多媒體概論

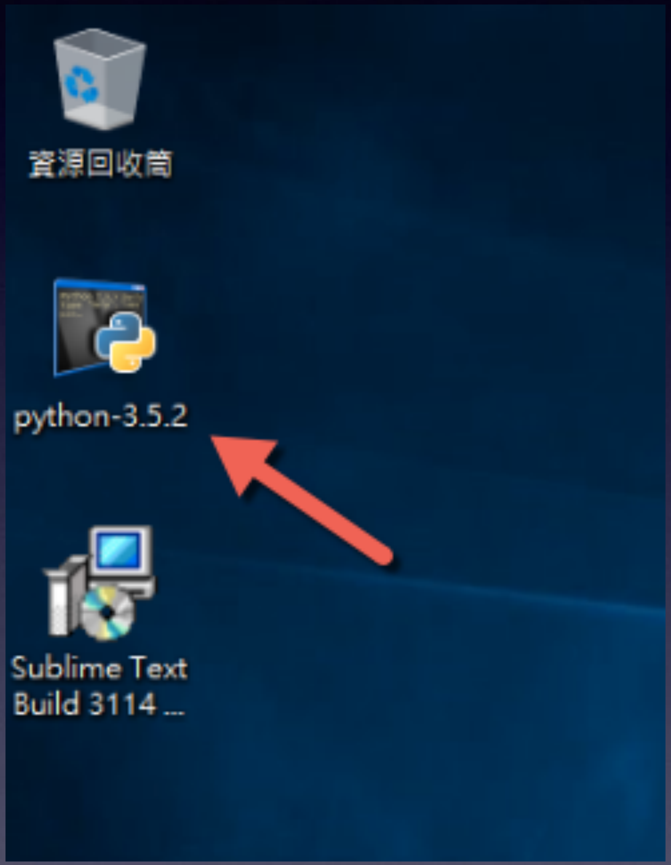
Working with Images

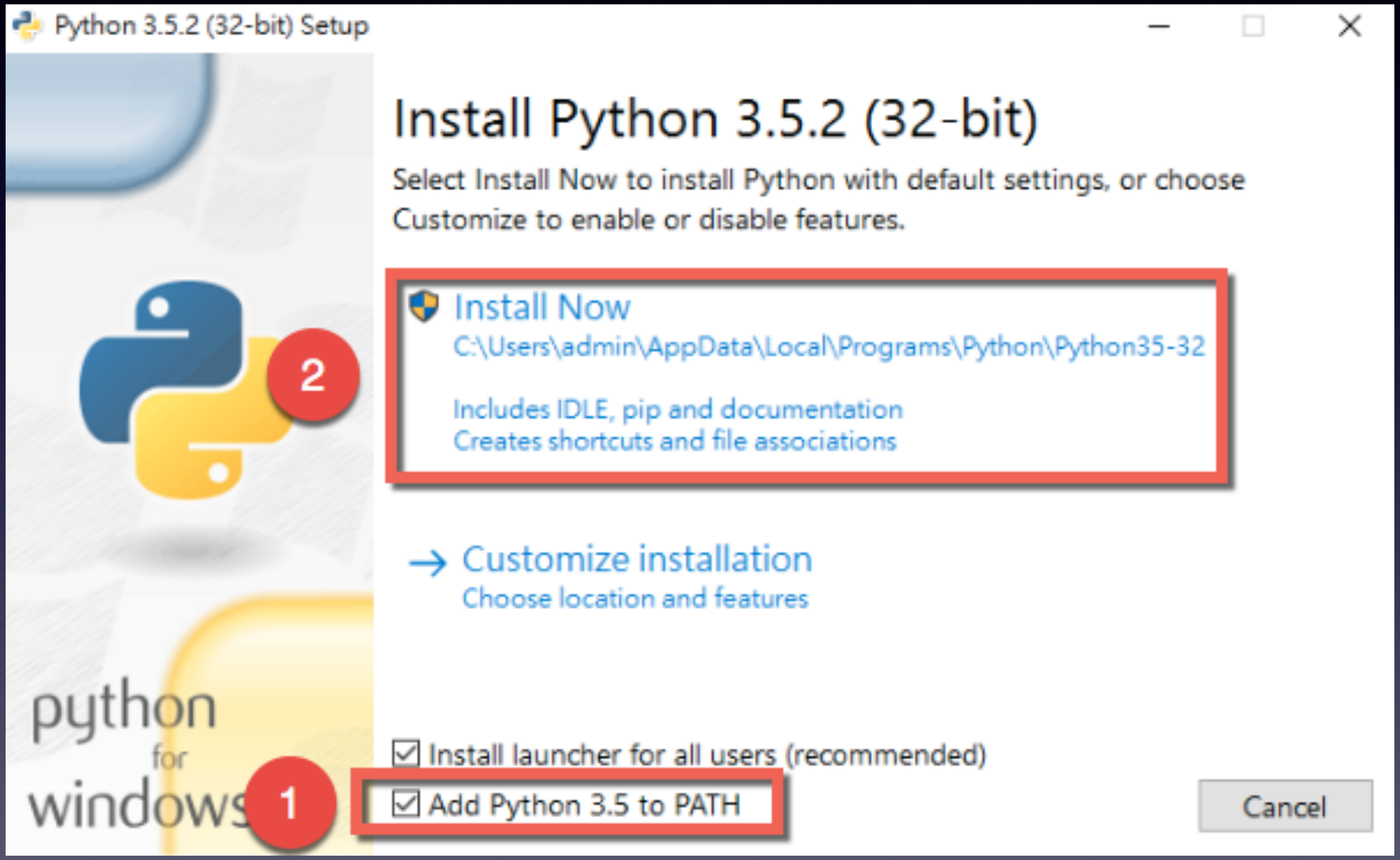
王昱景 Brian Wang
brian.wang.frontline@gmail.com

Installation prerequisites

- Python 3.5
- Sublime 3
- Python Imaging Library 3.3

- 安裝 Python 3.5


資源回收筒
python-3.5.2
Sublime Text Build 3114 ...



Install Python 3.5.2 (32-bit)

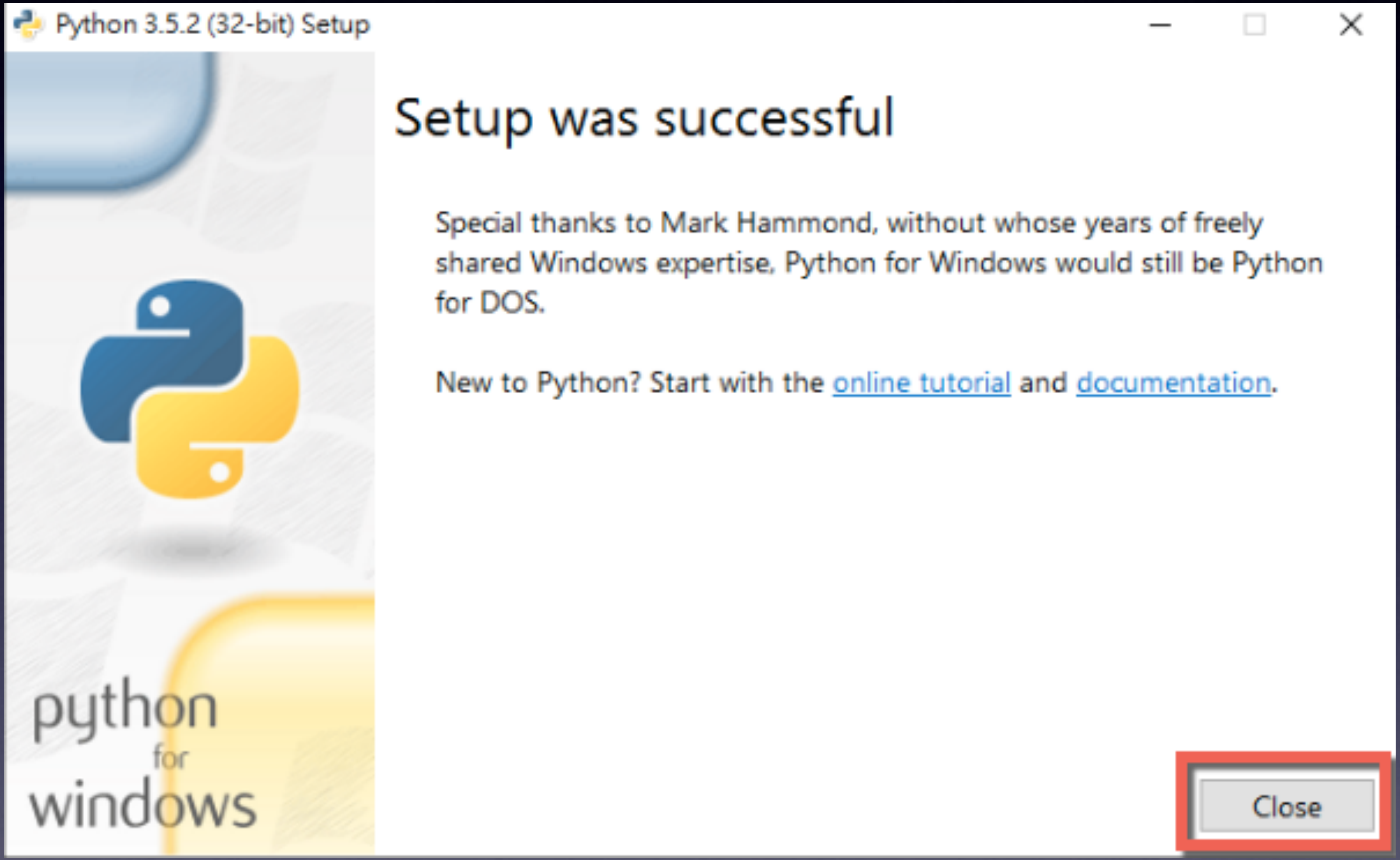
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\admin\AppData\Local\Programs\Python\Python35-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

- [✓] Install launcher for all users (recommended)
- [✓] Add Python 3.5 to PATH

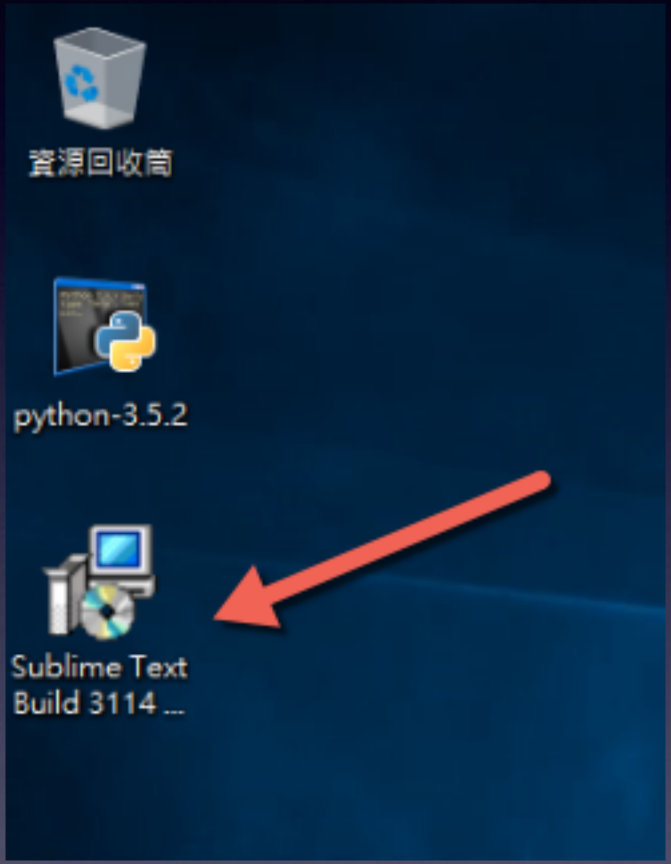


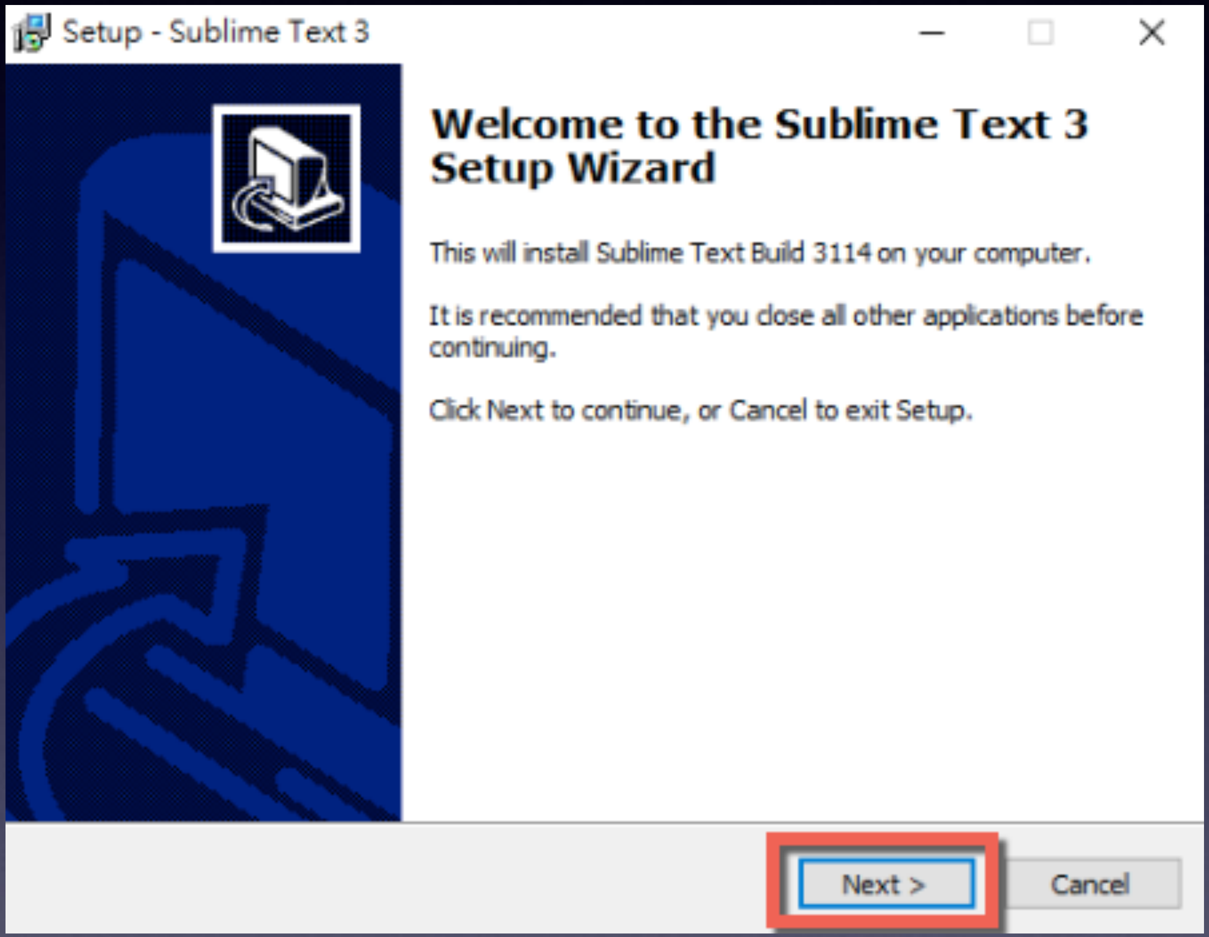
Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

- 安裝文字編輯器 Sublime 3


資源回收筒
python-3.5.2
Sublime Text Build 3114 ...

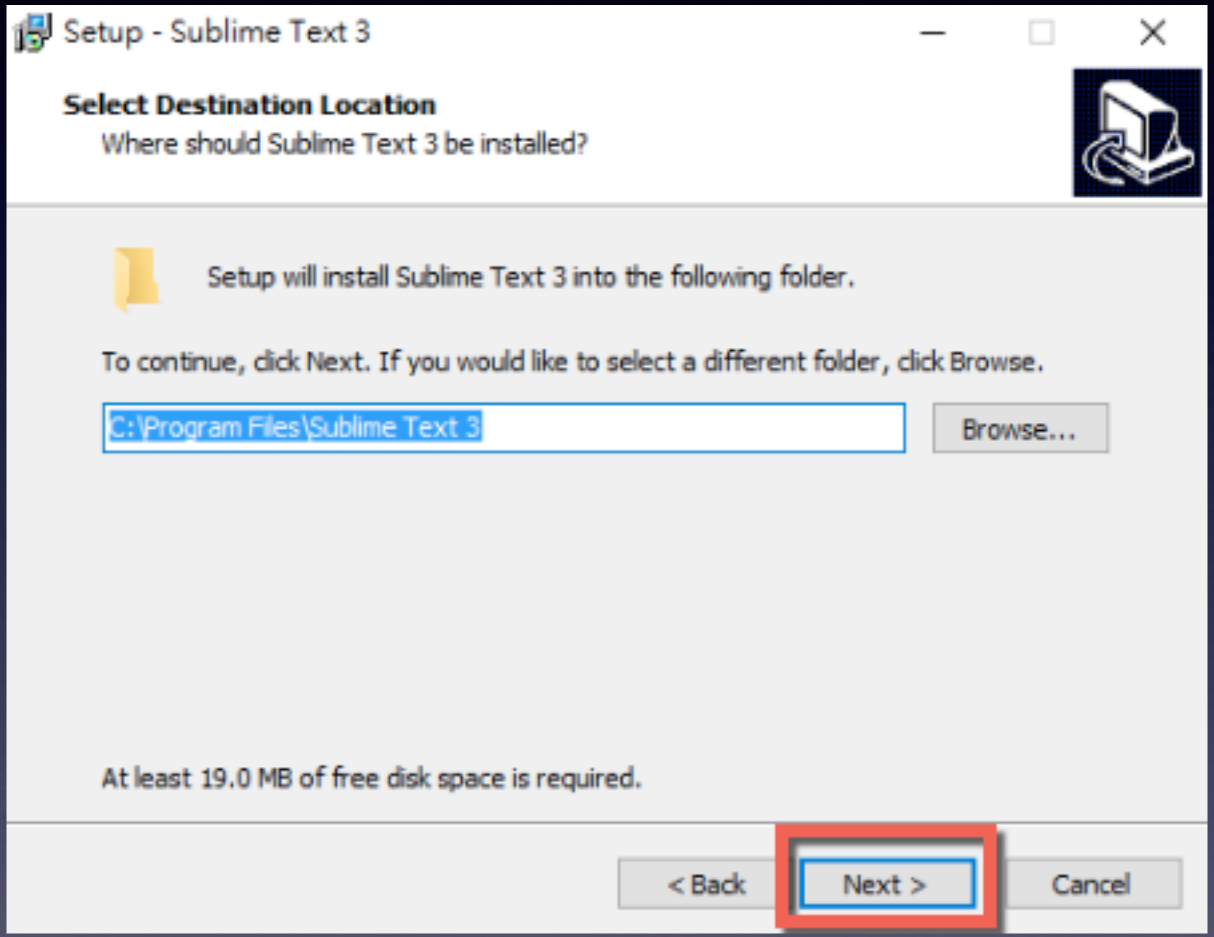

Setup - Sublime Text 3

Welcome to the Sublime Text 3 Setup Wizard

This will install Sublime Text Build 3114 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.



Select Destination Location

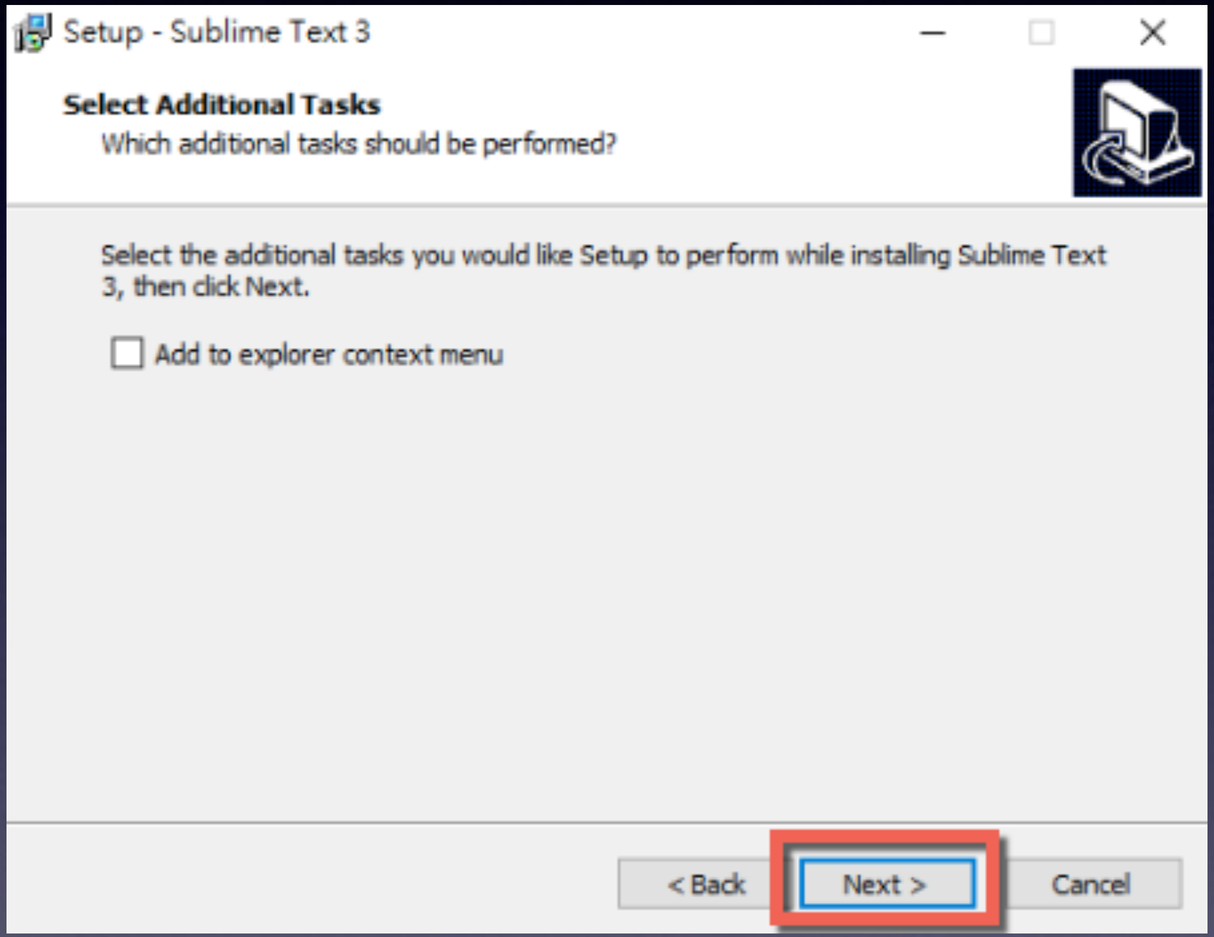
Where should Sublime Text 3 be installed?

Setup will install Sublime Text 3 into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Sublime Text 3

At least 19.0 MB of free disk space is required.

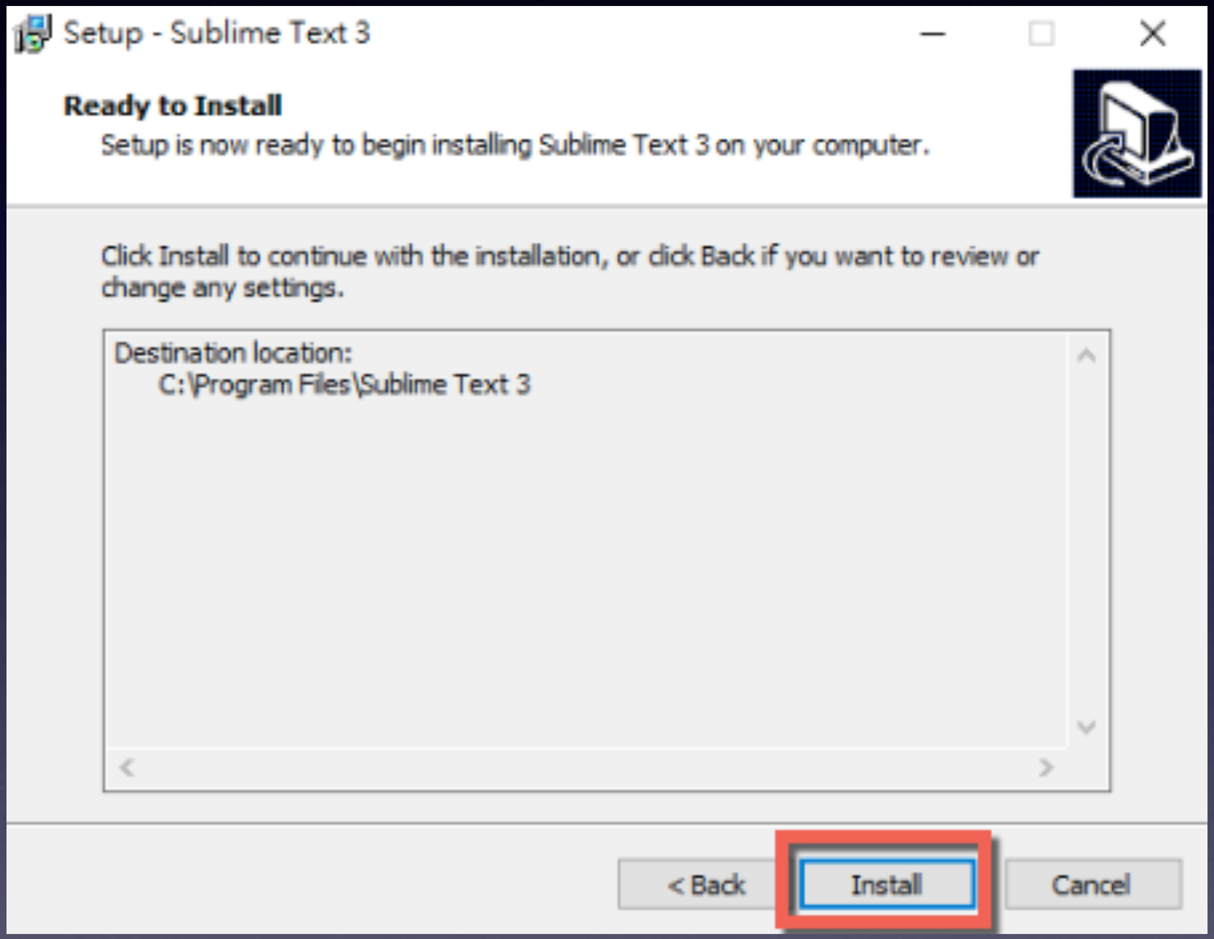

Setup - Sublime Text 3

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Sublime Text 3, then click Next.

☐ Add to explorer context menu

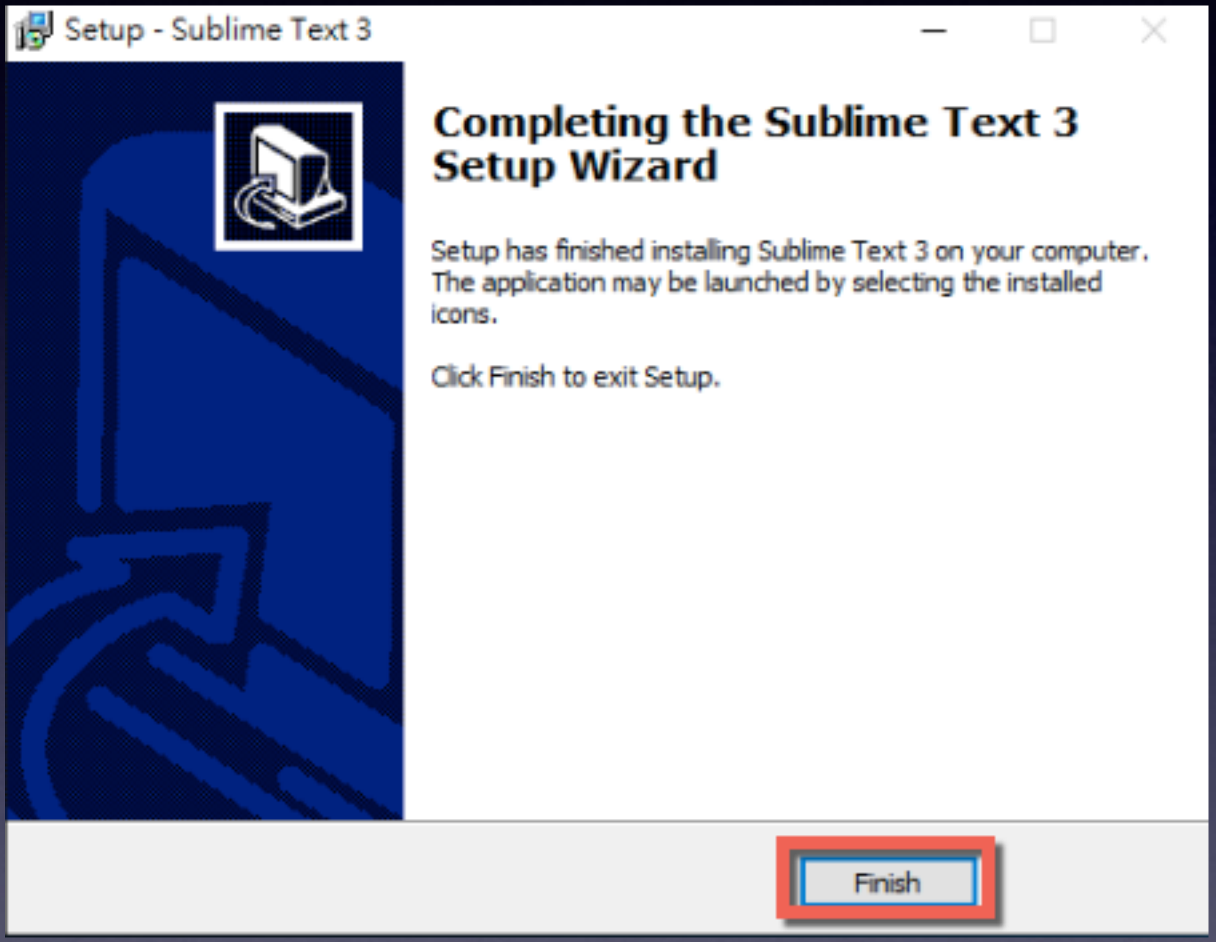


Ready to Install

Setup is now ready to begin installing Sublime Text 3 on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
    C:\Program Files\Sublime Text 3

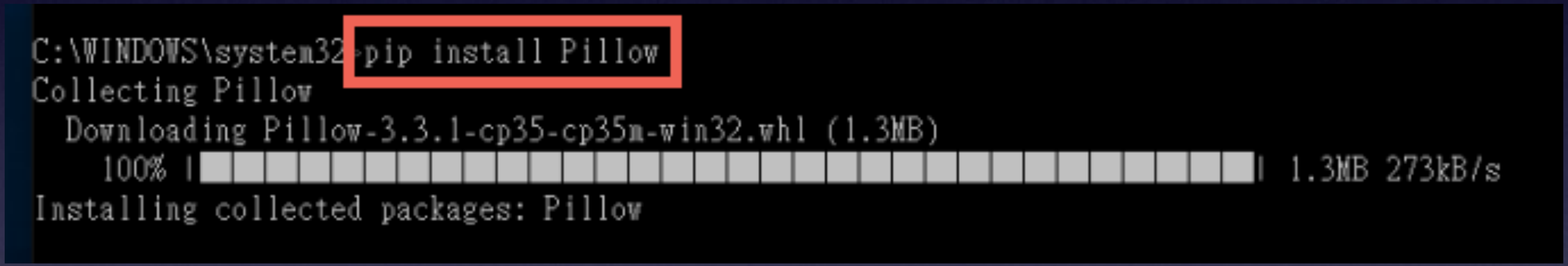


Completing the Sublime Text 3 Setup Wizard

Setup has finished installing Sublime Text 3 on your computer. The application may be launched by selecting the installed icons.

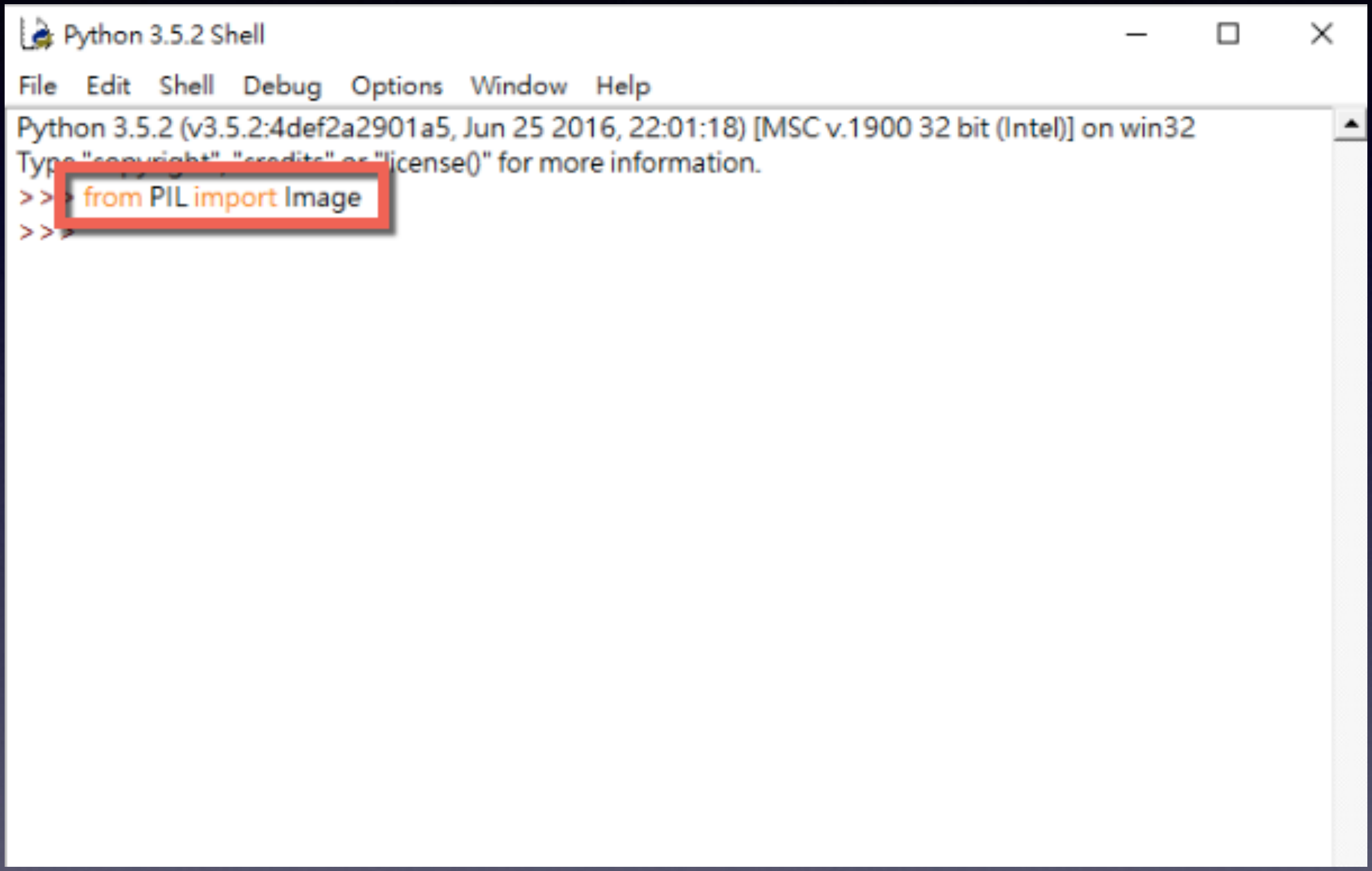
Click Finish to exit Setup.

- 安裝 Python Imaging Library（Pillow）模組
    - 開始 > 開啟 CMD > 輸入 pip install Pillow



```
C:\WINDOWS\system32>pip install Pillow
Collecting Pillow
  Downloading Pillow-3.3.1-cp35-cp35m-win32.whl (1.3MB)
    100% |████████████████████████████████| 1.3MB 273kB/s
Installing collected packages: Pillow
```

- 測試 Pillow 模組
    - 開啟 IDLE
    - 輸入 from PIL import Image



```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> from PIL import Image
>>>
```

駝峰命名法

- 各種名稱皆採駝峰命名法
- 名稱由多個文字組合時，第二個文字起，文字的第一個字母為大寫，其餘字母小寫
- 專案、類別等名稱採大寫駝峰命名法（第一個文字第一個字母大寫）
- 變數、函數等名稱採小寫駝峰命名法（第一個文字第一個字母小寫）

Reading and writing images

To manipulate an existing image, we must open it first for editing and we also require the ability to save the image in a suitable file format after making changes. The `Image` module in PIL provides methods to read and write images in the specified image file format. It supports a wide range of file formats.

To open an image, use `Image.open` method. Start the Python interpreter and write the following code. You should specify an appropriate path on your system as an argument to the `Image.open` method.

```
>>>import Image
>>>inputImage = Image.open("C:\\PythonTest\\image1.jpg")
```

This will open an image file by the name `image1.jpg`. If the file can't be opened, an `IOError` will be raised, otherwise, it returns an instance of class `Image`.

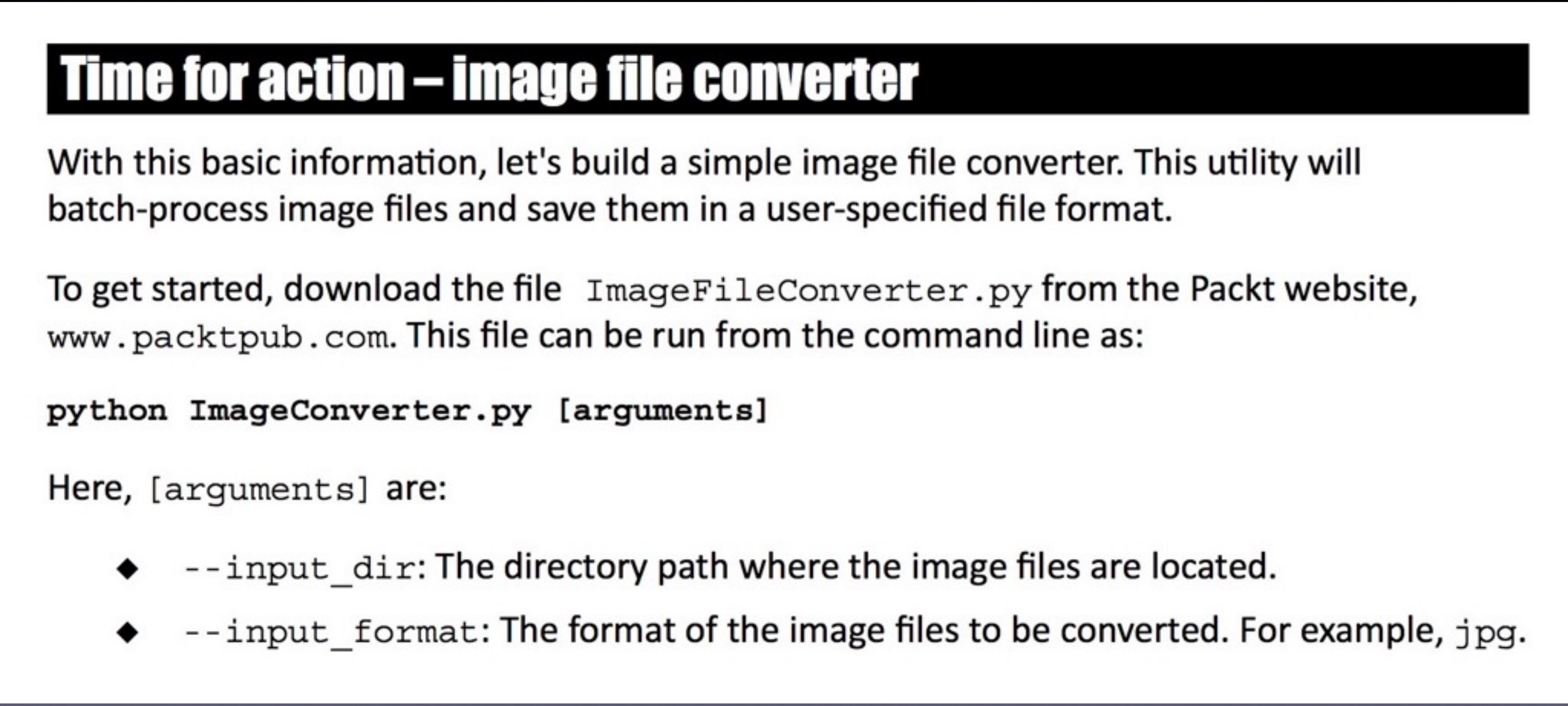
For saving image, use the `save` method of the `Image` class. Make sure you replace the following string with an appropriate `/path/to/your/image/file`.

```
>>>inputImage.save("C:\\PythonTest\\outputImage.jpg")
```

You can view the image just saved, using the `show` method of `Image` class.

```
>>>outputImage = Image.open("C:\\PythonTest\\outputImage.jpg")
>>>outputImage.show()
```

Here, it is essentially the same image as the input image, because we did not make any changes to the output image.



Time for action – image file converter

With this basic information, let's build a simple image file converter. This utility will batch-process image files and save them in a user-specified file format.

To get started, download the file `ImageFileConverter.py` from the Packt website, `www.packtpub.com`. This file can be run from the command line as:

```
python ImageConverter.py [arguments]
```

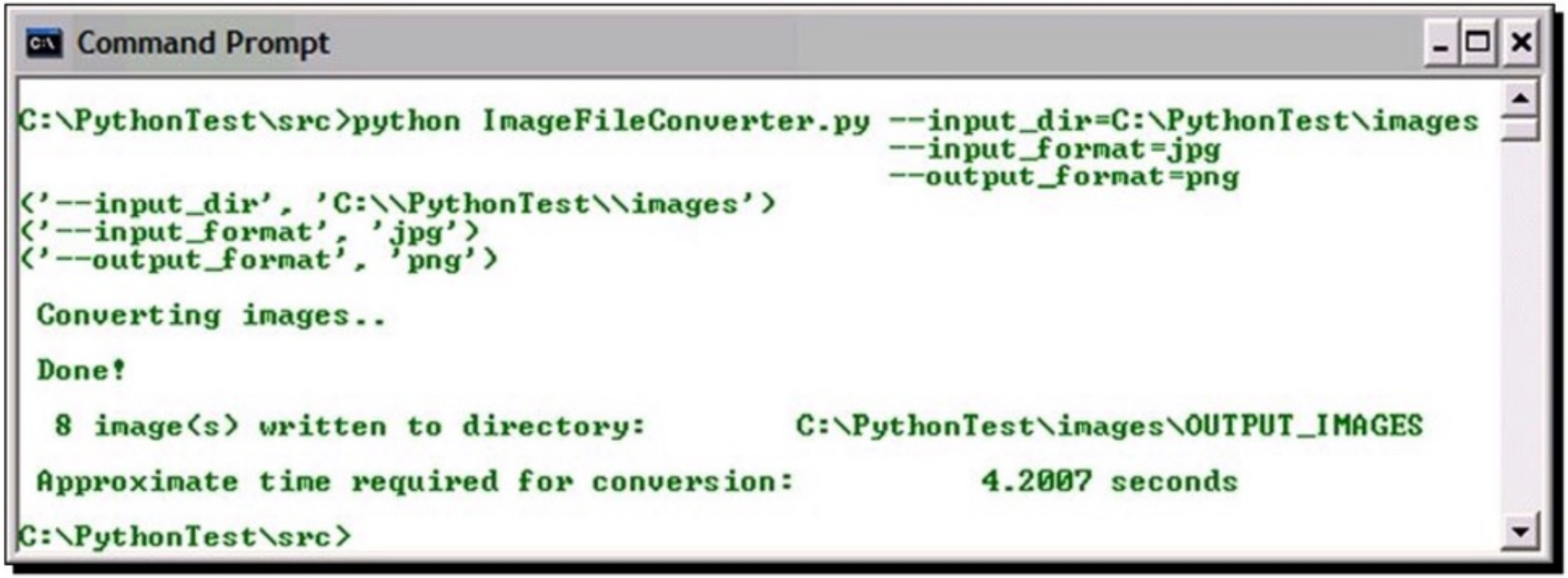
Here, `[arguments]` are:

- `--input_dir`: The directory path where the image files are located.
- `--input_format`: The format of the image files to be converted. For example, `jpg`.

- `--output_dir`: The location where you want to save the converted images.
- `--output_format`: The output image format. For example, `jpg`, `png`, `bmp`, and so on.

The following screenshot shows the image conversion utility in action on Windows XP, that is, running image converter from the command line.

Here, it will batch-process all the `.jpg` images within `C:\PythonTest\images` and save them in `png` format in the directory `C:\PythonTest\images\OUTPUT_IMAGES`.



```
C:\PythonTest\src>python ImageFileConverter.py --input_dir=C:\PythonTest\images
                                                --input_format=jpg
                                                --output_format=png
('--input_dir', 'C:\\PythonTest\\images')
('--input_format', 'jpg')
('--output_format', 'png')

 Converting images..

 Done!

  8 image(s) written to directory:        C:\PythonTest\images\OUTPUT_IMAGES

 Approximate time required for conversion:          4.2007 seconds

C:\PythonTest\src>
```

The file defines `class ImageConverter`. We will discuss the most important methods in this class.

- `def processArgs:` This method processes all the command-line arguments listed earlier. It makes use of Python's built-in module `getopts` to process these arguments. Readers are advised to review the code in the file `ImageConverter.py` in the code bundle of this book for further details on how these arguments are processed.
- `def convertImage:` This is the workhorse method of the image-conversion utility.

```
def convertImage(self):
  pattern = "*." + self.inputFormat
  filetype = os.path.join(self.inputDir, pattern)
  fileList = glob.glob(filetype)
  inputFileList = filter(imageFileExists, fileList)

  if not len(inputFileList):
    print "\n No image files with extension %s located \
    in dir %s"%(self.outputFormat, self.inputDir)
     return
   else:
```

```
  # Record time before beginning image conversion
  starttime = time.clock()
  print "\n Converting images.."

# Save image into specified file format.
for imagePath in inputFileList:
  inputImage = Image.open(imagePath)
  dir, fil = os.path.split(imagePath)
  fil, ext = os.path.splitext(fil)
  outPath = os.path.join(self.outputDir,
             fil + "." + self.outputFormat)
  inputImage.save(outPath)

endtime = time.clock()
print "\n Done!"
print "\n %d image(s) written to directory:\
%s" %(len(inputFileList), self.outputDir)
print "\n Approximate time required for conversion: \
%.4f seconds" % (endtime - starttime)
```