CSE 101
Introduction to Computers
Development / Tutorial / Lab Environment Setup

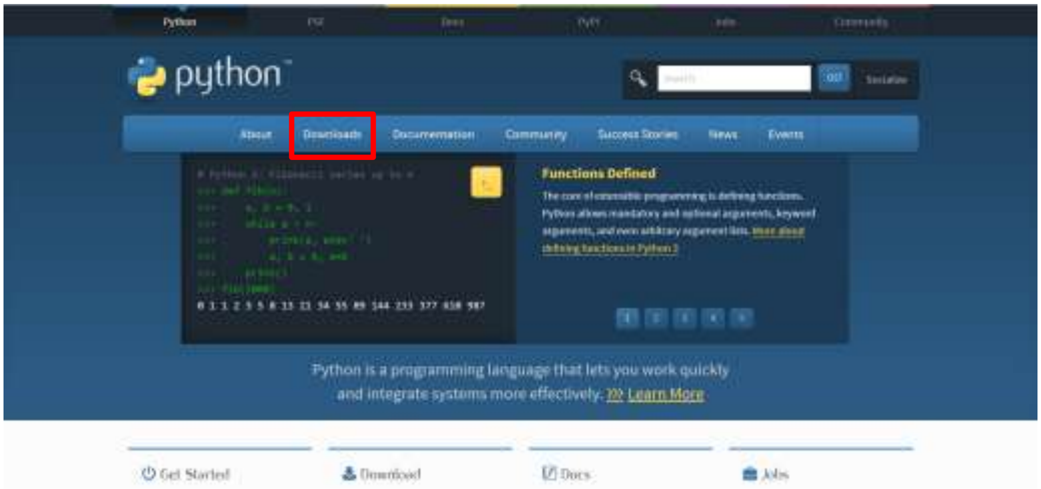
Purpose: The purpose of this lab is to setup software that you will be using throughout the term for learning about Python through tutorials, labs, and homework assignments.

Due: Ideally you should have this work completed by the end of the lab period. **If you don't complete it in the lab, you must finish it before the next class**; we will be using some of this software in our next class so it is important for you to have everything installed so you do not fall behind.

Please note that we have tried to make these instructions as clear as possible and test them on different operating systems, e.g., Windows 7 and 10, Mac OS X and Sierra, etc., however, some of you might encounter a problem and it might be a problem that we have not encountered before, so please work together on this lab as much as you can and help each other.

Step 1: Python Setup Instructions

1. Go to: https://www.python.org/ and select Downloads.



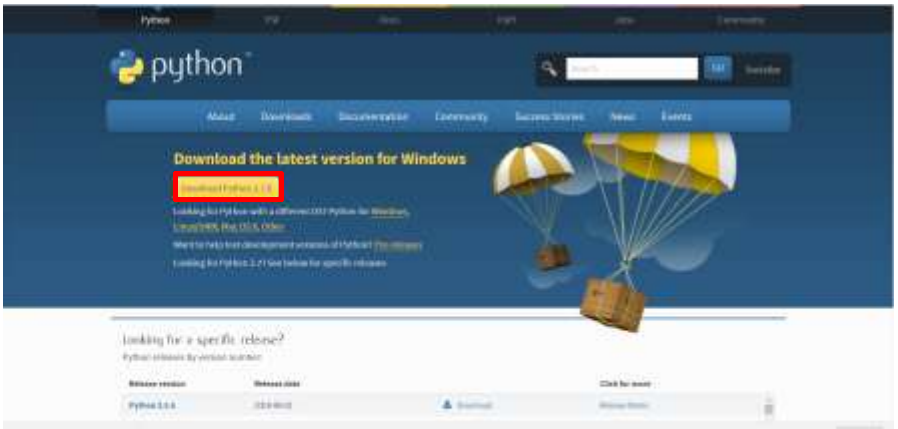
```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

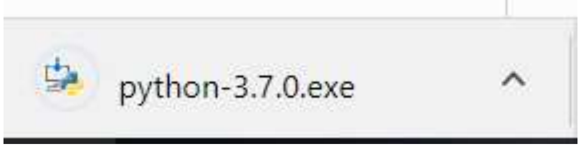
Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

This should bring you to a screen like this if you are running Windows:



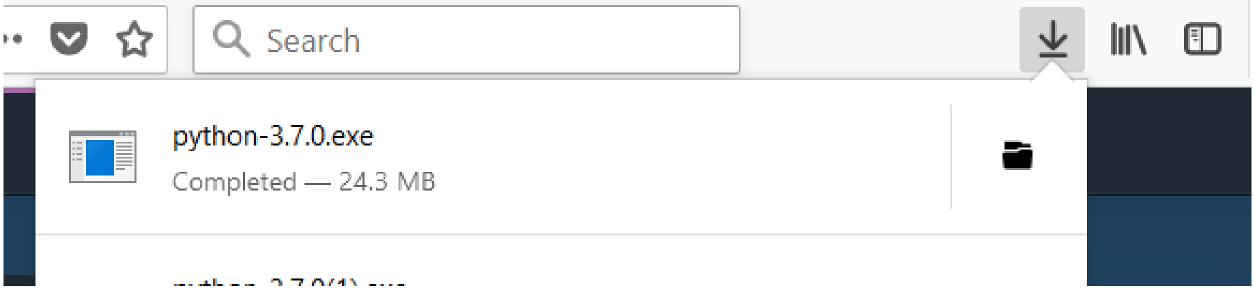
Click ‘Download Python 3.7.0’.

On Chrome, in the lower left corner of the browser screen, there will be a download indicator:

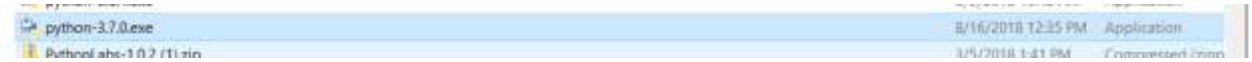


Click the up arrow on the right and select ‘Open’.

Note that Firefox indicates download with a down-pointing arrow in the top right of the browser window on the toolbar. In that case, click the down arrow, find the file ‘python-3.7.0.exe’ (which should be at the top of the list). Click the folder icon on the right.



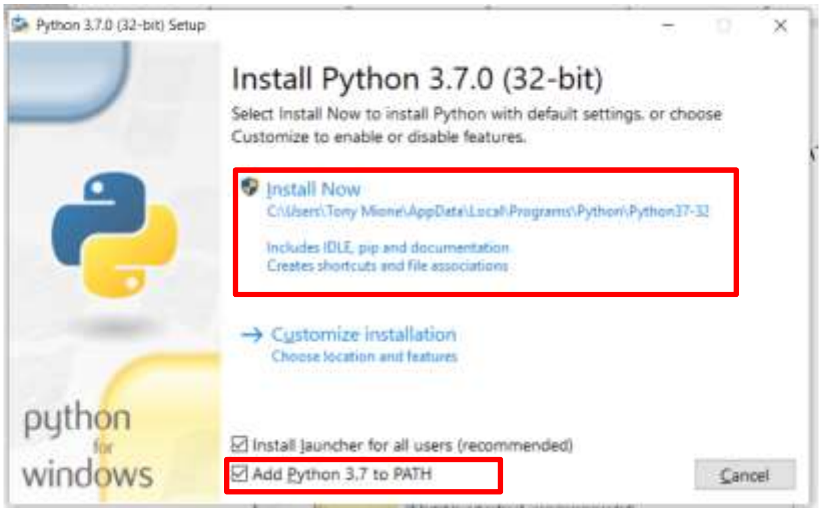
. Double click the file in the folder to start the install.



python-3.7.0.exe	8/16/2018 12:35 PM	Application

Step 1(a): Python Setup [Windows]

2. On the first screen of the setup program, make sure to check “Add Python 3.7 to PATH” and then select “Install Now”



Install Python 3.7.0 (32-bit)

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

Install Now
C:\Users\Tony Mione\AppData\Local\Programs\Python\Python37-32

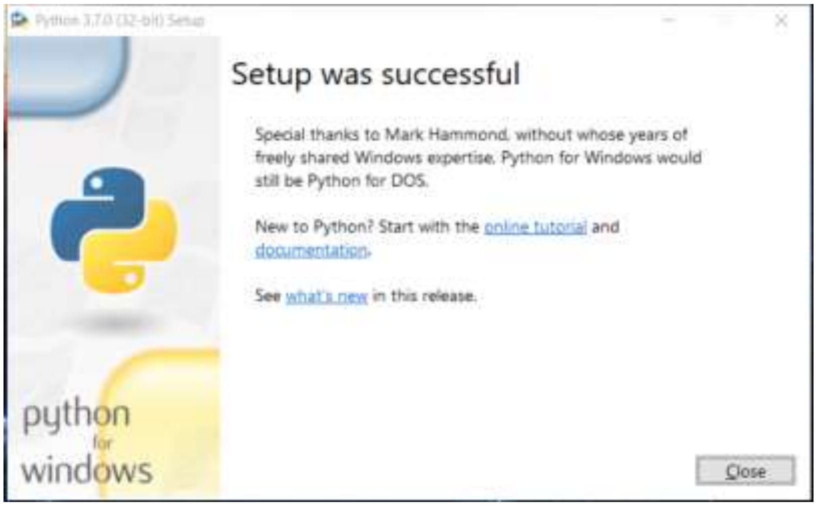
Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

☑ Add Python 3.7 to PATH

3. If the setup was successful you should see the following screen. If there was a problem please ask for help from your instructor, TA, or another student.



Setup was successful

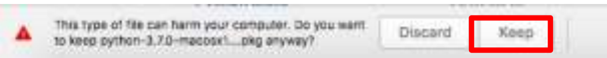
Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

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

See what's new in this release.

Step 1(b): Python Setup [MacOSX]

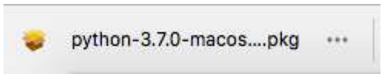
Follow the download instructions for Windows above. One difference is that in Chrome on Mac OSX, you may see this warning in the download location at the lower left of the browser window:



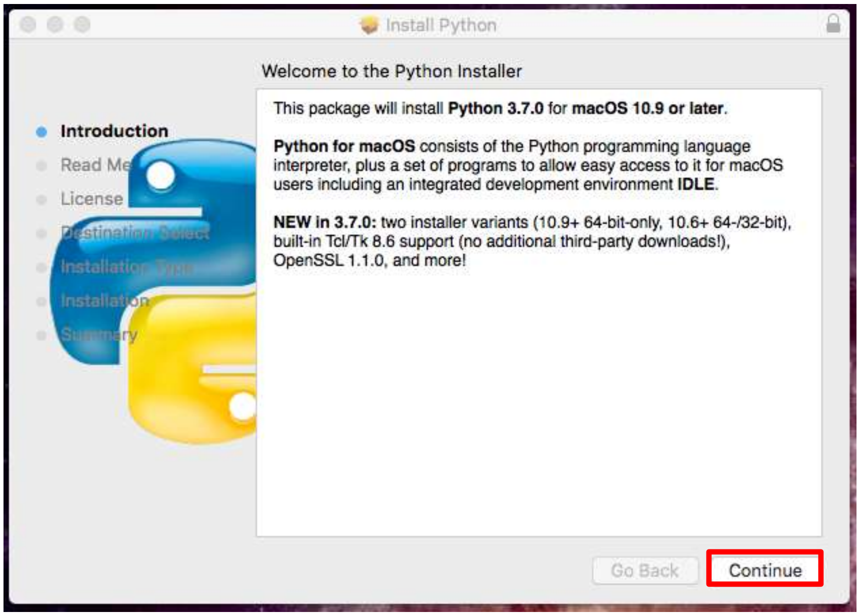
This type of file can harm your computer. Do you want to keep python-3.7.0-macosx1....pkg anyway?

Click ‘Keep’.

When the download is complete, you should see this in the lower right of the browser window:



After downloading the python 3.7.0 package, double click the icon to begin the install.



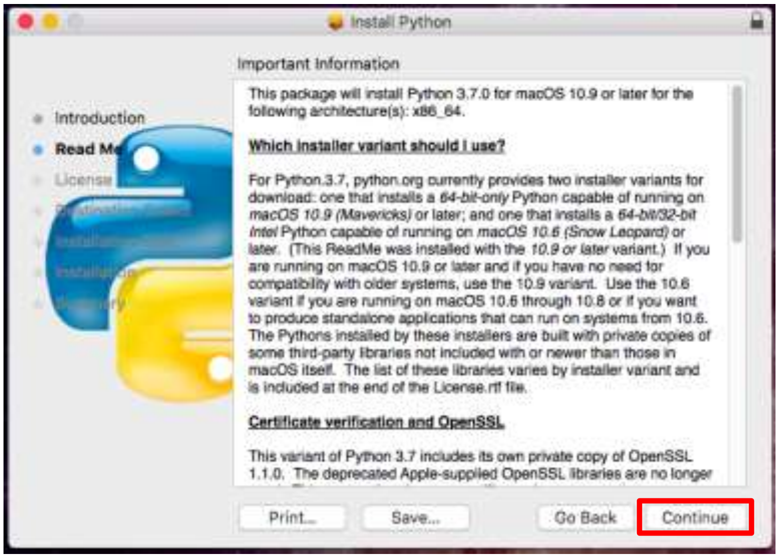
Welcome to the Python Installer

This package will install **Python 3.7.0** for **macOS 10.9 or later**.

Python for macOS consists of the Python programming language interpreter, plus a set of programs to allow easy access to it for macOS users including an integrated development environment **IDLE**.

NEW in 3.7.0: two installer variants (10.9+ 64-bit-only, 10.6+ 64-/32-bit), built-in Tcl/Tk 8.6 support (no additional third-party downloads!), OpenSSL 1.1.0, and more!

Click ‘Continue’.



Important Information

This package will install Python 3.7.0 for macOS 10.9 or later for the following architecture(s): x86_64.

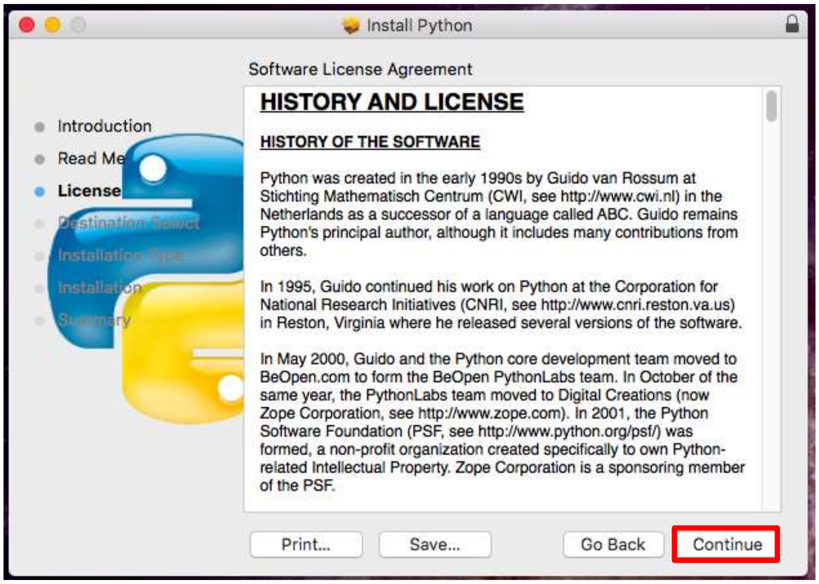
Which installer variant should I use?

For Python.3.7, python.org currently provides two installer variants for download: one that installs a *64-bit-only* Python capable of running on *macOS 10.9 (Mavericks)* or later; and one that installs a *64-bit/32-bit Intel* Python capable of running on *macOS 10.6 (Snow Leopard)* or later. (This ReadMe was installed with the *10.9 or later* variant.) If you are running on macOS 10.9 or later and if you have no need for compatibility with older systems, use the 10.9 variant. Use the 10.6 variant if you are running on macOS 10.6 through 10.8 or if you want to produce standalone applications that can run on systems from 10.6. The Pythons installed by these installers are built with private copies of some third-party libraries not included with or newer than those in macOS itself. The list of these libraries varies by installer variant and is included at the end of the License.rtf file.

Certificate verification and OpenSSL

This variant of Python 3.7 includes its own private copy of OpenSSL 1.1.0. The deprecated Apple-supplied OpenSSL libraries are no longer

Click ‘Continue’ on the ‘read me’ page. A license agreement dialog will pop up.



Software License Agreement

HISTORY AND LICENSE

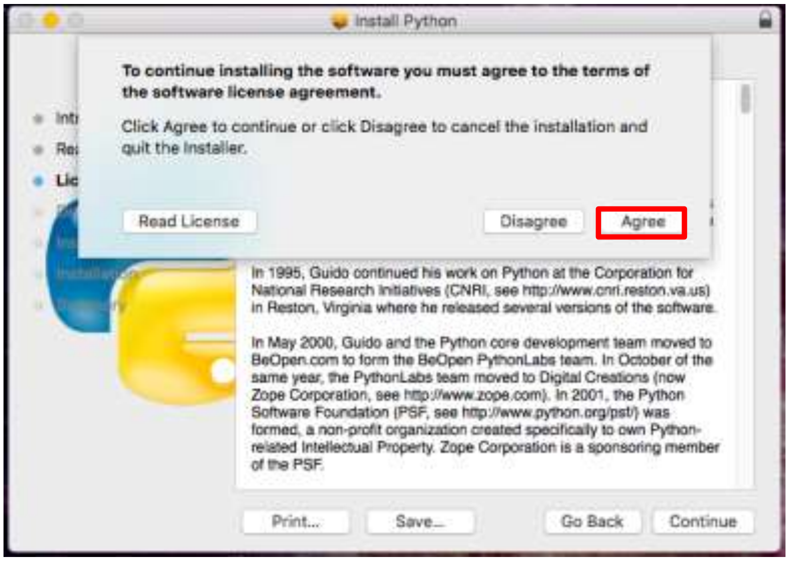
HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see http://www.zope.com). In 2001, the Python Software Foundation (PSF, see http://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

Click ‘Continue’ to get the actual License Agreement Dialog.



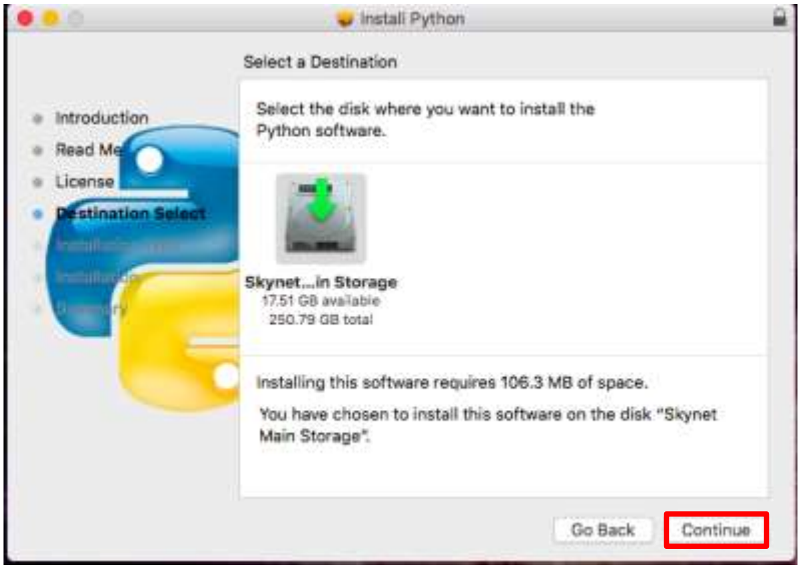
To continue installing the software you must agree to the terms of the software license agreement.

Click Agree to continue or click Disagree to cancel the installation and quit the Installer.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see http://www.zope.com). In 2001, the Python Software Foundation (PSF, see http://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

Click ‘Agree’.



Select a Destination

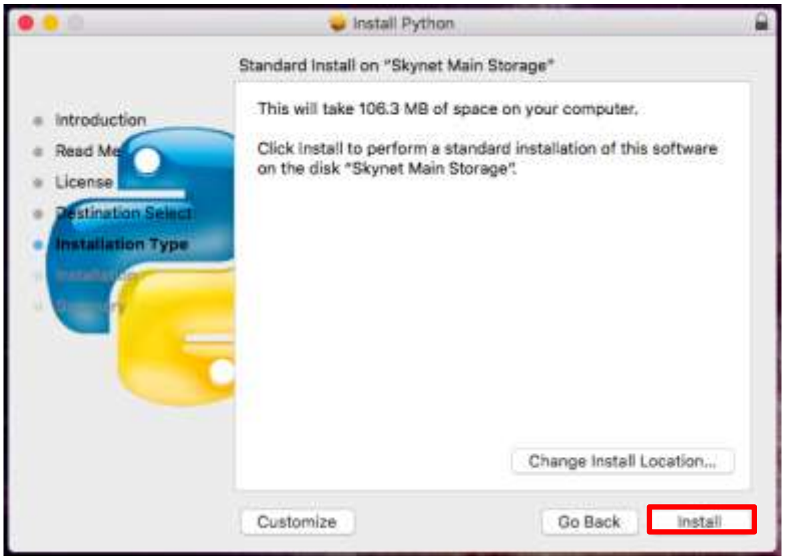
Select the disk where you want to install the Python software.

Skynet...in Storage
17.51 GB available
250.79 GB total

Installing this software requires 106.3 MB of space.

You have chosen to install this software on the disk “Skynet Main Storage”.

Click ‘Continue’.

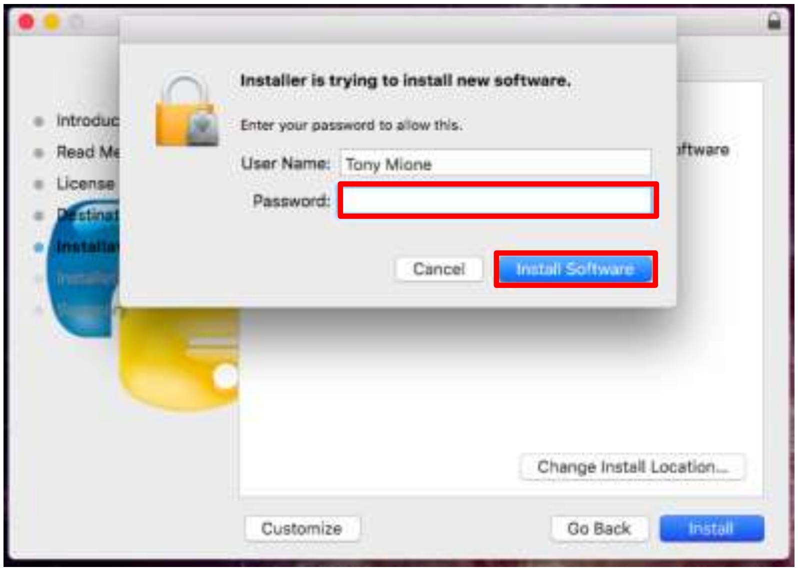


Standard Install on "Skynet Main Storage"

This will take 106.3 MB of space on your computer.

Click Install to perform a standard installation of this software on the disk "Skynet Main Storage".

This dialog gives the install location and amount of required storage space. You can change the install location or click 'Install' to accept the options shown.



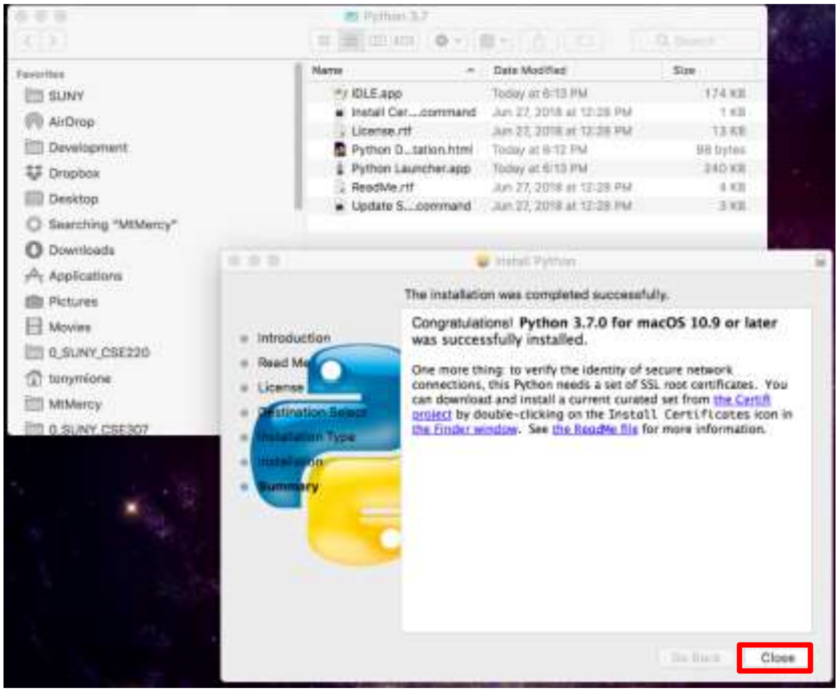
Installer is trying to install new software.

Enter your password to allow this.

User Name: Tony Mione

Password:

This dialog is prompting for an administrator password. Enter the password for your account and click 'Install Software'.



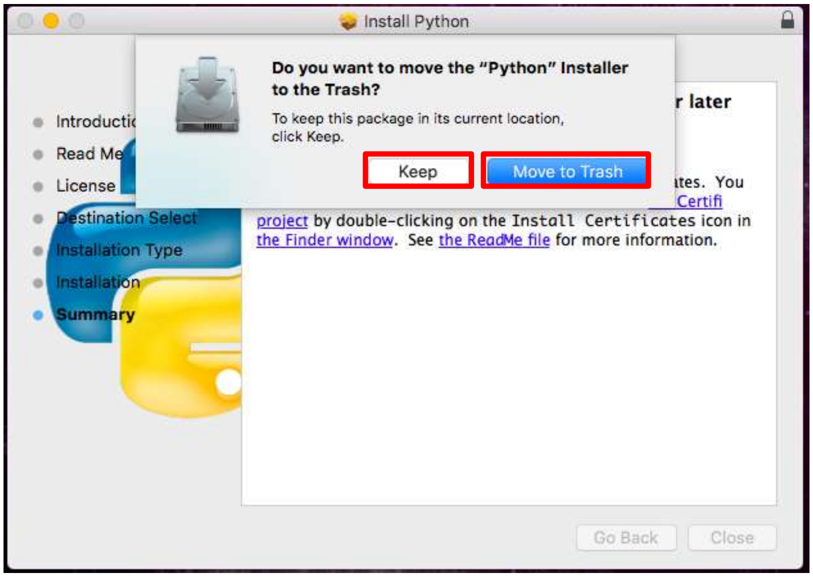
Name	Date Modified	Size
IDLE.app	Today at 6:13 PM	174 KB
Install Cer....command	Jun 27, 2018 at 12:28 PM	1 KB
License.rtf	Jun 27, 2018 at 12:28 PM	13 KB
Python D...tation.html	Today at 6:12 PM	98 bytes
Python Launcher.app	Today at 6:13 PM	340 KB
ReadMe.rtf	Jun 27, 2018 at 12:28 PM	4 KB
Update S....command	Jun 27, 2018 at 12:28 PM	3 KB

The installation was completed successfully.

Congratulations! **Python 3.7.0 for macOS 10.9 or later was successfully installed.**

One more thing: to verify the identity of secure network connections, this Python needs a set of SSL root certificates. You can download and install a current curated set from the Certifi project by double-clicking on the `Install Certificates` icon in the Finder window. See the ReadMe file for more information.

After installation, this dialog should indicate a successful install of Python 3.7.0 Click ‘Close’ to complete the installation.



Do you want to move the "Python" Installer to the Trash?

To keep this package in its current location, click Keep.

r later

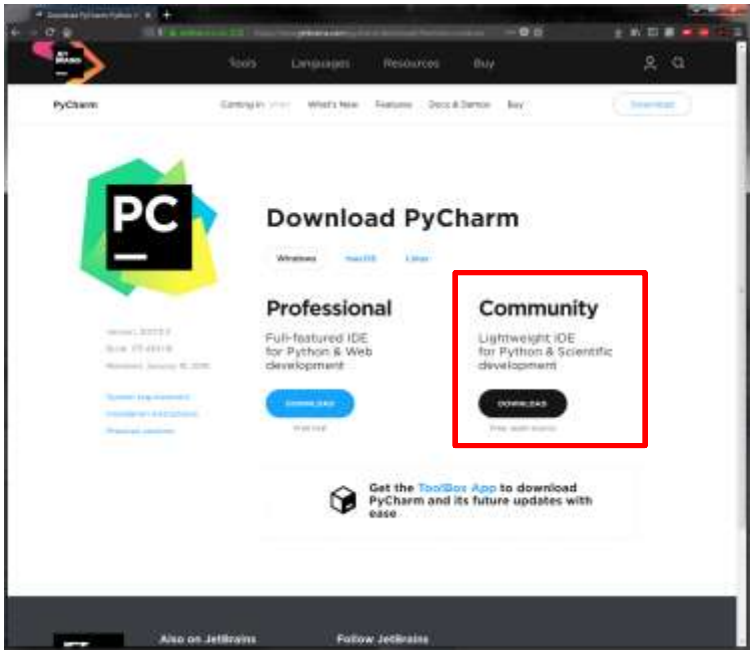
ates. You
Certifi
project by double-clicking on the `Install Certificates` icon in the Finder window. See the ReadMe file for more information.

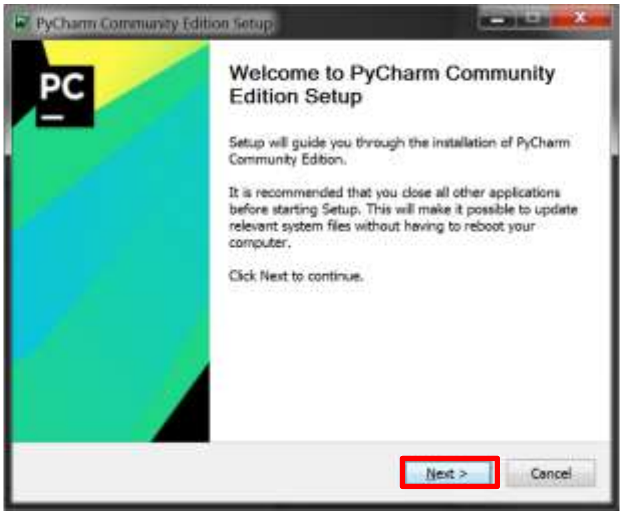
If you wish to discard the original install package, click ‘Move to Trash’. Otherwise, click ‘Keep’ if you think you may need to reinstall Python in the future.

Step 2(a): PyCharm Community Edition [Windows]

Python 3 includes a basic development environment called IDLE. IDLE is good for writing simple Python programs, but it lacks a lot of "comfort" features like live error-checking and debugging tools. As such, we recommend that you download the free, **Community Edition** version of the PyCharm IDE (integrated development environment) for use while completing your homework and lab assignments.

1. Go to http://www.jetbrains.com/pycharm/download to download the free **Community Edition** of PyCharm, and click on the download link. When the download completes, double-click the downloaded file to run the installer. The setup windows are shown below with options to select in red boxes:


Tools
Languages
Resources
Buy
PyCharm
What's New
Features
Docs & Demos
Buy
Download PyCharm
Windows
macOS
Linux
Professional
Full-featured IDE for Python & Web development
DOWNLOAD
Community
Lightweight IDE for Python & Scientific development
DOWNLOAD
Get the ToolBox App to download PyCharm and its future updates with ease
Also on JetBrains
Follow JetBrains

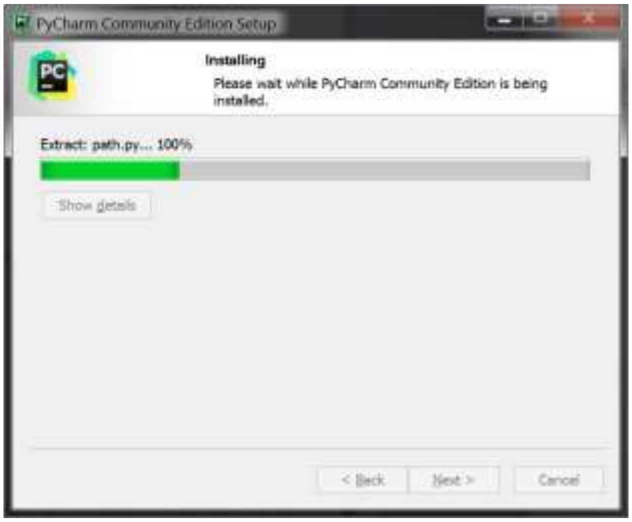


Welcome to PyCharm Community Edition Setup

Setup will guide you through the installation of PyCharm Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

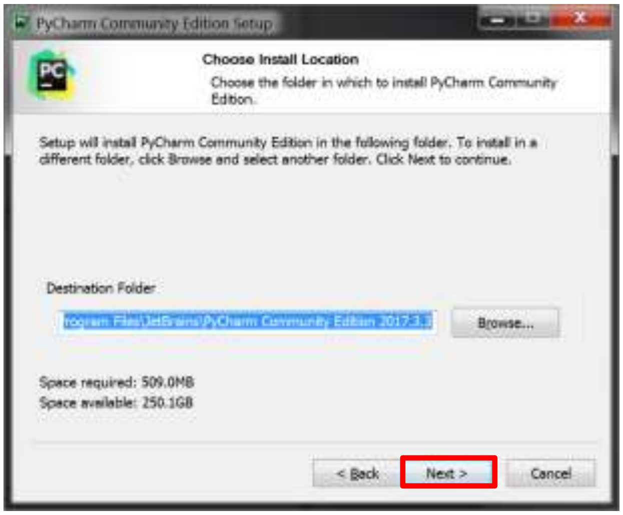
Click Next to continue.



Installing

Please wait while PyCharm Community Edition is being installed.

Extract: path.py... 100%



Choose Install Location

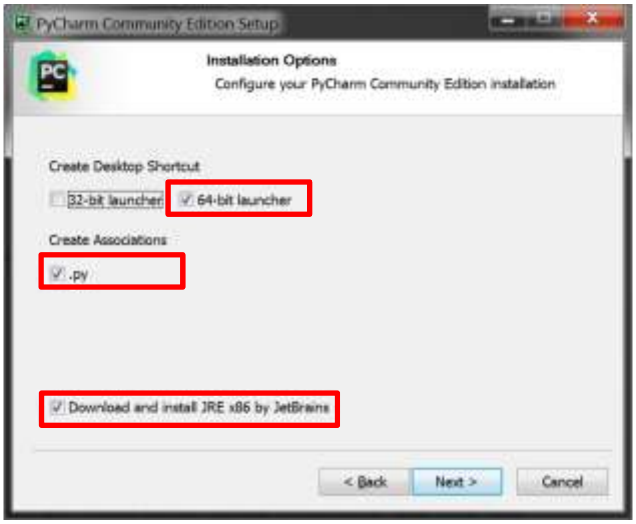
Choose the folder in which to install PyCharm Community Edition.

Setup will install PyCharm Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

rogram Files\JetBrains\PyCharm Community Edition 2017.3.3

Space required: 509.0MB
Space available: 250.1GB



Installation Options

Configure your PyCharm Community Edition installation

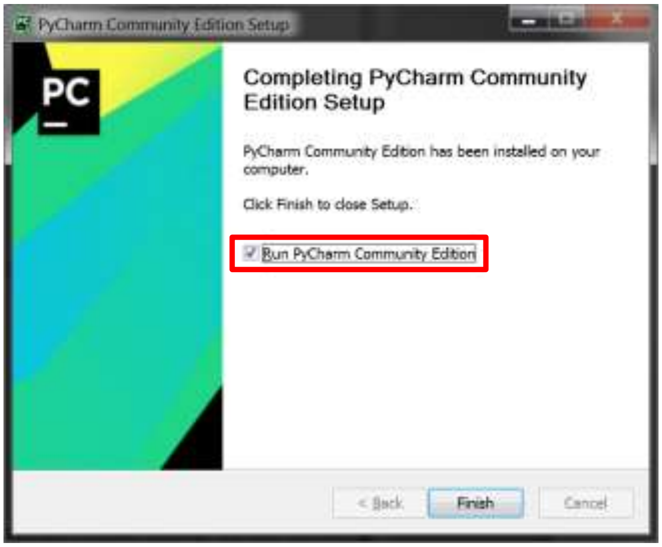
Create Desktop Shortcut

□ 32-bit launcher ☑ 64-bit launcher

Create Associations

☑ .py

☑ Download and install JRE x86 by JetBrains



Completing PyCharm Community Edition Setup

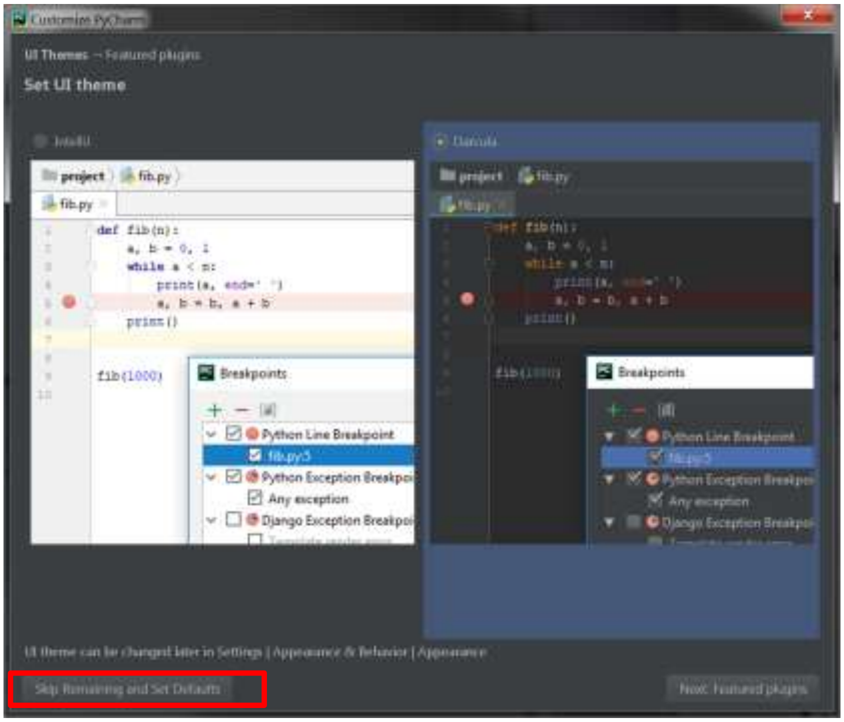
PyCharm Community Edition has been installed on your computer.

Click Finish to close Setup.

☑ Run PyCharm Community Edition

2. Configure PyCharm as follows:

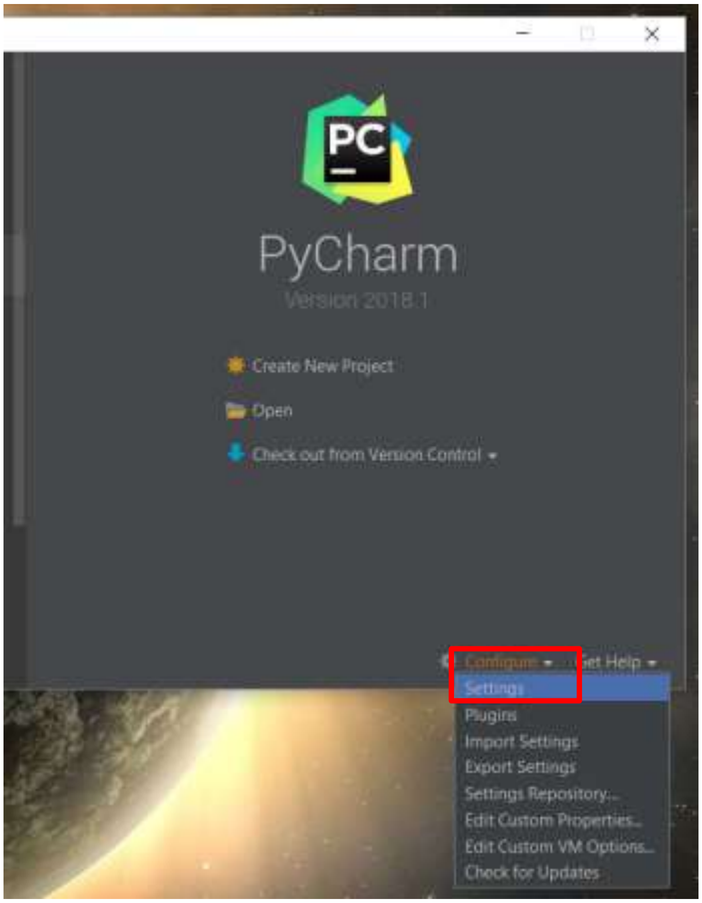
a) On initial startup, you can set a UI theme. Select whichever you like, IntelliJ or Dracula, then select “Skip Remaining and Set Defaults”



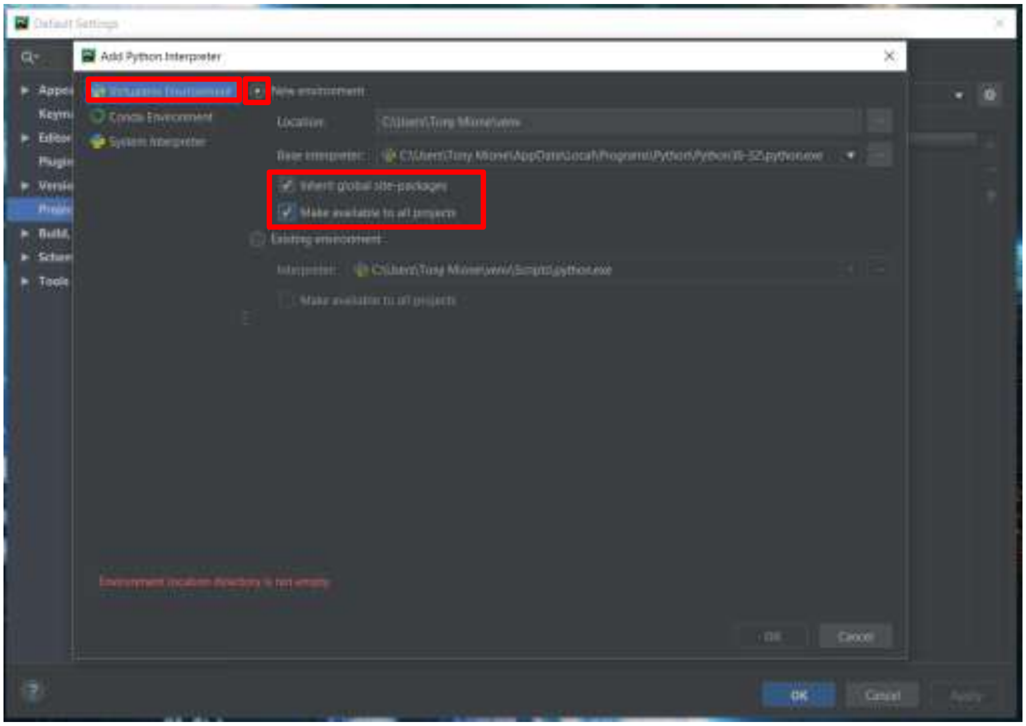
```
def fib(n):
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a + b
    print()

fib(1000)
```

UI theme can be changed later in Settings | Appearance & Behavior | Appearance

B) PyCharm will launch. Then select "Configure" from the bottom of the window that appears and choose “Settings” from the pop-up menu, and then click “Project Interpreter” on the left side of the Default Settings window



c) Then click the ⚙ in the upper right at the end of and select "Add…" and then "Virtualenv Environment" select "New environment", "Inherit global site-packages" and "Make available to all projects"



Default Settings

Add Python Interpreter

Virtualenv Environment

Conda Environment

System Interpreter

● New environment

Location: C:\Users\Tony Mione\venv

Base interpreter: C:\Users\Tony Mione\AppData\Local\Programs\Python\Python36-32\python.exe

☑ Inherit global site-packages

☑ Make available to all projects

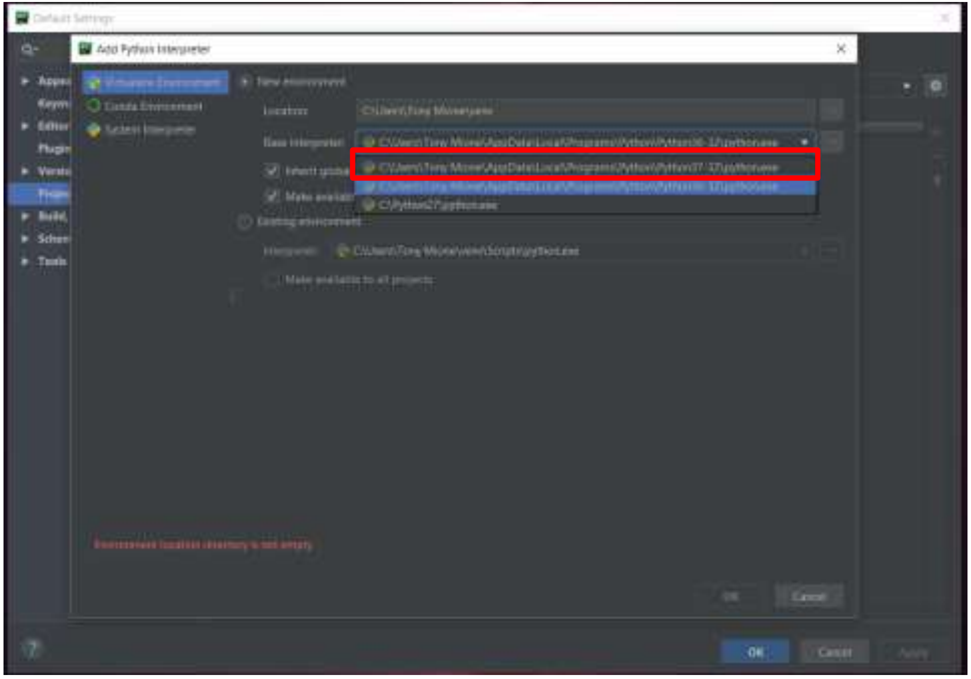
○ Existing environment

Interpreter: C:\Users\Tony Mione\venv\Scripts\python.exe

□ Make available to all projects

Environment location directory is not empty

The path to the Python interpreter that you just installed in Step 1 should be populated in the "Base interpreter" box. If it is, select "OK". If not, click the ▼ button next to the interpreter path. If the newest version of Python is listed in the Base interpreter dropdown, select it:



Default Settings

Add Python Interpreter

Virtualenv Environment

Conda Environment

System Interpreter

● New environment

Location: C:\Users\Tony Mione\venv

Base interpreter: C:\Users\Tony Mione\AppData\Local\Programs\Python\Python36-32\python.exe

C:\Users\Tony Mione\AppData\Local\Programs\Python\Python37-32\python.exe

C:\Users\Tony Mione\AppData\Local\Programs\Python\Python36-32\python.exe

C:\Python27\python.exe

☑ Inherit global

☑ Make availab

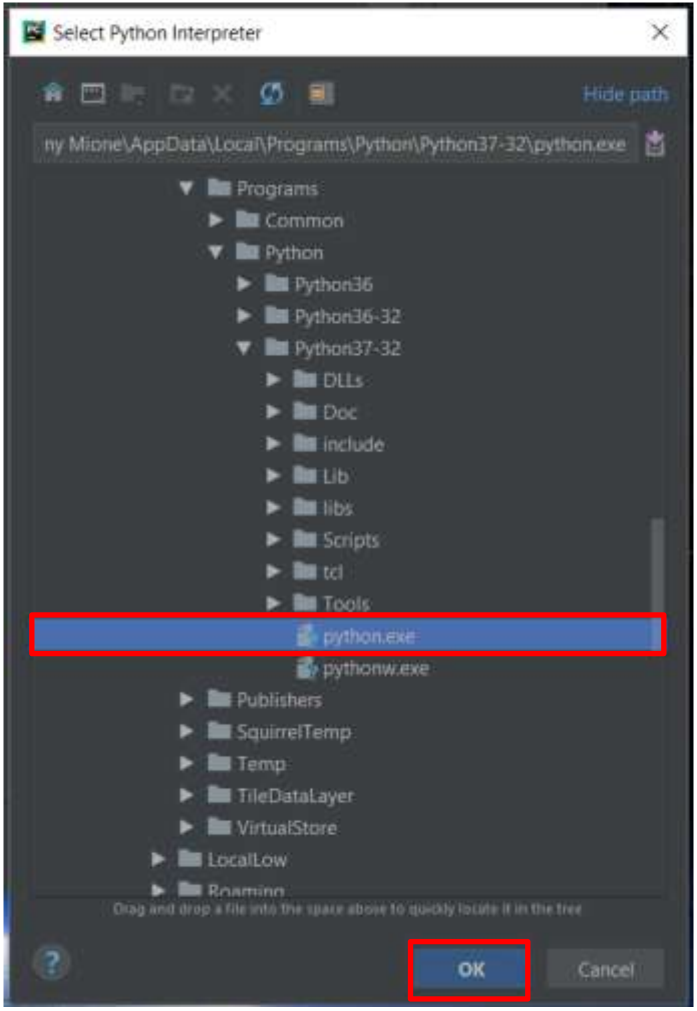
○ Existing environment

Interpreter: C:\Users\Tony Mione\venv\Scripts\python.exe

□ Make available to all projects

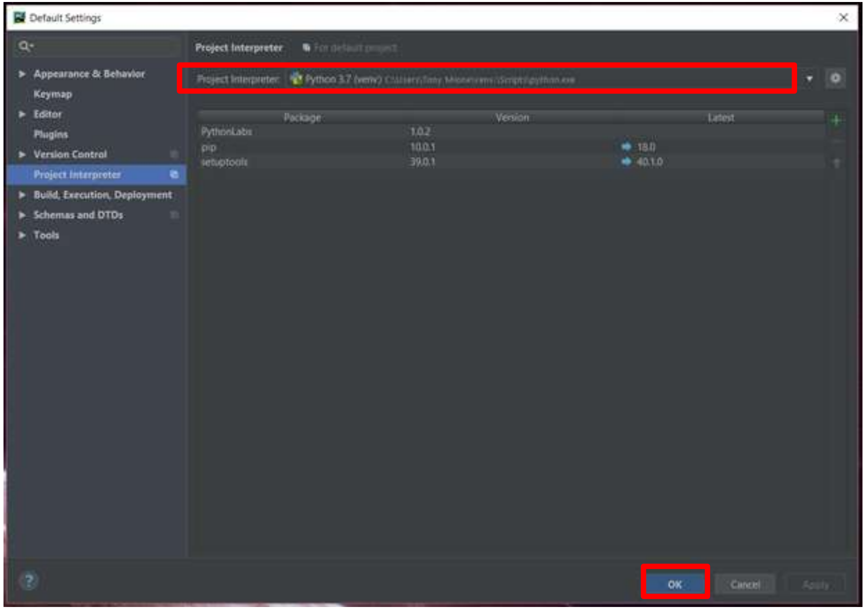
Environment location directory is not empty

If the box is blank select the ... and navigate to where the Python was installed, select it and then select OK as is shown below, note that your path may be different.



ny Mione\AppData\Local\Programs\Python\Python37-32\python.exe

d) After setting up correctly the Default Setting windows should look as follows:



Project Interpreter: Python 3.7 (venv) C:\Users\Tony Mione\venv\Scripts\python.exe

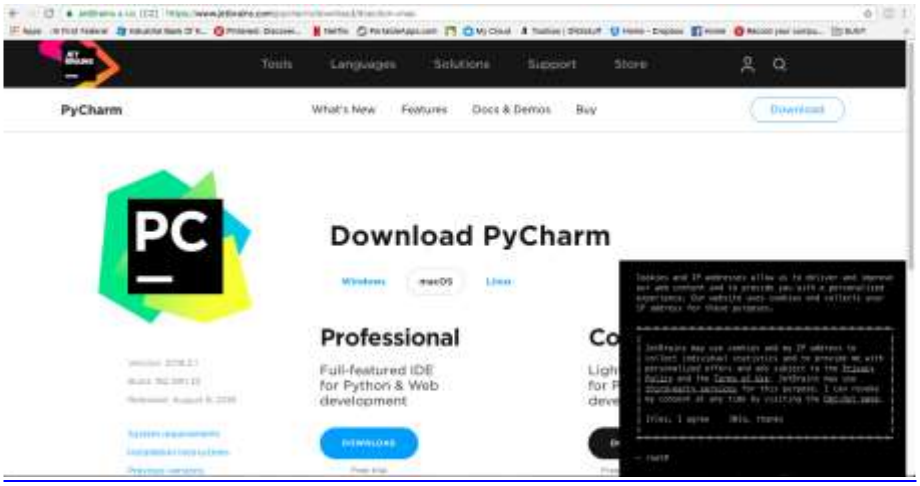
Package	Version	Latest
PythonLabs	1.0.2	
pip	10.0.1	18.0
setuptools	39.0.1	40.1.0

Select OK. Now PyCharm will always use the correct version of Python (Python 3) for your projects.

Step 2(b): PyCharm Community Edition [MacOS]

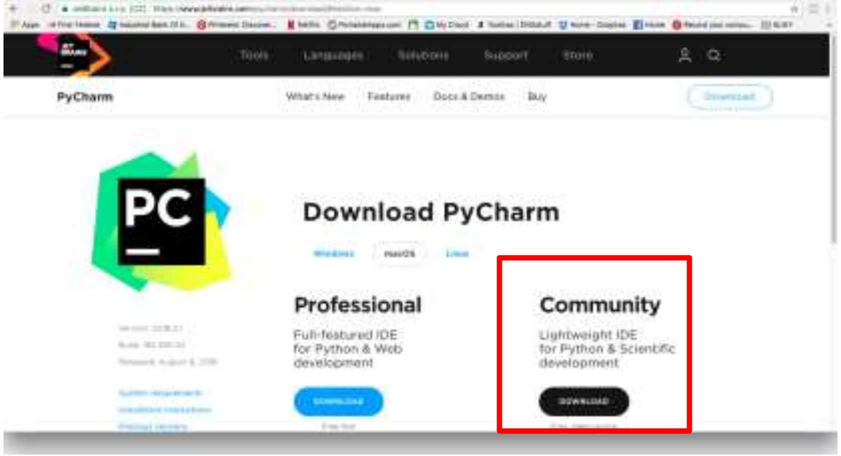
Python 3 includes a basic development environment called IDLE. IDLE is good for writing simple Python programs, but it lacks a lot of "comfort" features like live error-checking and debugging tools. As such, we recommend that you download the free, **Community Edition** version of the PyCharm IDE (integrated development environment) for use while completing your homework and lab assignments.

1. Go to http://www.jetbrains.com/pycharm/download

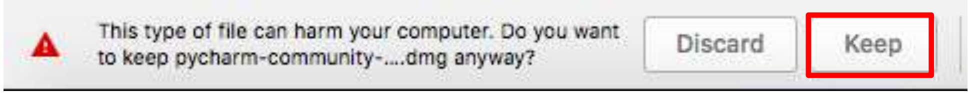


```
Cookies and IP addresses allow us to deliver and improve
our web content and to provide you with a personalized
experience. Our website uses cookies and collects your
IP address for these purposes.
-----------------------------------------------------------
| JetBrains may use cookies and my IP address to          |
| collect individual statistics and to provide me with    |
| personalized offers and ads subject to the Privacy      |
| Policy and the Terms of Use. JetBrains may use          |
| third-party services for this purpose. I can revoke     |
| my consent at any time by visiting the Opt-Out page.    |
|                                                         |
| [Y]es, I agree    [N]o, thanks                          |
-----------------------------------------------------------

~ root#
```

If there is a dark box with text in the lower right of the browser window, then click in the dark box on the lower right next to 'root#'. Type 'Y'

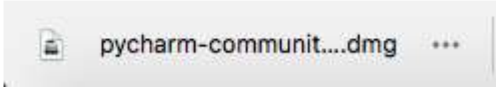


Next, click 'Download' under the description of the 'Community' edition which is free. This will download the free **Community Edition** of PyCharm, and click on the download link. Depending on settings on your mac, A dialog in the lower left corner may ask if you want to continue the download. If so, click 'Keep'.

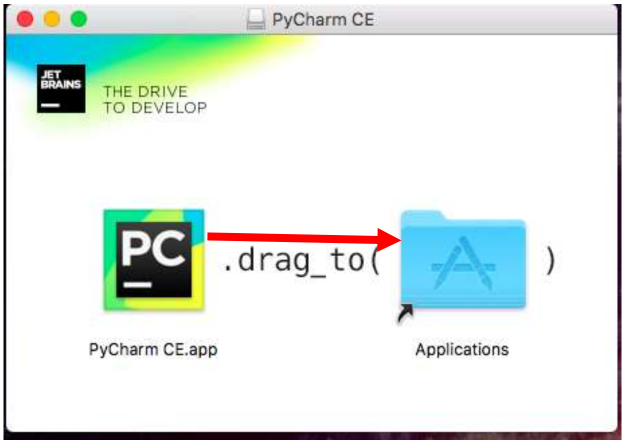


This type of file can harm your computer. Do you want to keep pycharm-community-....dmg anyway?

After the download completes, you should see this in the lower right.

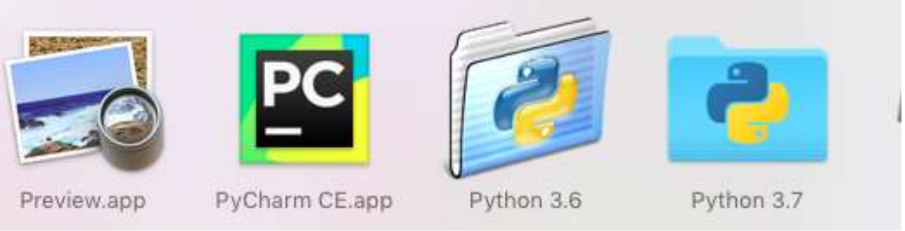


Double click on the name to start the install of Pycharm.

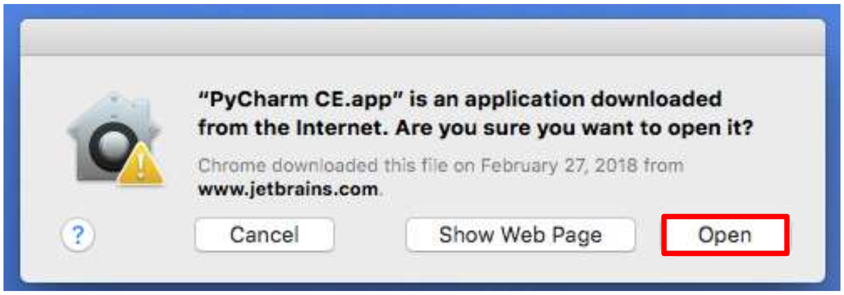


Press the left mouse button on the PyCharm icon to the left and drag the icon over the Application folder window on the right. An unpack progress bar will open up as PyCharm is copied to the Application folder and installed.

If you open the Application folder after the install is complete, you should see the PyCharm icon:



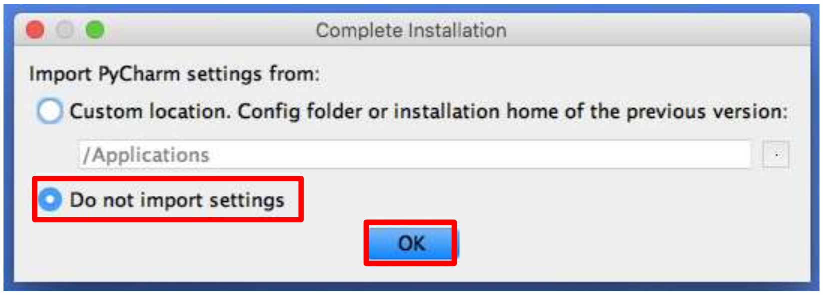
The first time you run PyCharm on the Mac, you will likely see a dialog indicating the application was downloaded from the network and it will ask to confirm that you want to run the application.


"PyCharm CE.app" is an application downloaded from the Internet. Are you sure you want to open it?

Chrome downloaded this file on February 27, 2018 from **www.jetbrains.com**.

Click 'Open' to allow PyCharm to start.

Next, PyCharm will ask if you want to import settings. Leave 'Do not import settings' selected and click 'Ok'.


Complete Installation

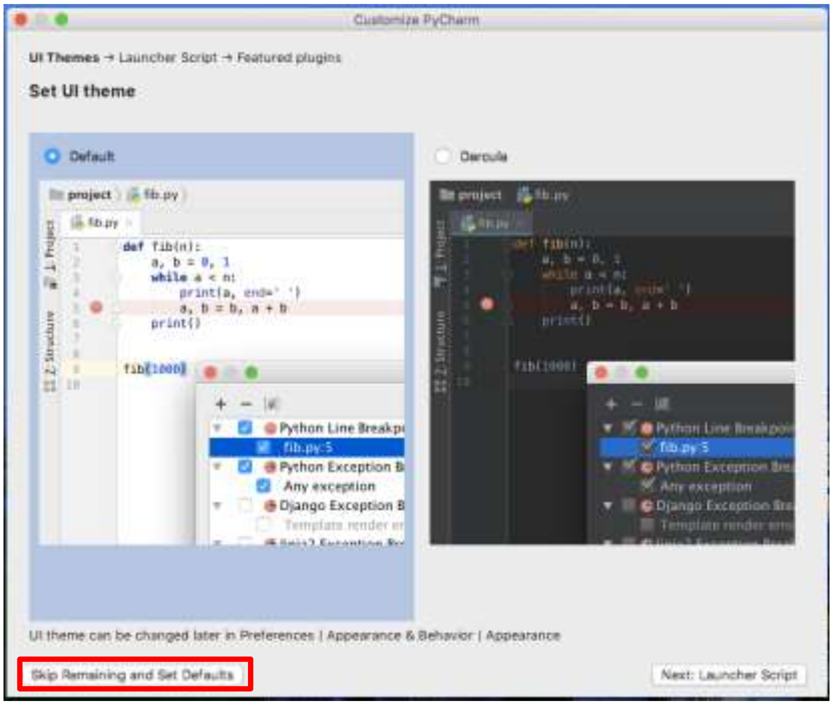
Import PyCharm settings from:

○ Custom location. Config folder or installation home of the previous version:

/Applications

● Do not import settings

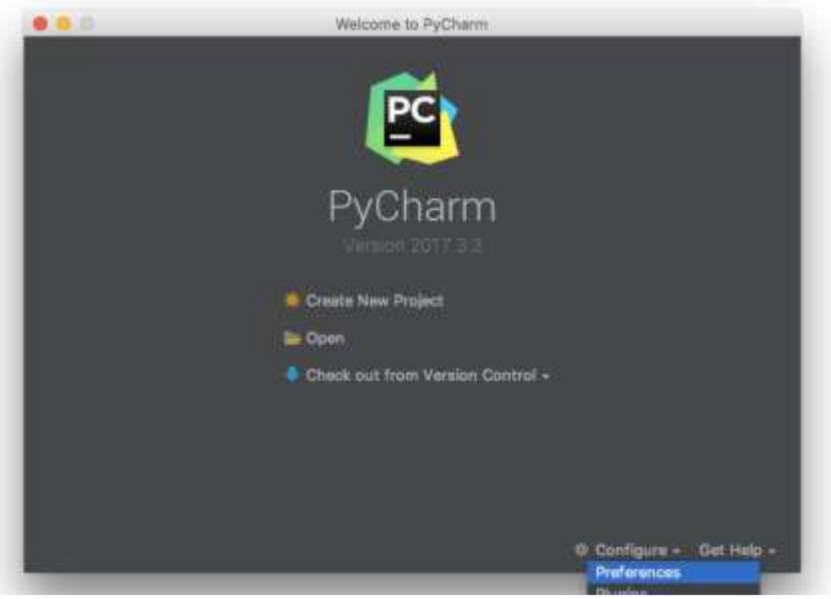
Next, the following dialog asks you to pick a UI theme.


Customize PyCharm
UI Themes → Launcher Script → Featured plugins
Set UI theme
Default
Darcula
project
fib.py
1: Project
2: Structure
Python Line Breakpoints
fib.py:5
Python Exception Breakpoints
Any exception
Django Exception Breakpoints
Template render error
Jinja2 Exception Breakpoints
UI theme can be changed later in Preferences | Appearance & Behavior | Appearance
Skip Remaining and Set Defaults
Next: Launcher Script

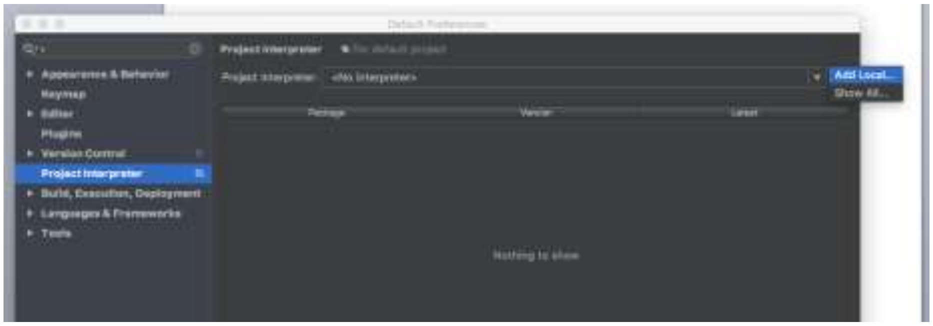
```
def fib(n):
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a + b
    print()


fib(1000)
```

You can pick the default which is already selected or pick ‘Darkula’. After this, click ‘Skip Remaining and Set Defaults’.

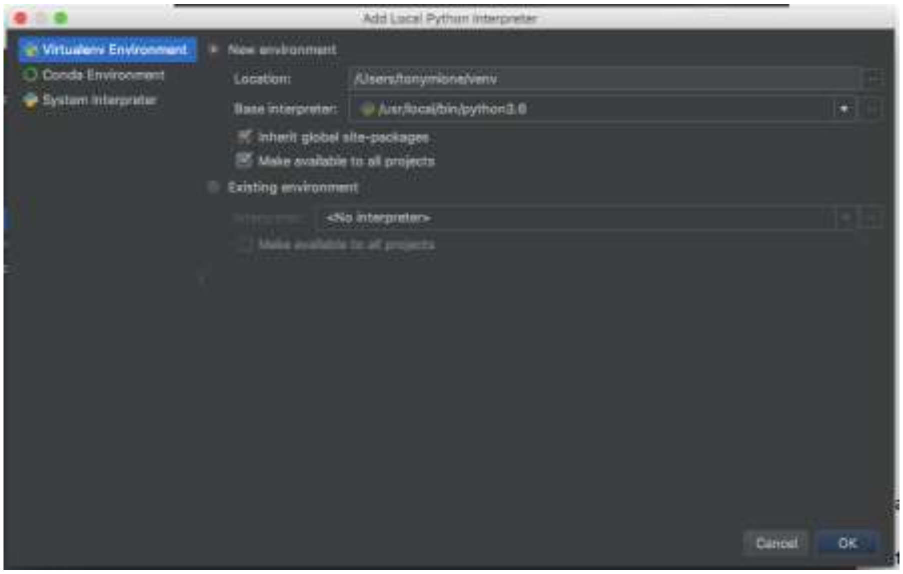
B) PyCharm will launch then select "Configure" from the bottom of the window that appears and choose “Preferences” from the pop-up menu, and then click “Project Interpreter” on the left side of the Default Settings window



c) Then click the ⚙ in the upper right at the end of and select “Add Local” and then “Virtualenv Environment” select “New environment”, “Inherit global site-packages” and “Make available to all projects”



Project interpreter: <No interpreter>



● New environment

Location: /Users/tonymione/venv

Base interpreter: /usr/local/bin/python3.6

☑ Inherit global site-packages

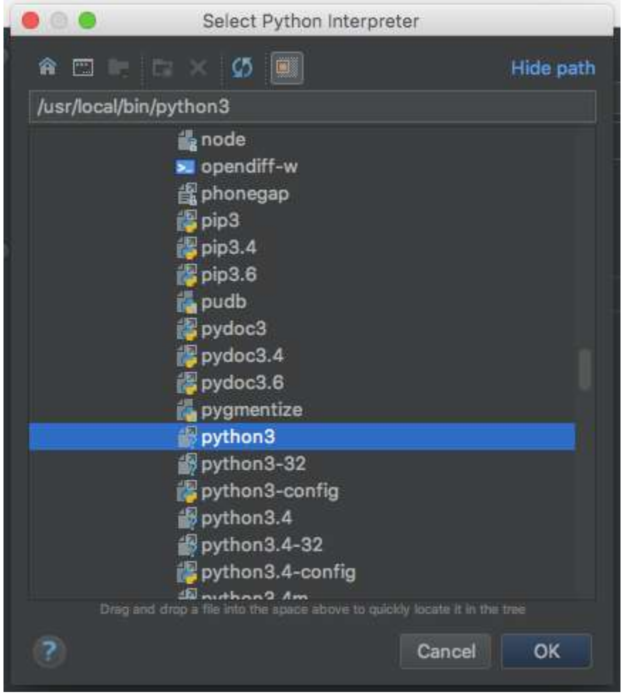
☑ Make available to all projects

○ Existing environment

Interpreter: <No interpreter>

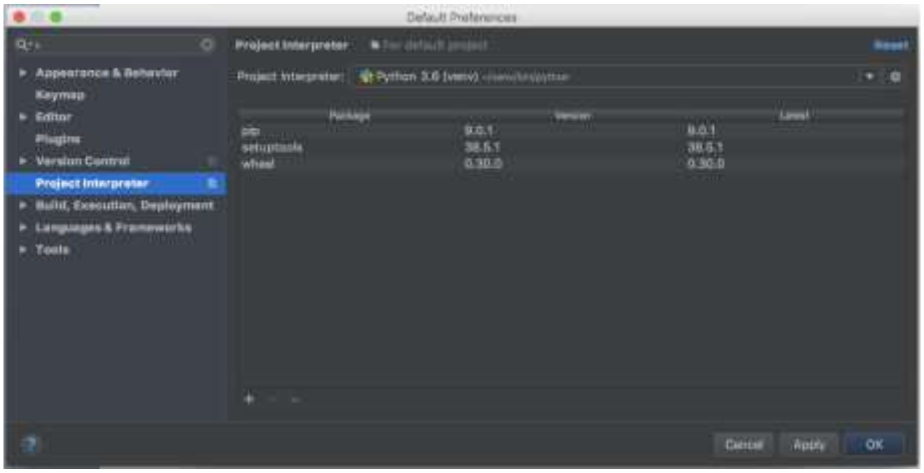
□ Make available to all projects

The path to the Python interpreter that you just installed in Step 1 should be populated in the "Base interpreter" box, if it is select "OK". If the box is blank select the ... and navigate to where the Python was installed, select it and then select OK as is shown below, note that your path may be different.



/usr/local/bin/python3

d) After setting up correctly the Default Setting windows should look as follows:



Project Interpreter

Project Interpreter: Python 3.6 (venv) ~/venv/bin/python

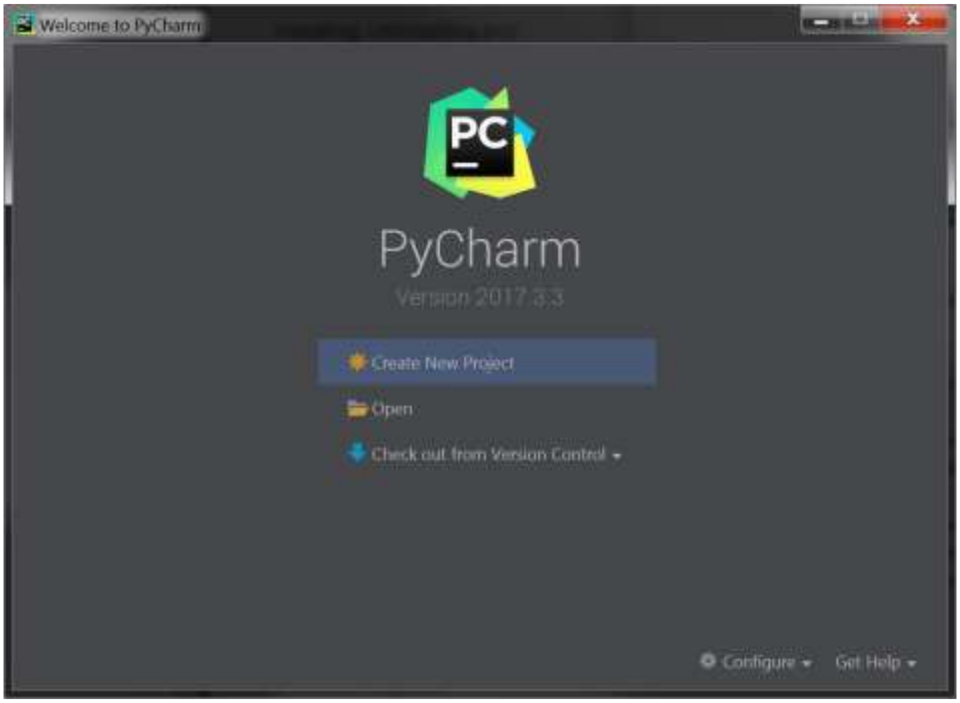
Package	Version	Latest
pip	9.0.1	9.0.1
setuptools	38.5.1	38.5.1
wheel	0.30.0	0.30.0

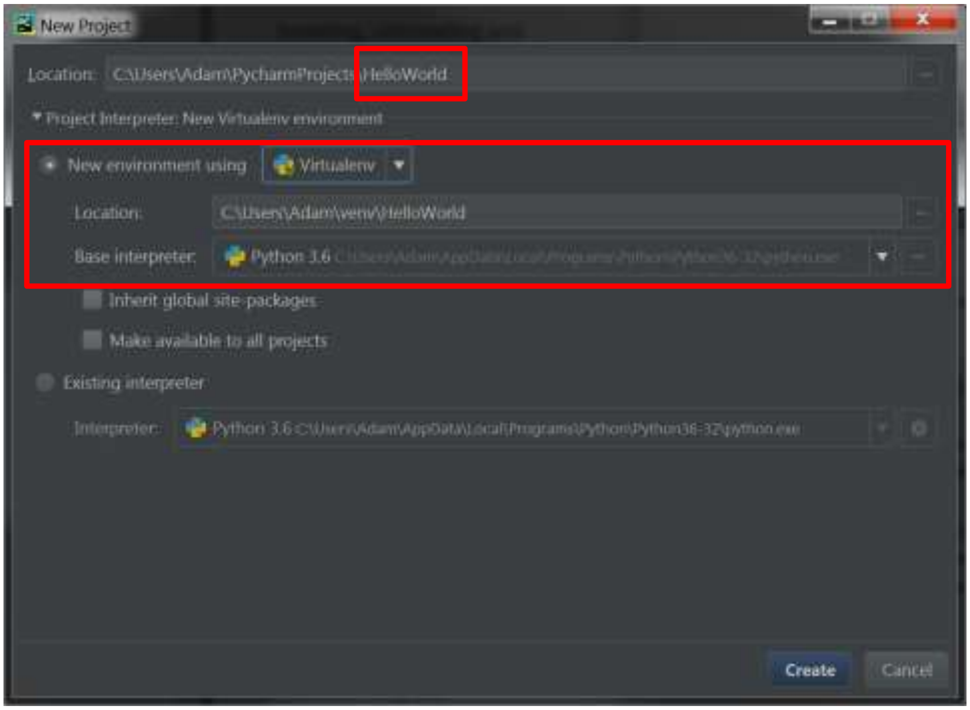
Select OK. Now PyCharm will always use the correct version of Python (Python 3) for your projects.

Step 3: Testing Your Development Environment

To test the Python and PyCharm environment we will create a pure Python project and write our first Python program. Normally, you will create new project for homework assignments that are assigned to you, specific details will be given to you for each assignment.

1) On the “Welcome to PyCharm” screen select “Create a New Project” and in the “Create Project” screen replace the “untitled” (or it might say “untitled1” or “untitled2” etc.) with the name of our project “HelloWorld”. Then click on the “Project Interpreter” item and select “New environment using Virtualenv” as is shown below and click “Create”





Location: C:\Users\Adam\PycharmProjects\HelloWorld

▾ Project Interpreter: New Virtualenv environment

● New environment using Virtualenv

Location: C:\Users\Adam\venv\HelloWorld

Base interpreter: Python 3.6 C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\python.exe

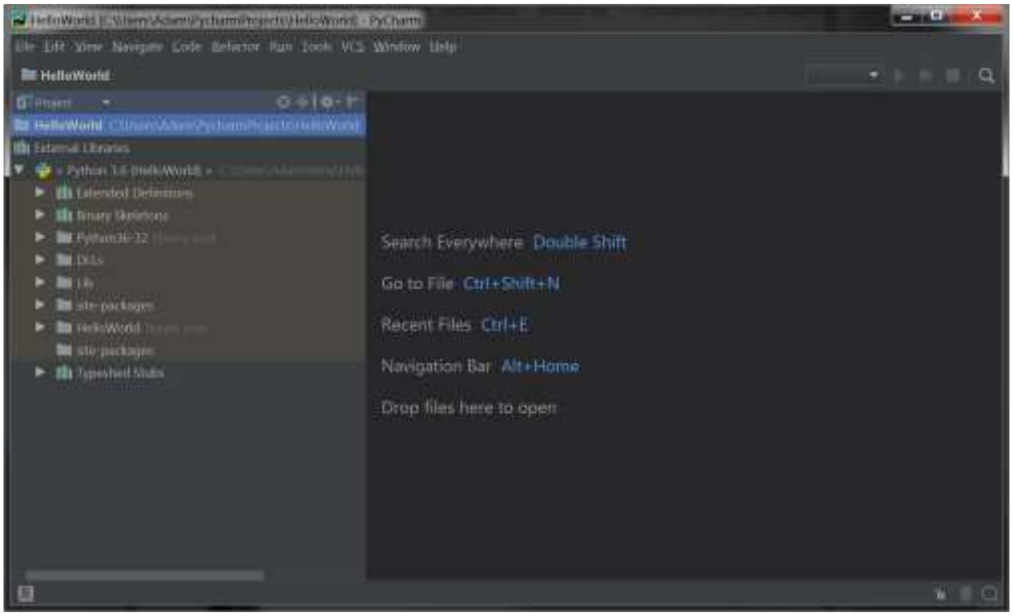
□ Inherit global site-packages

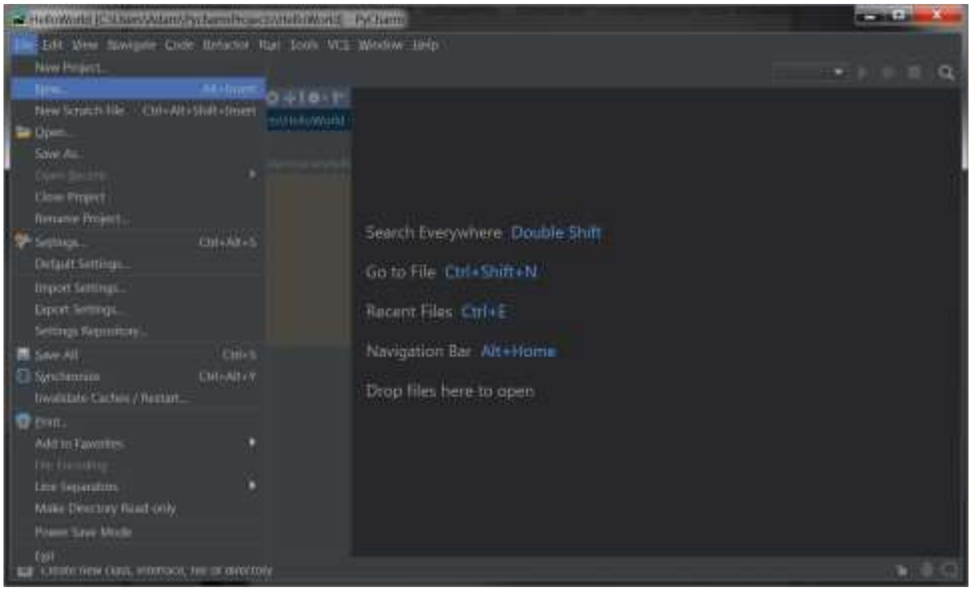
□ Make available to all projects

○ Existing interpreter

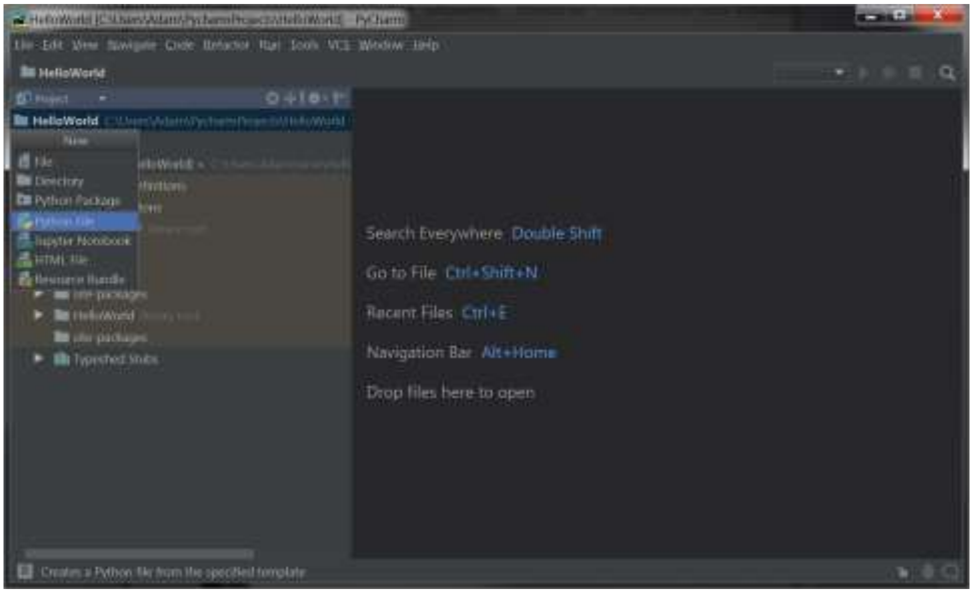
Interpreter: Python 3.6 C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\python.exe

2) Next we create a Python file. Select “File” from the menu bar, then “New” and then “Python File” and name the file “helloworld” as is shown in the figures below.

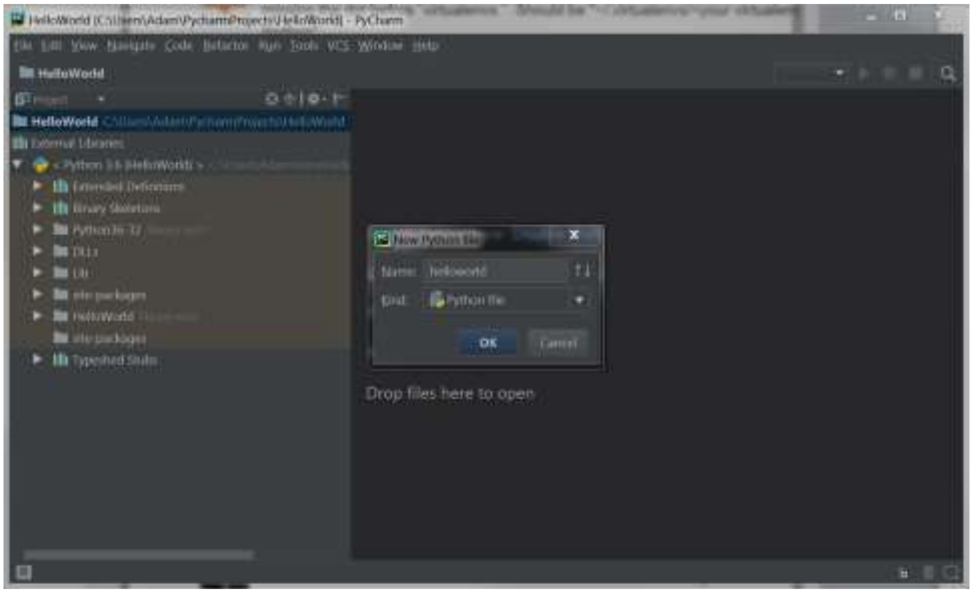




Create new class, interface, file or directory



Creates a Python file from the specified template


HelloWorld [C:\Users\Adam\PycharmProjects\HelloWorld] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
HelloWorld
Project
HelloWorld C:\Users\Adam\PycharmProjects\HelloWorld
External Libraries
< Python 3.6 (HelloWorld) >
Extended Definitions
Binary Skeletons
Python36-32
DLLs
Lib
site-packages
HelloWorld
site-packages
Typeshed Stubs
New Python file
OK
Cancel
Drop files here to open

Name: helloworld

Kind: Python file

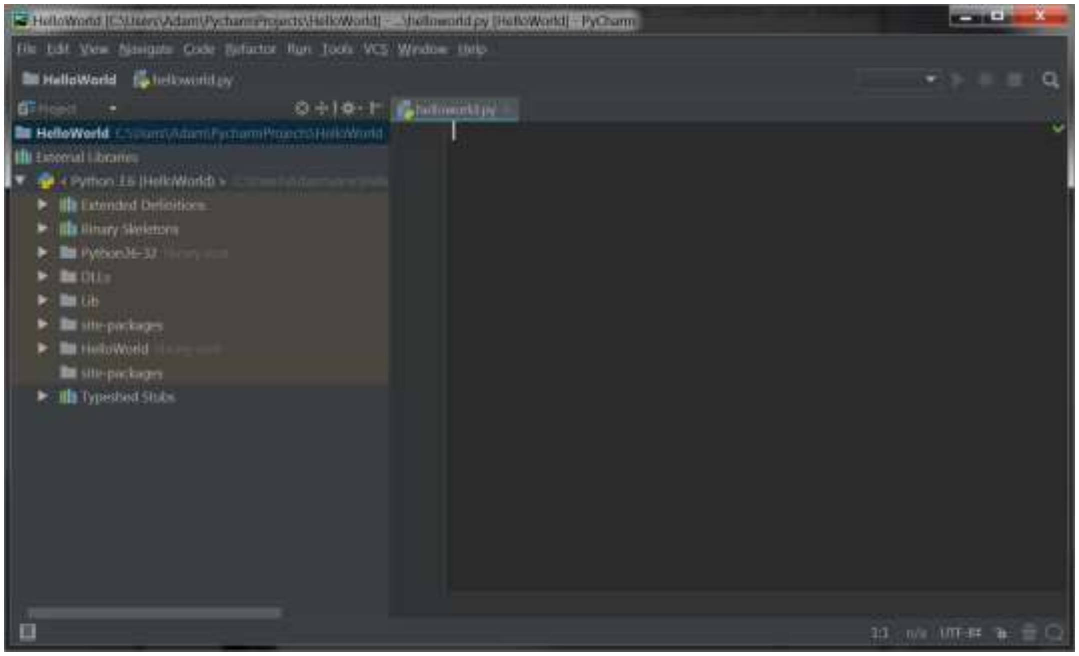
After you click “OK” you have created a Python file named “helloworld.py”. Python files must always have a “py” extension to be run by the Python interpreter.

Now let’s write our first Python program which is quite simple. We want it to create output that prints:

```
This is my first python program!
Hello, World!
```

While this is a very simple program, it will ensure that we installed Python and PyCharm correctly and that they are working together.

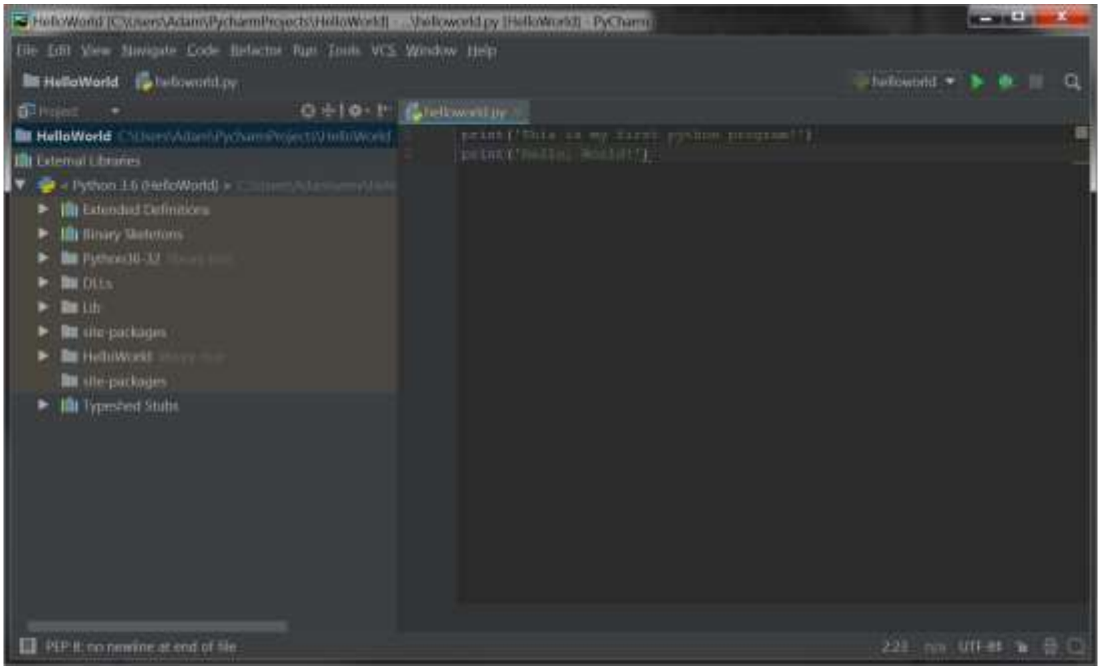
3) The tab on the right hand side labeled here as “helloworld.py” is where we are going to write our Python program.


HelloWorld [C:\Users\Adam\PycharmProjects\HelloWorld] - ...\helloworld.py [HelloWorld] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
HelloWorld
helloworld.py
Project
helloworld.py
HelloWorld C:\Users\Adam\PycharmProjects\HelloWorld
External Libraries
< Python 3.6 (HelloWorld) >
Extended Definitions
Binary Skeletons
Python36-32
DLLs
Lib
site-packages
HelloWorld
site-packages
Typeshed Stubs
1:1 n/a UTF-8

Type the following exactly:

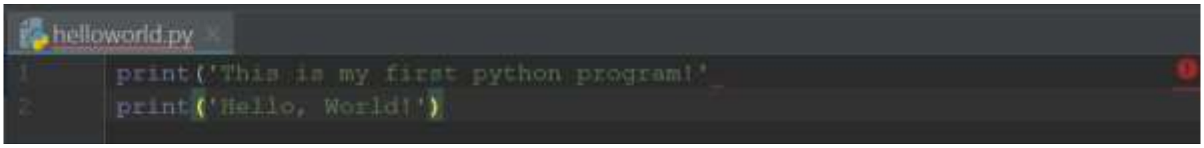
```
print('This is my first python program!')
print('Hello, World!')
```

As you type you will see that PyCharm will give you suggestions for auto completion, you can select a suggestion and autofill it in the editor window by hitting the tab key.


HelloWorld [C:\Users\Adam\PycharmProjects\HelloWorld] - ...\helloworld.py [HelloWorld] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
HelloWorld
helloworld.py
Project
HelloWorld C:\Users\Adam\PycharmProjects\HelloWorld
External Libraries
< Python 3.6 (HelloWorld) >
Extended Definitions
Binary Skeletons
Python36-32
DLLs
Lib
site-packages
HelloWorld
site-packages
Typeshed Stubs
helloworld
PEP 8: no newline at end of file
UTF-8

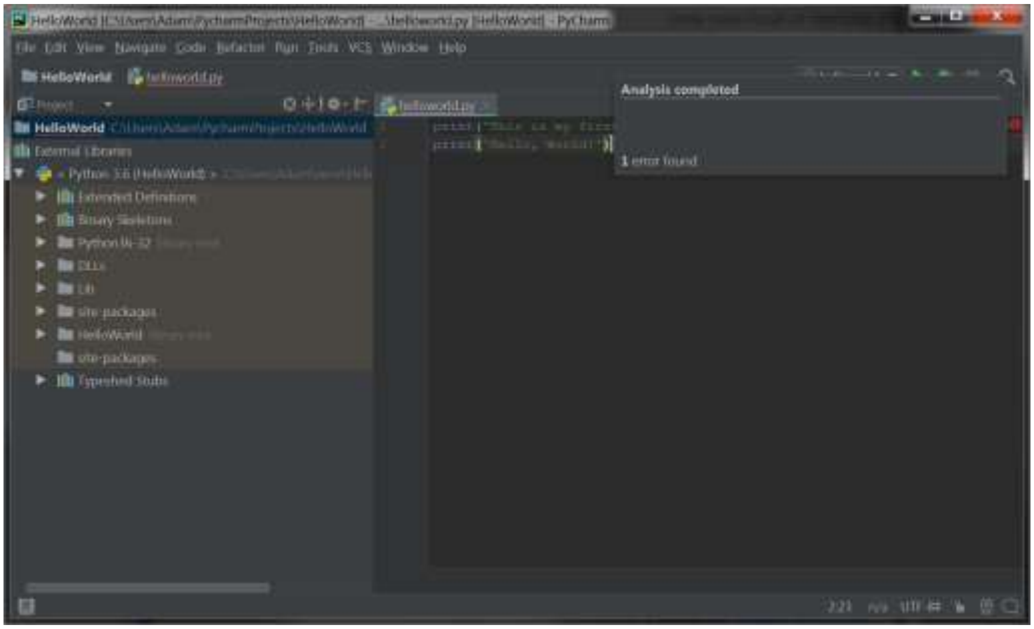
```
print('This is my first python program!')
print('Hello, World!')
```

PyCharm will evaluate your program while you type it for syntax errors, i.e., errors in the way the program was typed. For example, if you forget to put a parenthesis at the end of the first line of code you will be notified by a red exclamation mark in the upper right hand corner of the editor:



```
print('This is my first python program!'
print('Hello, World!')
```

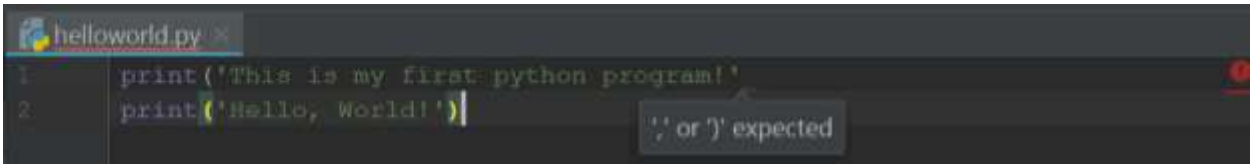
The red underscore shows you where the error occurred and clicking on the red exclamation mark will give you information on how many errors where found.


HelloWorld [C:\Users\Adam\PycharmProjects\HelloWorld] - ...\helloworld.py [HelloWorld] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
HelloWorld
helloworld.py
Project
HelloWorld C:\Users\Adam\PycharmProjects\HelloWorld
External Libraries
< Python 3.6 (HelloWorld) >
Extended Definitions
Binary Skeletons
Python36-32
DLLs
Lib
site-packages
HelloWorld
site-packages
Typeshed Stubs
UTF-8

Analysis completed

1 error found

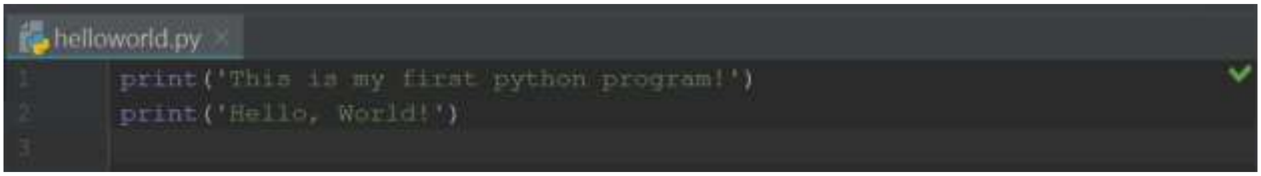
Placing your mouse pointer over the red underscore will give you more information on how to possibly fix the error. Here we see that we need to have either a comma or a parenthesis.



```
print('This is my first python program!'
print('Hello, World!')
```

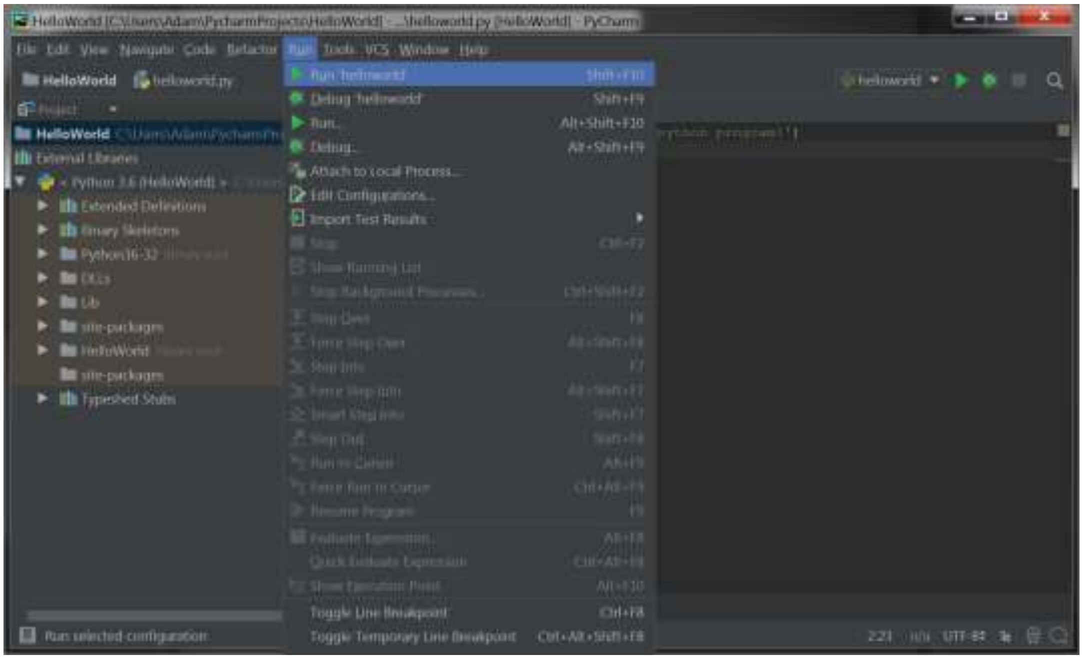
',' or ')' expected

A program that has no syntactic errors will be given a green check mark.

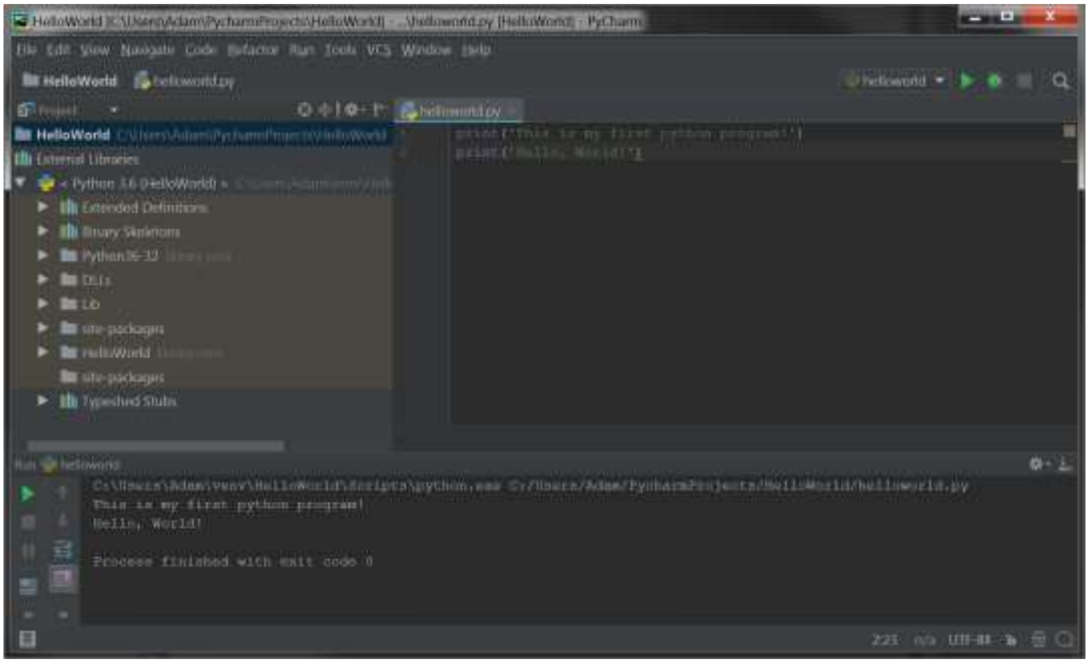


```
print('This is my first python program!')
print('Hello, World!')
```

4) To run our program we can select “Run” from the menu bar and “Run helloworld” or you can select the green arrow in the upper right.


HelloWorld [C:\Users\Adam\PycharmProjects\HelloWorld] - ...\helloworld.py [HelloWorld] - PyCharm
File
Edit
View
Navigate
Code
Refactor
Run
Tools
VCS
Window
Help
HelloWorld
helloworld.py
Project
HelloWorld C:\Users\Adam\PycharmPr
External Libraries
Python 3.6 (HelloWorld)
Extended Definitions
Binary Skeletons
Python36-32
DLLs
Lib
site-packages
HelloWorld
site-packages
Typeshed Stubs
Run 'helloworld' Shift+F10
Debug 'helloworld' Shift+F9
Run... Alt+Shift+F10
Debug... Alt+Shift+F9
Attach to Local Process...
Edit Configurations...
Import Test Results
Stop Ctrl+F2
Show Running List
Stop Background Processes... Ctrl+Shift+F2
Step Over F8
Force Step Over Alt+Shift+F8
Step Into F7
Force Step Into Alt+Shift+F7
Smart Step Into Shift+F7
Step Out Shift+F8
Run to Cursor Alt+F9
Force Run to Cursor Ctrl+Alt+F9
Resume Program F9
Evaluate Expression... Alt+F8
Quick Evaluate Expression Ctrl+Alt+F8
Show Execution Point Alt+F10
Toggle Line Breakpoint Ctrl+F8
Toggle Temporary Line Breakpoint Ctrl+Alt+Shift+F8
helloworld
Run selected configuration
2:21
n/a
UTF-8

A run window will be opened at the bottom of PyCharm window and you should see the output of the program you just wrote. The “Process finished with exit code 0” tells us that our program ran and finished with no errors.



```
print('This is my first python program!')
print('Hello, World!')
```

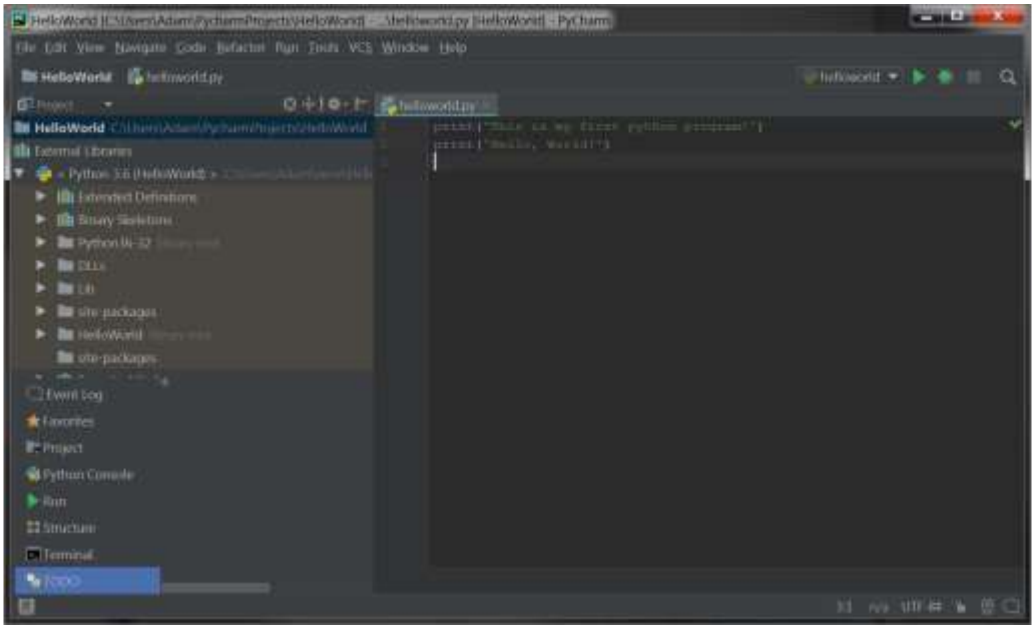
```
C:\Users\Adam\venv\HelloWorld\Scripts\python.exe C:/Users/Adam/PycharmProjects/HelloWorld/helloworld.py
This is my first python program!
Hello, World!

Process finished with exit code 0
```

Congratulations! You just wrote your first Python program! PyCharm is a very powerful IDE (Integrated Development Environment) and we will look at more ways it can be used during the course, but will not learn all of its features. If you want to learn more or have problems consider looking at the very extensive documentation at: https://www.jetbrains.com/pycharm/documentation/

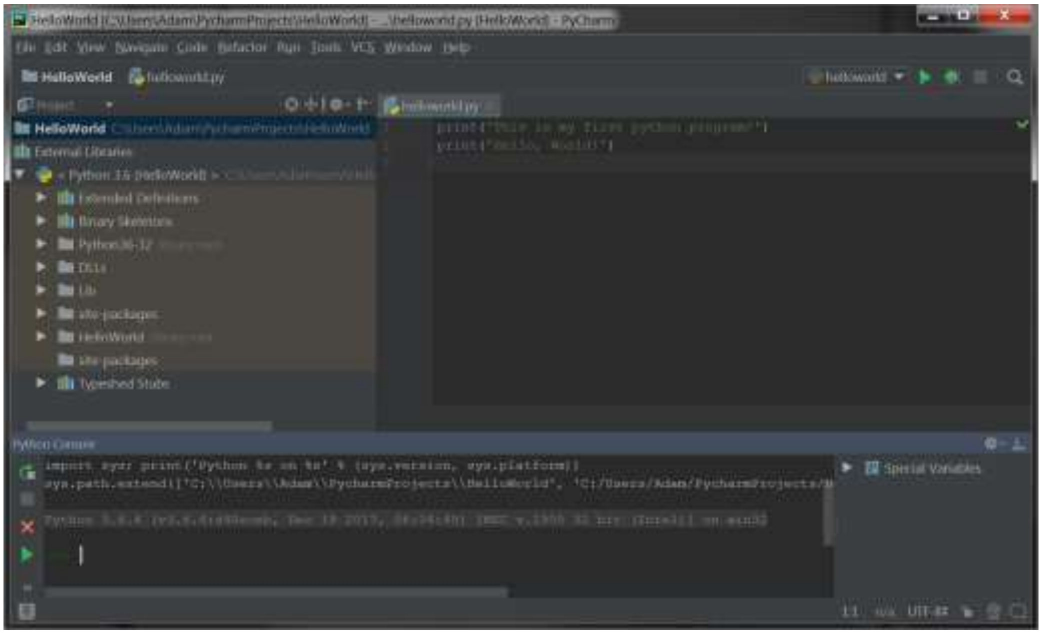
Step 4: PyCharm Python Console

Sometimes we don’t want to write a full program, but rather just test some code interactively in Python similar to IDLE (Python Shell) found in the Python. In PyCharm, we use the Python Console.

There are several ways to open a Python Console. The easiest is by placing your mouse pointer over the window icon in the lower left corner of the PyCharm window and selecting “Python Console”.



```
print('This is my first python program!')
print('Hello, World!')
```

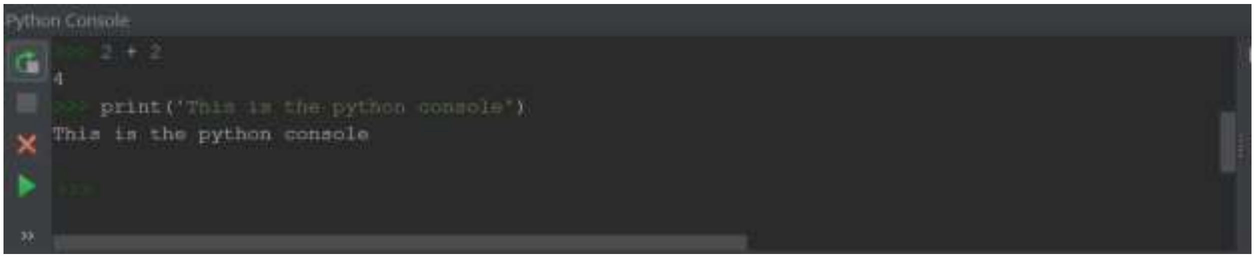


HelloWorld [C:\Users\Adam\PycharmProjects\HelloWorld] - ...\helloworld.py [HelloWorld] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
HelloWorld
helloworld.py
Project
HelloWorld C:\Users\Adam\PycharmProjects\HelloWorld
External Libraries
< Python 3.6 (HelloWorld) >
Extended Definitions
Binary Skeletons
Python36-32
DLLs
Lib
site-packages
HelloWorld
site-packages
Typeshed Stubs
Python Console
Special Variables

```
print('This is my first python program')
print('Hello, World!')
```

```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\Users\\Adam\\PycharmProjects\\HelloWorld', 'C:/Users/Adam/PycharmProjects/H
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
>>>
```

Here we can type expressions and they are evaluated as soon as we hit Enter as is shown below.



```
>>> 2 + 2
4
>>> print('This is the python console')
This is the python console
>>>
```

Again, we will explore more about PyCharm as we continue the course. For now let's exit PyCharm by selecting "File" and then "Exit" and move on to installing

Step 5: Installing PythonLabs

The PythonLabs software is a set of Python modules written specifically for the projects in this book. Here are the specific instructions for each operating system.

1) Windows OS:

a) Download and run http://ix.cs.uoregon.edu/~conery/eic/python/PythonLabs-1.0.2.win-amd64.exe . There are no options to specify, so when the installer exits the software should be in place.

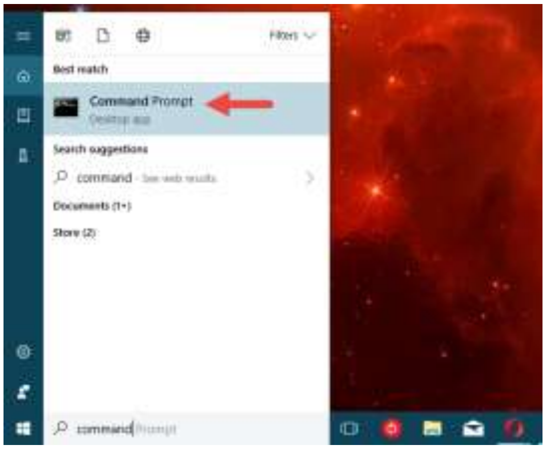
If the installer fails, download http://ix.cs.uoregon.edu/~conery/eic/python/PythonLabs-1.0.2.zip save it in your Downloads folder.

Start your File Browser. Go to the Downloads folder and find the zip file. Right-click and select 'Extract All'. In the dialog box, click the 'Extract All' button. You should now see a new folder with the same name, but without the .zip extension. Drag this folder to your Desktop.

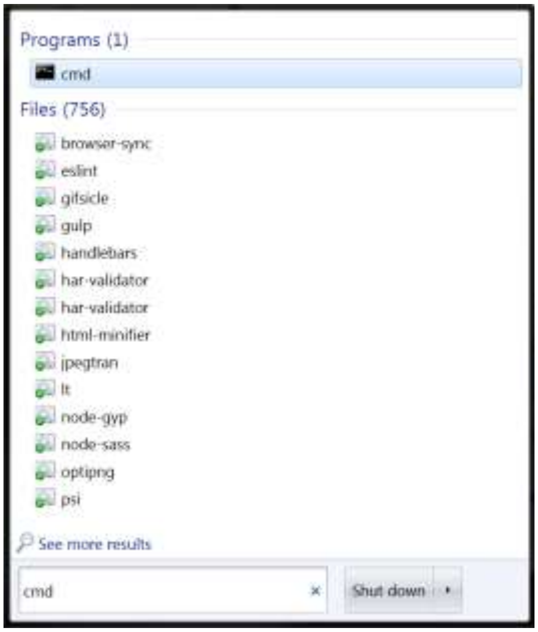
Next start a Command Prompt and type the following commands to install the lab modules:

```
cd Desktop\PythonLabs-1.0.2
python setup.py install
```

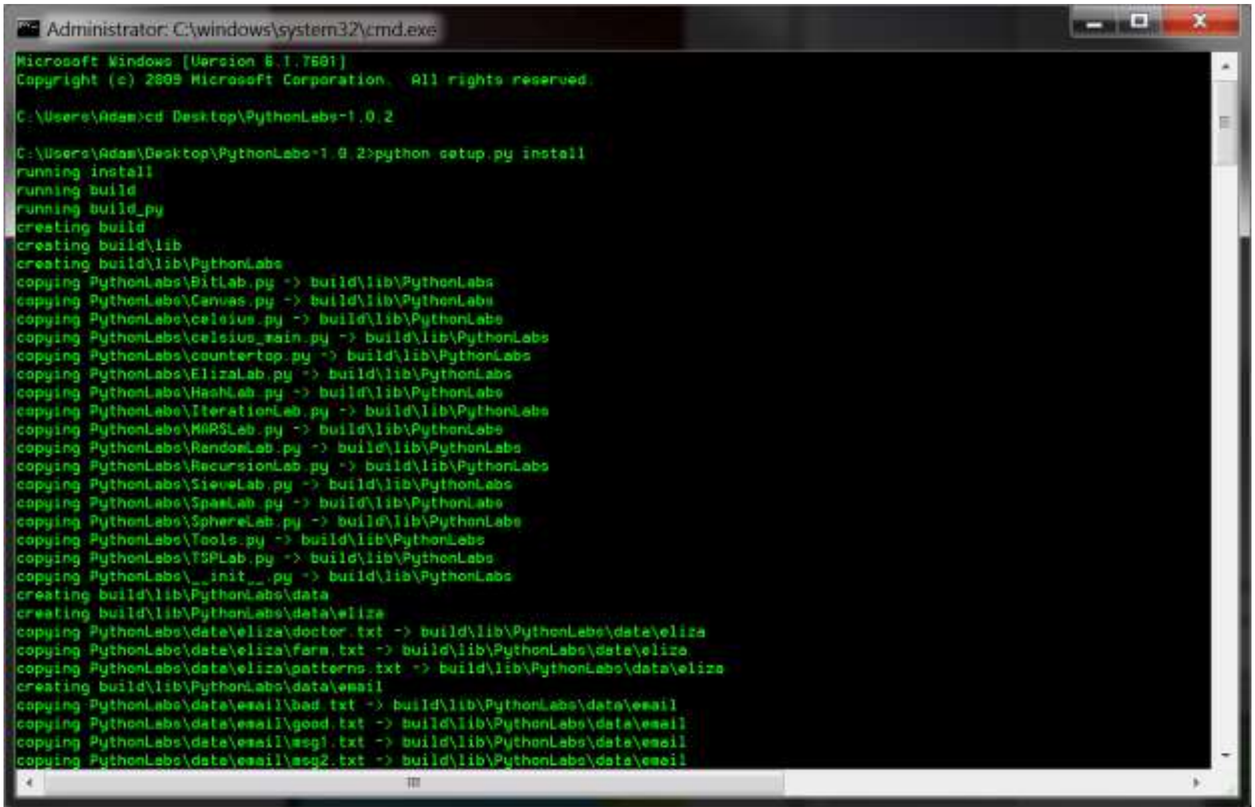
In Windows 10, one of the fastest ways to launch Command Prompt is to use search. Inside the search field from your taskbar, enter command or cmd. Then, select the Command Prompt result.



In Windows 7, go to the Start Menu and search for “command” or “cmd” and then select the Command Prompt result.

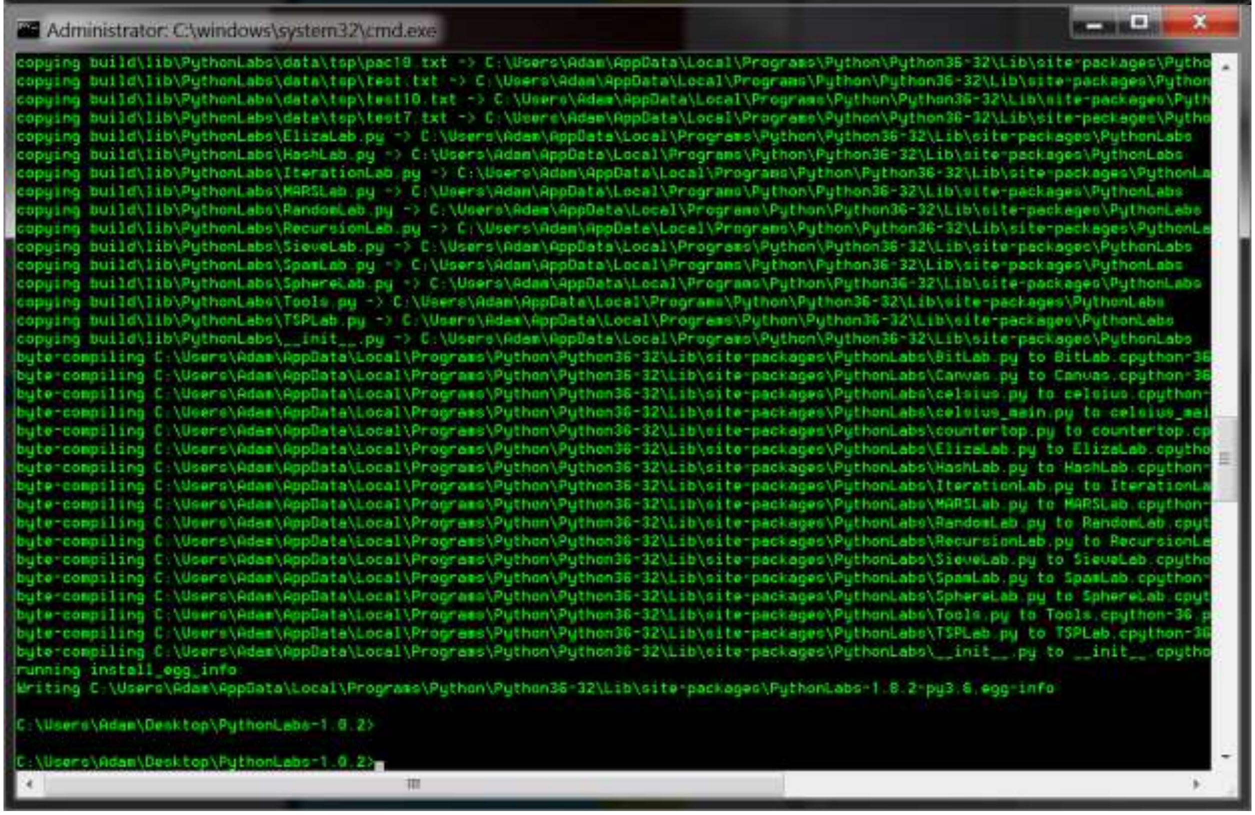


After running the commands above, you should see something like the following in the Command window.



Administrator: C:\windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Adam>cd Desktop\PythonLabs-1.0.2

C:\Users\Adam\Desktop\PythonLabs-1.0.2>python setup.py install
running install
running build
running build_py
creating build
creating build\lib
creating build\lib\PythonLabs
copying PythonLabs\BitLab.py -> build\lib\PythonLabs
copying PythonLabs\Canvas.py -> build\lib\PythonLabs
copying PythonLabs\celsius.py -> build\lib\PythonLabs
copying PythonLabs\celsius_main.py -> build\lib\PythonLabs
copying PythonLabs\countertop.py -> build\lib\PythonLabs
copying PythonLabs\ElizaLab.py -> build\lib\PythonLabs
copying PythonLabs\HashLab.py -> build\lib\PythonLabs
copying PythonLabs\IterationLab.py -> build\lib\PythonLabs
copying PythonLabs\MARSLab.py -> build\lib\PythonLabs
copying PythonLabs\RandomLab.py -> build\lib\PythonLabs
copying PythonLabs\RecursionLab.py -> build\lib\PythonLabs
copying PythonLabs\SieveLab.py -> build\lib\PythonLabs
copying PythonLabs\SpamLab.py -> build\lib\PythonLabs
copying PythonLabs\SphereLab.py -> build\lib\PythonLabs
copying PythonLabs\Tools.py -> build\lib\PythonLabs
copying PythonLabs\TSPLab.py -> build\lib\PythonLabs
copying PythonLabs\__init__.py -> build\lib\PythonLabs
creating build\lib\PythonLabs\data
creating build\lib\PythonLabs\data\eliza
copying PythonLabs\data\eliza\doctor.txt -> build\lib\PythonLabs\data\eliza
copying PythonLabs\data\eliza\farm.txt -> build\lib\PythonLabs\data\eliza
copying PythonLabs\data\eliza\patterns.txt -> build\lib\PythonLabs\data\eliza
creating build\lib\PythonLabs\data\email
copying PythonLabs\data\email\bad.txt -> build\lib\PythonLabs\data\email
copying PythonLabs\data\email\good.txt -> build\lib\PythonLabs\data\email
copying PythonLabs\data\email\msg1.txt -> build\lib\PythonLabs\data\email
copying PythonLabs\data\email\msg2.txt -> build\lib\PythonLabs\data\email
```



Administrator: C:\windows\system32\cmd.exe

```
copying build\lib\PythonLabs\data\tsp\pac10.txt -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\Pytho
copying build\lib\PythonLabs\data\tsp\test.txt -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\Python
copying build\lib\PythonLabs\data\tsp\test10.txt -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\Pyth
copying build\lib\PythonLabs\data\tsp\test7.txt -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\Pytho
copying build\lib\PythonLabs\ElizaLab.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs
copying build\lib\PythonLabs\HashLab.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs
copying build\lib\PythonLabs\IterationLab.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLa
copying build\lib\PythonLabs\MARSLab.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs
copying build\lib\PythonLabs\RandomLab.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs
copying build\lib\PythonLabs\RecursionLab.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLa
copying build\lib\PythonLabs\SieveLab.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs
copying build\lib\PythonLabs\SpamLab.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs
copying build\lib\PythonLabs\SphereLab.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs
copying build\lib\PythonLabs\Tools.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs
copying build\lib\PythonLabs\TSPLab.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs
copying build\lib\PythonLabs\__init__.py -> C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\BitLab.py to BitLab.cpython-36
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\Canvas.py to Canvas.cpython-36
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\celsius.py to celsius.cpython-
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\celsius_main.py to celsius_mai
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\countertop.py to countertop.cp
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\ElizaLab.py to ElizaLab.cpytho
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\HashLab.py to HashLab.cpython-
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\IterationLab.py to IterationLa
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\MARSLab.py to MARSLab.cpython-
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\RandomLab.py to RandomLab.cpyt
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\RecursionLab.py to RecursionLa
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\SieveLab.py to SieveLab.cpytho
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\SpamLab.py to SpamLab.cpython-
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\SphereLab.py to SphereLab.cpyt
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\Tools.py to Tools.cpython-36.p
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\TSPLab.py to TSPLab.cpython-36
byte-compiling C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs\__init__.py to __init__.cpytho
running install_egg_info
Writing C:\Users\Adam\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\PythonLabs-1.0.2-py3.6.egg-info

C:\Users\Adam\Desktop\PythonLabs-1.0.2>

C:\Users\Adam\Desktop\PythonLabs-1.0.2>
```

1) Mac OS:

a) Download the lab modules http://ix.cs.uoregon.edu/~conery/eic/python/PythonLabs-1.0.2.tar.gz

If your browser asks you what you want to do with the file tell it you want to save it in your Downloads folder.

Start Terminal and type this cd command to navigate to your Downloads folder:

```
cd Downloads
```

Type the following commands to unpack and install the modules:

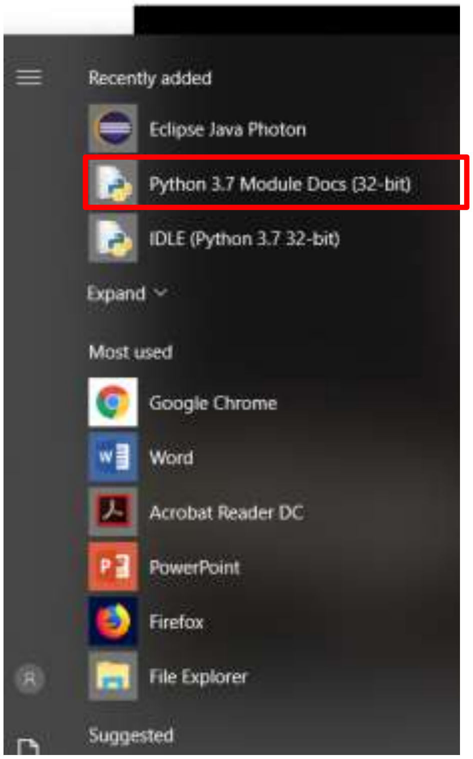
```
mv PythonLabs-1.0.2.tar.gz PythonLabs-1.0.2.tar
tar xf PythonLabs-1.0.2.tar
cd PythonLabs-1.0.2
sudo python3 setup.py install
```

You will see a lot of messages flash by on the screen, similar to the pictures above, but you can ignore them.

Step 6: Testing PythonLabs

1) Start an IDLE Python (Shell) session

a) In Windows OS you should be able to launch this from the Start menu:



b) In Mac OS, open a terminal window and type the following command to start an interactive session with Python: `python3`

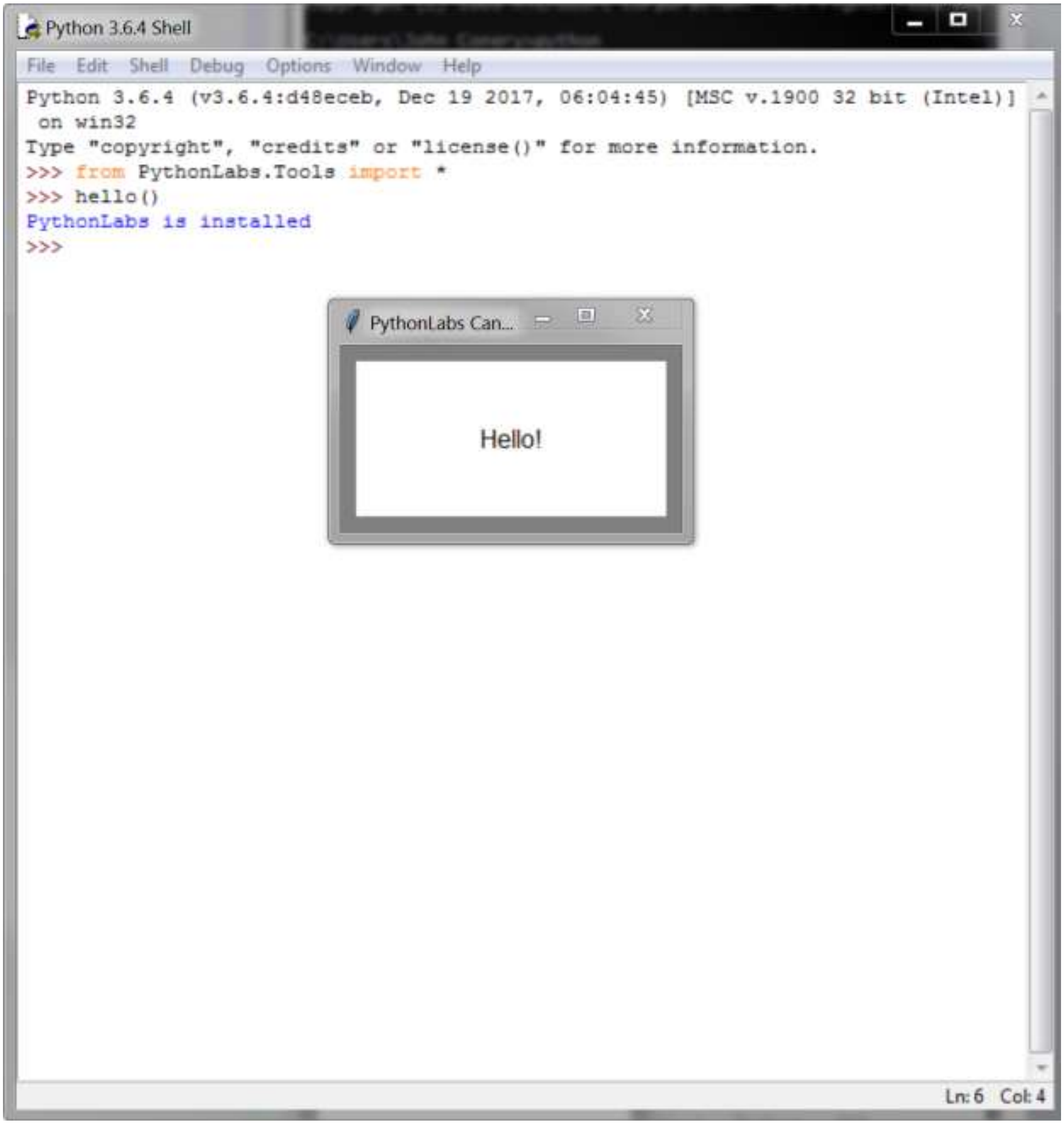
2) When you see Python's prompt (three greater-than symbols) type this command to load one of the PythonLabs modules:

```
from PythonLabs.Tools import *
```

Next type this command to test the installation:

```
hello()
```

If everything is working you will see a message that says "PythonLabs is installed" and a new window will pop up with the word "Hello" in it, as is shown below.



```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]
 on win32
Type "copyright", "credits" or "license()" for more information.
>>> from PythonLabs.Tools import *
>>> hello()
PythonLabs is installed
>>>
```

Step 7: Setting up IPython (Jupyter) Notebooks

The tutorial exercises for the projects in each chapter of the text are available as IPython (Jupyter) Notebooks. The notebooks allow you to define and test functions and run interactive experiments (including experiments that use the PythonLabs canvas) in your web browser.

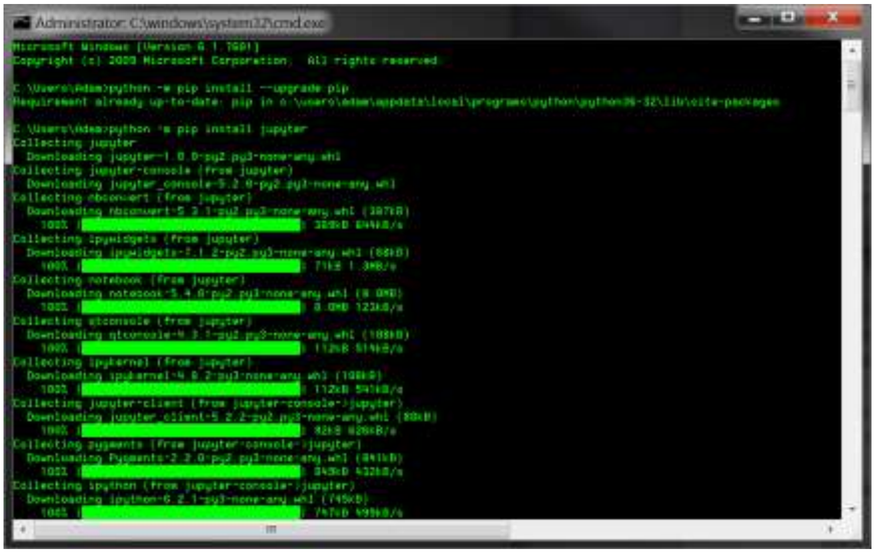
To install Jupyter Notebooks we will use a program in Python called pip, a package management system used to install and manage software packages written in Python. The steps are simple:

1) Windows OS:

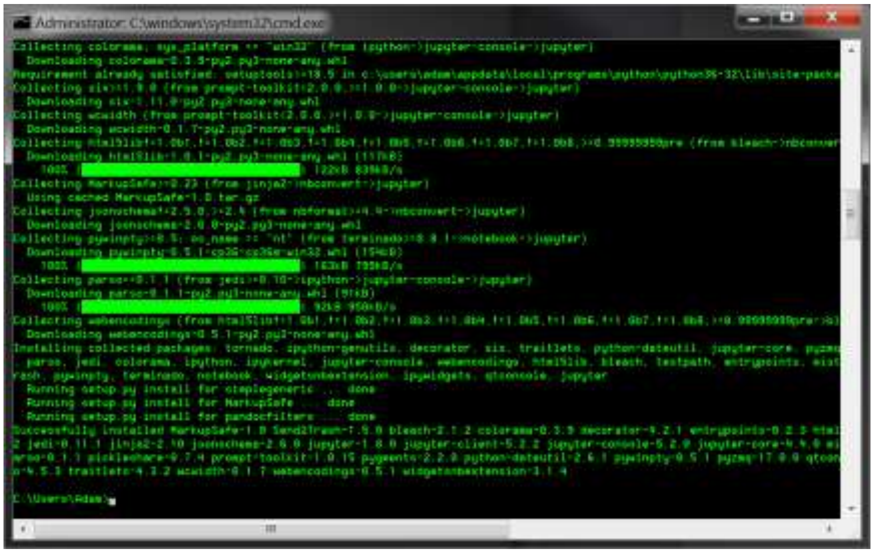
Start a Command Prompt and type the following commands to install the lab modules:

```
python -m pip install --upgrade pip
python -m pip install jupyter
```

If you need a review on how to start a Command Prompt in Windows review Step 5 above. Jupyter will start to install, it will take a bit of time – be patient and wait!



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Adam>python -m pip install --upgrade pip
Requirement already up-to-date: pip in c:\users\adam\appdata\local\programs\python\python36-32\lib\site-packages

C:\Users\Adam>python -m pip install jupyter
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl
Collecting jupyter-console (from jupyter)
  Downloading jupyter_console-5.2.0-py2.py3-none-any.whl
Collecting nbconvert (from jupyter)
  Downloading nbconvert-5.3.1-py2.py3-none-any.whl (387kB)
    100% |████████████████████████████████| 389kB 644kB/s
Collecting ipywidgets (from jupyter)
  Downloading ipywidgets-7.1.2-py2.py3-none-any.whl (68kB)
    100% |████████████████████████████████| 71kB 1.3MB/s
Collecting notebook (from jupyter)
  Downloading notebook-5.4.0-py2.py3-none-any.whl (8.0MB)
    100% |████████████████████████████████| 8.0MB 123kB/s
Collecting qtconsole (from jupyter)
  Downloading qtconsole-4.3.1-py2.py3-none-any.whl (108kB)
    100% |████████████████████████████████| 112kB 514kB/s
Collecting ipykernel (from jupyter)
  Downloading ipykernel-4.8.2-py3-none-any.whl (108kB)
    100% |████████████████████████████████| 112kB 541kB/s
Collecting jupyter-client (from jupyter-console->jupyter)
  Downloading jupyter_client-5.2.2-py2.py3-none-any.whl (80kB)
    100% |████████████████████████████████| 82kB 828kB/s
Collecting pygments (from jupyter-console->jupyter)
  Downloading Pygments-2.2.0-py2.py3-none-any.whl (841kB)
    100% |████████████████████████████████| 849kB 432kB/s
Collecting ipython (from jupyter-console->jupyter)
  Downloading ipython-6.2.1-py3-none-any.whl (745kB)
    100% |████████████████████████████████| 747kB 499kB/s
```



```
Collecting colorama; sys_platform == "win32" (from ipython->jupyter-console->jupyter)
  Downloading colorama-0.3.9-py2.py3-none-any.whl
Requirement already satisfied: setuptools>=18.5 in c:\users\adam\appdata\local\programs\python\python36-32\lib\site-packa
Collecting six>=1.9.0 (from prompt-toolkit<2.0.0,>=1.0.0->jupyter-console->jupyter)
  Downloading six-1.11.0-py2.py3-none-any.whl
Collecting wcwidth (from prompt-toolkit<2.0.0,>=1.0.0->jupyter-console->jupyter)
  Downloading wcwidth-0.1.7-py2.py3-none-any.whl
Collecting html5lib!=1.0b1,!=1.0b2,!=1.0b3,!=1.0b4,!=1.0b5,!=1.0b6,!=1.0b7,!=1.0b8,>=0.99999999pre (from bleach->nbconver
  Downloading html5lib-1.0.1-py2.py3-none-any.whl (117kB)
    100% |████████████████████████████████| 122kB 839kB/s
Collecting MarkupSafe>=0.23 (from jinja2->nbconvert->jupyter)
  Using cached MarkupSafe-1.0.tar.gz
Collecting jsonschema!=2.5.0,>=2.4 (from nbformat>=4.4->nbconvert->jupyter)
  Downloading jsonschema-2.6.0-py2.py3-none-any.whl
Collecting pywinpty>=0.5; os_name == "nt" (from terminado>=0.8.1->notebook->jupyter)
  Downloading pywinpty-0.5.1-cp36-cp36m-win32.whl (154kB)
    100% |████████████████████████████████| 163kB 795kB/s
Collecting parso==0.1.1 (from jedi>=0.10->ipython->jupyter-console->jupyter)
  Downloading parso-0.1.1-py2.py3-none-any.whl (91kB)
    100% |████████████████████████████████| 92kB 958kB/s
Collecting webencodings (from html5lib!=1.0b1,!=1.0b2,!=1.0b3,!=1.0b4,!=1.0b5,!=1.0b6,!=1.0b7,!=1.0b8,>=0.99999999pre->bl
  Downloading webencodings-0.5.1-py2.py3-none-any.whl
Installing collected packages: tornado, ipython-genutils, decorator, six, traitlets, python-dateutil, jupyter-core, pyzmq
, parso, jedi, colorama, ipython, ipykernel, jupyter-console, webencodings, html5lib, bleach, testpath, entrypoints, mist
rash, pywinpty, terminado, notebook, widgetsnbextension, ipywidgets, qtconsole, jupyter
  Running setup.py install for simplegeneric ... done
  Running setup.py install for MarkupSafe ... done
  Running setup.py install for pandocfilters ... done
Successfully installed MarkupSafe-1.0 Send2Trash-1.5.0 bleach-2.1.2 colorama-0.3.9 decorator-4.2.1 entrypoints-0.2.3 html
2 jedi-0.11.1 jinja2-2.10 jsonschema-2.6.0 jupyter-1.0.0 jupyter-client-5.2.2 jupyter-console-5.2.0 jupyter-core-4.4.0 mi
arso-0.1.1 pickleshare-0.7.4 prompt-toolkit-1.0.15 pygments-2.2.0 python-dateutil-2.6.1 pywinpty-0.5.1 pyzmq-17.0.0 qtcon
o-4.5.3 traitlets-4.3.2 wcwidth-0.1.7 webencodings-0.5.1 widgetsnbextension-3.1.4

C:\Users\Adam>
```

1) Mac OS:

Start Terminal and type the following commands to install the lab modules:

```
python3 -m pip install --upgrade pip
python3 -m pip install jupyter
```

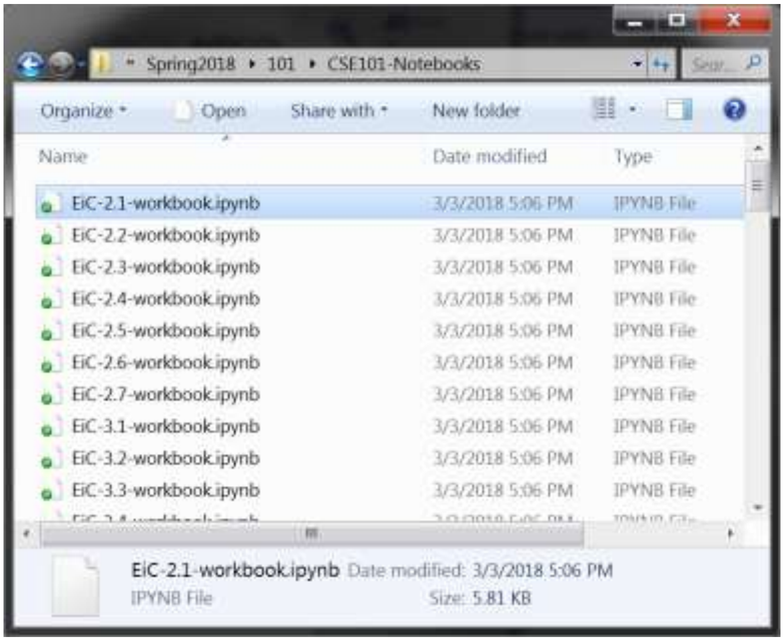
Jupyter should start to install. The output will look similar to the above and it will take a bit of time – be patient and wait!

After Jupyter installs, we want to test our installation by first downloading the notebook content for our textbook

2) Create a folder named “CSE101-Notebooks”

3) Download ONE of the following. Extract and move the contents to the “CSE101-Notebooks” you just created:

http://ix.cs.uoregon.edu/~conery/eic/python/ipython/all_notebooks.zip
http://ix.cs.uoregon.edu/~conery/eic/python/ipython/all_notebooks.tar



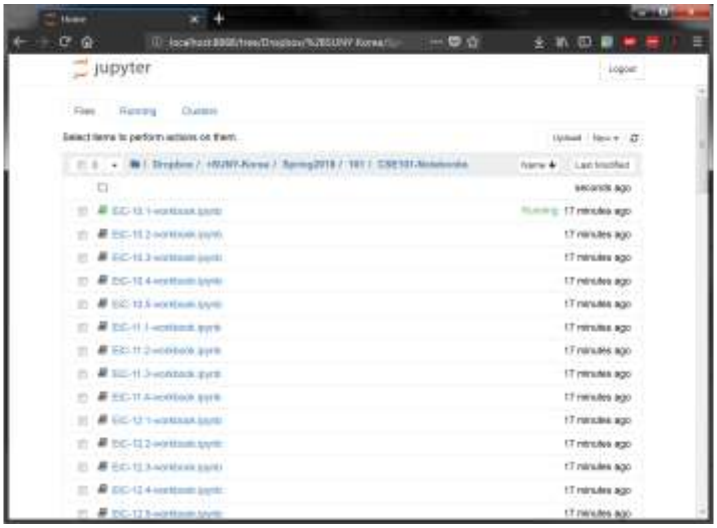
Name	Date modified	Type
EiC-2.1-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File
EiC-2.2-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File
EiC-2.3-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File
EiC-2.4-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File
EiC-2.5-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File
EiC-2.6-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File
EiC-2.7-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File
EiC-3.1-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File
EiC-3.2-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File
EiC-3.3-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File
EiC-3.4-workbook.ipynb	3/3/2018 5:06 PM	IPYNB File

EiC-2.1-workbook.ipynb Date modified: 3/3/2018 5:06 PM
IPYNB File Size: 5.81 KB

4) Next, in the Command Prompt (Windows) or the Terminal (Mac OS) type the following command:

```
jupyter notebook
```

This should open a webpage in your browser with a directory listing. Navigate to the “CSE101-Notebooks”

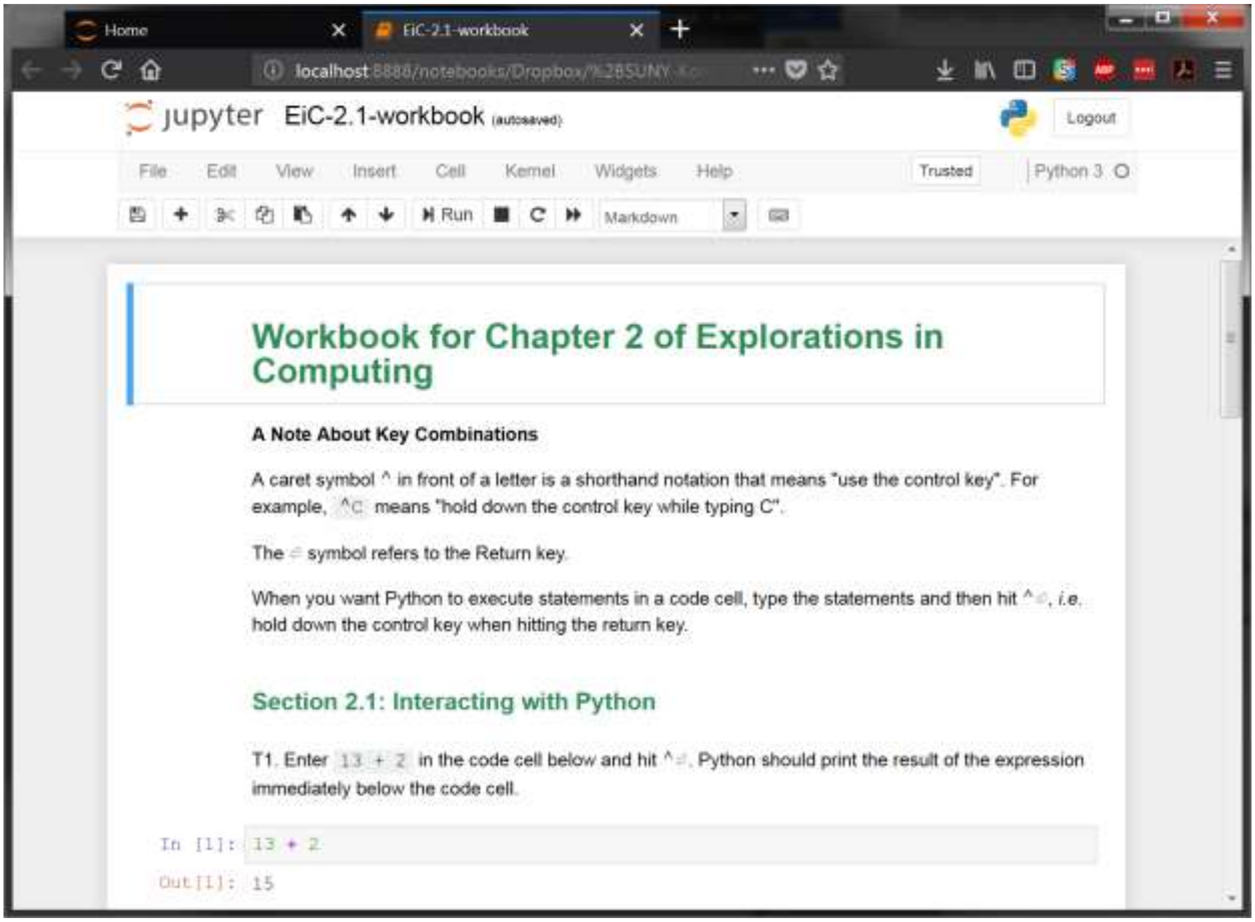


Select items to perform actions on them.

/ Dropbox / +SUNY-Korea / Spring2018 / 101 / CSE101-Notebooks

	Name	Last Modified
		seconds ago
EiC-10.1-workbook.ipynb	Running	17 minutes ago
EiC-10.2-workbook.ipynb		17 minutes ago
EiC-10.3-workbook.ipynb		17 minutes ago
EiC-10.4-workbook.ipynb		17 minutes ago
EiC-10.5-workbook.ipynb		17 minutes ago
EiC-11.1-workbook.ipynb		17 minutes ago
EiC-11.2-workbook.ipynb		17 minutes ago
EiC-11.3-workbook.ipynb		17 minutes ago
EiC-11.4-workbook.ipynb		17 minutes ago
EiC-12.1-workbook.ipynb		17 minutes ago
EiC-12.2-workbook.ipynb		17 minutes ago
EiC-12.3-workbook.ipynb		17 minutes ago
EiC-12.4-workbook.ipynb		17 minutes ago
EiC-12.5-workbook.ipynb		17 minutes ago

Select "EIC-2.1-workbook.ipynb"



Workbook for Chapter 2 of Explorations in Computing

A Note About Key Combinations

A caret symbol ^ in front of a letter is a shorthand notation that means "use the control key". For example, `^C` means "hold down the control key while typing C".

The ⏎ symbol refers to the Return key.

When you want Python to execute statements in a code cell, type the statements and then hit ^⏎, *i.e.* hold down the control key when hitting the return key.

Section 2.1: Interacting with Python

T1. Enter `13 + 2` in the code cell below and hit ^⏎. Python should print the result of the expression immediately below the code cell.

```
In [1]: 13 + 2

Out[1]: 15
```

Follow the instructions for T1. You should get the output of 15.

Make sure you use the key combination of the Ctrl (control) key and Enter if you want if you want to enter evaluate the values that you have typed.

Congratulations, you have the notebooks working. To end your session, go to “File” and select “Close and Halt” and then you can close the webpage. In the Command Prompt or Terminal window use Ctrl-C to end the jupyter process.

Final Notes:

We have installed quite a few programs and it can seem confusing right now, but as you make your way through the first set of tutorials and the first homework assignment it will become more clear on which program you should be using. Make sure you read the directions and are completely clear on what we are asking you to do and if you have questions please ask.

Welcome to CSE101 and good luck!