If Python is not installed on your system, or if you see a version earlier than Python 3.9, you need to download a Python installer for Windows. Go to https://python.org and hover over the Downloads link. You should see a button for downloading the latest version of Python. Click the button, which should automatically start downloading the correct installer for your system. After you’ve downloaded the file, run the installer. Make sure you select the option Add Python to PATH, which will make it easier to configure your system correctly. Figure 1-1 shows this option selected

To check python install on window wirte in commond prompt

Python enter

Then tun a cod

print("Hello Python interpreter!")

to close terminal press cntr Z and enter or enter commond exit()

nstalling VS Code

You can download an installer for VS Code at <https://code.visualstudio.com>.

Click the Download for Windows button and run the installer.

Open VS Code, and close the Get Started tab if it’s still open. Make a new file by clicking File4New File or pressing CTRL-N (⌘-N on macOS). Save the file as hello\_world.py in your python\_work folder. The extension .py tells VS Code that your file is written in Python, and tells it how to run the program and highlight the text in a helpful way. After you’ve saved your file, enter the following line in the editor: hello\_world.py print("Hello Python world!") To run your program, select Run4Run Without Debugging or press CTRL-F5. A terminal screen should appear at the bottom of the VS Code window, showing your program’s output:

STRING

Print(name.title())

Print(name.upper())

Print(name.lower())

4

The Style Guide When someone wants to make a change to the Python language, they write a Python Enhancement Proposal (PEP). One of the oldest PEPs is PEP 8, which instructs Python programmers on how to style their code. PEP 8 is fairly lengthy, but much of it relates to more complex coding structures than what you’ve seen so far. The Python style guide was written with the understanding that code is read more often than it is written. You’ll write your code once and then start reading it as you begin debugging. When you add features to a program, you’ll spend more time reading your code. When you share your code with other programmers, they’ll read your code as well. Given the choice between writing code that’s easier to write or code that’s easier to read, Python programmers will almost always encourage you to write code that’s easier to read. The following guidelines will help you write clear code from the start.

Indentation PEP 8 recommends that you use four spaces per indentation level. Using four spaces improves readability while leaving room for multiple levels of indentation on each line

The Style Guide

When someone wants to make a change to the Python language, they write a Python Enhancement Proposal (PEP). One of the oldest PEPs is PEP 8, which instructs Python programmers on how to style their code. PEP 8 is fairly lengthy, but much of it relates to more complex coding structures than what you’ve seen so far. The Python style guide was written with the understanding that code is read more often than it is written. You’ll write your code once and then start reading it as you begin debugging. When you add features to a program, you’ll spend more time reading your code. When you share your code with other programmers, they’ll read your code as well. Given the choice between writing code that’s easier to write or code that’s easier to read, Python programmers will almost always encourage you to write code that’s easier to read. The following guidelines will help you write clear code from the start.

Indentation

PEP 8 recommends that you use four spaces per indentation level. Using four spaces improves readability while leaving room for multiple levels of indentation on each line. In a word processing document, people often use tabs rather than spaces to indent. This works well for word processing documents, but the Python interpreter gets confused when tabs are mixed with spaces. Every text editor provides a setting that lets you use the TAB key but then converts each tab to a set number of spaces. You should definitely use your TAB key, but also make sure your editor is set to insert spaces rather than tabs into your document. Mixing tabs and spaces in your file can cause problems that are very difficult to diagnose. If you think you have a mix of tabs and spaces, you can convert all tabs in a file to spaces in most editors. Working with Lists   69

Line Length

Many Python programmers recommend that each line should be less than 80 characters. Historically, this guideline developed because most computers could fit only 79 characters on a single line in a terminal window. Currently, people can fit much longer lines on their screens, but other reasons exist to adhere to the 79-character standard line length. Professional programmers often have several files open on the same screen, and using the standard line length allows them to see entire lines in two or three files that are open side by side onscreen. PEP 8 also recommends that you limit all of your comments to 72 characters per line, because some of the tools that generate automatic documentation for larger projects add formatting characters at the beginning of each commented line. The PEP 8 guidelines for line length are not set in stone, and some teams prefer a 99-character limit. Don’t worry too much about line length in your code as you’re learning, but be aware that people who are working collaboratively almost always follow the PEP 8 guidelines. Most editors allow you to set up a visual cue, usually a vertical line on your screen, that shows you where these limits are.

Blank Lines

To group parts of your program visually, use blank lines. You should use blank lines to organize your files, but don’t do so excessively. By following the examples provided in this book, you should strike the right balance. For example, if you have five lines of code that build a list and then another three lines that do something with that list, it’s appropriate to place a blank line between the two sections. However, you should not place three or four blank lines between the two sections. Blank lines won’t affect how your code runs, but they will affect the readability of your code. The Python interpreter uses horizontal indentation to interpret the meaning of your code, but it disregards vertical spacing.

Styling Your if Statements

In every example in this chapter, you’ve seen good styling habits. The only recommendation PEP 8 provides for styling conditional tests is to use a single space around comparison operators,

such as ==, >=, and <=.

For example: if age < 4: is better than: if age<4:

Class

you write classes that represent real-world things and situations,

and you create objects based on these classes.

When you write a class, you define the general behavior that a whole category of objects can have.

When you create individual objects from the class, each object is automatically equipped with the general behavior;

you can then give each object whatever unique traits you desire.

Making an object from a class is called instantiation,

and you work with instances of a class.

**In this chapter you’ll write classes and create instances of those classes**

You’ll specify the kind of information that can be stored in instances,

and you’ll define actions that can be taken with these instances.

You’ll also write classes that extend the functionality of existing classes,

so similar classes can share common functionality, and you can do more with less code.

You’ll store your classes in modules

and import classes written by other programmers into your own program files.