PYTHON SETUP GUIDE

EVERYTHING YOU NEED TO KNOW TO GET UP AND RUNNING WITH PYTHON

02

WHAT'S INSIDE

03

INTRODUCTION

There is nothing quite like seeing a Python program that you have created run, and perform as expected (or not work as expected as is often the case). Sadly, too many new developers never get to this point. They get stuck on downloading, and installing Python, setting up a code editor, and if they jump these hurdles, how to actually run their program. The frustration at simply getting up and running can sometimes be enough to make you quit. I know, because I have taught over 60,000 students how to program with Python and they've all had the same early frustrations as you.

The two most common questions from new students are:

- 'How do I install Python?'
- ‘My Python installation is not working, what did I do wrong?’

You might have heard it before, but when a rocket ship is lifting off from earth most of the fuel is used up at launch. When learning to code, most of your time, effort, and frustration will be at the beginning. But once you clear the launch pad you will find Python an easy to learn, use, and understand programming language.

04

This guide will walk you through liftoff. It will show you how to get writing Python programs as quickly as possible, with as little frustration as possible.

Another common question from new developers is, 'what code editor should I use?'. There is no easy answer to this question. Right now, heated debates are raging online about the right code editor. For many developers this is a zero or one answer, if you don't use 'Code Editor A' you have no idea what you are doing.

I am not one of those developers. In my humble opinion there is no right code editor to use, only the right code editor for the job you are currently doing. A code editor is a tool, one of many in your developers toolbox. Just like a physical toolbox that contains different tools for different jobs, you will use different code editors for different coding jobs. For example, if you are building a website you might use Sublime Text Editor, if you are analysing large amounts of data, you might use Jupyter Notebook and if you are working on something that is mission critical you will need a code editor with strong version control.

My advice is to try as many code editors as possible. Learning to code is fun and exciting, don't beat yourself up because you're not using the same code editor as some random person in an online forum. Eventually, a code editor will rise to the top and become your go-to choice.

05

REPL.IT

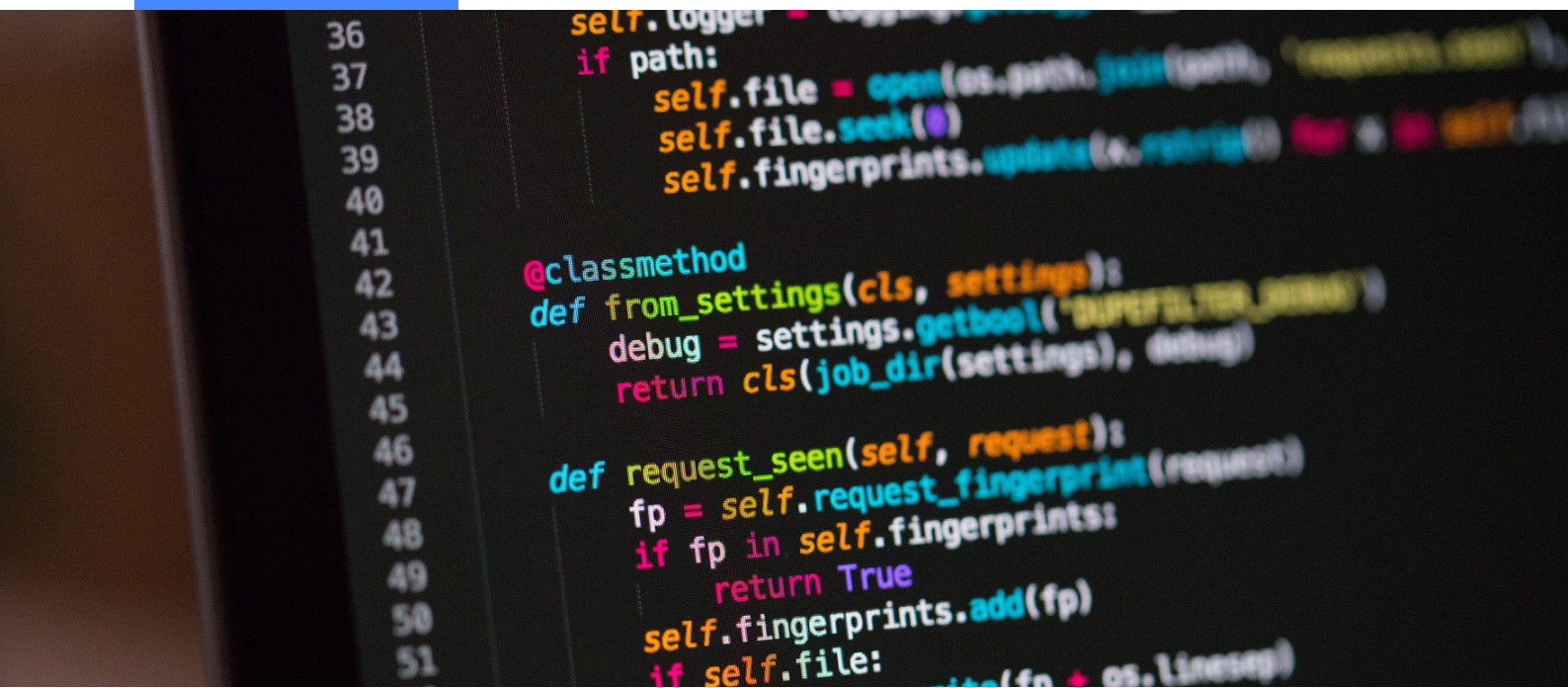
If you're looking to start coding Python with no setup, no configuration on your computer and no installations of any kind then take a look at Repl.it

Repl.it is a free, in-browser Integrated Development Environment for Python, and 50+ other programming languages.

Repl.it runs in your browser so you can get started coding in seconds. You can use Repl.it on your macOS, Windows, Linux and Chromebook.

Repl.it is a great choice for new developers as it significantly reduces the barriers to coding and lowers the learning curve so you can spend more time coding.

It's free so check it out at, Repl.it



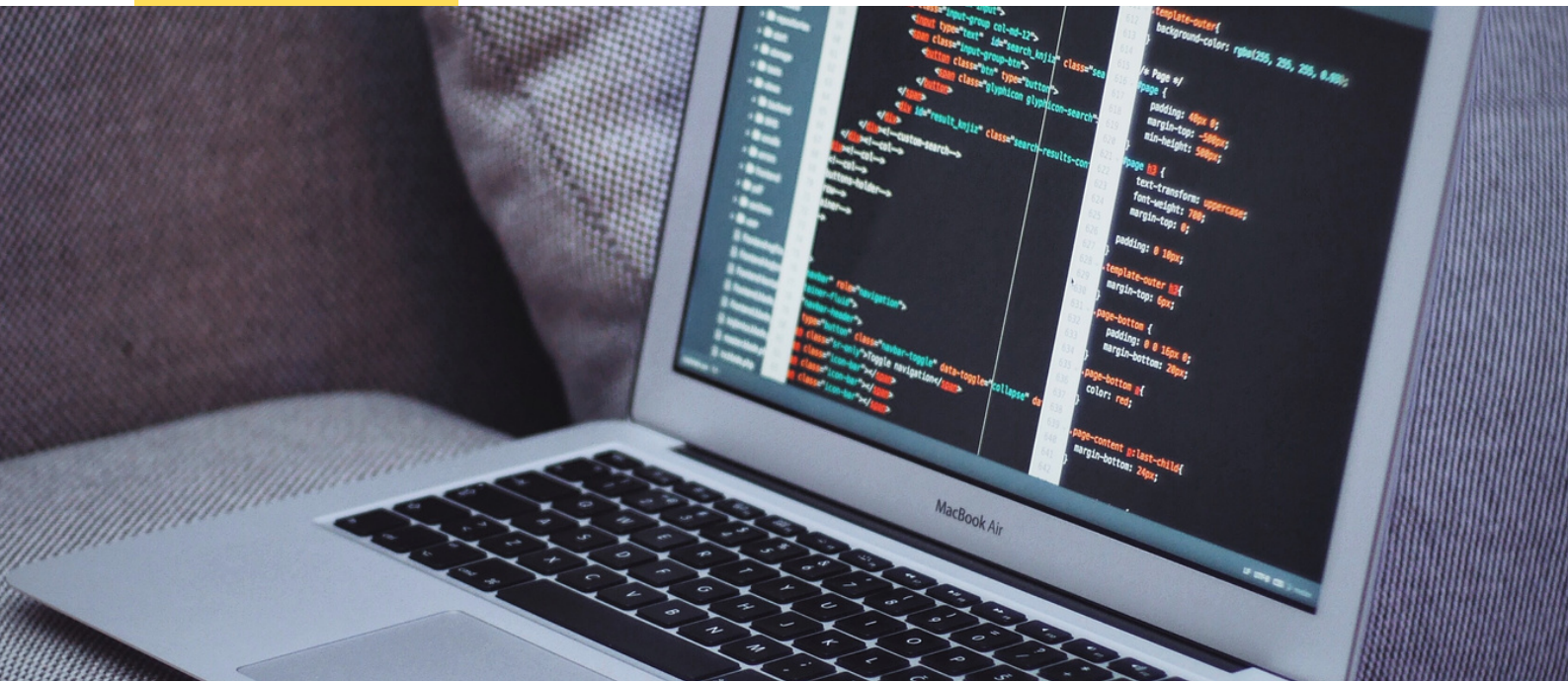
06

ANACONDA

Anaconda, is the world's most popular data science platform. But don't let that scare you off, it's not just for data scientists. It has become one of the go to sources for those who want to learn Python with an option that is easy to install and setup. Anaconda is available for Windows, MacOS, and Linux. When you install Anaconda not only do you get Python but the installation also includes multiple libraries and packages to help you on your coding journey. One such package is the Jupyter Notebook which is discussed in section two.

At the time of writing, there are three editions of Anaconda:

- Individual Edition
- Team edition
- Enterprise Edition



07

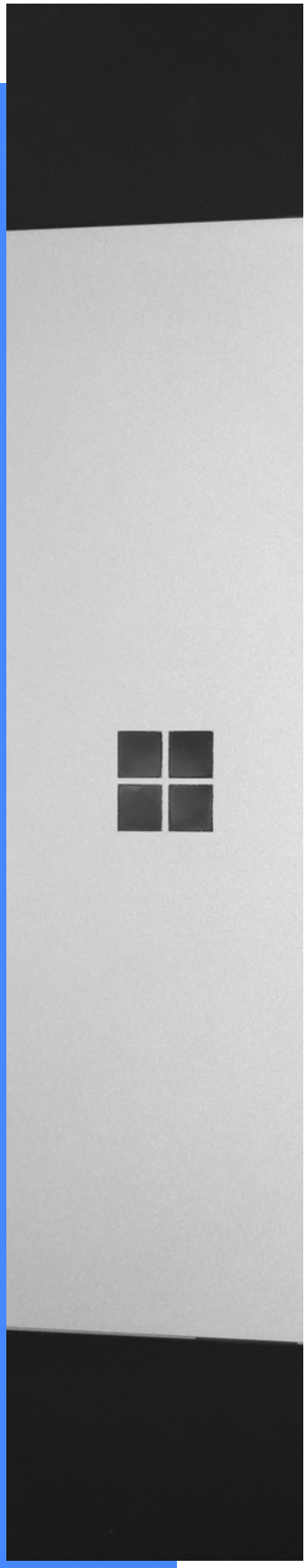
We're going to learn how to install the Individual Edition, which is Anaconda's free Python distribution for solo Python practitioners, researchers and students working on their own.

With over 20 million users worldwide, what makes Anaconda so popular?

- It's a free, open source Python distribution
- It's available for several operating systems, including Windows, macOS and Linux
- When you download and install Anaconda you don't just get Python you also get dozens of Python packages which may be relevant to your work as a developer such as (flask, pandas, numpy and many more) See here for the list of packages available for various operating systems
- Anaconda helps you create and manage virtual environments. A virtual environment is a container where you can install a specific version of Python and related packages without impacting your original Python installation.
- Anaconda installs several tools to help you write Python code such as Jupyter Notebook and Spyder

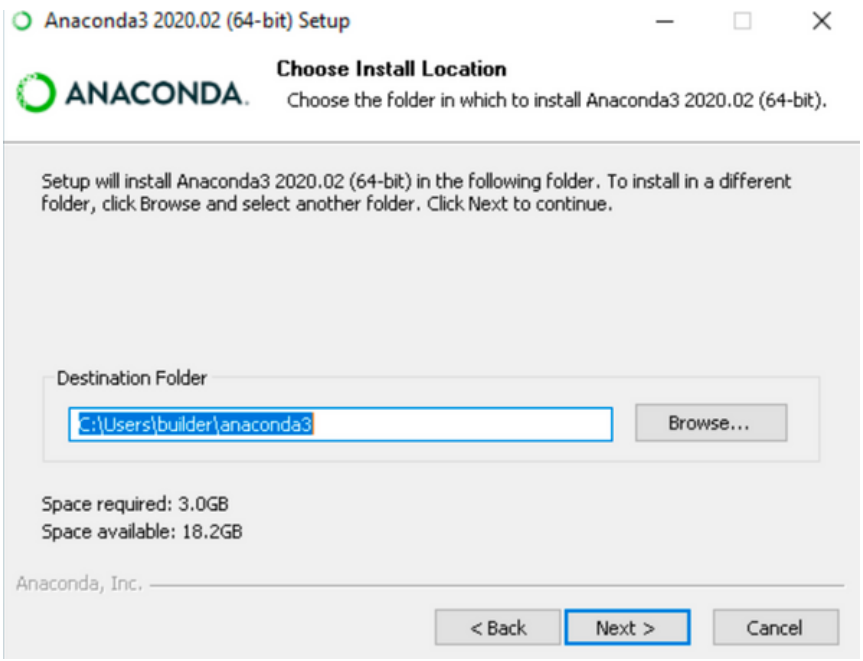
There are lots of other reasons to choose Anaconda but I think that covers the main points.

08

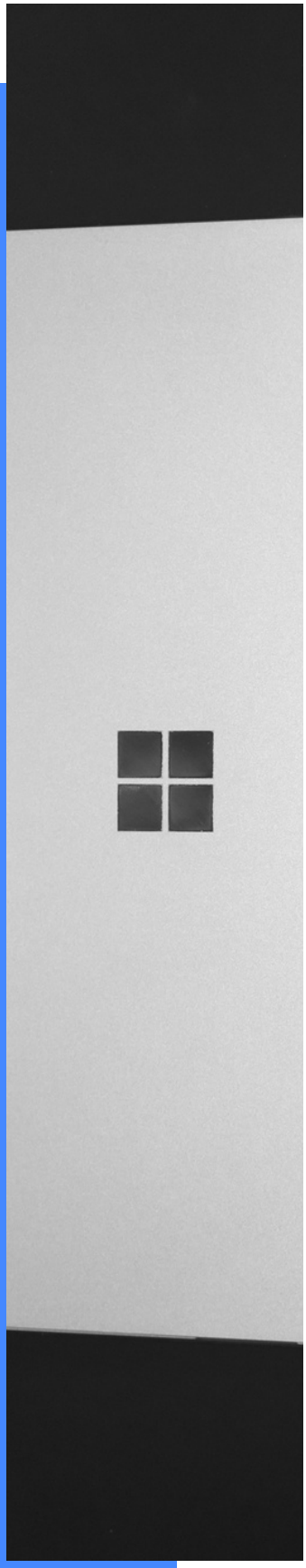


WINDOWS INSTALLATION

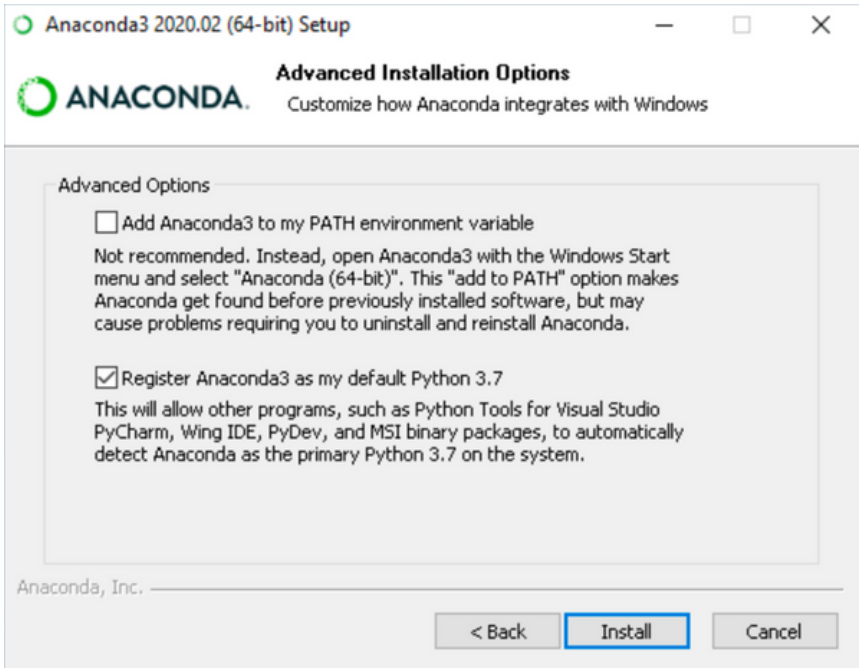
- Download the Anaconda Installer
- Double click the installer file to launch it
- Click Next
- Read the licensing terms and click 'I Agree'
- Select an install for 'Just Me', unless you are installing it for all users of your computer, in which case you will need administrator access to continue



09

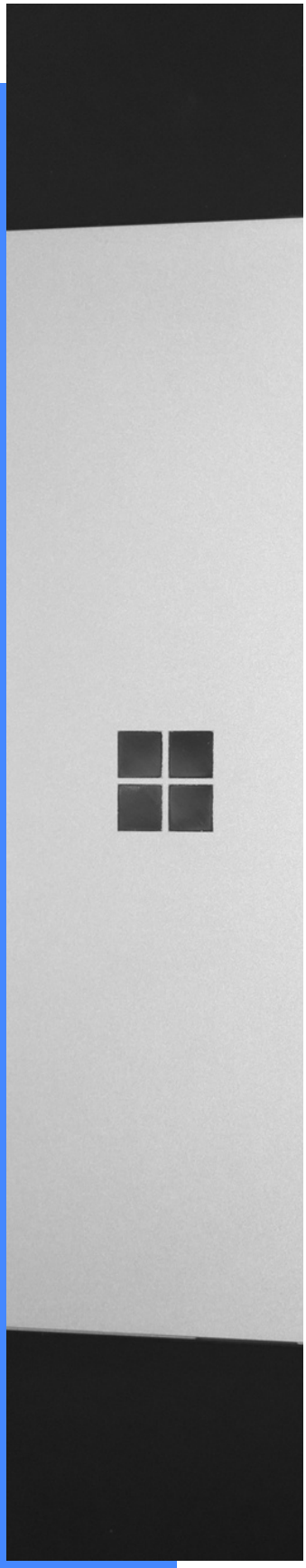


- Choose if you want to add Anaconda to your PATH environment variable. **I recommend that you do check this option even though the Anaconda documentation recommends that you do not**. When you do check the box the text will turn red, don't worry, and proceed to the next step.

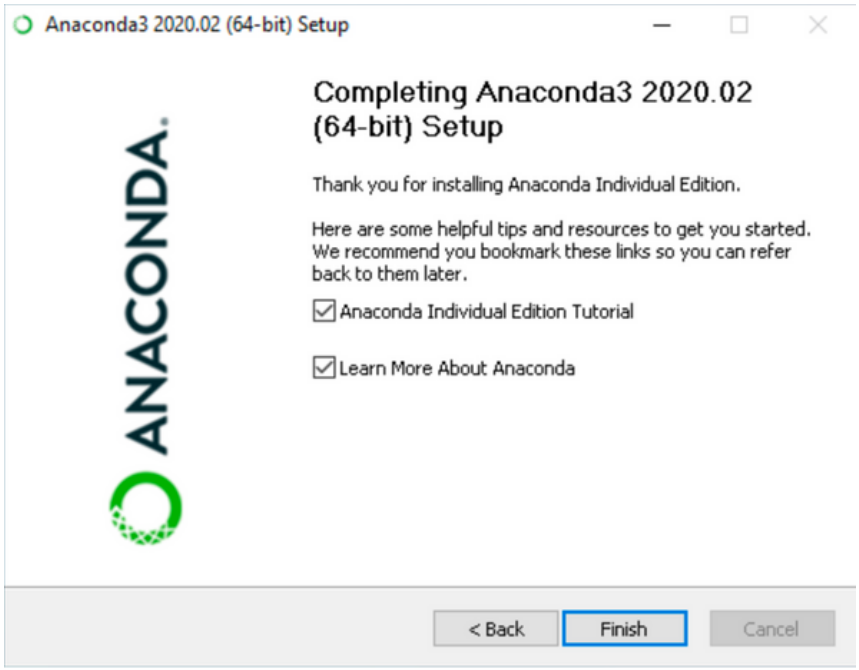


- Choose whether to register Anaconda as your default version of Python. Unless you plan on running multiple versions of Python, accept the default values and leave this box checked.
- Click the Install button
- You have the option of installing PyCharm for Anaconda by clicking the link https://www.anaconda.com/pycharm. If you would like to proceed without installing PyCharm, click the Next button.

10



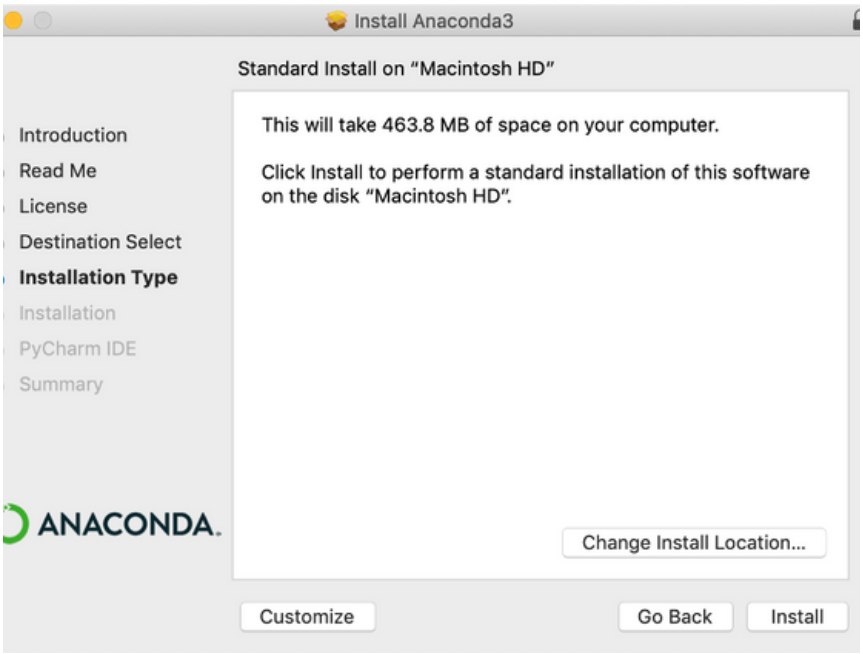
- When the installation has completed and been successful you will see the 'Thanks for installing Anaconda' prompt



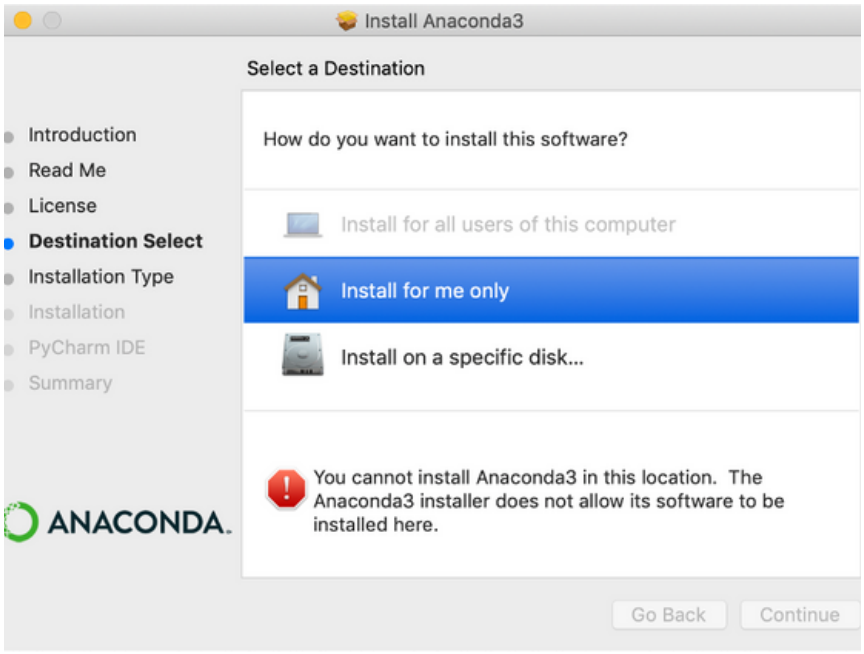
- For more information go to the Anaconda documentation for Windows installation
- Verify your installation by following steps in the **Verify Your Installation** section below.

MACOS INSTALLATION

- Download the graphical installer
- Double click the downloaded file and click continue to start the installation process
- Answer the prompts on the Introduction, Read Me, and License screens
- Click the Install button to install Anaconda in the recommended directory (changing the directory is not recommended)

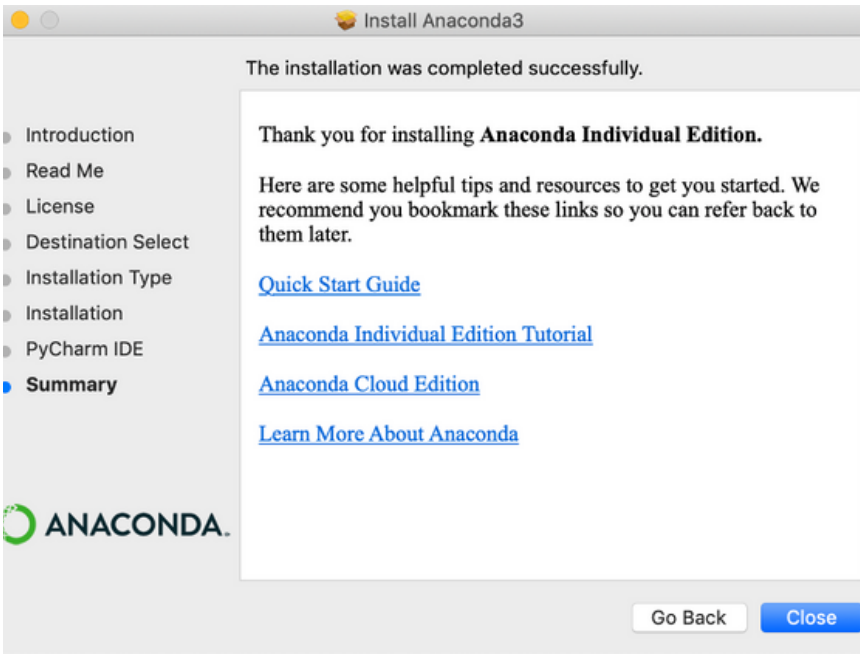


12



- Click the continue button
- You have the option of installing PyCharm for Anaconda by clicking the link https://www.anaconda.com/pycharm. If you would like to proceed without installing PyCharm, click the Continue button.
- When the installation has been successfully completed the following screen appears:

13

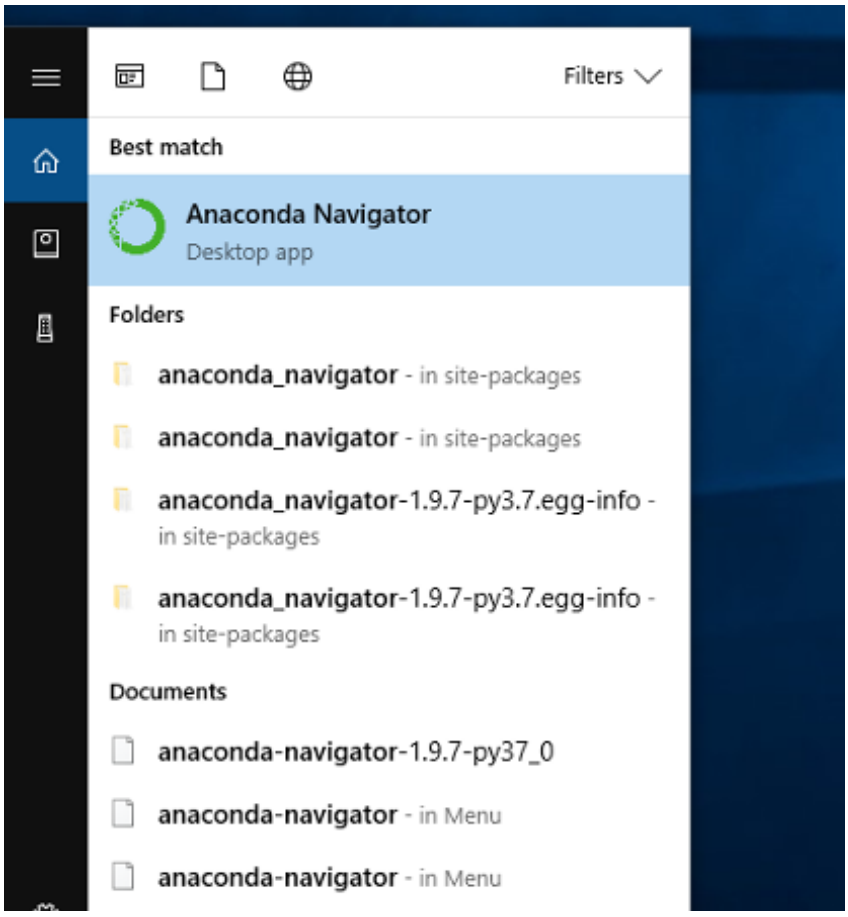


- Verify your installation by following the steps in **Verify Your Installation** section below

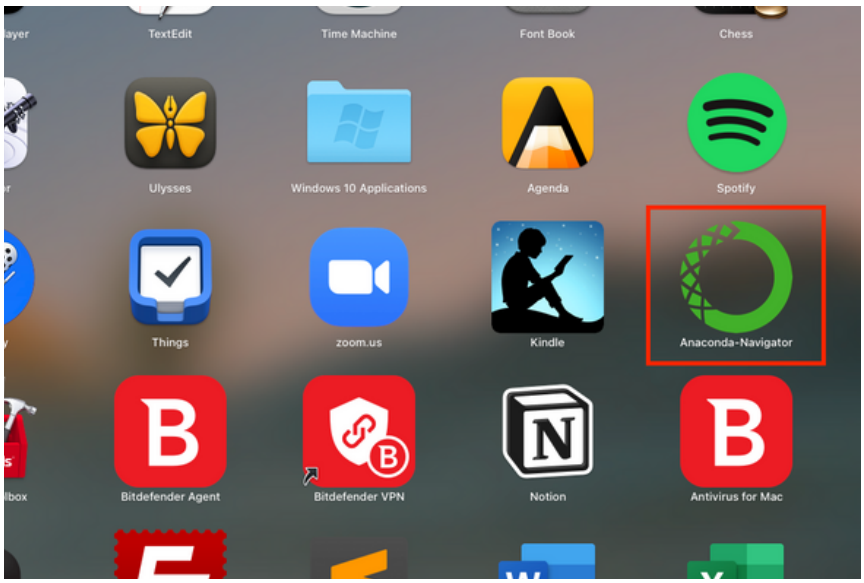
VERIFY YOUR INSTALLATION

You can confirm that Anaconda has been installed and working by opening up the Anaconda Navigator. The Anaconda Navigator is a graphical user interface that comes with your installation of Anaconda.

- **Windows**: Click Start, search or select Anaconda Navigator from the menu



- **macOS**: Open Launchpad, select Anaconda Navigator or use cmd+space to open Spotlight Search and type 'Navigator' to open the program.



JUPYTER NOTEBOOK

When you install Anaconda you automatically get Jupyter Notebook. The Jupyter Notebook is an open source web application that runs locally on your computer. This means that you do not have to be connected to the Internet to use it. You can use Jupyter to create and share documents that contain real code, which in our case is Python. For more information on the Jupyter Notebook go to Project Jupyter. If you want to go straight to the documentation you can find it here. It's a great idea to check out the documentation as it covers more topics than the brief introduction material to follow.

Where does the name Jupyter come from? It's actually a combination of the main programming languages that it supports, Julia, Python and R. Jupyter comes with the IPython kernel, which is going to allow us to write and run Python programs directly in a single document.

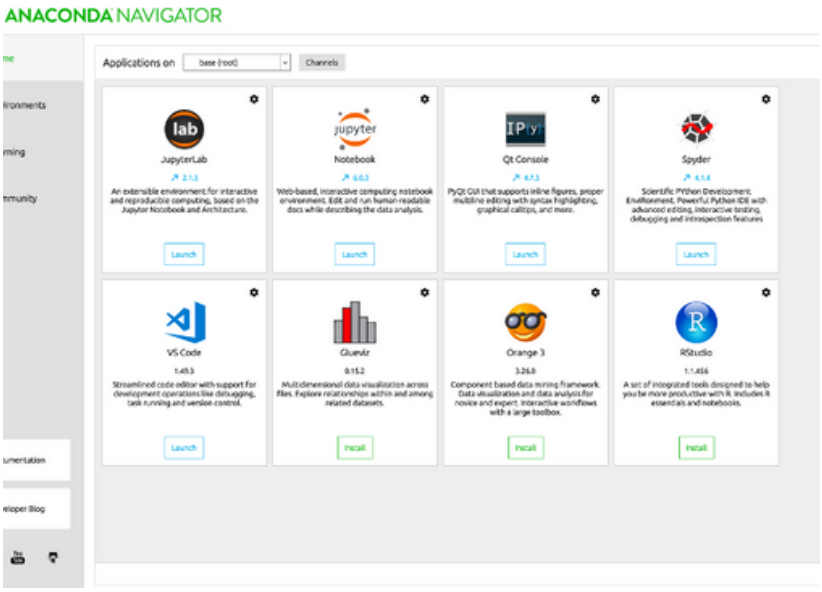
PYTHON CODE ACADEMY

STARTING ANACONDA

As mentioned above, Jupyter comes with the installation of Anaconda. To access Jupyter you need to have installed Anaconda. If you haven't already done go to page six of this guide for instructions. Once Anaconda has been successfully installed click on the Anaconda Navigator icon to start it up.

When Anaconda starts you will see the home screen with a menu on the left-hand side and a selection of applications in the main window.

18



Select Jupyter Notebook and it will automatically start up in a window of your default web browser. When Jupyter opens you will see a file tree of your main or home directory. From here you can choose where to save your Jupyter Notebook. I recommend you save it in an easy to reach location with a descriptive file name such as Python Projects or Jupyter Projects. At this point when you are in the file tree of Jupyter, you are running what's called the Notebook Server, not an actual Jupyter Notebook.

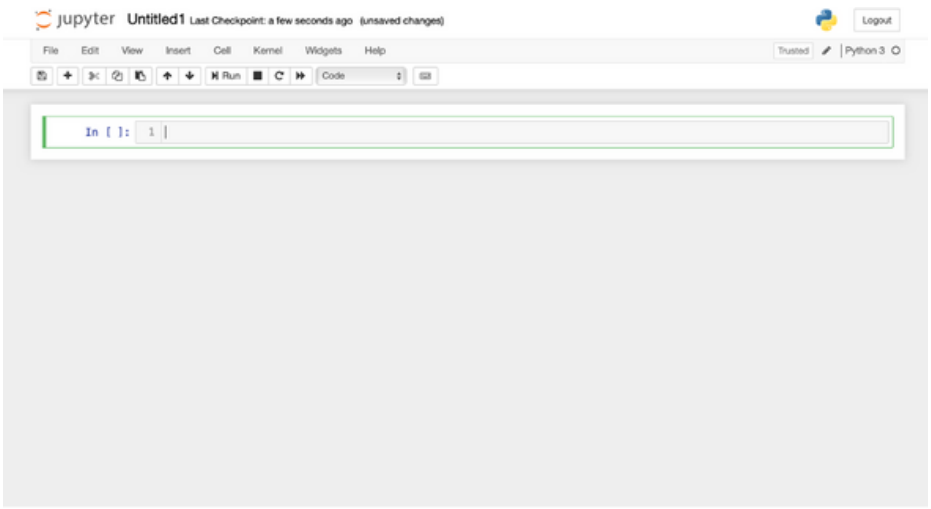
CREATING A NOTEBOOK

Let's learn how to create a Jupyter Notebook. From what is almost the top right-hand side of the Notebook Server window, click on the New button.

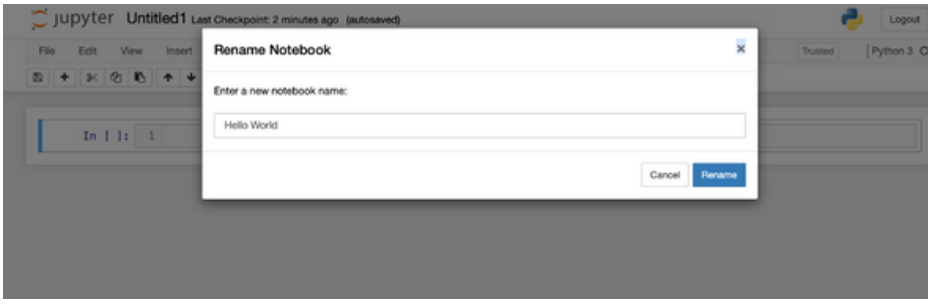
Depending on the versions of Python that you have installed on your computer your choices may be different from the image above. On my computer I only have Python 3 installed so that's the only choice I get. As Python 3 is the most recent version of Python that's what I'll select.

You should now see a web page that looks like the image below:

20



The first thing that you should do, which I often neglect, is to give your Notebook a name. At the moment it's called Untitled. Move your mouse over the text Untitled and click on it. A browser window should pop-up with a text field asking you to enter a new name for your notebook.

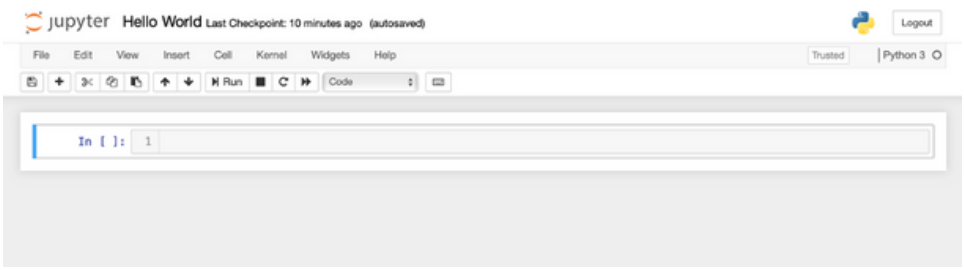


I've named mine Hello World. Rename yours now and then click the button Rename in the lower right corner of the pop-up window.

RUNNING CELLS

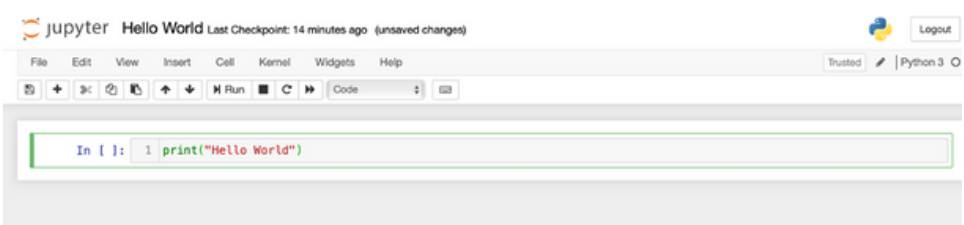
In a Jupyter Notebook you enter code, text, images and more in what are called Cells. By default when you start a new notebook the first cell will be set to use code. When I created a notebook in the previous step I selected Python 3 which means our cell can receive and run code using the Python 3 kernel.

You can tell what a cell is set to, by selecting it, and looking the at the list of icons in the main menu. Towards the end there is a dropdown list, which at this moment is set to **code**.



Currently our cell is empty, let's enter some code. Type in:

```
print("Hello World!")
```



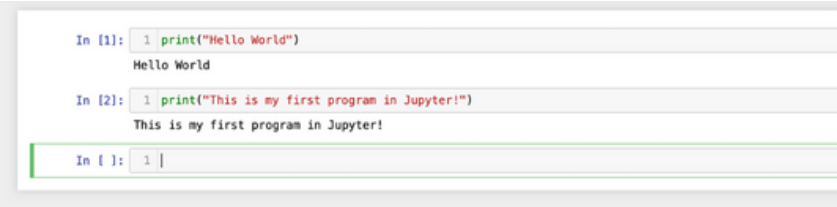
22

As you start to write longer programs in Jupyter you'll start to use multiple cells. You can run cells in order and share variables across cells.

Before you run a cell you'll notice on its left-hand side that there is the word **In** followed by a set of empty square brackets. After you run a cell the square brackets will auto-fill with a number which indicates the order that you ran the cells in.

You can execute the code in a cell in one of two ways:

- Press **Shift + Enter** keys
- Select the cell and click **Run** from the main menu bar



THE JUPYTER MENU

At the top of every Jupyter Notebook is the main menu. You can interact with this menu to do more with your notebook and I encourage you to experiment with it as there are some options in there that will make using Jupyter a lot easier. For example, inserting cells above or below other cells, running all cells at once or removing the output from all cells. At the time of writing, the menu includes:

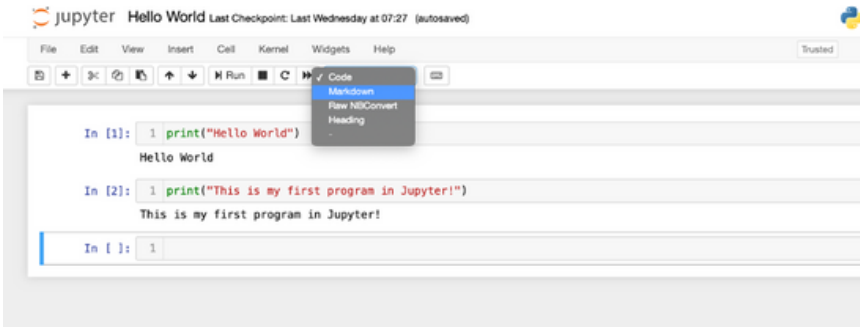
- File
- Edit
- View
- Insert
- Cell
- Kernel
- Widgets
- Help

ADDING TEXT CONTENT

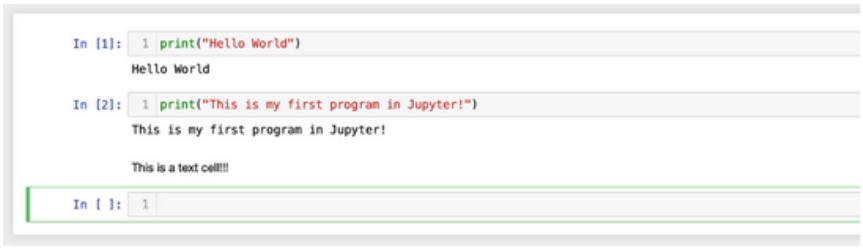
As well as adding Python code to cells within a Jupyter Notebook you can also add text content. This is one of the most powerful features of Jupyter as it allows you to add text to explain your code. In particular text cells are used by data scientists as they are often trying to communicate a story with their code and data analysis.

24

To start adding text to a cell, first select the cell and then go to the main menu and select Markdown from the cell type drop down.

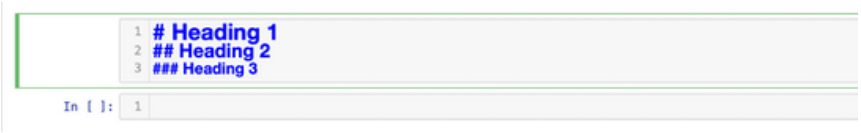


Type some text into the cell and run it.



As you can see from the screenshot above, when you run a text cell, the text is returned.

You can style the text within cells using Markdown. For example, to add a header you use the hash(#) symbol.



Which when run will output the following:

Heading 1

Heading 2

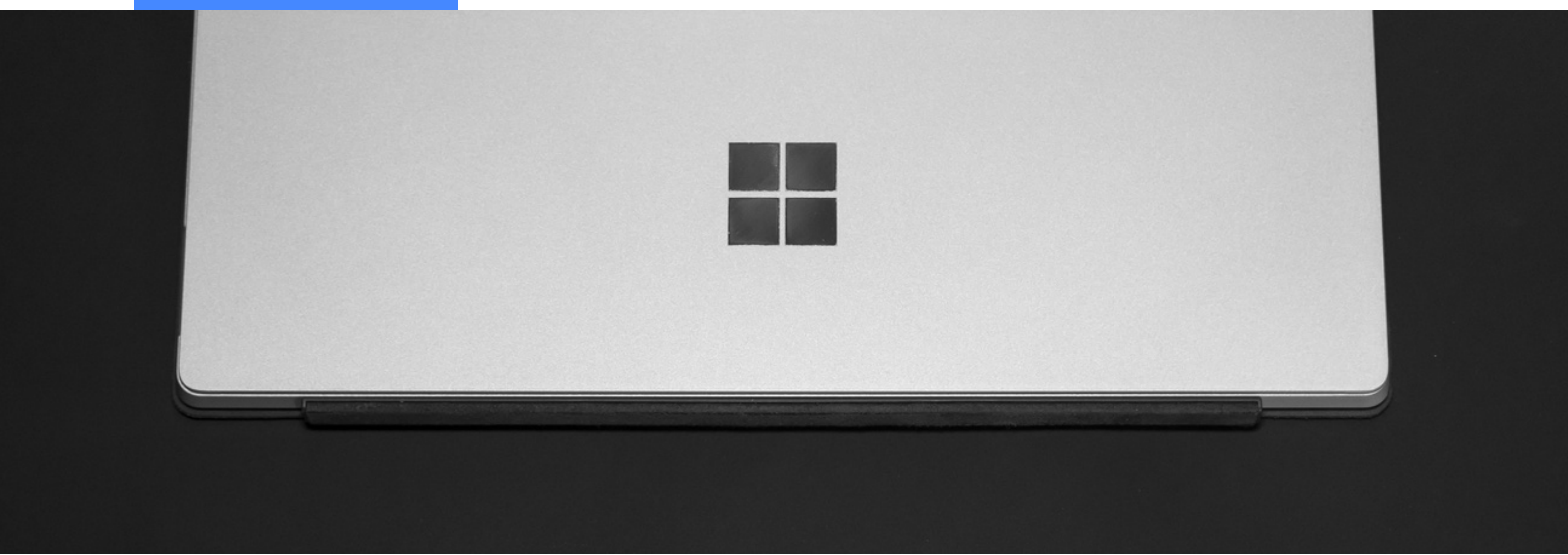
Heading 3

For more information on how to add Markdown and more examples of styling, take a look at the documentation here.

PYTHON ON WINDOWS

The Anaconda installation can take up a lot of space on your computer and this may not appeal to some users. You can install Python individually without Anaconda. The first thing to do is check if Python is already installed on your computer. To do this open up a command window, you can do this two different ways:

- Click the search icon in the lower right side of your window and type command or cmd for short
- Hold down the SHIFT key while right-clicking on your desktop and selecting Open Command Window here from the menu

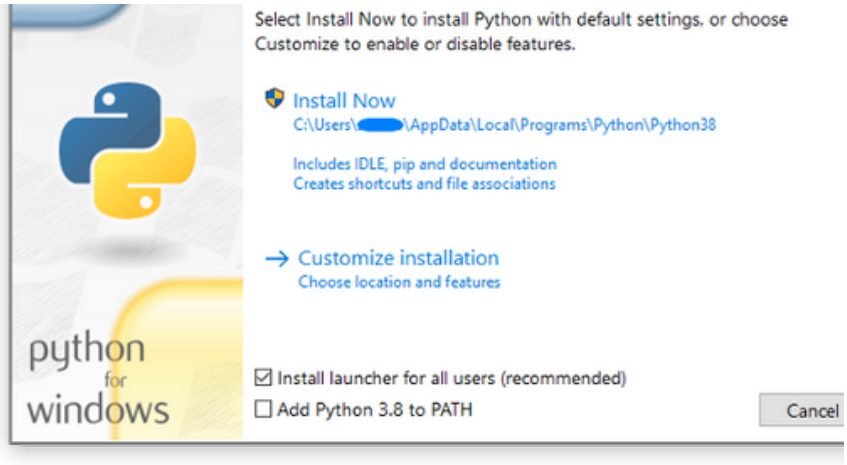


In the command window, type 'python', all lowercase. If you get the Python prompt (>>>), then Python is installed on your computer. If you get an error message telling you that python is not a recognised command, then Python is not installed on your computer. In this case you will need to install Python.

If you see a Python version lower then 3.6 you will also need to install a more up to date version of Python.

INSTALLING PYTHON

Go to https://python.org/ and move your move mouse over the Downloads menu option. You should see a button to the latest version of Python. You'll also notice that the link automatically detects what operating system you are on. Click on the button, and your Python download should start automatically. After the download has been complete, run the installer by double clicking on the downloaded file. **During the installation process, make sure that you select the option of 'Add Python to PATH'**. This will make running your Python programs a lot easier.



After the Python installation has been complete open up a command window and once more type python in lowercase. If Python has been successfully installed you should see the Python prompt (>>>) which means Python has been installed successfully and Windows has found the version that you have just installed.

VERIFYING YOUR INSTALLATION

Open a command window and enter the following line of code:

```
>>> print("Hello World!")
Hello World!
```

If you see the output “Hello World”, Python has been correctly installed on Windows.

PYTHON ON MACOS

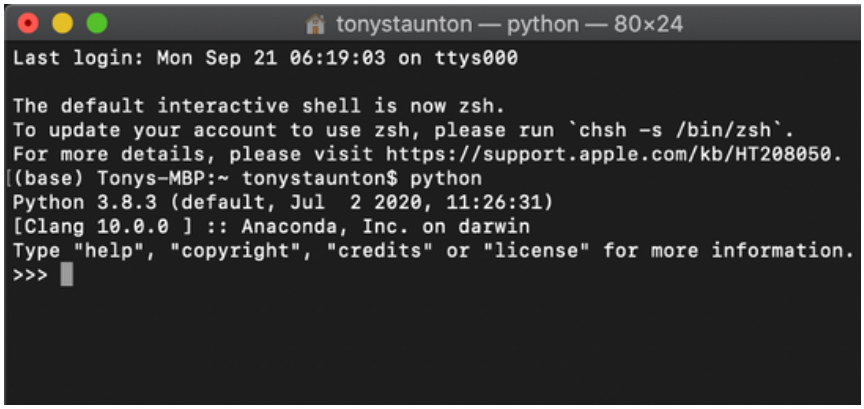
On macOS, Python is already installed but it is an outdated version needed to support some older programs. You do not want to learn using this version of Python.

Let's check what version of Python is installed by opening up a terminal window. You can open terminal in two ways:

- Go to, Applications >> Utilities >> Terminal
- Press the Command button and Spacebar at the same time to open Spotlight. Type terminal and press enter

When the terminal window opens type 'python', all lowercase, to see which version of Python is installed. You should see output telling you about the version of Python installed followed by the Python prompt (>>>).

30



Press CTRLand D keys together to exit Python.

To check if you have Python 3 installed, type 'python3', all lowercase, into the terminal window. Most likely you will get an error message, which is telling you that you do not have any version of Python 3 installed.

INSTALLING PYTHON

Go to https://python.org and move your mouse over the downloads menu option. You should see a button to download the latest version of Python. Also note that your operating system is automatically detected. Click on the button and your download should start automatically. After the download has finished run the installer.

When the installation has been completed enter the following command at the Python prompt:

```
>>> python3 --version
```

You should see output similar to:

```
Python 3.8.3
```

Your version number may be different depending on the version of Python available at your time of install.

From now on when you want to run Python in a terminal window make sure you type python3 to ensure that you are using Python 3.

VERIFY YOUR PYTHON INSTALLATION

Open a terminal window and type python3, all lowercase. At the Python prompt (>>>), type:

```
>>> print("Hello World!")
```

```
Hello World!
```

You should see the same output.

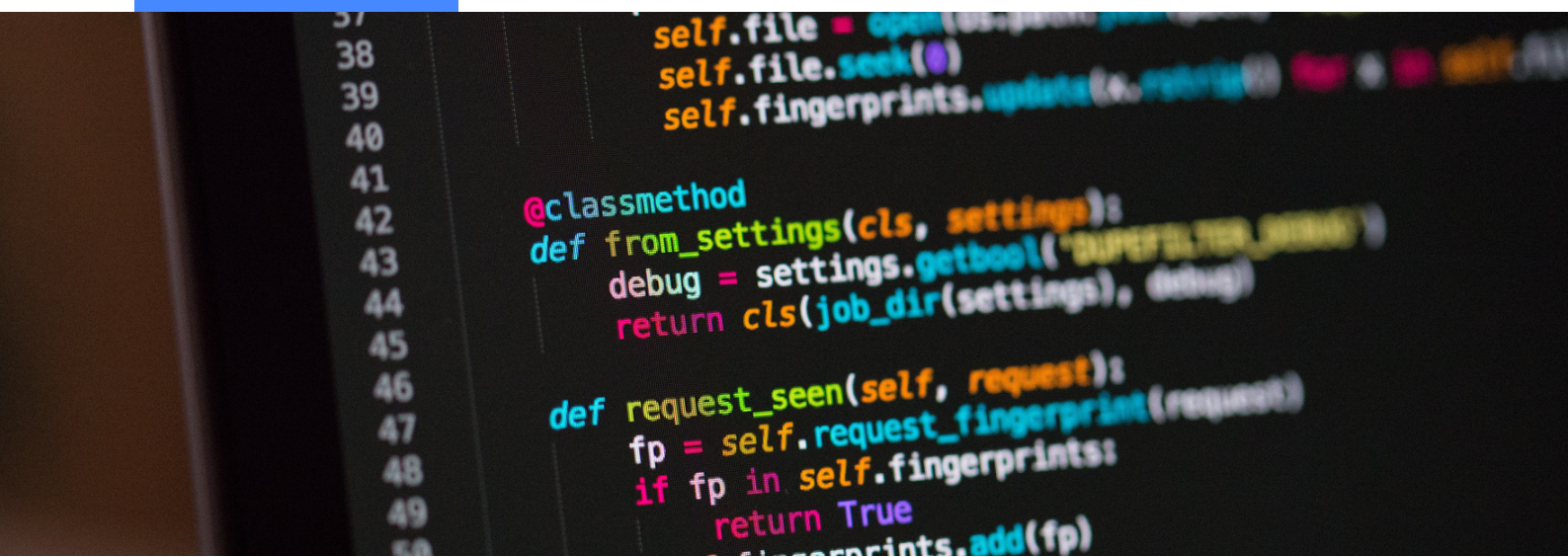
To close the Python session type CTRL and D or exit().

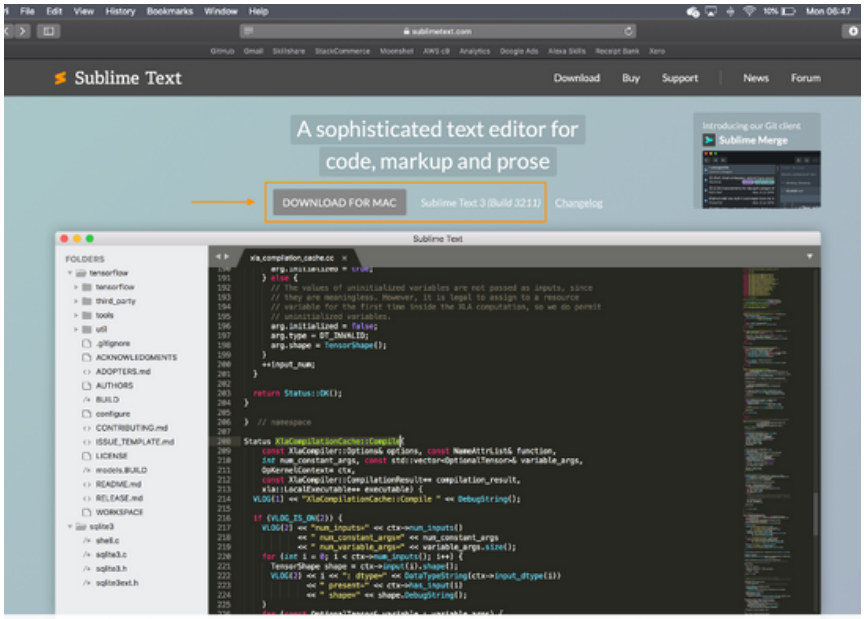
INSTALLING SUBLIME TEXT EDITOR

After you install Python you will need a code editor to write your code. Choosing a code editor is a very personal choice. Most developers experiment with a few editors, find what they like and don't like, and then make a decision on what feels right for them and helps to make their code writing as productive as possible.

If you already have a code editor and are happy with it then I would encourage you to stick with it. When it comes to using a code editor the choice is yours.

The section below discusses how to install and use Sublime Text Editor. First go to http://sublimetext.com/. Just below the main menu and heading you'll see the option to download. Note that the download option automatically detects your operating system.





WINDOWS

Once the download has finished run the installer and accept all its defaults.

MACOS

Once the download has finished, open it and then drag the Sublime icon into your Applications folder.

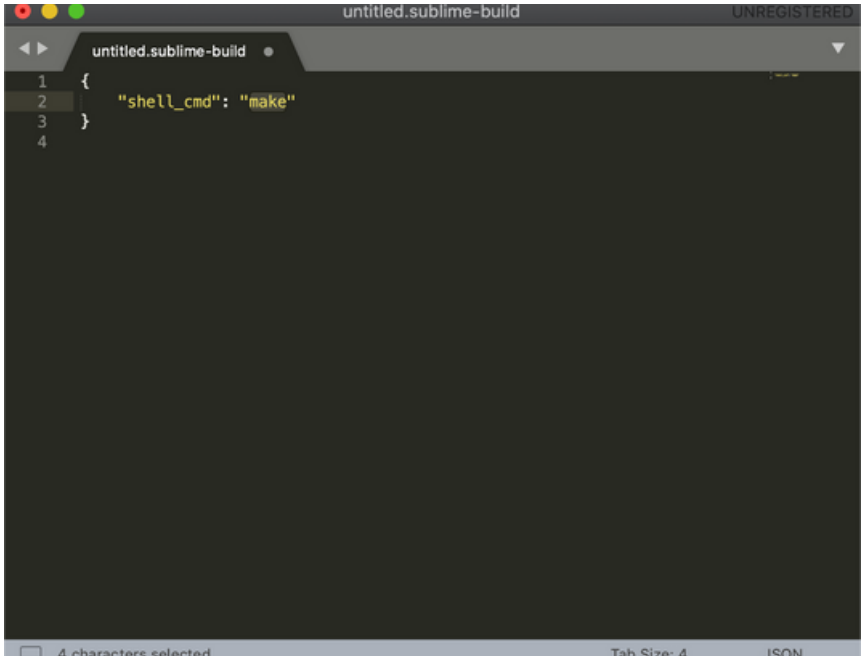
CONFIGURE SUBLIME TO USE THE CORRECT VERSION OF PYTHON

If, when you type 'python' into your terminal window and it automatically runs Python 3 then you don't need to follow this next step.

If, when you type 'python' into your terminal window and Python 2 is returned, you will need to configure Sublime Text to use the correct version of Python which is Python 3.

Launch Sublime Text Editor and go to Tools > Build System > New Build System. This will open up a new configuration file for you.

36



Delete whatever you see in this file and enter:

```
{
        "cmd": ["python3", "-u", "$file"],
}
```

This code block tells Sublime Text to use your system's python3 command when running your Python program files.

Save this file as Python3.sublime-build in the default directory that Sublime Text opens up when you choose Save As.

RUNNING A PROGRAM IN SUBLIME TEXT

Open up Sublime Text. The first thing to do is to save an empty file as a Python file by selecting File > Save as. Give your file the name helloworld.py

The .py extension tells Sublime Text the code in this file is written in Python. This will also allow Sublime Text to run the program correctly and perform text highlighting. After you have saved your file, enter:

```
print("Hello World!")
```

To run your program, select Tools > Build from the main menu or by pressing:

- CTRL-B on Windows
- Command-B on macOS

If you had to configure Sublime Text to work with Python 3, then to run your first program go to Tools > Build System > Python 3.
After you have done this once you can then use the shortcut keys:

- CTRL-B on Windows
- Command-B on macOS

After you run your program a terminal screen should appear at the bottom of Sublime Text, displaying your programs output.

RUNNING PYTHON FROM A TERMINAL

The majority of programs that you run will be from within your text editor. Sometimes, however, it may be necessary to run a program from the terminal. For example, you may want to run a program without editing it or viewing the code, you simply want it to run.

You can run a Python program from within the terminal if you have Python installed and you know how to navigate or access the directory where the file is stored.

Open up your terminal or command window and navigate to the location where your Python file is located. To change directory locations you can use the terminal command **cd** which means, change directory.

On Windows the command **dir**, which means directory, will show you a list of all the files that exist in the current directory. On macOS the same result is achieved using the **ls**, which stands for list. You can then use cd to navigate to the directory containing your Python file. When you get there, type:

python hello_world.py

You should see the output of your Python file in terminal/command window.

41

SUMMARY

When it comes to writing Python there are so many options available to you as a developer that it is impossible to put them all into an introduction guide. What I have tried to do with this guide is to give you some simple, easy to use options so that you can get up and running as quickly as possible.

There is no right or wrong way to get up and running with Python only the way that works best for you. I use a mix of Jupyter Notebook and Sublime Text Editor. These two tools are sufficient for almost 100% of my day-to-day coding tasks.

Learning to code is suppose to be fun and exciting so don't stress yourself out thinking too much about your Python environment setup. Remember, it's what works best for you and allows to you to write Python as productively as possible.

I hope you found this guide useful.

Tony