

Engenheiros que dizem: “Nih!”

Giovani Zanelatto, Victor Bergossi,
Gabriel Selow, Leo, 41 9687-1698, 41
9846-7571



Apostila de Python Para Iniciantes

UTFPR

Engenheiros que dizem: “Nih!”

Giovani Zanelatto, Victor Bergossi,
Gabriel Selow, Leo, 41 9687-1698, 41
9846-7571



Apostila de Python Para Iniciantes

UTFPR

© 2020 Engenheiros que dizem: “Nih!”
Giovani Zanelatto, Victor Bergossi, Gabriel Selow, Leo, 41 9687-1698, 41 9846-7571 & UTFPR
Qualquer parte desta publicação pode ser reproduzida, desde que citada a fonte.

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

Zanelatto, Giovani.
Apostila de Python Para Iniciantes. / Engenheiros que dizem: “Nih!”
Giovani Zanelatto, Victor Bergossi, Gabriel Selow, Leo, 41 9687-1698, 41 9846-7571 . – Curitiba:
UTFPR 2020.

Bibliografia.
ISBN XXXX-XXXX-XX.

1. Programas de computador. 2. Python. 3. Fundamentos. 4. Receitas.

Lista de ilustrações

Lista de tabelas

Sumário

Introdução	7
0.1 Mas o que é Python?	7
0.2 Mas por que Python?	8
1 Programação	9
1.1 Filosofia de Programação	9
1.2 Sequencia lógica de ações	10
1.3 Instalando o que é preciso para Python	11
2 Elementos básicos de Python	13
2.1 Primeiro programa	13
 Referências	 15

Introdução

PHYTON foi criado em torno de 1990 por Guido van Rossum. O nome Python foi uma homenagem a um seriado de comédia da rede BBC intitulado “Monty Python’s Flying Circus”(LUTZ, 2006). Curiosamente muitos autores e desenvolvedores preferem ser associados a cobras e serpentes do que a um grupo de humor, e cada vez menos se observam referências ao programa de humor na literatura.

Desde sua estréia pública em 1991, Python tem crescido e se atualizado; passando de uma linguagem de script para uma ferramenta bem estabelecida e amplamente utilizada por todo o globo. *Elaborar mais isto aqui, encher linguagem com o que for informação bacana, mesmo que inútil, mas sem ser chato ou se alongar demais...*

Esta apostila tem como objetivo iniciar os que ainda não programam em linguagem alguma, e capacitá-los a compreender e executar uma série de receitas para resolução de problemas práticos. Referências básicas e avançadas serão indicadas para o estudante que quiser se aprofundar mais ou estiver curioso para saber o que há debaixo do capô.

0.1 Mas o que é Python?

Python é uma linguagem de computador interpretada. Isto significa que o computador, para executar um programa escrito em Python, precisa necessariamente ter um interpretador de Python instalado. Linguagens compiladas não necessitam disto, embora eventualmente contem com a presença de bibliotecas como as do sistema operacional.

O fato de Python ser interpretado não deve ser interpretado como sinal de fraqueza. Linguagens compiladas são, de fato, mais versáteis e podem ser otimizadas a níveis que o Python não compete. Entretanto o Python contém uma quantidade enorme de módulos, escritos em linguagens compiladas e otimizados, de tal forma que boa parte das operações para as quais um módulo é utilizado, o interpretador não penaliza a velocidade.

Portanto, antes de começar a desbravar o mundo da programação de computadores, necessitaremos preparar o terreno e instalar alguma distribuição de Python. Muitas estão disponíveis, algumas com mais ou menos capacidades. Indicamos uma distribuição em

particular pois é mais fácil seguir o tutorial enquanto não precisamos ficar perdendo tempo com questões do tipo: “onde está o terminal de saída?”.

0.2 Mas por que Python?

- Python é fácil de aprender.
- Python é versátil.
- Python é poderoso.

Elaborar mais a lista de vantagens...

Programação

1.1 Filosofia de Programação

Apesar de toda sua complexidade, um computador é tão inteligente quanto o programa que roda nele. A grosso modo podemos dizer que um computador não passa de um “estúpido muito rápido”. Um computador não é capaz, por si só, de descobrir o que o usuário quer que ele faça. Ao ligarmos um computador um pequeno programa na placa-mãe do computador faz uma checagem dos componentes instalados e inicia a execução de um programa no dispositivo de “boot”, que carrega o que normalmente é o sistema operacional, que passa a ter o controle do sistema todo.

O sistema operacional, seja ele qual for, estabelece a interface entre o usuário e o computador e também de interface entre programas e periféricos. Programas que executam sobre o sistema operacional não precisam saber qual a taxa de atualização dos pente de memória A, B ou C. Cabe ao sistema operacional servir de interface entre um programa que manda desenhar uma figura na tela e a efetiva associação de valores na memória de vídeo que faz com que a imagem de fato apareça desenhada na tela.

Cabe aqui uma ilustração do programa com uma figura e uma tela com a figura impressa.

Para organizarmos nossa contabilidade mensal com um computador, não basta ligá-lo e colocar a pilha de extratos e informações sobre o teclado ou monitor. Ligamos o computador, que se inicializa e carrega um sistema operacional e então escolhemos e executamos um programa aonde colocamos os dados e com o qual fazemos operações matemáticas e armazenamos o resultado. Programas complexos como o Microsoft Word, Windows 10, Ubuntu, IOS e Android, podem ter centenas de milhares de linhas de código. Programas não precisam ser complexos para serem úteis. Muito se pode fazer com poucas linhas de código, organização e imaginação.

Com Python, podemos criar programas que acessam planilhas Excel e executam operações nestes valores sem a necessidade de invocar comandos Excel. Podemos também criar arquivos Excel para testar se uma planilha realmente está funcionando conforme programado.

Quando falamos em programas de computador, falamos em um conjunto lógico de instruções que instruem os vários macaquinhos do processador a fazer o que queremos que eles façam. A linguagem de programação estabelece a interface entre o que sabemos escrever e o que os macaquinhos do processador compreendem.

O sucesso e a eficiência de um programa dependem essencialmente da habilidade lógica de quem define as operações necessárias e sua ordem sequencial. Bons programadores procuram fazer códigos eficientes, o que é aprendido na prática com a prática frequente da programação. (NORTON et al., 2005)

1.2 Sequencia lógica de ações

If we took the bones out, it wouldn't be crunchy, would it? I told you to lay off the beans, you whore! At this time, a friend shall lose his friend's hammer and the young shall not know where lieth the things possessed by their fathers that their fathers put there only just the night before, about eight o'clock. Well, had I got as far as the penis entering the vagina? Pero las llamas son peligrosas. Si usted ve una llama donde hay gente nadando, usted gritar: ¡Cuidado! ¡Llamas! And Dinsdale says 'I hear you've been a naughty boy, Clement', and he splits me nostrils open, saws me leg off and pulls me liver out. And I tell him 'My name's not Clement', and then he loses his temper and nails my head to the floor. Well, I think I should point out first, Brian, in all fairness, we are not, in fact, the rescue committee. However, I have been asked to read the following prepare statement on behalf of the movement. "We the People's Front of Judea, brackets, officials, end brackets, do hereby convey our sincere fraternal and sisterly greetings to you, Brian, on this, the occasion of your martyrdom."

Oh, waiter! This conversation isn't very good. I'm not a roman mum, I'm a kike, a yid, a heebie, a hook-nose, I'm kosher mum, I'm a Red Sea pedestrian, and proud of it! Well, er, yes Mr. Anchovy, but you see your report here says that you are an extremely dull person. You see, our experts describe you as an appallingly dull fellow, unimaginative, timid, lacking in initiative, spineless, easily dominated, no sense of humour, tedious company and irrepressibly drab and awful. And whereas in most professions these would be considerable drawbacks, in chartered accountancy, they're a positive boon. Shut your festering gob, you tit! Your type really makes me puke you vacuous, toffy-nosed, malodorous pervert!

It's not pining, it's passed on! This parrot is no more! It has ceased to be! It's expired

and gone to meet its maker! This is a late parrot! It's a stiff! Bereft of life, it rests in peace! If you hadn't nailed it to the perch, it would be pushing up the daisies! It's metabolic processes are now history! He's off the twig! He's kicked the bucket, he's shuffled off the mortal coil, rung down the curtain and joined the choir invisible. This is an ex-parrot!

1.3 Instalando o que é preciso para Python

Elementos básicos de Python

2.1 Primeiro programa

```
def f(x):  
    return x
```


Referências

LUTZ, M. *Programming Python, 3rd Edition*. [S.l.]: O'Reilly, 2006. Citado na página [7](#).

NORTON, P. et al. *Beginning Python*. [S.l.]: Wrox, 2005. Citado na página [10](#).