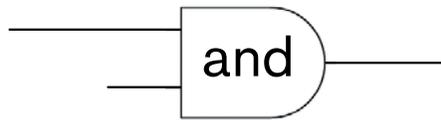


Coding in **Python**



Elements of Discrete Mathematics

Appendix A Getting Started with Python

Maria Litvin

Phillips Academy, Andover, Massachusetts

Gary Litvin

Skylight Software, Inc.

Skylight Publishing
Andover, Massachusetts

Skylight Publishing
9 Bartlet Street, Suite 70
Andover, MA 01810

web: <http://www.skylit.com>
e-mail: sales@skylit.com
support@skylit.com

**Copyright © 2019 by Maria Litvin, Gary Litvin, and
Skylight Publishing**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the authors and Skylight Publishing.

Library of Congress Control Number: 2019905086

ISBN 978-0-9972528-4-2

The names of commercially available software and products mentioned in this book are used for identification purposes only and may be trademarks or registered trademarks owned by corporations and other commercial entities. Skylight Publishing and the authors have no affiliation with and disclaim any sponsorship or endorsement by any of these products' manufacturers or trademarks' owners.

1 2 3 4 5 6 7 23 22 21 20 19

Printed in the United States of America

Getting Started with Python

- A.1 [Introduction](#) 3
- A.2 [Installing and Configuring under Windows](#) 3
- A.3 [Installing and Configuring under Mac OS X](#) 5
- A.4 [Running Python Programs](#) 6
- A.5 [Where to Go from Here](#) 7

A.1 Introduction

Python is distributed under the open source license that makes it free to use, even for commercial products. There are two versions of Python, Python 2 and Python 3, which are not 100% upward compatible. We will work with Python 3.

To obtain Python go to www.python.org and click Downloads. Download the current production version for your operating system and CPU, for example, Python 3.7.3 Windows x86-64 executable installer (`python-3.7.3-amd64.exe`) for a 64-bit Windows system.

The Python distribution includes the Python interpreter, a very simple development environment, called IDLE, libraries, tools, documentation (tutorials, manuals) and a few demo programs.

Python is preinstalled on Mac systems, but it may be an old version, and a convenient interactive development environment may not be present by default. Windows does not come with Python installed. The configuration of IDLE and file path syntax differ slightly between operating systems.

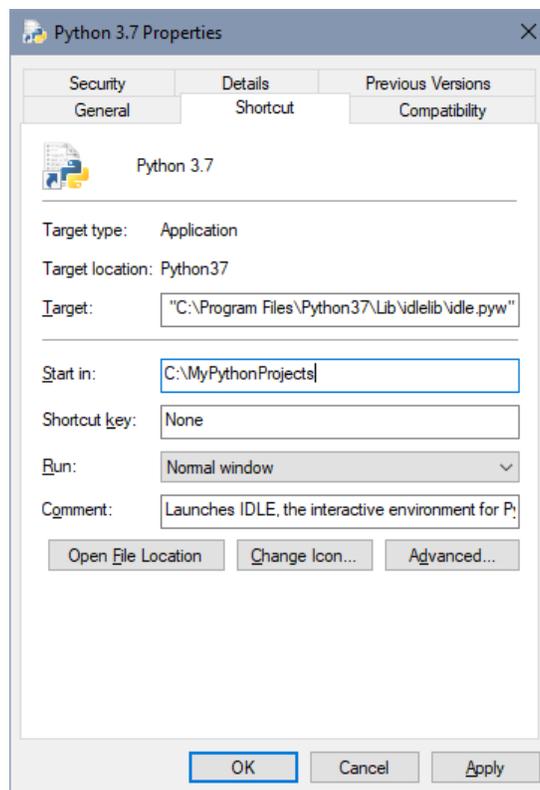
A.2 Installing and Configuring Under Windows

Download the appropriate Python installer file and save it in a folder of your choice or to the Desktop. Double-click on the saved `.exe` file to run the installer. Accept the defaults, only consider installing Python in `C:\Program Files\Python37`. The installation includes the Python interpreter, IDLE — a very simple development environment with a graphical user interface (GUI), libraries, tools, and documentation (a tutorial, a reference manual) and a few demo programs. There is no need to install docs and manuals: you can always find them online.

It is possible to run Python in a Command Prompt window, but for most purposes it is more convenient to use IDLE, rather than the “raw” command-line user interface. To run IDLE, on Windows 10 type “python” in the search box and click on “IDLE desktop app” to run.

It is better to create a shortcut to IDLE on the desktop. To do that, type “python” in the search box, right click on “IDLE desktop app” and choose “Open file location.” Drag IDLE shortcut to the desktop while holding the `Ctrl` key down to create a copy (you should see a small “+” symbol next to the icon).

As other programs, IDLE uses the concept of the working directory — the folder where the interpreter looks for Python programs and data files. By default, the working directory is set to the same folder where Python is installed. It is not a good idea, though, to place your files into the installation folder. Instead, reconfigure the shortcut you created to specify your own working directory. To do that, right-click on the shortcut and choose `Properties`. In the “Start In:” line, enter your work folder, for example `C:\MyPythonProjects`. Click `OK`.



(If your working directory name has spaces in it, put it in double quotes.)

Check your configuration:

Double-click on the shortcut icon to start the interpreter

At the prompt, enter:

```
>>> from os.path import abspath
>>> abspath('.')
```

Python responds:

```
'C:\\MyPythonProjects'
```

Here `\\` stands for one backslash. `abspath('.')` returns the absolute path of the current directory.

It is possible to change the working directory from the interpreter itself. For example:

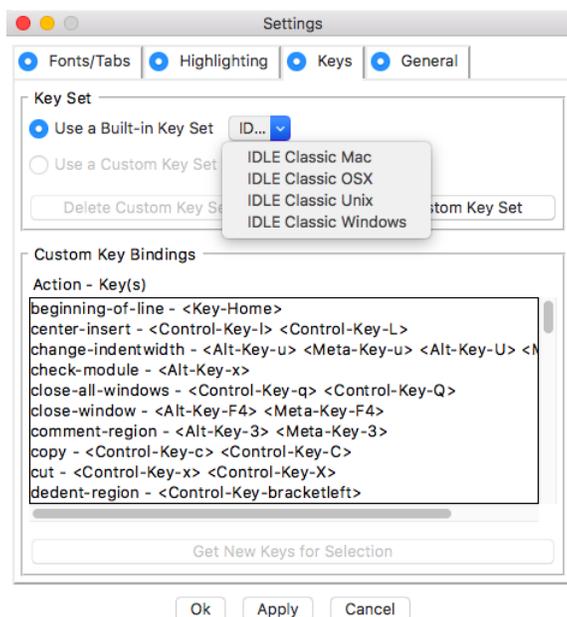
```
>>> from os import chdir
>>> chdir('C:/MyOtherProjects')
```

By default, Windows operating systems hide file extensions. When you are developing apps, it is more convenient to see the extensions. To enable extensions in Windows 10, click `View` on any folder and check the “File name extensions” box.

A.3 Installing and Configuring under Mac OS X

Python 2 is installed on all Mac systems, but historically it has been several years out of date, and IDLE, the simple development tool, might be not included. You should download and install the current release. Download and install the latest version of Python and IDLE (for example, 3.7.3 as of June 2019), which will create a Python application in your Application folder. You can drag an icon out of this folder to your desktop for convenient access. If you have more than one version of Python and IDLE installed, make sure that you use Python 3 when working with this book.

If you use IDLE on several operating systems at the same time, you may find it confusing if the shortcut key bindings for IDLE commands and menu choices differ between those systems. IDLE allows you to select whether these bindings follow Mac, UNIX, Windows or other historical key bindings. Select the key bindings you prefer in the IDLE `Options->Keys` (Or `IDLE Preferences->Keys` on a Mac) dialog:



A.4 Running Python Programs

A short program can be typed directly into the interpreter:

```
>>> print('Hello, World!')
Hello, World!
```

Or:

```
>>> sum_to_n = 0
>>> for n in range(1, 6):
    sum_to_n += n
    print("{0:3d}: {1:5d}".format(n, sum_to_n))

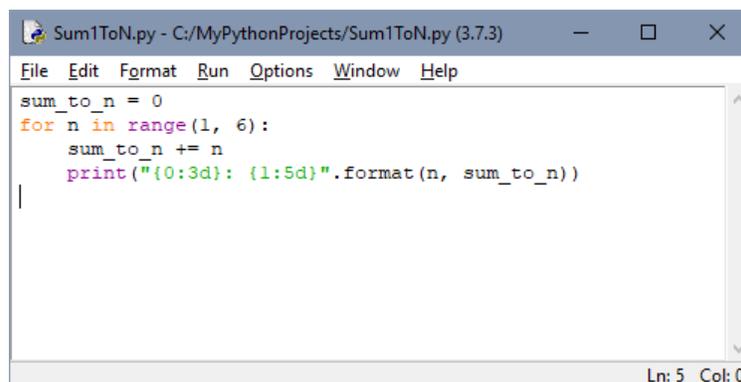
1:      1
2:      3
3:      6
4:     10
5:     15
```

This is not very practical, though, because you have to reenter every statement to run the program again or to make a small change to it. In IDLE, you can copy one of the previous statements: click on it or move the cursor up to the appropriate line and press `<Enter>`. You can then edit the statement. Still reentering every statement of a program would be too tedious.

It is more practical to save the program statements in a file and execute the program from the file. The text of a program is called *source code*, so the file will hold the source code for your program. Python source code files usually have the extension `.py`.

You can create a source file using any text editor, for example Notepad. You could even use a word processor — just make sure you save your file as a “text only” file, and that you replace the default `.txt` extension in the file name with `.py`. However, the easiest way to write a short Python program is by using IDLE’s own built-in editor.

To open a new editor window in IDLE, choose `New File` from the `File` menu (or press `Ctrl-N`). Type in the text of your program.



```
Sum1ToN.py - C:/MyPythonProjects/Sum1ToN.py (3.7.3)
File Edit Format Run Options Window Help
sum_to_n = 0
for n in range(1, 6):
    sum_to_n += n
    print("{0:3d}: {1:5d}".format(n, sum_to_n))
|
Ln: 5 Col: 0
```

The IDLE editor automatically increases indentation for statements after a colon: `def`, `while`, `if`, and so on. Press `Backspace` to decrease the indentation level. Choose `Save` or `Save As...` from the `File` menu or press `Ctrl+Shift+S` to save the program in a file. Use the `.py` extension with the file name. Save the file in a folder of your choice, for example, `C:\MyPythonProjects`.

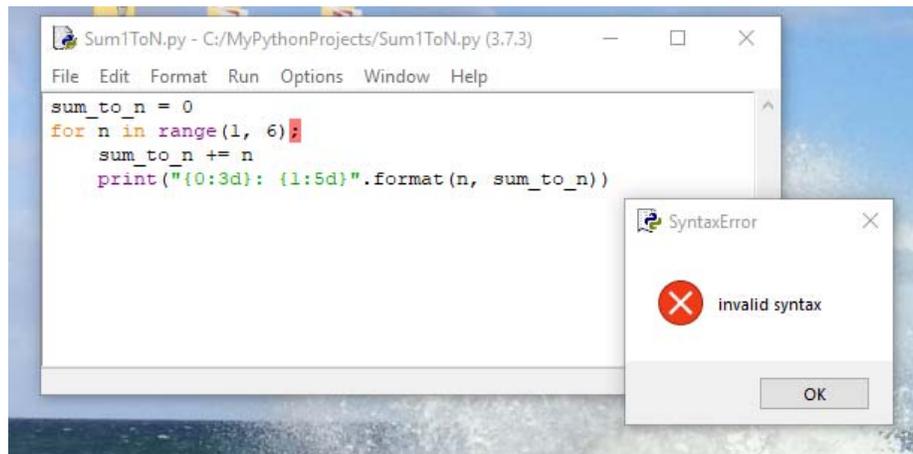
While an IDLE editor window is open and active, you can test your program by choosing `Run Module` from the `Run` menu or simply by pressing `F5`.

Every time you run a program after making a change to it, IDLE will ask you whether you wish to save it. You can disable this by going to `Options/Configure IDLE` and choosing the “No Prompt” option under the “General” tab.

Other IDLE configuration defaults are quite reasonable.

IDLE uses a default monospace font. You may want to use a more modern monospace font. For example, the “Andale Mono” font distinguishes the numeral zero from the letter “O”, and the numeral one from the letter “l”; it is designed specifically for coding. The IDLE `Options->Fonts/Tabs` (or `IDLE Preferences` on a Mac) dialog will allow you to set it as IDLE's default font.

If your program has syntax errors, Python will alert you to that and highlight the first error:



(semicolon instead of a colon).

You can have several editor windows open at once and cut and paste text within the same window or from one window to another. Highlight the text you want to copy, press `Ctrl-C` to copy the text, position the cursor at the insertion point, press `Ctrl-V` to paste the text.

A.5 Where to Go from Here

The IDLE environment is totally adequate for working on programs in our *Coding in Python and Elements of Discrete Mathematics* book. There are many professional development tools for Python programmers, of course (see <http://wiki.python.org/moin/DevelopmentTools>).