumas pessoas veem Python como uma linguagem de script, mas ela é muito mais, realmente. (O capítulo 18 lhe mostrará uma breve ideia das aplicações que dependem de Python para funcionarem.)

Sobre Este Livro

Começando a Programar em Python Para Leigos é um guia para iniciar e executar Python em um instante. Você quer aprender a linguagem rapidamente para que possa tornar-se produtivo e usá-la para executar seu trabalho, que pode ser qualquer coisa. Diferentemente dos principais livros sobre o assunto, este mostra, logo no começo, o que faz Python ser diferente das outras linguagens e como ela pode ajudá-lo a executar bem um trabalho em um emprego que não seja de programação. Como resultado, você ganha entendimento do que você precisa desde o início, usando exemplos práticos e fazendo, em boa parte do tempo, tarefas realmente produtivas. Você até ganha ajuda na instalação de Python no seu sistema particular.

Quando você tem uma boa instalação, independente da plataforma que esteja usando, você começa com o básico e trabalha para incrementá-la. Quando você acabar de trabalhar com os exemplos deste livro, você saberá escrever programas simples e executar tarefas, como encaminhar um e-mail usando Python. Não, você não será um especialista, mas será capaz de usar Python para suprir necessidades específicas em seu ambiente de trabalho.

Para absorver os conceitos de forma mais fácil, este livro usa as seguintes convenções:

- ✔ O texto que você deve digitar, assim como aparece neste livro, está em negrito. A exceção é quando você está trabalhando em etapas: Porque cada passo está em negrito, porém o texto a digitar não está .
- ✔ Quando as palavras estiverem em *itálico*, como parte de uma sequência, você precisa trocá-las por informações válidas para você. Por exemplo, se você ver “Digite Seu Nome e teclle Enter,” você precisa trocar **Seu Nome** pelo seu nome real.
- ✔ Endereços de Web e códigos de programação, aparecem em Courier New. Se você estiver lendo a versão digital deste livro em um dispositivo conectado a internet, note que você pode clicar no endereço para visitar aquele website.
- ✔ Quando você precisa digitar sequências de comandos, você os vê separados por uma seta especial, assim: Arquivo ⇒ Novo Arquivo. Neste caso, você vai para o menu Arquivo primeiro e então seleciona a entrada do Novo Arquivo naquele menu. O resultado é que você vê um novo arquivo criado.

Exceto pelo material disponibilizado em www.altabooks.com.br, os sites indicados estão em inglês e a editora não se responsabiliza pelo conteúdo, material ou permanência de sites de terceiros.

Penso Que...

Você deve achar difícil acreditar que eu imagine qualquer coisa sobre você — afinal, eu nem te conheci ainda! Mesmo considerando que boa parte dessa imaginação não seja verdade, eu criei uma imagem sua para estabelecer um ponto inicial para o livro.

É importante que você esteja familiarizado com a plataforma que você quer usar, pois este livro não orienta qualquer direcionamento neste sentido. (O Capítulo 2 fornece instruções de instalação do Python para várias plataformas.) Para proporcionar o máximo de informação sobre Python, este livro não discute qualquer situação específica sobre plataforma. Você realmente precisa saber como instalar aplicações, usar aplicações e trabalhar normalmente com a plataforma escolhida antes de começar a trabalhar com este livro.

Este livro também supõe que você pode achar coisas na internet. Em todo lugar, existem numerosas referências de material online que vão aprimorar sua experiência de aprendizado. No entanto, estas fontes adicionadas são úteis somente se você realmente as encontrar e usar.

Ícones Utilizados Neste Livro

Conforme você lê este livro, você verá ícones nas margens que indicam material de interesse (ou não, como preferir). Esta seção descreve brevemente cada ícone neste livro.



Dicas são boas porque auxiliam você a economizar tempo ou executar alguma tarefa sem muito tempo extra de trabalho. As dicas neste livro são técnicas para economizar tempo ou indicações de recursos que você deveria experimentar para tirar o máximo proveito do Python.



Eu não quero parecer com pais aborrecidos ou algum tipo de maníaco, mas você deve evitar fazer qualquer coisa marcada com o ícone de advertência. Caso contrário, você pode descobrir que o seu programa serve, somente, para confundir os usuários que, então, vão evitar de trabalhar com ele.



Sempre que você ver este ícone pense em dica ou técnica avançada. Você pode achar chato estes pedaços de informações úteis ou elas podem conter a solução que você precisa para executar o programa. Pule estas poucas informações sempre que quiser.



Se você não se beneficiar de nada em um capítulo ou uma seção em particular, ao menos lembre-se sobre o material sinalizado por este ícone. Este texto normalmente contém uma parte essencial ou um pouco de informação que você deve conhecer para escrever programas em Python com sucesso.

Além Deste Livro

Este livro não é o fim da sua experiência com programação Python — é somente o início. Eu forneço conteúdo online para fazer este livro mais flexível e para melhor atingir suas necessidades. Desta forma, quando eu receber um e-mail seu, eu posso lhe fazer perguntas e lhe dizer como atualizações para Python ou suas bibliotecas associadas afetam o conteúdo do livro. De fato, você ganha acesso a todas estas adições especiais:

- ✔ **Folha de cola:** Você se lembra das pequenas anotações na escola para tirar uma melhor nota em um teste, não? Bem, uma folha de cola é, de alguma forma, parecida com aquilo. Ela vai lhe fornecer algumas anotações especiais sobre tarefas que você pode fazer com Python que nem todos os desenvolvedores conhecem. Você pode encontrar a folha de cola para este livro em www.altabooks.com.br, procurando pelo título do livro. Ela contém informações importantes, como os dez principais erros que os desenvolvedores cometem quando trabalham com Python e algumas sintaxes de Python que dão maiores problemas.

- ✔ **Artigos Online em Dummies.com:** Muitos leitores estavam pulando as páginas complementares do livro, então eu decidi fazer algumas mudanças. Agora você realmente tem uma boa razão para ler as páginas complementares, isto é, o conteúdo online. Cada página complementar tem um artigo associado a ela, fornecendo informações adicionais interessantes que não caberiam no livro. Você pode achar os artigos para este livro em <http://dummies.com/extras/beginningprogrammingwithpython>.
- ✔ **Atualizações:** Algumas vezes, mudanças acontecem. Por exemplo, eu posso não ter visto uma mudança iminente quando olhei dentro de minha bola de cristal enquanto escrevia este livro. No passado, isto simplesmente significaria que o livro ficaria desatualizado e menos útil, mas agora você pode achar atualizações para o livro em <http://dummies.com/extras/beginningprogrammingwithpython>. Em adição a essas atualizações, confira os posts do blog com respostas a questões de leitores e demonstrações de técnicas úteis relacionadas ao livro em <http://blog.johnmuellerbooks.com/>.
- ✔ **Arquivos Companheiros:** Ei! Quem realmente quer digitar todo o código neste livro? A maioria dos leitores preferiria gastar seu tempo de fato trabalhando por meio dos exemplos de códigos, em vez de digitá-los. Para sua sorte, o código-fonte está disponível para baixar, então tudo o que você precisa fazer é ler o livro para aprender técnicas de codificação Python. Cada exemplo do livro diz precisamente qual projeto de exemplo usar. Você pode encontrar esses arquivos em `arquivos` em www.altabooks.com.br, procurando pelo título do livro.

De Lá Para Cá, Daqui Para Lá

É hora de iniciar a sua aventura em Programação com Python! Se você é um completo novato em programação, você deve iniciar pelo Capítulo 1 e progredir com o livro em um ritmo que permita você absorver o máximo possível do material.

Se você é um novato e está na correria para entrar no mundo de Python o mais rápido possível, pode pular para o Capítulo 2, mas sabendo que você pode achar, mais tarde, alguns tópicos um pouco confusos. Ir para o Capítulo 3 é possível se você já tem o Python instalado, mas tenha certeza de, pelo menos, folhear o Capítulo 2 para que você saiba quais assuntos foram abordados ao longo deste livro.

Os leitores que têm alguma vivência com Python podem economizar algum tempo e pular diretamente para o Capítulo 5. Você sempre pode voltar aos capítulos anteriores quando for necessário ou tiver dúvidas. No entanto, é importante que você entenda como cada exemplo funciona antes de ir para o próximo. Cada um contém uma lição importante e você pode perder conteúdo vital se começar a pular muita informação.

Parte I

Iniciando com Python

getting started
with

Python

Nesta parte...

- ✔ Descubra tudo sobre programação e porque você precisa do Python para programar.
- ✔ Obtenha a sua própria cópia do Python e a instale no seu sistema.
- ✔ Trabalhe com o ambiente interativo que o Python proporciona.
- ✔ Crie a sua primeira aplicação usando Python.
- ✔ Entenda os benefícios de adicionar comentários à sua aplicação.

Capítulo 1

Falando com Seu Computador

Neste Capítulo

- ▶ Falando com seu computador
 - ▶ Criando programas para conversar com seu computador
 - ▶ Entendendo o que um programa faz e porque você quer criá-lo
 - ▶ Considerando porque você quer usar Python como sua linguagem de programação
-

Ter uma conversa com seu computador pode parecer como o roteiro de um filme de ficção científica. Afinal, os membros da *Enterprise* em *Star Trek* normalmente falavam com seus computadores. De fato, o computador frequentemente falava de volta. No entanto, com o crescimento do Siri da Apple (<http://www.apple.com/ios/siri/>) e outros softwares interativos, talvez você realmente não ache que uma conversa desse tipo seja tão inacreditável.



Perguntar ao computador uma informação é uma coisa, porém, fazer isto com instruções é outra. Este capítulo analisa porque você quer instruir o seu computador e que benefícios você ganha com isso. Você também descobre a necessidade de uma linguagem especial quando executa esse tipo de comunicação e o porquê de você querer usar Python. No entanto, o principal a se extrair deste capítulo é que programação é simplesmente uma forma de comunicação semelhante as outras formas que você já teve em seu computador.

Entendendo Por Que Você Quer Falar com Seu Computador

Falar com a máquina pode parecer estranho no início, mas isto é necessário porque um computador não pode ler sua mente — ainda. Mesmo que o computador realmente lesse sua mente, ele ainda estaria se comunicando com você. Nada pode ocorrer sem uma troca de informação entre a máquina e você. Atividades como

- Ler o seu e-mail
- Escrever sobre suas férias
- Achar o melhor presente no mundo

são todas exemplos de comunicação que ocorrem entre o computador e você. Que o computador, além disso, se comunica com outras máquinas ou pessoas para encaminhar tarefas que você quer, simplesmente estende a ideia básica que a comunicação é necessária para produzir qualquer resultado.

Na maioria dos casos, a comunicação acontece de uma maneira que é quase invisível, a não ser que você realmente pense sobre isto. Por exemplo, quando você visita uma sala de bate-papo online, você pode pensar que você está se comunicando com outra pessoa. No entanto, você está se comunicando com o seu computador, o seu computador está se comunicando com o computador de outra pessoa, por meio da sala de bate-papo, e o computador da outra pessoa está se comunicando com ela. A Figura 1-1 dá a você uma ideia do que realmente está acontecendo.

Figura 1-1:
A comunicação com o seu computador pode ser invisível, a não ser que você realmente pense sobre isto.



Observe a nuvem no centro da Figura 1-1. A nuvem pode conter qualquer coisa, mas você sabe que, pelo menos, ela contém outros computadores executando outras aplicações. Estes computadores tornam possível você e seus amigos baterem papo. Agora, pense como todo o processo parece fácil, quando você está usando a aplicação de conversa. Apesar de todas estas coisas estarem acontecendo no plano de fundo, parece que você está simplesmente conversando com seu amigo e o processo em si é invisível.

Sabendo que uma Aplicação é uma Forma de Comunicação

A comunicação entre computadores ocorre mediante o uso de aplicações. Você usa uma aplicação para responder um e-mail, outra para comprar mercadorias e uma outra para criar uma apresentação. Uma *aplicação* (algumas vezes chamada de *app*) fornece o meio para expressar ideias humanas ao computador de forma que ele possa definir as ferramentas necessárias para dar formato aos dados usados para a comunicação de maneiras específicas. Os dados usados para expressar o conteúdo de uma apresentação são diferentes dos dados usados para comprar um presente para sua mãe. A forma que você vê, usa e entende o dado é diferente para cada tarefa, então você deve usar diferentes aplicações para interagir com os dados de forma que o computador e você possam se entender.

É possível obter aplicações para atender a qualquer necessidade que você puder imaginar hoje. Na verdade, você provavelmente tem acesso a aplicações sobre as quais ainda não pensou a respeito. Os programadores estiveram ocupados criando milhões de aplicações de todos os tipos por muitos anos, então deve ser difícil entender o que você pode realizar criando algum novo método para falar com seu computador através de uma aplicação. A resposta se resume a pensar nos dados e como você quer interagir com eles. Alguns dados simplesmente não são comuns o suficiente para atrair a atenção de um programador ou você pode precisar dos dados em um formato que a aplicação normalmente não suporta. Então você não tem nenhuma forma de dizer ao computador o que você precisa, a não ser que você crie uma aplicação customizada para isso.

As próximas seções mostrarão aplicações a partir de uma perspectiva de trabalhos com dados únicos de uma maneira, de certa forma, especial. Por exemplo, você pode ter acesso à biblioteca do banco de dados de vídeos, mas não existe uma forma confortável de acesso para você. Os dados são únicos e as suas necessidades de acesso são especiais, então você pode querer criar uma aplicação que satisfaça tanto os dados quanto suas necessidades.

Pensando sobre procedimentos que você usa diariamente

Um *procedimento* é simplesmente um conjunto de passos que você segue para executar uma tarefa. Por exemplo, ao fazer uma torrada, você provavelmente deve usar esse procedimento:

1. Pegar o pão e a manteiga da geladeira.
2. Abrir a sacola do pão e pegar dois pedaços.
3. Remover a tampa da torradeira.

4. Colocar cada pedaço de pão em seu compartimento.
5. Colocar a alavanca da torradeira para baixo para iniciar a torração do pão.
6. Esperar que o processo de torração se complete.
7. Remover a torrada da torradeira.
8. Colocar a torrada no prato.
9. Colocar a manteiga na torrada.

Seu procedimento pode variar do apresentado aqui, mas é pouco provável que você coloque manteiga na torrada antes de colocá-la na torradeira. É claro que você realmente tem que remover o pão da embalagem antes de torr-lo (colocar o pão, a embalagem e tudo, dentro da torradeira, produziria resultados indesejados). A maioria das pessoas realmente nunca pensou sobre o procedimento de fazer torrada. Logo, você usa um procedimento como esse, mesmo que você não pense a respeito.



Os computadores não podem executar tarefas sem um procedimento. Você deve dizer ao computador quais passos seguir e a ordem em que devem ser executados. Quaisquer exceções à regra podem causar falhas. Todas estas informações (e mais) aparecem dentro de uma aplicação. Resumindo, uma aplicação é simplesmente um procedimento escrito usado para dizer ao computador o que e como fazer. Como você tem usado procedimentos a vida toda, o que você realmente precisa fazer é aplicar o conhecimento que já possui, de forma que o computador saiba realizar as tarefas específicas.

Escrevendo procedimentos

Quando eu estava no ensino fundamental, nossa professora pediu que escrevêssemos um artigo sobre fazer torrada. Depois de entregarmos nossos artigos, ela levou uma torradeira e alguns pães para a sala. Cada artigo foi lido e demonstrado. Nenhum de nossos procedimentos funcionaram como esperado, mas todos renderam resultados hilariantes. No meu caso, eu esqueci de dizer à professora para retirar o pão da embalagem, então ela tentou, com entusiasmo, enfiar o pedaço de pão, embalagem e tudo dentro da torradeira. Aquilo me paralisou. Escrever sobre procedimentos pode ser um tanto difícil, pois nós sabemos exatamente o que queremos fazer, porém, muitas vezes deixamos passos de fora — nós supomos que a outra pessoa também saiba precisamente o que fazer.

Muitas experiências na vida giram em torno de procedimentos. Pense sobre o *check-list* usado pelos pilotos antes do avião decolar. Sem um bom procedimento, o avião poderia cair. Aprender a escrever um bom procedimento leva tempo, mas é viável. Você deve tentar várias vezes antes de chegar a um procedimento que funcione completamente, mas você, eventualmente, criará um. Escrever procedimentos não é o suficiente, no

entanto — você também precisará testá-lo, usando alguém que não está familiarizado com a tarefa envolvida. Ao trabalhar com computadores, o computador é sua cobaia no assunto.

Vendo aplicações como se fossem qualquer outro procedimento

Um computador age como a professora do ensino fundamental no meu exemplo, na seção anterior. Quando você usa uma aplicação, você está usando um procedimento que define a série de passos que o computador deve executar para realizar tarefas que você tem em mente. Se você deixa de fora um passo, os resultados não serão o que você espera. O computador não saberá o que você quer dizer ou que você pretendia que ele executasse certas tarefas automaticamente. A única coisa que o computador saberá é que você forneceu um procedimento específico e ele precisa executá-lo.

Entendendo que os computadores entendem coisas literalmente

As pessoas eventualmente se acostumam aos procedimentos que você cria. Elas automaticamente compensam as deficiências do seu procedimento ou fazem anotações sobre coisas que você deixou de fora. Em outras palavras, as pessoas estabilizam os problemas dos procedimentos que você escreve.



Quando você começa a escrever programas de computador, você vai ficar frustrado porque os computadores executam tarefas precisamente e leem suas instruções literalmente. Por exemplo, se você diz ao computador que certo valor deve ser igual a 5, o computador vai olhar para o valor como, exatamente, 5. Um humano pode ver 4,9 e achar que o valor é bom o suficiente, mas o computador não enxerga desta forma. Ele vê o valor de 4,9 e decide que não é igual a 5. Resumindo, os computadores são inflexíveis, não-intuitivos e sem imaginação. Quando você escreve um procedimento para um computador, ele vai executar o que foi pedido de forma precisa absolutamente todas as vezes e nunca modificará seu procedimento ou notará que você queria que ele fizesse uma outra tarefa.