Escola Nacional de Ciências Estatísticas

# Apostila de Introdução à Programação
# Prof. Eduardo Corrêa

## Capítulo I – Introdução

## Sumário

# Capítulo I. Introdução

A disciplina **Introdução à Programação** tem por objetivo capacitar os alunos a desenvolver algoritmos e programas de computadores com o emprego de técnicas básicas de programação estruturada, manipulação de estruturas de dados e manipulação de arquivos. A linguagem adotada no curso é **Python**, que além de ser muito popular na área de Estatística, é considerada uma das mais simples e didáticas linguagens de programação já criadas.

Este capítulo introduz o conceito de programa de computador e, em seguida, apresenta os passos necessários para você instalar o ambiente Python em seu computador, mostrando ainda a receita básica que você poderá adotar nesta disciplina para criar e executar os seus programas. O capítulo encerra apresentado um exemplo comentado de programa Python contendo instruções de entrada e saída.

## I.1. O que é um programa de computador?

Para que um computador possa realizar qualquer tarefa solicitada por um ser humano é preciso que ele receba instruções. Um programa de **programa de computador** consiste em um conjunto de instruções que conseguem ser entendidas e executadas por um computador. Ao longo de seu curso de estatística, você desenvolverá programas para extrair estatísticas sobre bases de dados, gerar gráficos, produzir tabulações e construir modelos de *machine learning*, entre outras coisas. Seja qual for o objetivo do seu programa, de uma maneira geral ele funcionará de acordo com o fluxo apresentado na Figura 1.


**Figura 1. Um programa de computador recebe dados de entrada e produz uma saída**

A figura ilustra que um programa toma algum valor, ou conjunto de valores como **entrada** (estes valores podem, por exemplo, ser digitados pelo usuário via teclado ou podem estar disponibilizados em um arquivo ou banco de dados). Em seguida, o programa **executa as tarefas** (instruções) implementadas pelo programador levando em conta o que foi passado como entrada. Como resultado, será produzido algum valor ou conjunto de valores que serão apresentados ao usuário. Estes valores para apresentação são chamados de **saída** do programa. Tipicamente, a saída é mostrada no vídeo, mas também pode ser direcionada para a impressora. Em programas de análise estatística e ciência de dados, também é muito comum que a saída seja direcionada para um arquivo (como, por exemplo, uma planilha ou um gráfico), que posteriormente poderá ser aberto e ter o seu conteúdo exibido no vídeo. A saída de um programa é sempre função da entrada (ou seja, entradas diferentes resultam em saídas diferentes).

A seguir são apresentados dois exemplos de programas e suas respectivas entradas e saídas.

**Exemplo 1** - Uma universidade possui um sistema onde o aluno pode digitar a sua matrícula e senha para consultar a sua situação (notas e faltas) em cada uma das disciplinas em que ele se inscreveu.

**Entrada**: matrícula e senha.

**Saída**: planilha contendo as notas e faltas do aluno por disciplina (exibida na tela do computador ou celular).

**Exemplo 2** - Um sistema de reconhecimento facial, que analisa a imagem do rosto de uma pessoa e informa se essa pessoa é um adulto ou uma criança.


**Entrada**: foto do rosto (pode ser obtida com uma câmera)

**Saída**: programa informa se é 'criança' ou 'adulto' (no exemplo acima, o sistema informa que o rosto é de uma criança)

O ser humano responsável por escrever um programa é chamado de **programador**. Por sua vez, a pessoa que adquire e usa um programa é chamada de **usuário**. O processo de escrita de um programa de computador também é conhecido como implementação ou desenvolvimento.

## I.2. O que é Python?

Python é uma linguagem de programação de **propósito geral**, o que significa que ela pode ser empregada nos mais diferentes tipos de projetos, variando desde aplicações Web até sistemas de inteligência artificial. A linguagem foi criada no ano de 1991, tendo como principal filosofia priorizar a construção de programas simples e legíveis – aquilo que os *pythonistas* (programadores Python) chamam de programas "bonitos" sobre a construção de programas que, ainda que velozes, sejam complicados e pouco legíveis (os ditos programas "feios"). Devido a esta característica, nas últimas duas décadas a linguagem tem sido largamente adotada para o ensino de programação em cursos de nível superior. Além de versátil, a linguagem é muito poderosa, possuindo um grande número de pacotes para ciência de dados (pacotes de estatística, matemática, mineração de dados, inteligência artificial, aprendizado de máquina, cálculo numérico, etc.). Por isso, vem sendo utilizada por um número cada vez maior de profissionais da área de Estatística. A próxima seção fornece as instruções para você instalar o ambiente Python em seu computador.

## I.3 Download e Instalação do Python

O texto a seguir apresenta os passos necessários para instalar a linguagem **Python**, obtendo o instalador a partir do website oficial do Python (www.python.org). Em seguida, mostramos como instalar o ambiente de desenvolvimento **Thonny**, que será utilizado em nossa disciplina para facilitar o desenvolvimento, teste e execução de seus programas. Após a instalação do Python e do Thonny, mostramos como criar o seu primeiro programa. Assim você poderá conhecer a receita básica a ser utilizada para criar e executar os demais programas que precisará elaborar ao longo do curso.

### (A). Fazendo o download e Instalação do Python

Para realizar a instalação do Python em sua máquina execute os seguintes passos:

- PASSO 1: **acessar o site** www.python.org e clicar em Downloads (Figura 2).

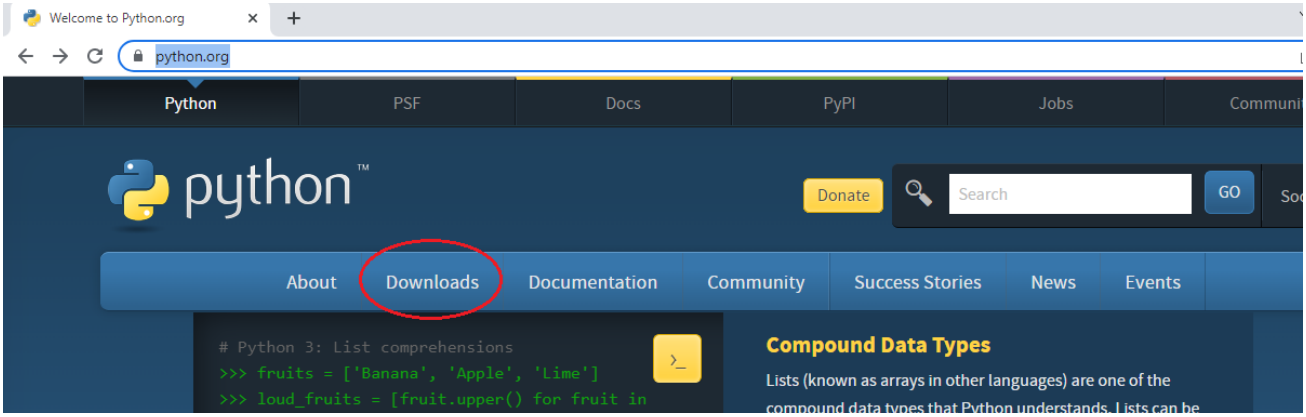


**Figura 2. Web site do Python**

- PASSO 2: clicar no botão para efetuar o download do arquivo de instalação da versão mais recente do Python, que no momento da elaboração desta apostila era **3.10.4** (Figura 3).


**Figura 3. Botão para download do Python**

- PASSO 3: após o fim do download, execute o arquivo baixado. Assim, o Python começará a ser instalado em sua máquina. **É IMPORTANTE** marcar a opção "Add Python 3.10 to PATH", que vem desmarcada por padrão (veja a parte destacada na Figura 4).

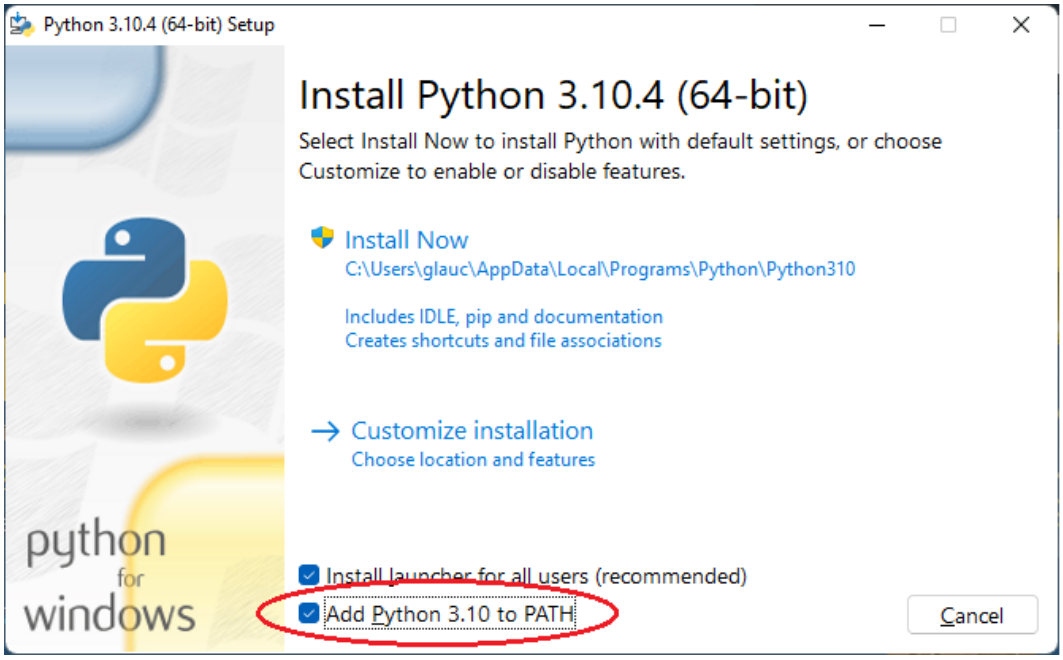


**Figura 4. Tela inicial do instalador do Python – é recomendado marcar a opção Add Python 3.10 to PATH**

- PASSO 4: **este passo é OPCIONAL, ou seja, você não precisa executar.** Caso você deseje, poderá modificar o caminho da instalação clicando em "Customize Installation" (Figura 5), depois no botão "Next" (Figura 6) e alterando a pasta indicada em "Customize install location" (Figura 7). Veja que eu mudei para "C:\Python" (dependendo do computador que você utilizar, alterar o caminho poderá não ser permitido). Após alterar clique no botão "Back" e, na tela seguinte, em "Back" de novo para retornar a tela principal.

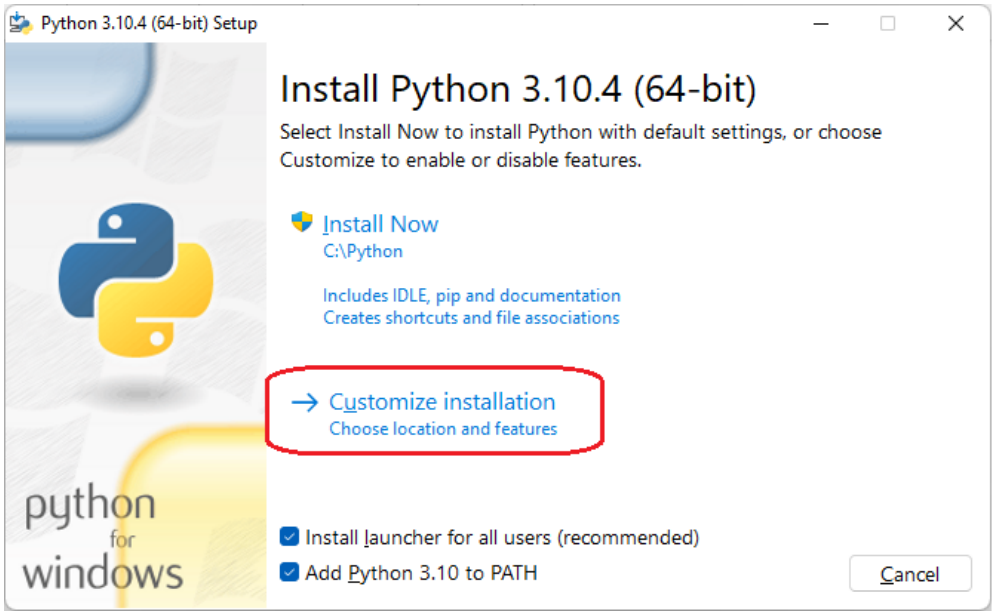


**Figura 5. Alterando a pasta onde o Python será instalado (PARTE 1)**

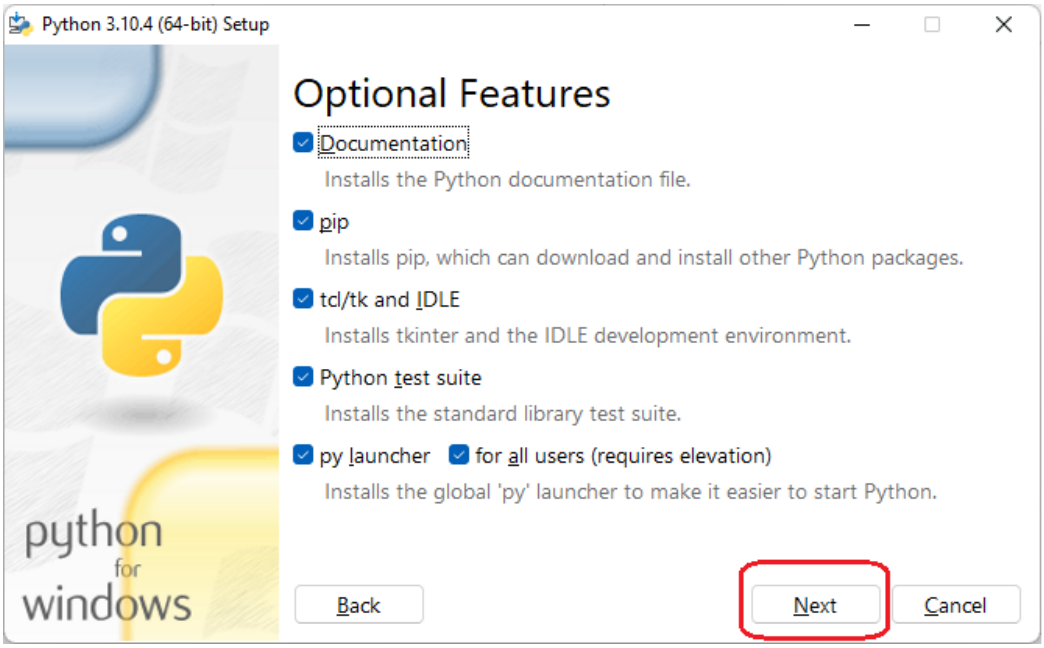


**Figura 6. Alterando a pasta onde o Python será instalado (PARTE 2)**

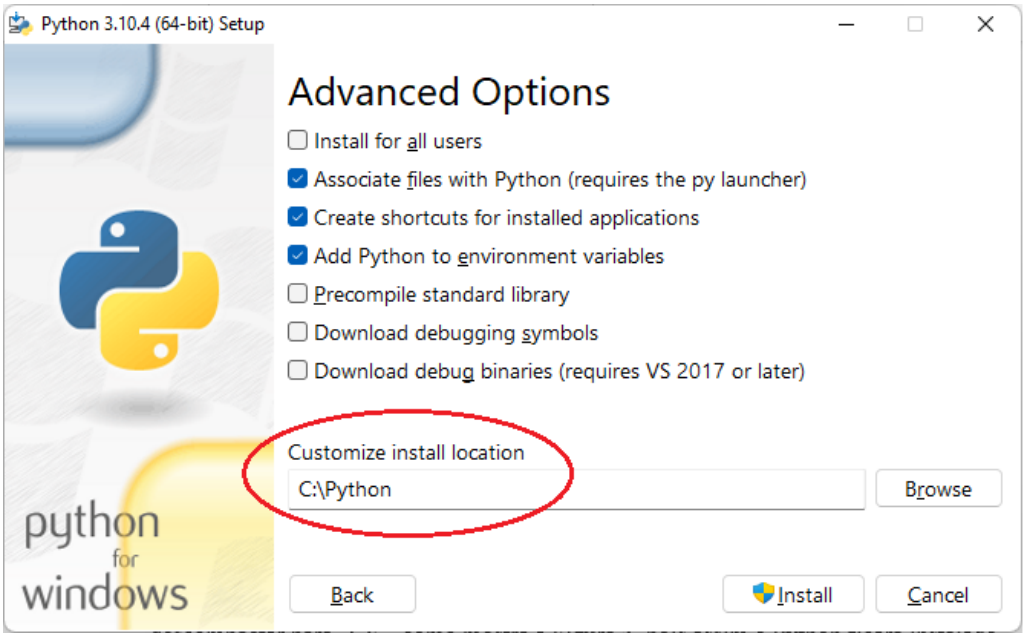


**Figura 7. Alterando a pasta onde o Python será instalado (PARTE 3)**

- PASSO 5: Clique em "Install Now" (Figura 8). Após alguns minutos a instalação será concluída (Figura 9).

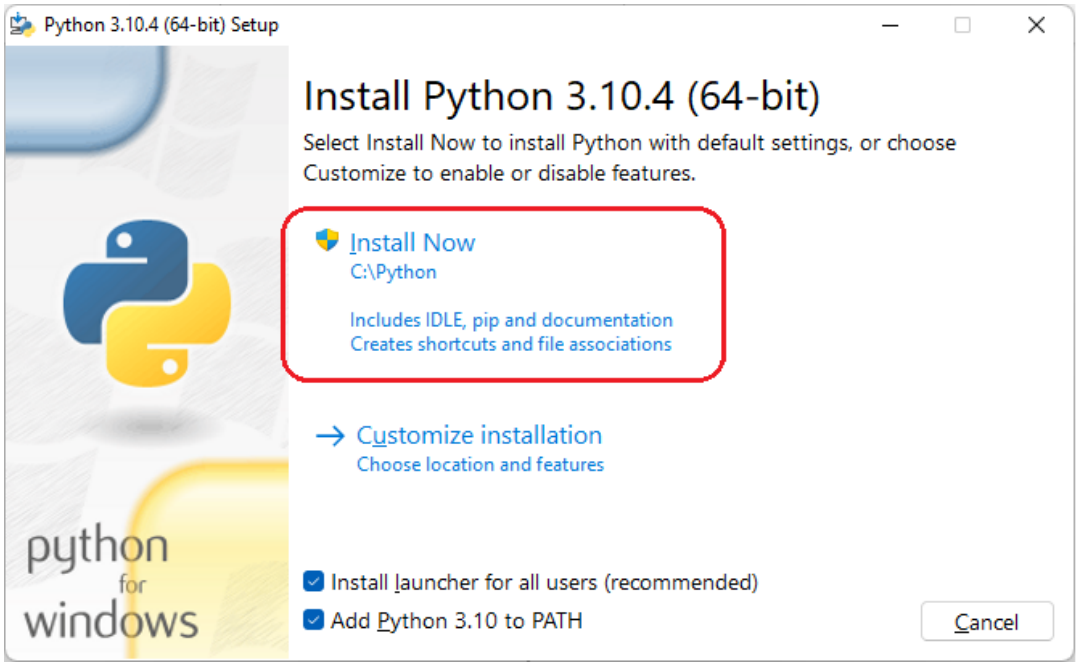


**Figura 8. Iniciando a instalação**

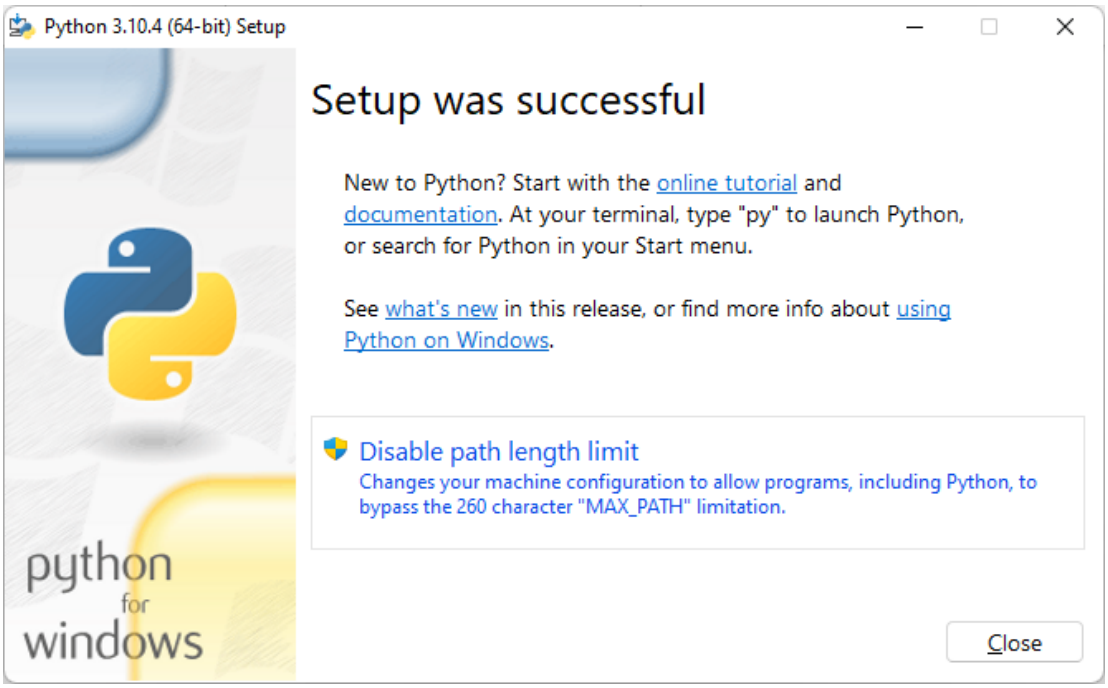


**Figura 9. Instalação finalizada**

**(B). Testando a Instalação do Python**

- PASSO 1: Abra uma janela de prompt de comando. Para fazer isso no Windows, digite a tecla WINDOWS junto com a tecla R para abrir a janela "Executar". Em seguida digite **cmd** e clique no botão OK (Figura 10).

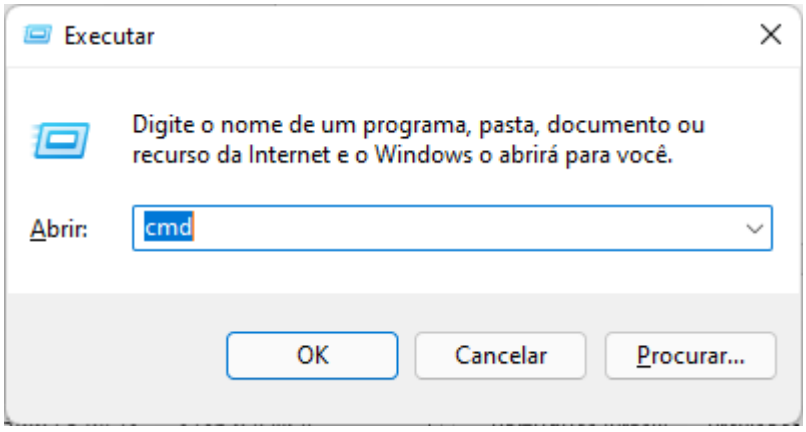


**Figura 10. Executando o prompot de comando**

- PASSO 2: No prompt de comando, digite a instrução a seguir:

```
python --version.
```

Se aparecer a mensagem **Python 3.10.4**, como na Figura 11, é porque a instalação foi realizada com sucesso.

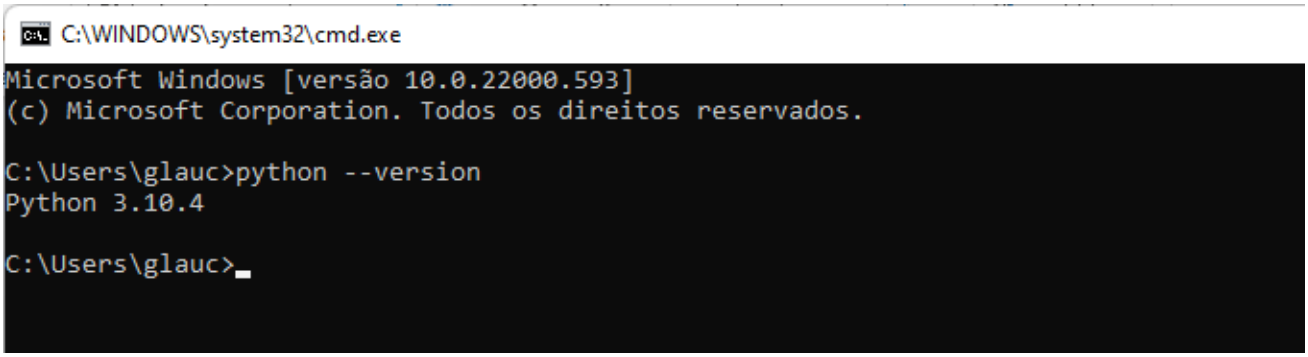


**Figura 11. Testando se a instalação foi bem-sucedida**

### (C). Instalando o ambiente de desenvolvimento Thonny

- PASSO 1: Ainda no prompt de comando, digite a instrução abaixo, que instala a última versão do Thonny e cria um atalho na Área de Trabalho e no Menu Iniciar.

  ```
  pip install thonnyapp
  ```

  Durante o processo de instalação você verá uma tela similar à mostrada na Figura 12.

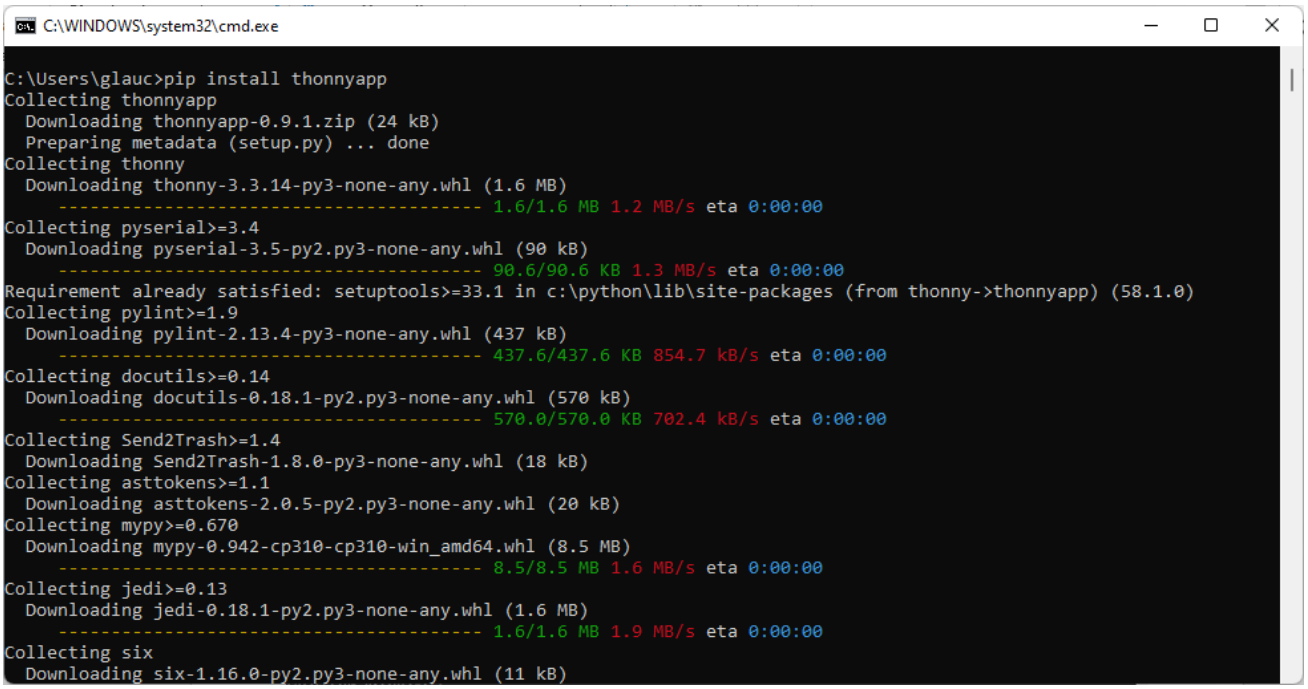


**Figura 12. Instalando o ambiente Thonny**

- PASSO 2: Ao final da instalação (que poderá demorar alguns minutos), verifique se o ícone do Thonny foi adicionado a sua Área de Trabalho (Figura 13)


**Figura 13. Ícone do Thonny na área de trabalho**

**(D). Criando e executando o seu primeiro programa Python no Thonny**

- PASSO 1: Efetue o duplo clique no ícone do Thonny e você verá a tela da Figura 14. A parte **superior** é o **editor de código**, onde você escreverá seus programas. A área inferior, denominada **shell**, é onde os resultados serão apresentados. Na shell você também pode usar o Python de forma interativa, como será mostrado em um capítulo posterior.

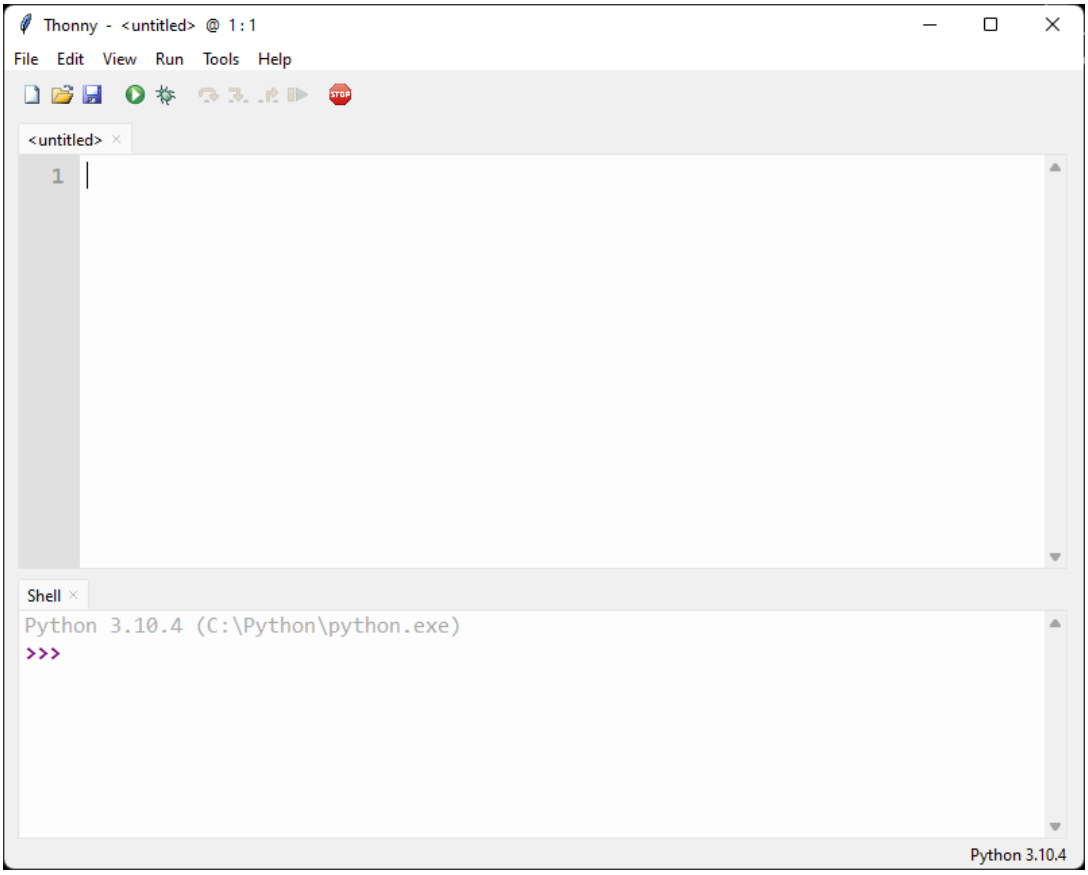


**Figura 14. Tela principal do Thonny**

- PASSO 2: digite o programa mostrado na Figura 15 e depois no botão com o ícone da setinha (Run Current Script).

  Vai ser aberta uma janela para que você salve o programa digitado (Figura 16). Escolha uma pasta destino e um nome para o programa (por exemplo, eu escolhi "hello" e salvei na pasta "C:\progs"). IMPORTANTE: todo programa Python é salvo com a extensão ".py", ou seja, se você atribui o nome "hello", ele será salvo como "hello.py".

  Após ser salvo, o programa será executado e exibirá a mensagem "Olá ENCE!" na área de resultados (Figura 17).

  Se você alterar o programa e o executar novamente, as suas modificações serão salvas automaticamente.

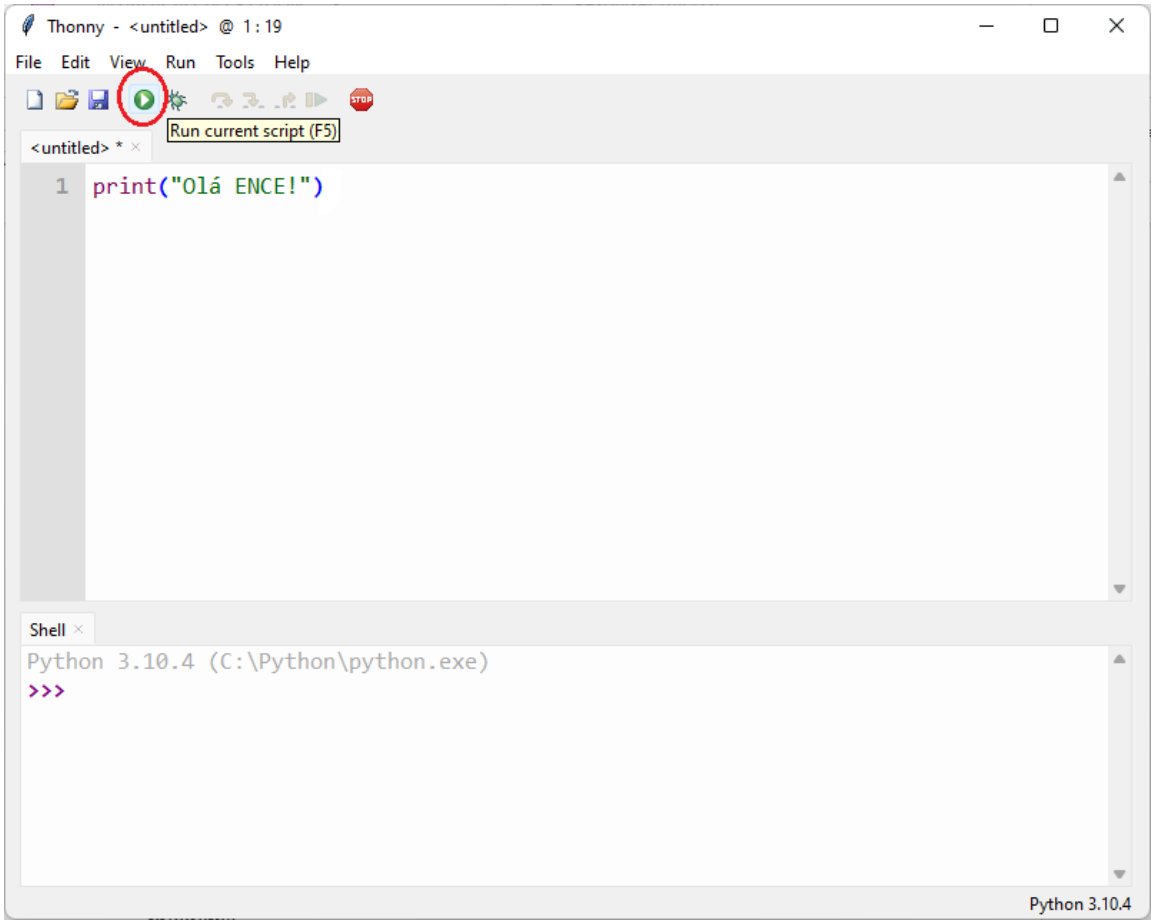


**Figura 15. Tela principal do Thonny**

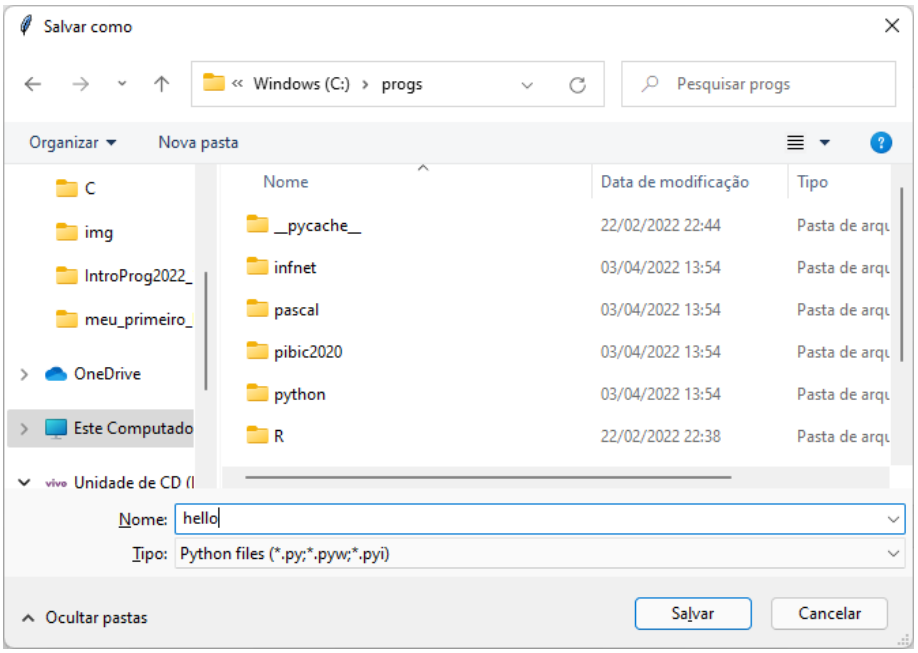


**Figura 16. Salvando o programa**

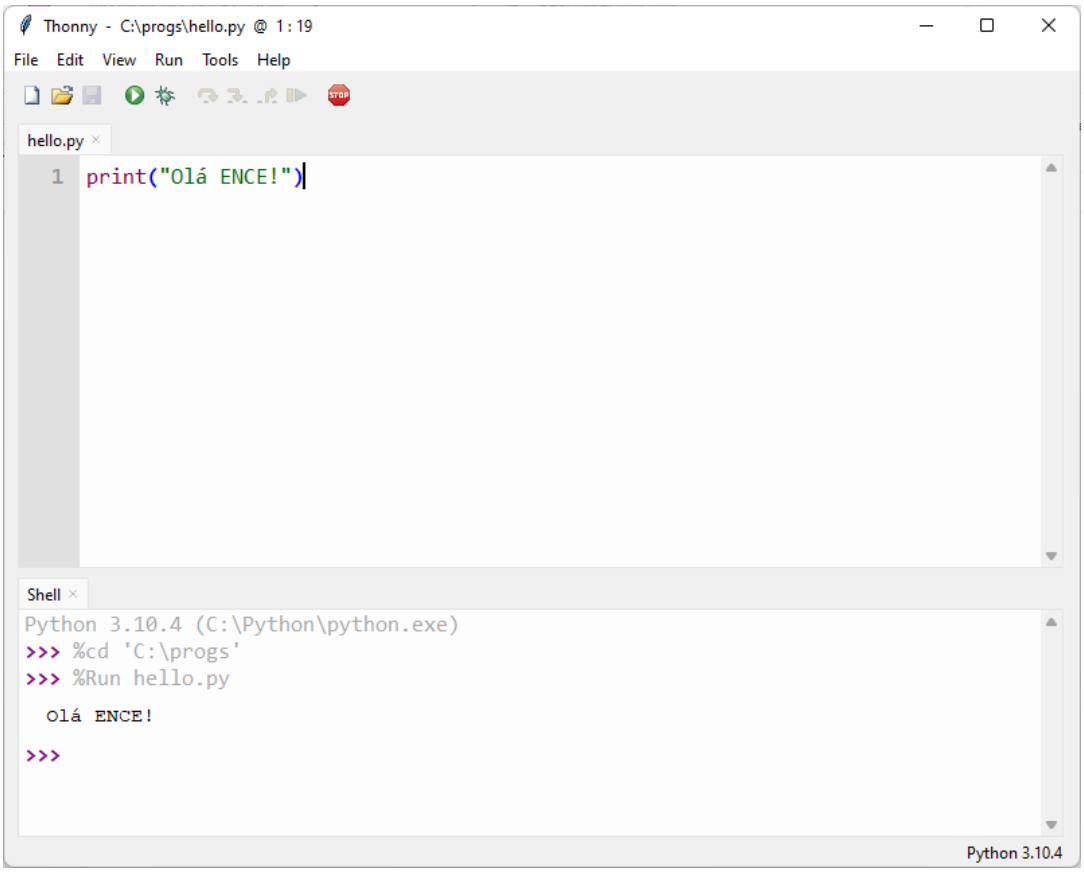


**Figura 17. Resultado exibido na parte inferior do Thonny**

Pronto! Você acabou de instalar o Python, além de ter criado, salvo e executado o seu primeiro programa!

**IMPORTANTE – PLANO B**

Se você não conseguir instalar utilizando as instruções acima, uma alternativa que poderá utilizar é baixar o arquivo ZIP com o Thonny + Python que está no meu Google Drive

https://drive.google.com/drive/folders/1jEvAOeXnx8N8F1D1ziRFEDwz0B2WrcE1

Depois de baixar, basta descompactar para qualquer pasta do seu computador. Este foi o arquivo utilizado para a instalação no laboratório. Ele tem uma versão um pouco mais antiga do Python, mas ela é suficiente para rodar todos os programas que veremos em nosso curso.

## I.4 Criando (e entendendo!) um programa em Python

Para facilitar o entendimento sobre o conceito de programa de computador e a forma como estes manipulam dados de entrada e produzem dados de saída, será um apresentado um novo exemplo. Desta vez, o exemplo é um pouco mais elaborado do que o programa que imprimiu a mensagem "Olá ENCE!". Para digitar o programa você pode escolher a opção File / New no menu do Thonny e depois proceder da mesma maneira como apresentado na seção anterior.

A Figura 18 apresenta um pequeno programa Python composto por 5 linhas. Este programa realiza a seguinte tarefa: recebe um número inteiro como entrada e, como saída, exibe no vídeo o valor correspondente ao dobro do número digitado.

```
#PROGRAMA QUE CALCULA O DOBRO DE UM NÚMERO
print('Digite um número inteiro:')
n = int(input())
dobro = n * 2
print('O dobro do número é:', dobro)
```

**Figura 18. Programa que recebe um número inteiro e calcula e imprime o seu dobro**

O programa da Figura 18 contém uma sequência de instruções (ou comandos) escritas na linguagem Python, a linguagem usada em nosso curso. Todo programa de computador é sempre escrito em alguma **linguagem de programação**. Além do Python existem diversas outras linguagens[1] para a programação de computadores, como C, Java, Julia, R, SAS e PHP, por exemplo. Por enquanto não se preocupe em tentar entender o motivo de existirem tantas linguagens de programação; também não se preocupe em desvendar a forma pela qual o computador consegue decodificar (entender) as instruções programadas. Estes temas serão

[1] Consulte o ranking das linguagens mais populares em https://www.tiobe.com/tiobe-index/

abordados posteriormente. Nos parágrafos a seguir, será realizada apenas uma explicação sobre o programa da Figura 18, para que você possa se familiarizar com a maneira pela qual os dados são recebidos, armazenados, processados e disponibilizados aos usuários. Então lá vai a explicação:

- As palavras apresentadas em **negrito** são **funções** da biblioteca padrão da linguagem Python. De uma maneira simplificada, pode-se dizer que estas palavras representam instruções que conseguem ser decodificadas pelo computador.

- A primeira linha do programa contém o símbolo tralha "#" seguido pelo texto "PROGRAMA QUE CALCULA O DOBRO DE UM NÚMERO". Esta linha consiste apenas em um **comentário** utilizado para documentar o programa. Ela é ignorada pelo computador. Sendo assim, se você quiser colocar comentários em qualquer programa, basta utilizar # seguido de um texto. Comentários são opcionais, você usa se quiser (ou seja, você não é obrigado a colocar um comentário como primeira linha de seu programa e nem em nenhuma outra).

- A linha 2 contém a função **print()**. Esta é a principal função de saída da linguagem Python. Isto quer dizer que ela é a responsável por enviar uma informação ao usuário. No caso da linha 2, a seguinte instrução está sendo fornecida ao computador: "escreva a mensagem '*Digite um número inteiro:*' no vídeo". Sempre que o computador "vê" a função print() em um programa ele "sabe" que precisa gerar uma saída para o usuário.

- A linha 3 faz várias coisas! Basicamente ela serve para receber uma informação digitada pelo usuário – que esperamos ser um número inteiro, já que foi isso que pedimos para ele – e armazenar esse valor em uma **variável** chamada "n". Vamos explicar com calma:

    - Veja que no lado direito do sinal de igualdade, dentro dos parênteses mais internos aparece a função **input()**, que é utilizada para a entrada de dados na linguagem Python. É ela que permite que o usuário digite dados, possibilitando um "diálogo com o computador". Quando o computador "vê" a instrução **input**(), ele "sabe" que deve esperar o usuário digitar um algo no teclado.

    - Porém, toda informação lida do teclado através do input() é, por padrão, considerada como um texto (dado alfanumérico) pelo Python. Isto é: mesmo que o usuário digite um número, como 100 por exemplo, o Python inicialmente vai considerar que se trata de um texto. Para informar ao Python que você deseja tratar a informação digitada como um número inteiro, é necessário utilizar função **int()**. Essa função serve para converter um texto em número inteiro. Sendo assim a instrução **int(input())** literalmente significa: "aguarde o usuário digitar algo e depois que ele tiver digitado, considere que a informação digitada é um número inteiro e não um texto".

- Se o usuário digitar alguma coisa que não possa ser convertida para um número inteiro, como “ENCE”, “bossa nova”, 3.14, “bla bla bla” etc., **um erro será disparado** pelo Python e a execução do programa será encerrada. Mas se ele digitar algo que possa ser convertido para inteiro, como “100”, “-1”, “0”, “9999”, o programa vai executar corretamente. Em um capítulo posterior da apostila abordaremos as técnicas que podem ser utilizadas para tratar erros.

- O dado digitado como entrada é sempre armazenado em alguma **variável**. No exemplo acima uma variável chamada “n”, definida no lado esquerdo do sinal de igualdade. Mas o que é uma variável?

  - As variáveis consistem em um recurso disponibilizado por qualquer linguagem para armazenar os dados que serão utilizados pelas instruções de um programa. O conceito de variável é um dos mais importantes no mundo da programação de computadores e será exaustivamente trabalhado ao longo deste curso. Por agora, é importante que o aluno saiba que qualquer dado de entrada **precisa sempre** ser armazenado numa variável; e que qualquer dado que precise ser utilizado em mais de uma linha do programa também deve ser armazenado numa variável. Observe que o programa da Figura 18 utiliza duas variáveis, batizadas como “n” e “dobro”. Estas variáveis são do **tipo** inteiro (**int**). O tipo de uma variável indica a espécie de valor que ela pode armazenar. Existem outros tipos como **float**, para armazenar números reais e **str**, para armazenar textos.

- A linha 4 contém um comando de atribuição, caracterizado pelo uso do símbolo igual: “ = ”. A instrução nesta linha multiplica o conteúdo da variável “n” – valor digitado pelo usuário – por 2 (“n * 2”, a parte que está à direita do “=”) e armazena o resultado na variável “dobro” (que está à esquerda do “=”). Note que o símbolo “ * ” é utilizado como operador de multiplicação.

- A última instrução do programa está contida na linha 5. Nela, a função print() é mais uma vez utilizada para produzir uma saída ao usuário. Desta vez o programador ordena que o programa exiba a mensagem “*O dobro do número é* ” seguida do valor de “dobro” (variável que armazena o valor do dobro de “n”).

Caso você não tenha entendido com clareza as instruções que formam o programa da Figura 18, não fique tão preocupado. A programação Python é abordada com detalhes a partir do Capítulo 3 até o final da apostila. Neste momento, o mais importante a ser assimilado é que: (i) um programa é composto por instruções que conseguem ser entendidas pelo computador; (ii) estas instruções sempre devem ser ordenadas de maneira adequada. Veja que o programador colocou tudo na ordem certa no programa exemplo: primeiro é exibida uma mensagem que solicita com que o usuário digite um número inteiro (linha 2); em seguida está o comando de entrada, que recebe o número digitado pelo usuário e armazena numa

variável (linha 3); depois de receber o número, o programa o multiplica por 2 (linha 4). Por fim, o resultado do processamento é apresentado ao usuário (linha 5).

## Exercícios propostos

**(1)** Para que são utilizadas as variáveis em um programa de computador?

**(2)** Em um programa, qual o papel dos comandos de entrada? E para que servem os comandos de saída?

**(3)** O *site* **países** do IBGE ( https://paises.ibge.gov.br/#/ ) possui um mapa onde o usuário pode clicar sobre a área de um país e, em resposta a esta ação, o *site* apresenta estatísticas oficiais sobre o país. **Responda**: qual é a entrada e quais são as saídas deste programa?

**(4)** Quais os principais tipos de programa desenvolvidos por um cientista de dados? Na sua opinião, são os mesmos tipos de programas criados por um desenvolvedor de aplicativos?

**(5)** Escreva um programa em Python que exiba a seguinte mensagem na tela: "**Preciso fazer todos os exercícios para aprender**". **Obs.:** para fazer este e os outros programas propostos neste capítulo, oriente-se pelo programa apresentado na Figura 18.

**(6)** Desenvolva um programa Python que leia um número inteiro e exiba o seu sucessor e o seu antecessor.

**(7)** Descreva o propósito do programa abaixo.

```
print('Digite o valor da base:')
base = int(input())
print('Digite o valor da altura:')
altura = int(input())
area = (base * altura) / 2
print(area)
```

**(8)** Escreva um programa em Python que peça ao usuário com que forneça dois números inteiros; a seguir o programa deve exibir a soma dos dois números, o produto, a diferença do primeiro pelo segundo e o quociente do primeiro pelo segundo. **Obs.**: Em Python os caracteres +, -, * e / são utilizados com operadores de soma, subtração, multiplicação e divisão, respectivamente.