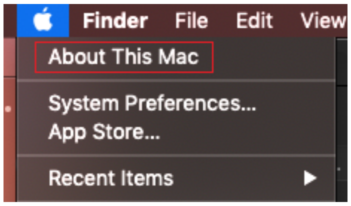
Updated June 2019

Installation for Mac

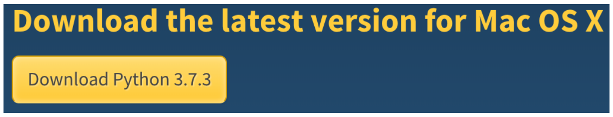
A. Installing Python

1. Make sure your Mac OS is update to date, or at least at 10.9. Click on the Apple Logo on the top-left corner of your screen, then click on "**About this Mac**" to verify the system version.

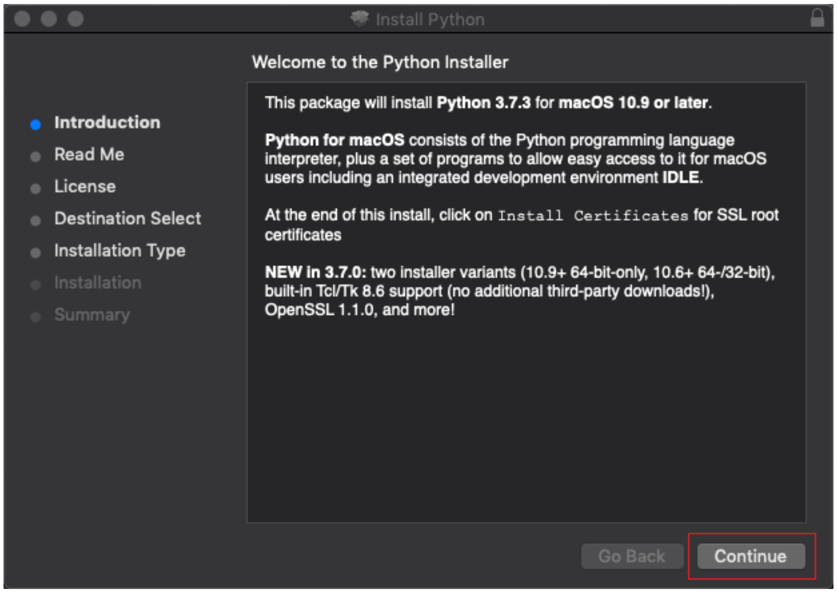


If the system does not meet this requirement, go to "**System Preferences**", then "**Software Updates**" to update.

2. Go to https://www.python.org/downloads/ and click on the Button "**Download Python 3.7.3**" The download should start automatically.



3. Open the installer and you shall see the screen below, follow the red rectangles on each screen to finish the installation with all default settings.



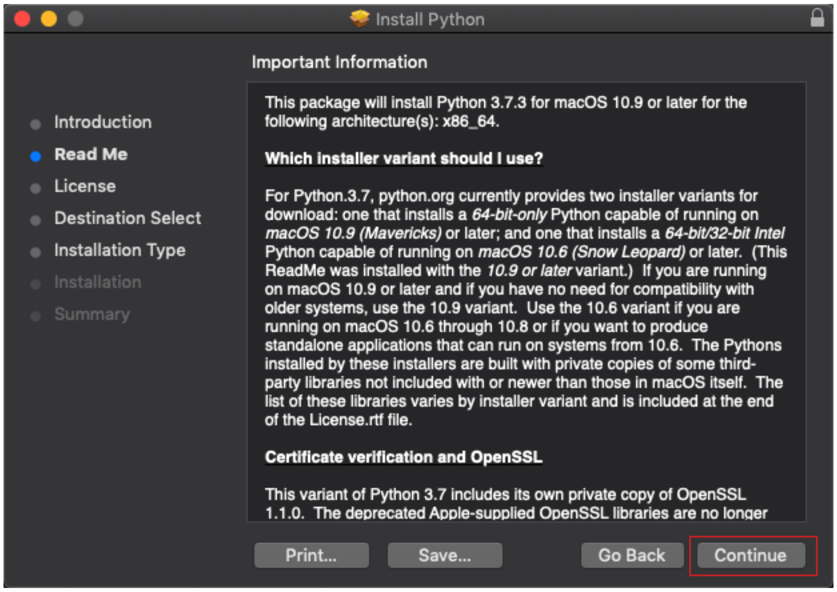
Welcome to the Python Installer

This package will install **Python 3.7.3** 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**.

At the end of this install, click on `Install Certificates` for SSL root certificates

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!



Important Information

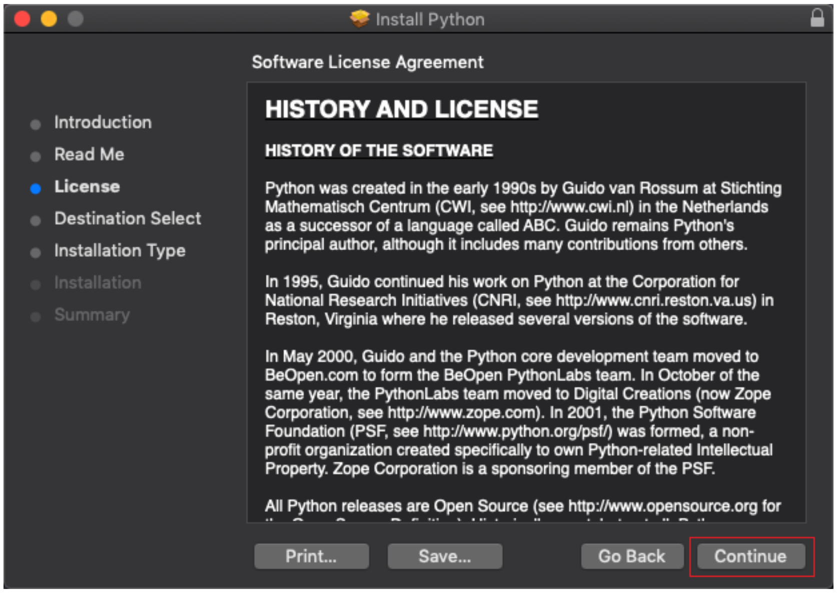
This package will install Python 3.7.3 for macOS 10.9 or later for the following architecture(s): x86_64.

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



Software License Agreement

HISTORY AND LICENSE

HISTORY OF THE SOFTWARE

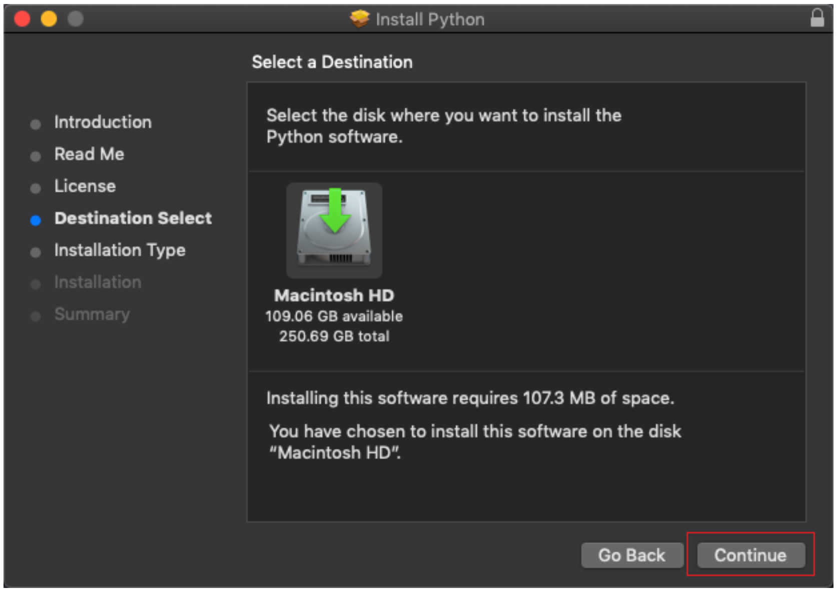
Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

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.

All Python releases are Open Source (see http://www.opensource.org for

If a window pops up for Confirmation of the Agreements, click "**Agree**."



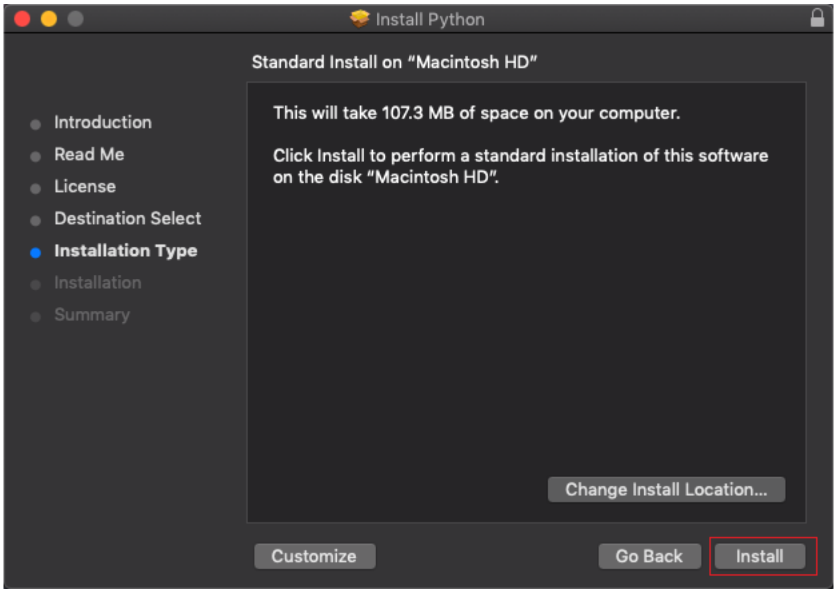
Select a Destination

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

Macintosh HD
109.06 GB available
250.69 GB total

Installing this software requires 107.3 MB of space.

You have chosen to install this software on the disk "Macintosh HD".



Standard Install on "Macintosh HD"

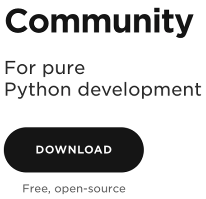
This will take 107.3 MB of space on your computer.

Click Install to perform a standard installation of this software on the disk "Macintosh HD".

Wait for the installation to finish, then exit the installer.

B. Installing PyCharm

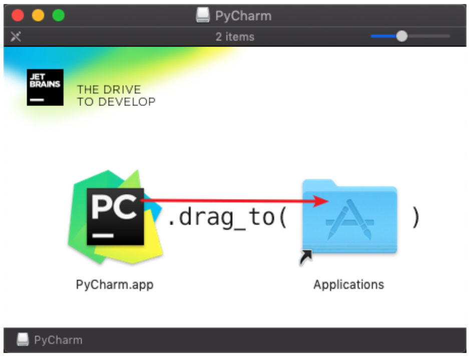
1. Download Mac OS X of PyCharm from: https://www.jetbrains.com/pycharm/download/#section=mac, select the "**Community**" version to download.



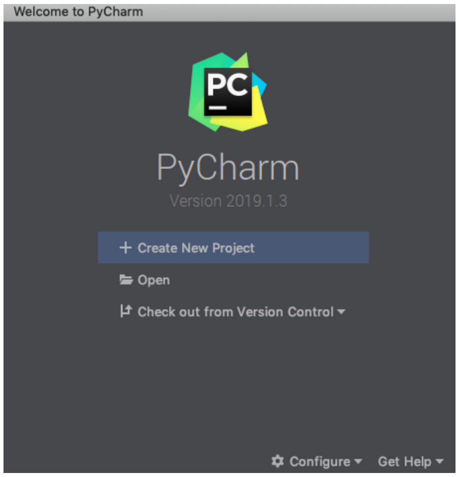
Community

For pure
Python development

2. Click to run the installer and drag the "PyCharm.app" on the left to "Applications" folder on the right. Wait for the auto-installation to finish.

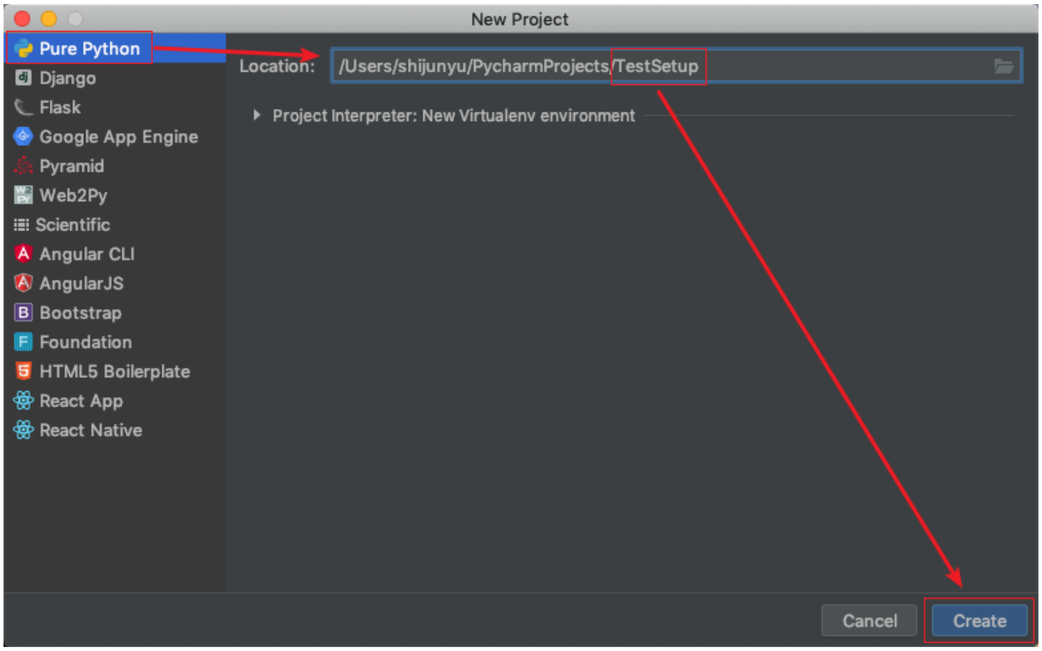


3. Launch PyCharm from Launchpad and add it to your Dock by right clicking on the icon and choosing **Options, Keep in Dock**. Running PyCharm for the first time will ask you to configure the editor, follow the screens below to finish setup. [Insert Screenshots Here]
   - Import PyCharm Settings: "**Do not import settings**," then "**OK**".
   - Select the user interface theme: Choose a theme that you like, then "**OK**".
   - Download and install additional plugins: No additional plugins are needed; we can just click "**OK**".
   - Start a project in PyCharm: "**Create New Project**". Then go to **section C**.



C. Configuring PyCharm to work with Python and Pygame

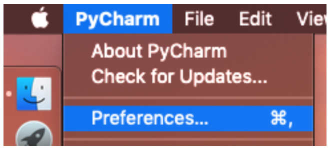
1. Choose “Pure Python,” then select a desired location for your projects, and give it a name such as “TestSetup” in the last tag. (the example directory below is “PycharmProjects/TestSetup”, in which the project name(folder) is separated by a ‘/’ from the parent directory)



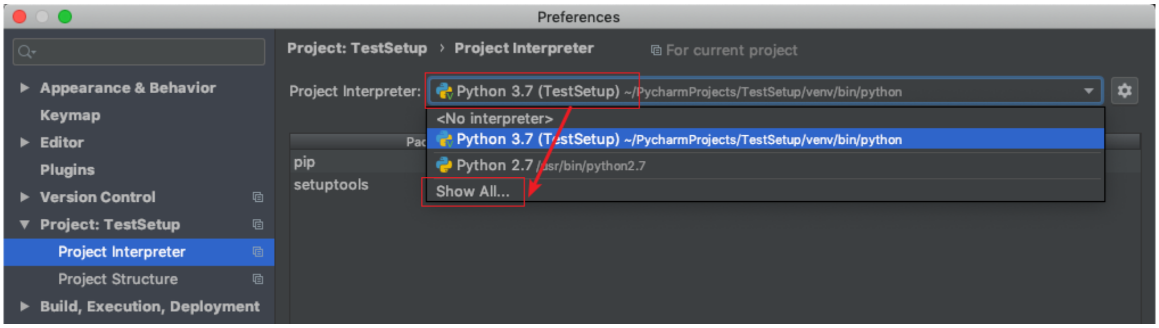
Location: /Users/shijunyu/PycharmProjects/TestSetup

Project Interpreter: New Virtualenv environment

2. Click on “**PyCharm**” on the top-left menu bar, then click “Preferences”

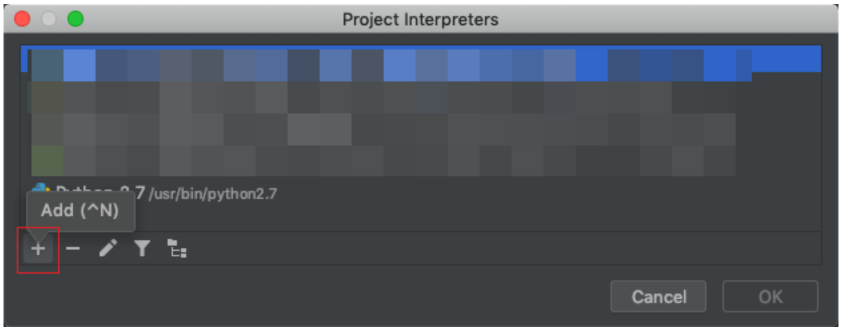


3. On the left navigation panel, go to **Project: [YourProjectName]** and expand it, select “Project Interpreter”. (See screen below)
Verify the Project Interpreter to be Python 3.7: if not, click on the upper rectangle area to open the drop-down menu, if no Python 3.7 is available, then click “Show All…”



Project Interpreter: Python 3.7 (TestSetup) ~/PycharmProjects/TestSetup/venv/bin/python

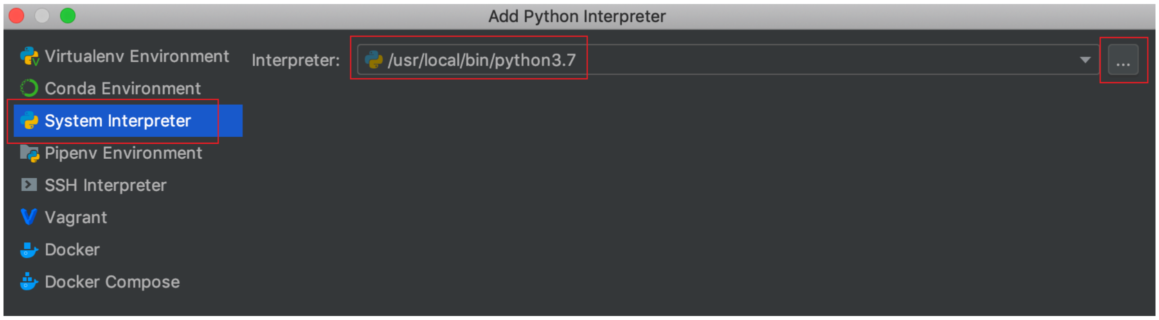
Click “+” button to add a new Interpreter.



[redacted]

Python 2.7 /usr/bin/python2.7

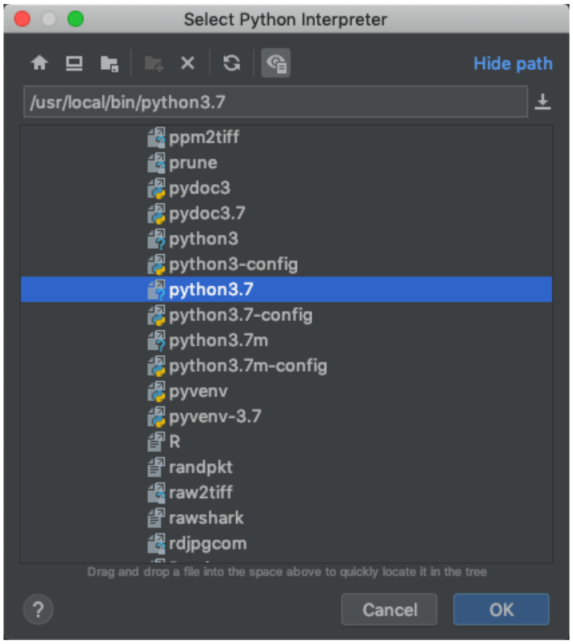
On the left Navigation panel, click on “**System Interpreter**”, then click the interpreter input box to open the drop-down menu, look for Python 3.7 options, which is usually located as **“/usr/local/bin/python3.7**”



Interpreter: /usr/local/bin/python3.7

If there is no option of python 3.7, click on the “**…**” on the right end of “Interpreter” and find the python3.7 file in the following directory: **“/usr/local/bin/python3.7**”, then click “**OK**”.

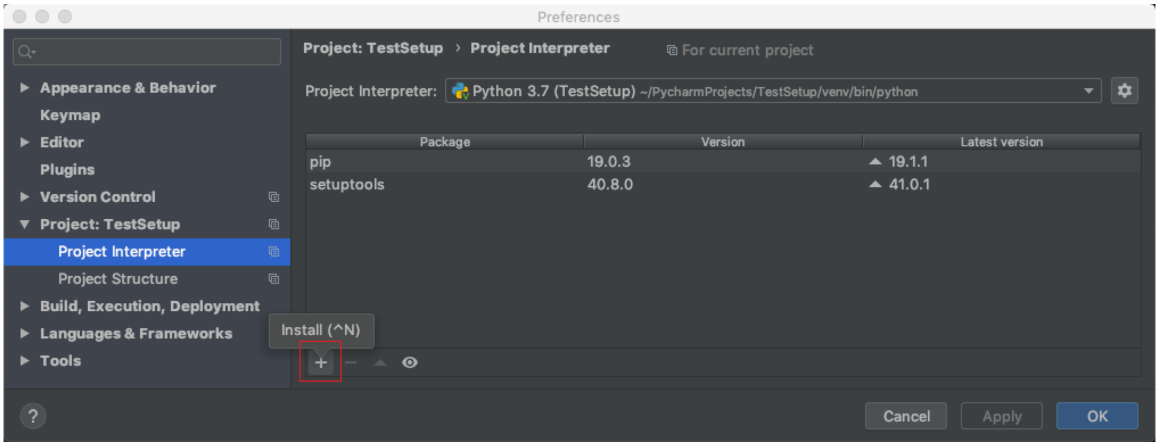
Then click “**OK**” again on the “Add Python Interpreter” screen.



/usr/local/bin/python3.7

Drag and drop a file into the space above to quickly locate it in the tree

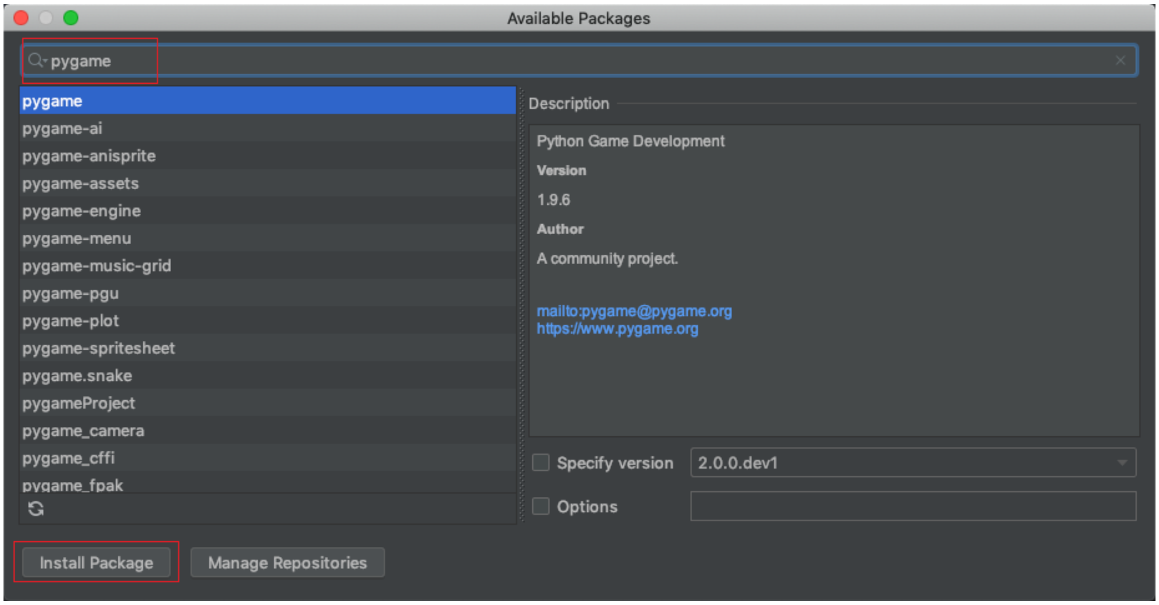
4. Now we are back to Preference screen under “Project Interpreter” section. Click “+” button to add a new package to the project.



Project Interpreter: Python 3.7 (TestSetup) ~/PycharmProjects/TestSetup/venv/bin/python

Package	Version	Latest version
pip	19.0.3	▲ 19.1.1
setuptools	40.8.0	▲ 41.0.1

Search for pygame and click “Install Package” on the package with the exact name.



Python Game Development

Version

1.9.6

Author

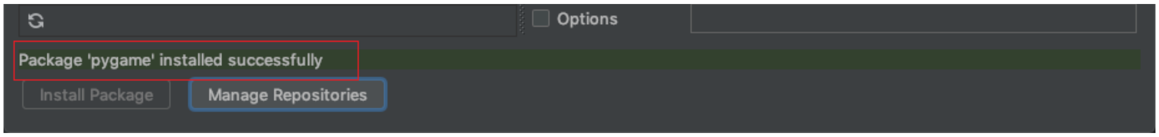
A community project.

mailto:pygame@pygame.org
https://www.pygame.org

☐ Specify version 2.0.0.dev1

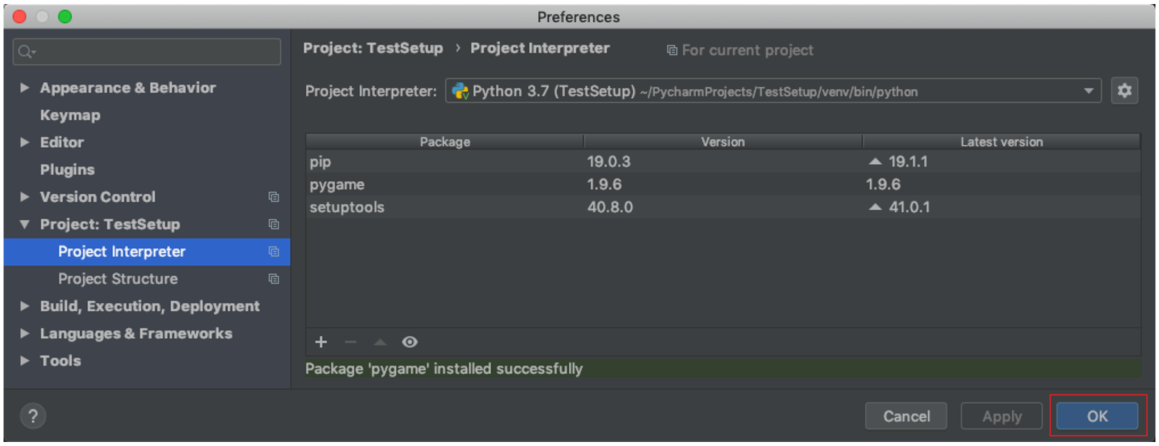
☐ Options

It will say “Package ‘pygame’ installed successfully”, then we can close the window.



Package 'pygame' installed successfully

Back to Preference screen, now click “OK”. Wait for the process to finish on the bottom progress bar.

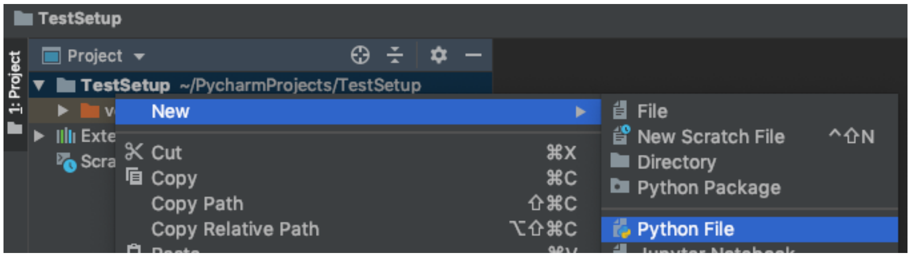


Project Interpreter: Python 3.7 (TestSetup) ~/PycharmProjects/TestSetup/venv/bin/python

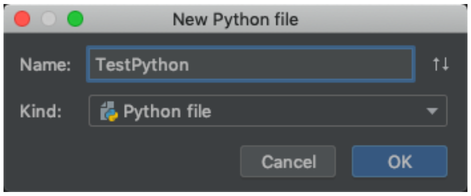
Package	Version	Latest version
pip	19.0.3	▲ 19.1.1
pygame	1.9.6	1.9.6
setuptools	40.8.0	▲ 41.0.1

Package 'pygame' installed successfully

5. Create a new Python file in the project by **right-clicking** the root(upper-most) folder, then “**New**”, “**Python File**”.



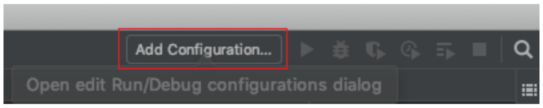
Give your new python file a name:



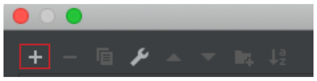
Name: TestPython

Kind: Python file

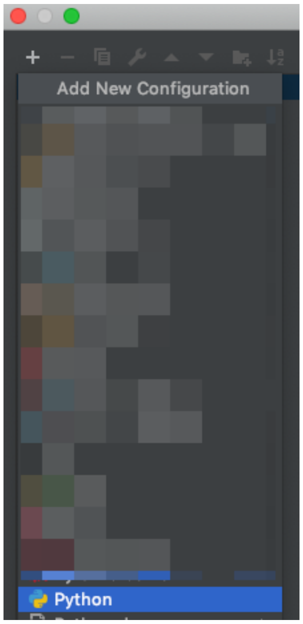
6. On the top-right corner, click on “**Add Configuration**” button.



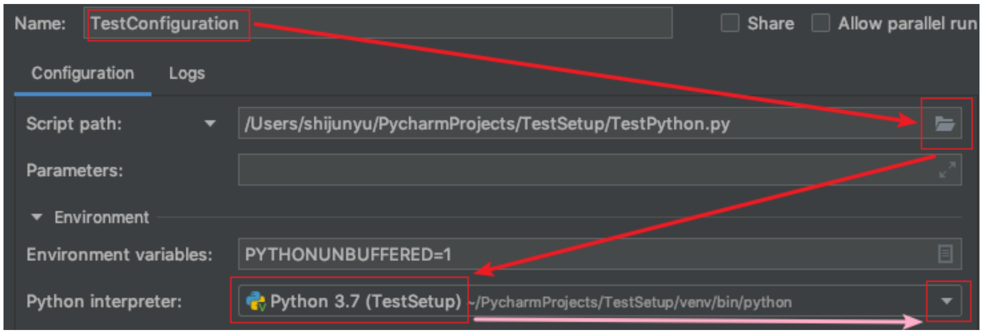
Click “+” to add a new configuration on the top-left corner of the new screen.



Select “Python” in the window under the “+” button after clicking on it.



Give your configuration a name, such as “TestConfiguration”; then click on the Folder Icon on the right end of Script path, select the python file that we just created.
Next verify the python interpreter is “Python 3.7”: if not, open the drop-down menu by clicking the triangle on the right (follow the pink arrow), and select Python 3.7. (If no Python 3.7 interpreter is available, refer to step 4)



Name: TestConfiguration

□ Share □ Allow parallel run

Configuration Logs

Script path: /Users/shijunyu/PycharmProjects/TestSetup/TestPython.py

Parameters:

Environment

Environment variables: PYTHONUNBUFFERED=1

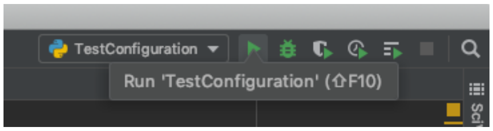
Python interpreter: Python 3.7 (TestSetup) ~/PycharmProjects/TestSetup/venv/bin/python

7. Now all the setup part is done, let’s test out Python and Pygame works with a short script!
   Put the following code (you can copy and paste) into the python file we created earlier (TestPython.py in my example).

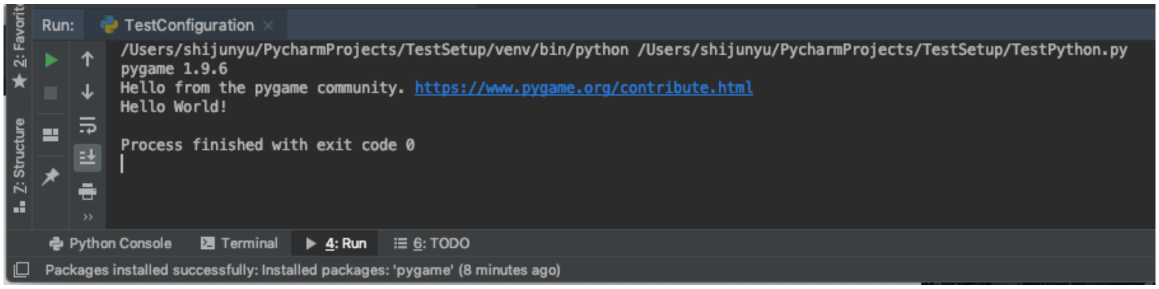
```
import pygame

print('Hello World!')
```

Go to the top-right corner, we can run our Configuration by clicking the green Triangle next to the Configuration name.



Check the bottom left-corner, if your program runs and print message from Pygame and “Hello World”, then you are all set!



```
/Users/shijunyu/PycharmProjects/TestSetup/venv/bin/python /Users/shijunyu/PycharmProjects/TestSetup/TestPython.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Hello World!

Process finished with exit code 0
```

Packages installed successfully: Installed packages: 'pygame' (8 minutes ago)

Now sit back and relax while waiting for other people to finish. Feel free to play around with more python commands as well!