

# **Introdução à Programação com Python**

**Algoritmos e lógica de programação para iniciantes**

**Nilo Ney Coutinho Menezes**

Copyright © 2010 da Novatec Editora Ltda.

Todos os direitos reservados e protegidos pela Lei 9610 de 19/02/1998.

É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates

Revisão gramatical: Patrizia Zagni

Editoração eletrônica: Camila Kuwabata

Capa: Victor Bittow

ISBN: 978-85-7522-250-8

Histórico de impressões:

Novembro/2010 Primeira edição

Novatec Editora Ltda.

Rua Luís Antônio dos Santos 110

02460-000 – São Paulo, SP – Brasil

Tel.: +55 11 2959-6529

Fax: +55 11 2950-8869

E-mail: [novatec@novatec.com.br](mailto:novatec@novatec.com.br)

Site: [novatec.com.br](http://novatec.com.br)

Twitter: [twitter.com/novateceditora](https://twitter.com/novateceditora)

Facebook: [facebook.com/novatec](https://facebook.com/novatec)

LinkedIn: [linkedin.com/in/novatec](https://linkedin.com/in/novatec)

**Dados Internacionais de Catalogação na Publicação (CIP)**  
(Câmara Brasileira do Livro, SP, Brasil)

Menezes, Nilo Ney Coutinho  
Introdução à programação com Python :  
algoritmos e lógica de programação para  
iniciantes / Nilo Ney Coutinho Menezes. --  
São Paulo : Novatec Editora, 2010.

Bibliografia  
ISBN 978-85-7522-250-8

1. Python (Linguagem de programação para  
computadores) 2. Web sites - Desenvolvimento  
I. Título.

10-10552

CDD-005.133

Índices para catálogo sistemático:

1. Python : Desenvolvimento de aplicações Web :  
Programação : Processamento de dados  
005.133  
VC20101116

*Então, você quer aprender a programar?*

Programar computadores é uma tarefa que exige tempo e dedicação para ser corretamente aprendida. Muitas vezes não basta estudar e fazer os exemplos, mas deixar a mente se acostumar com a nova forma de pensar. Para muitas pessoas, o mais difícil é continuar gostando de programar. Muitas pessoas desistem nas primeiras dificuldades e não voltam mais a estudar. Outras são mais pacientes e aprendem a não se irritar com a máquina e a assumir seus erros.

Para não sofrer dos males de quem não aprendeu a programar, você precisa responder algumas perguntas antes de começar:

1. Você quer aprender a programar?
2. Como está seu nível de paciência?
3. Quanto tempo você pretende estudar?
4. Qual o objetivo de programar?

## **1.1 Você quer aprender a programar?**

Responda a essa questão, mas pense um pouco antes de chegar à resposta final. A maneira mais difícil de aprender a programar é não querer programar. Essa vontade deve vir de você, e não de um professor ou de um amigo. Programar é uma arte e precisa de dedicação para ser dominada. Como tudo que é desconhecido, é muito difícil quando não entendemos, mas se torna mais simples à medida que aprendemos.

Se você já decidiu aprender a programar, siga para a próxima parte. Se ainda não se convenceu, continue lendo. Programar pode se tornar um novo *hobby* e até mesmo uma profissão. Se você estuda computação, precisa saber programar.

Isso não significa que você será um programador por toda a vida, ou que isso limitará seu crescimento dentro da área de informática. Uma desculpa que já ouvi muitas vezes é *“eu sei programar, mas não gosto”*. Vários alunos de computação terminam seus cursos sem saber programar. Isto é, sem realmente saber programar. Programar é como andar de bicicleta, você não esquece, mas só aprende fazendo. Ao trocar de uma linguagem de programação para outra, se você realmente aprendeu a programar, terá pouca dificuldade para aprender a nova linguagem. Diferentemente de saber programar, a sintaxe de uma linguagem de programação é esquecida muito facilmente. Não pense que saber programar é decorar todos aqueles comandos, parâmetros e nomes estranhos. Programar é saber utilizar uma linguagem de programação para resolver problemas, ou seja, saber expressar uma solução usando uma linguagem de programação.

## 1.2 Como está seu nível de paciência?

Seja paciente.

Outro caso para quem estuda programação é querer fazer coisas difíceis logo de início.

Qual será seu primeiro programa? Um editor de textos? Uma planilha eletrônica? Uma calculadora?

Não! Será algo bem mais simples... Como somar dois números.

É, isso mesmo: somar dois números!

Com o tempo, a complexidade e o tamanho dos programas aumentarão.

Seja paciente.

Programar exige muita paciência e, principalmente, atenção a detalhes. Uma simples vírgula no lugar de um ponto ou aspas esquecidas podem arruinar seu programa. No início é comum perder a calma ou mesmo se desesperar até aprender a ler o que realmente escrevemos em nossos programas. Nessas horas, paciência nunca é demais. Leia novamente a mensagem de erro ou pare para entender o que não está funcionando corretamente. E nunca pense que o computador está contra você, nem culpe o dia ou o destino.

Seja paciente.

## 1.3 Quanto tempo você pretende estudar?

Pode-se aprender a programar em poucas horas. Se você é o sujeito que programa o micro-ondas de sua tia, abre vidros de remédio lendo as instruções da tampa ou que brincou de Lego, programar é seu segundo nome.

Todos nós já programamos algo na vida, nem que seja ir ao cinema no sábado. A questão é: quanto tempo você dedicará para aprender a programar computadores?

Como tudo na vida, nada de exageros. Na realidade, isso varia muito de pessoa para pessoa. Algumas rendem mais estudando em grupo. Outras gostam de ter aulas.

O importante é incluir o estudo da programação em seu estilo preferido. Não tente aprender tudo ou entender tudo rapidamente. Se isso ocorrer, parabéns, há muito pela frente. Caso contrário, relaxe. Se não entender na segunda tentativa, pare e volte a tentar amanhã.

Quando encontrar um problema, tenha calma. Veja o que você escreveu. Verifique se você entende o que está escrito. Um erro comum é querer programar sem saber escrever as instruções. É como querer escrever sem saber falar.

Inicie com sessões de uma ou no máximo duas horas por dia. Depois ajuste esse tempo a seu ritmo.

## 1.4 Qual o objetivo de programar?

Se você não precisa programar para seu trabalho ou estudo, vejamos algumas outras razões:

### 1.4.1 Escrever páginas web

Hoje, todos estão expostos à web, à Internet e a seus milhares de programas. A web só funciona porque permite a publicação de páginas e mais páginas de textos e imagens com apenas um editor de textos. A mais complexa página que você já acessou é um conjunto de linhas de texto reunidos para instruir um programa, o navegador (*browser*), sobre como apresentar seu conteúdo.

## 1.4.2 Acertar seu relógio

Você conhece aquelas pessoas que nunca aprenderam a acertar seus relógios? Consigo me lembrar de várias delas...

Seguir instruções é muito importante para tarefas tão simples quanto essas. A sequência de passos para ajustar as horas, minutos e até mesmo a data de seu relógio podem ser encaradas como um programa. Normalmente, aperta-se o botão de ajuste até que um número comece a piscar. Depois, você pode usar um botão para mudar a hora ou ir direto para o ajuste dos minutos. Isso se repete até que você tenha ajustado todos os valores como segundos, dia, mês e, às vezes, o ano.

## 1.4.3 Aprender a usar mapas

Já se perdeu em uma cidade estranha? Já fez uma lista de passos para chegar a algum lugar? Então você já programou antes. Só de procurar um mapa você já mereceria um prêmio. Ao traçar um caminho de onde você está até onde deseja chegar, você normalmente relaciona uma lista de ruas ou marcos desse caminho. Normalmente é algo como passar três ruas à esquerda, dobrar à direita, dobrar à esquerda... Ou algo como seguir reto até encontrar um sinal de trânsito ou um rio. Isso é programar. Seguir seu programa é a melhor forma de saber se o que você escreveu está correto ou se você agora está realmente perdido.

## 1.4.4 Mostrar para seus amigos que você sabe programar

Esta pode ser a razão mais complicada. Vamos ver isso como um subproduto do aprendizado, e não seu maior objetivo. Se essa for a razão de aprender a programar, é melhor continuar lendo e arranjar outra.

Programar é um esforço para você realizar algo. É uma tarefa que exige dedicação e que traz muita satisfação pessoal. Seus programas podem ser legais, mas aí já serão seus programas e não somente você.

## 1.4.5 Parecer estranho

Agora estamos conversando. Entender o que milhares de linhas de programa significam pode realmente gerar uma fama como essa entre os leigos. Se esse é seu objetivo, saiba que há maneiras mais fáceis, como parar de tomar banho, deixar as unhas crescerem, ter um cabelo de cor laranja ou roxa, parecer um roqueiro sem nunca ter tocado em uma banda etc.

Embora boa parte dos programadores que conheço não seja exatamente o que eu classifico de 100% normal, ninguém o é.

Saber programar não significa que você seja louco ou muito inteligente. Saber programar também não significa que você não seja louco ou que não seja muito inteligente. Imagine aprender a programar como qualquer outra coisa que você já aprendeu.

Um dia, sua mente poderá ter pensamentos estranhos, mas, quando esse dia chegar, pode apostar que você saberá.

### 1.4.6 Entender melhor como seu computador funciona

Programando você pode começar a entender por que aquela operação falhou ou por que o programa simplesmente fechou sem nem mesmo avisar o motivo.

Programar também pode ajudá-lo a utilizar melhor sua planilha ou editor de textos. O tipo de pensamento que se aprende programando servirá não só para fazer programas, mas também para usar programas.

### 1.4.7 Cozinhar

Uma vez precisei cozinhar um prato, mas as instruções estavam escritas em alemão. Não sei nada de alemão. Peguei o primeiro dicionário e comecei a traduzir as principais palavras. Com as palavras traduzidas, tentei entender o que deveria realmente fazer.

Naquela noite, o jantar foi apenas uma sopa instantânea. Receitas podem ser vistas como programas. E como programas, só é possível segui-las se você entender aquilo que foi escrito.

A simples sequência de instruções não ajuda uma pessoa que não entenda seus efeitos.

Para algumas pessoas, programar é mais fácil que aprender alemão (ou qualquer outro idioma estrangeiro). E como qualquer outra língua, não se aprende apenas com um dicionário.

Idiomas humanos são ricos em contextos, e cada palavra costuma ter múltiplos significados. A boa notícia: linguagens de programação são feitas para que máquinas possam entender o que ali foi representado. Isso significa que entender um programa é muito fácil, quase como consultar um dicionário. Outra boa notícia é que a maioria das linguagens contém conjuntos pequenos de “palavras”.

### 1.4.8 Salvar o mundo

Uma boa razão para aprender a programar é salvar o mundo. Isso mesmo!

Todos os dias, milhares de quilos de alimento são desperdiçados ou não chegam onde deveriam por falta de organização. Programando você pode ajudar a criar sistemas e até mesmo programas que ajudem outras pessoas a se organizar.

Outra boa ação é ajudar um projeto de software livre. Isso permitirá que muitas pessoas que não podem pagar por programas de computador se beneficiem deles sem cometer crime algum.

### 1.4.9 Software livre

Aliás, você tem licença de uso para todos os seus programas?

Se a resposta é não, saiba que programadores aprendem Linux e outros sistemas operacionais muito mais rapidamente. Também conseguem tirar maior proveito desses sistemas porque conseguem programá-los.

Se não existe, crie. Se é ruim, melhore.

Separar a programação em um mundo à parte pode ser sua primeira ideia estranha entre muitas.

Criar mundos dentro de programas e computadores pode ser a segunda.

## 1.5 Por que Python?

A linguagem de programação Python é muito interessante como primeira linguagem de programação devido à sua simplicidade e clareza. Embora simples, é também uma linguagem poderosa, podendo ser usada para administrar sistemas e desenvolver grandes projetos. É uma linguagem clara e objetiva, pois vai direto ao ponto, sem rodeios.

Python é software livre, ou seja, pode ser utilizada gratuitamente, graças ao trabalho da Python Foundation<sup>1</sup> e de inúmeros colaboradores. Você pode utilizar Python em praticamente qualquer arquitetura de computadores ou sistema operacional, como Linux<sup>2</sup>, FreeBSD<sup>3</sup>, Microsoft Windows ou Mac OS X<sup>4</sup>.

---

1 <http://www.python.org>

2 <http://www.kernel.org> ou <http://www.ubuntu.com> para obter o pacote completo

3 <http://www.freebsd.org>

4 <http://www.apple.com/macosex>



Python vem crescendo em várias áreas da computação, como inteligência artificial, banco de dados, biotecnologia, animação 3D, aplicativos móveis (celulares), jogos e mesmo como plataforma web. Isso explica porque Python é famosa por ter “*batteries included*”, ou seja, baterias inclusas, fazendo referência a um produto completo que pode ser usado prontamente (quem nunca ganhou um presente de Natal que veio sem pilhas?). Hoje é difícil encontrar uma biblioteca que não tenha *bindings* (versão) em Python. Esse fato torna o aprendizado da linguagem muito mais interessante, uma vez que aprender a programar em Python é poder continuar a utilizar os conhecimentos adquiridos mesmo depois de aprender a programar para resolver problemas reais.

Uma grande vantagem de Python é a legibilidade dos programas escritos nessa linguagem. Outras linguagens de programação utilizam inúmeras marcações, como ponto (.) ou ponto e vírgula (;), no fim de cada linha, além dos marcadores de início e fim de bloco como chaves ({ }) ou palavras especiais (*begin/end*). Esses marcadores tornam os programas um tanto mais difíceis de ler e felizmente não são usados em Python. Veremos mais sobre blocos e marcações nos capítulos seguintes.

Outro bom motivo para aprender Python é poder obter resultados em pouco tempo. Como Python é uma linguagem completa, contando com bibliotecas para acessar bancos de dados, processar arquivos XML, construir interfaces gráficas e mesmo jogos, podemos utilizar muitas funções já existentes escrevendo poucas linhas de código. Isso aumenta a produtividade do programador, pois ao utilizarmos bibliotecas usamos programas desenvolvidos e testados por outras pessoas. Isso reduz o número de erros e permite que você se concentre realmente no problema que quer resolver.

Vejamos um pequeno programa escrito em Python na listagem 1.1.

### Listagem 1.1 – Programa Olá Mundo

```
print ("Olá!")
```

A listagem do programa 1.1 tem apenas uma linha de código. A palavra **print** é uma função utilizada para enviar dados para a tela do computador. Ao escrevermos **print** ("Olá"), ordenamos ao computador que exiba o texto “Olá!” na tela. Veja o que seria exibido na tela ao se executar esse programa no computador:

```
Olá!
```

Observe que as aspas (“”) não aparecem na tela. Esse é um dos detalhes da programação: precisamos marcar ou limitar o início e o fim de nossas mensagens com um símbolo, nesse caso, aspas. Como podemos exibir praticamente qualquer texto na tela, as primeiras aspas indicam o início da mensagem, e as seguintes, o fim. Ao programar, não podemos esquecer as limitações do computador. Um computador não interpreta textos como seres humanos. A máquina não consegue diferenciar o que é programa ou mensagem. Se não utilizarmos as aspas, o computador interpretará nossa mensagem como um comando da linguagem Python, gerando um erro.

O interpretador Python é uma grande ferramenta para o aprendizado da linguagem. O interpretador é o programa que permite digitar e testar comandos escritos em Python e verificar os resultados instantaneamente. Veremos como utilizar o interpretador na seção 2.2.

A linguagem Python foi escolhida por simplificar o trabalho de aprendizado e fornecer grande poder de programação. Como é um software livre, disponível praticamente para qualquer tipo de computador, sua utilização não envolve a aquisição de licenças de uso, muitas vezes a um custo proibitivo.

**Bem-vindo ao mundo da programação!**