

Installing Python and the PyCharm Editor/IDE

Goals for Getting Started:

- Install the Python interpreter for running programs
- Install the PyCharm text editor/IDE for writing programs
- Download and install the course lab files

Before our first class, please complete these instructions. They may seem long, but you should be able to get through them in less than an hour. Issues? dbb212@nyu.edu.

Sometimes it seems more efficient to "skim" long instructions and skip ahead on your own; **please don't do that here.** Instead, please read through and follow each step, and let me know if you encounter any problems or if you have questions.

Keep in mind that any divergence from these instructions can create problems now, or later in the course. One of the first lessons that I can offer is the necessity for attention to detail in this environment.

(If you are already equipped to write and run Python programs, you may not need these instructions. If you would like to use your own setup for writing and running programs, please contact me to discuss.)

Install the Python Interpreter from Miniconda

Miniconda is a Python distribution that will make it easy for us to install additional modules if and when we need them. If for some reason you don't want to install Miniconda, you can install the Standard Distribution of Python, or the Anaconda distribution – contact me for more information. Once you have installed Python, you may advance to **Installing PyCharm**, below.

Please note that you never need to uninstall any version of Python! Multiple versions can be, and often are, installed on the same system (I currently have four versions installed on my computer). If we find that the wrong version of Python is run when you try to use it, we will simply change the 'default' version for your system. (One exception might be to free up space.)

Visit <https://docs.conda.io/projects/miniconda/en/latest/>. Scroll down to see a series of links appearing under "Latest Miniconda installer links".

1. Mac:

- a. Choose **Miniconda3 macOS Intel x86 64-bit pkg** (for older, Intel-based Macs) or **macOS Apple M1 64-bit pkg** (for newer, "M" series Macs). If you're not sure which type of Mac you have, click on the Apple icon, choose "About This Mac" and look at "Processor" – you'll see "Intel" or "M". (Please make sure not to choose **64-bit bash**.)

- b. The installer will begin to download, usually to your **Downloads** folder.
- c. When complete, find and double-click the installer.
- d. Click **Continue** or **Agree** over the next 4 dialogs.
- e. At **Installation Type** (highlighted on the left), the installer will announce **Standard Install on "Macintosh HD"**.
 - i. Please click the button at the lower left labeled **Change Install Location...**
 - ii. At **Select a Destination**, click **Install on a specific disk...**
 - iii. Select your main disk, which should be **Macintosh HD**
 - iv. Click **Choose Folder...**
 - v. Select your home directory and click **Choose**
 - vi. Returning to the **Select a Destination** dialog, you should see "You have chosen to install this software in the folder "[homedir]" on the disk "Macintosh HD". **[homedir]** should be the name of your home directory. Click **Continue...**
 - vii. Back at **Installation Type** click **Install**.
- f. **PLEASE NOTE** if the Miniconda installer asks for your permission to make changes (requesting a password), please stop installation and start again, making sure to follow the instructions for **Installation Type** above. The installer should note that Miniconda will be installed in your home directory.
- g. If it appears that the installation location is not your home directory, please contact me.
- h. Installation may take several minutes. When done, the installer will tell you installation is complete.

2. Windows:

- a. Choose **Miniconda3 Windows 64-bit**.
- b. The installer will begin to download, usually to your **Downloads** folder.
- c. When complete, locate and double-click the installer.
- d. On "Open File – Security Warning" click **Run**.
- e. On "Welcome to Miniconda3" click **Next**.
- f. On "License Agreement" click **I Agree**.
- g. On "Select Installation Type" make sure **Just Me** is selected; click **Next**.

- h. On “Choose Installation Location” make sure the path is as shown here:

C:\Users\[homedir]\miniconda3

...where **[homedir]** is the name of your home directory.

If the directory shown there is different from above, please do not accept as is – instead, please edit the path so it is as shown above (of course replacing **[homedir]** with your home directory). **Please save this path for later in these instructions.**

- i. On “Advanced Installation Options” leave the default check marked boxes and click **Install**.
- j. Installation may take several minutes. When done, the installer will tell you installation is complete. (If you leave the two “resources” checkboxes checked, Miniconda will open browser windows with background on Conda and Anaconda. You may uncheck these boxes if you prefer.)

Confirm the Path to Python

1. Test whether your system can find Python by name.

- a. Windows: use the Windows search blank (usually at the bottom left of your Windows screen) to search for and open a **new** “Anaconda Prompt” window.

At the prompt in the newly opened window, type **where python** and hit **[Enter]**. You will likely see two paths listed.

If the *first* path is **C:\Users\[homedir]\miniconda3\python.exe** (where **[homedir]** is the name of your home directory), then your path is correctly set. You can skip to “Installing the PyCharm Editor\IDE” below.

If you can't find the Anaconda prompt, or do not see the **miniconda3** path, or it is not the first path listed, please contact me.

- b. Mac: use the Spotlight search (the magnifying glass at the top right of your Mac screen) to search for an open a Terminal window. Make sure this is a newly opened Terminal window, not an existing one.

At the prompt in the newly opened window, type **which python**. If you see one of these paths:

/Users/[homedir]/miniconda3/bin/python

= or =

/Users/[homedir]/opt/miniconda3/bin/python

(where **[homedir]** is the name of your home directory), then your path is correctly set. You can skip to “Installing the PyCharm Editor\IDE” below. If not, please proceed to the next numbered step. **Please save this path for later in these instructions.**

2. Mac: verify the install path for miniconda.

(This step is only needed if you are on the Mac and the **miniconda** path did not show up under **which python**.)

- a. Open a *new* Command Prompt or Terminal window. *(It must be a new window.)*
- b. Mac: type **ls**.
- c. Look through the files in this listing, which should be in alphabetical order (although it may show uppercase and lowercase in separate sections of this listing).
 - a. Look for the item **miniconda3**. If you see this item, then your correct path is this (where **[homedir]** is the name of your home directory):

/Users/[homedir]/miniconda3/bin

- b. If you don't see **miniconda3** in your home directory, look for the item **opt**. If you see this item, then please issue this command at the command line: **cd opt**. Following that, again please issue the command **ls**.

Look through the files in this listing for the item **miniconda3**. If you see this item, then your correct path is this (where **[homedir]** is the name of your home directory):

/Users/[homedir]/opt/miniconda3/bin

Please save the path you found for later in these instructions.

If you do not see **miniconda3** or **opt/miniconda3** in your home directory, and you are sure you have run the miniconda installer, please get in touch with me.

3. Mac: set the PATH environment variable.

This step is needed only if you are on the Mac and your version of Python did not show up in response to **which python**.

Use the spotlight search (the magnifying glass at the top of the screen) to search for **Terminal**, and select it, or you may use the existing Terminal window you opened earlier.

- i. In the opened terminal window, click right after the prompt and type **cd**, then hit **[Enter]**.
- ii. If your Terminal prompt ends in a % sign, type **touch .zshrc** and hit **[Enter]**. If the prompt ends in a \$ sign, type **touch .bash_profile** and hit **[Enter]**. (No output should result.)

- iii. Still at the prompt, type **open .zshrc** or **open .bash_profile**, choosing the filename that you touched above. Your mac's default text editor should open.
- iv. If you see a ruler and formatting buttons at the top of your edit window, from the menu please select **Format > Make Plaintext**. The ruler and buttons should disappear.
- v. If the file is not empty, scroll all the way to the bottom of the file.
- vi. Type one of the following paths into the file, on its own line, based on the path that you found earlier:

```
export PATH=/Users/[homedir]/miniconda3/bin:$PATH
```

=Or=

```
export PATH=/Users/[homedir]/opt/miniconda3/bin:$PATH
```

[homedir] represents the name of your home directory.

The **export** command tells your system to look in that directory when attempting to execute the command **python**.

- vii. Save the file and close out the editor. Make sure to close out your existing Terminal window(s), as only newly opened windows will see the change.

Again, please *close all Terminal windows* and return to the top of this section (“Test whether your system can find Python by name”) and test that the PATH environment variable has been set.

Install the PyCharm Editor/IDE

Special Notes on Text Editor / IDEs:

- if you have PyCharm already installed, you can skip this section on installing, but we will also make sure that PyCharm is using Miniconda as its Python version
- if you prefer to use another IDE or text editor (such as VSCode, IDLE, Sublime, Eclipse, etc.) you may do so, but I strongly recommend that you proceed *only* if you're familiar with using this tool to perform the following: writing and running Python programs, and inputting text at the keyboard with the **input()** function. If you're not sure about any of these items and still want to use a different IDE, please contact me.
- please do not expect to use Jupyter notebook as your main text editor/IDE. We may make some use of Jupyter for in-class exercises and tests only. Jupyter is not appropriate for project work!

1. Download and Install PyCharm.

- a. Visit the download page at <https://www.jetbrains.com/pycharm/download>
- b. **Do not download PyCharm Professional! This will download a paid version on a 30-day trial.** Instead, scroll down to PyCharm Community Edition.
- c. Under 'PyCharm Community Edition', click the down arrow next to **Download** to select the version appropriate for your system.

For Windows, your likely choice is 'Windows (.exe)'. There are some windows machines that have ARM chips, but if you're not sure, it's unlikely you have one.

For Mac, older machines are Intel-based (choose macOS (dmg)) and newer ones are 'Apple Silicon', i.e. the M series chips. If you're not sure which type of Mac processor you have, click on the Apple icon, choose "About This Mac" and look at "Processor" – you'll see "Intel" (choose **dmg**) or "M1", "M2" or "M3" (for these choose 'Apple Silicon').

- d. Click the link and download will begin.
- e. Once downloaded, double-click the downloaded file:

i. On Mac:

- a. a window will appear suggesting that you drag the PyCharm icon ("PC") to your Applications Folder -- you can do so right in that window (you may need to enter a password, or confirm that this program is safe to run -- it is!). Additionally, when you first run the program, you will see a warning; simply Click 'Open' or otherwise affirm your intention.
- b. once the program is in **Applications**, you can also create a shortcut to the application if you wish, by dragging it from Applications to your Dock (the horizontal bar at the bottom of your screen with icons that launch programs). You can also create a Desktop shortcut by dragging the icon from Applications and holding down the Cmd+Option keys (so that a swoopy arrow indicating a shortcut is displayed) before releasing on the Desktop. (If you don't see a swoopy arrow, the application itself will be moved to the Desktop, which you do not want - it belongs in the Applications folder.)

ii. On Windows:

- a. follow the installer steps and accept all defaults.
- b. During the install process you can choose to have a Desktop Shortcut placed on your Desktop. (You will also be able to launch PyCharm using the search or **Start > Run...** box.) You can also choose to associate **.py** with PyCharm, but I do not recommend this.

- c. Continue to click **Next** until installation is complete. You may also choose to launch PyCharm for the first time here.

2. Launch PyCharm and respond to initial settings questions.

For Mac, find the PyCharm **PC** icon in **Applications** and double-click it, or if you placed an icon in the Dock at the bottom of your screen or on the Desktop, you can simply click that.

For Windows, use the icon you placed on your Desktop or go to the search box and type **PyCharm**; you should see your program listed as **JetBrains PyCharm**; click it.

PyCharm may ask a series of questions, for example if you would like to import settings (please do not import any settings), and share usage data with JetBrains to help them improve the product (this is your choice). PyCharm may also ask you to select a Keymap (choose the one appropriate for your platform), a Light or Dark color scheme (recommended: "High Contrast", but you may also select a light color scheme), a launcher script (skip this), and download plugins (skip this). These questions may or may not appear during this step.

Finally, you can click to launch PyCharm.

3. Check out the PyCharm Welcome Screen. This medium-sized box shows the PyCharm **PC** icon and "Welcome to PyCharm" at the top. This will be the starting point when we create new projects in PyCharm.
 - a. Projects is where we can quickly open projects that we have worked with.
 - b. Customize allows us to set color theme and font sizes. We will set these as part of the next step.
4. Set theme (this may have been accomplished earlier). The following steps specify a high-contrast color scheme. We should make PyCharm as easy on the eyes as possible.
 - a. On the left, click **Customize**. Three settings appear on the right.
 - b. On the right, click the **Color Theme** dropdown. Select **High Contrast** (recommended), or choose the theme that you prefer (the color scheme will be displayed as you select each option).

Launch PyCharm, Create a Project, Write and Run a Python Program

1. Start a new PyCharm project. A PyCharm "project" is a folder which will serve as the home directory for files that are part of a single software project.
 - a. Return to PyCharm (if it is still open), or Launch PyCharm, either by double-clicking the **PC** desktop icon, clicking the **PC** dock icon, or typing **PyCharm** into the Magnifying Glass or Windows search box and selecting **PyCharm CE** (Mac) or **JetBrains PyCharm Community Edition** (Windows).

After PyCharm launches (it will temporarily display a brightly colored banner), you should see the medium-sized box titled "Welcome to PyCharm" and the PC logo.

If you see "Tip of the Day", close this box. You can also uncheck the box that sets PyCharm to display tips of the day each time it starts. (Or you can enjoy the tips if you want to learn more about PyCharm.)

If you see a large screen with multiple panels, you are looking at an open project. Close this project with **File > Close Project**. You should then see the "Welcome to PyCharm" medium-sized box.

- b. Click on the **New Project** button. PyCharm displays a **Name** blank, a **Location** blank, an **Interpreter Type** box selection and a **Path to conda** blank.
 - i. Look at the path to the folder that PyCharm intends to create (but do not click **Create**). PyCharm suggests a location in a special folder called **PycharmProjects** in your home directory. The folder is initially titled one of these:

C:\Users\homedir\PycharmProjects (Windows)

/Users/homedir/PycharmProjects (Mac)

...where **homedir** is the name of your computer's home directory. If you prefer to choose a different location to save your projects, click the folder icon to the right of the **Location** blank and select a folder location. You can also type out another full path. PyCharm will keep track of this location, but you may want to note it down for future reference.

- ii. In the **Name** blank, name your project **session test**. Do not click **Create** yet -- if you did, chose **File > Close Project** and start **Create New Project** again.

Special notes on naming your project:

- do not click **Create** until after you've chosen an interpreter, below -- if you do, chose **File > Close Project** and start **Create New Project** again.
- never use a space in any filename, directory name or project name you create -- this can cause problems in other contexts. You can use an underscore instead.
- please do not put a **.py** at the end of your project name -- this is reserved for Python files, not projects

- iii. Select the Python interpreter. Now we'll tell PyCharm where to find Python, and this choice will be remembered for future projects as well.

- 1) At **Interpreter type**, please select **Base conda**. This is highlighted very lightly but is important.

2) The **Path to conda** should be one of the following:

Windows: C:\Users\[homedir]\miniconda3\condabin\conda.bat

Mac: /Users/[homedir]/miniconda3/bin/conda =or=
 /Users/[homedir]/opt/miniconda3/bin/conda

...where **[homedir]** refers to your directory's name. If you do not see one of the paths above, try to add this path as shown here. If that path is rejected, please get in touch with me right away.

c. Click **Create**.

If you see the dialog titled **Create Project** with the message "The directory xxx is not empty" (or similar), this means that you specified a project name that already exists. Choose a new name. (Do not click **Create** yet -- if you do, chose **File > Close Project** and start **Create New Project** again.)

d. On Windows, if you see a message starting with **Windows Defender might be...** you may choose to **Fix** and **Configure automatically**, or **Don't show again for this project**. I am uncertain as to how either choice will impact your experience, but it does not appear to present a problem either way. The inferred choice is **Fix and Configure**.

e. Wait for PyCharm to finish indexing. When opening or launching a new project, PyCharm needs to prepare the project by searching through it. This is marked by a swirly progress bar and other activity at the bottom of the PyCharm screen. Wait for this activity to subside before proceeding.

f. Check out the PyCharm project view. You should see a left side rectangular window, and a larger main window to its right. If you don't see the left side window, click the folder icon at the upper left margin of the window.

The left-side window is the *project view* -- there you can see the project folder (a little folder icon with the same name as the name you gave the project, such as **session_test**). You can use the triangle next to the project folder to view files inside the folder (after we create some). **External Libraries** lists any installed Python libraries you may be using in the project; this and **Scratches and Consoles** are not meaningful to us at this stage.

If you see a **venv** folder in the project view, you should close this project and start again, choosing **Base conda** as your **Interpreter type**.

g. **You can open and close the project view with Alt-1 (Windows) or Cmd-1 (Mac).**

h. At the very top of the project window, note the location of your project folder. You can see its location, starting with **C:\Users\[homedir]** (Windows) or **~/** (Mac) at all times when you are working. You may need to remember this path if you want to access these files outside of PyCharm.

(On Mac, the tilde (~) represents your home directory, which you recorded earlier and is usually `/Users/[homedir]` where *homedir* is your or your computer's homedir.)

2. Turn off warnings and suggestions. By default, PyCharm will mark up your code as you type, providing suggestions and warnings about your code. Besides being distracting, these warnings may also be misleading in the early stages of our learning. At this stage, it's essential that we be free to evaluate our code independently.

Accordingly, please select **File > Power Save Mode**. If after doing this you see underlinings, light bulbs or other suggestions, please let me know!

3. Write a new Python program. Program source code is written in "plaintext" (unformatted text). We will write our source code inside PyCharm, then send the code to Python to read and run the program.

- a. Locate the project folder. In the project view (the left-hand window of PyCharm), look again for the *project folder icon*, which looks like a file folder and has the project name (**session_test**).

- b. **Right-click** the project folder icon (next to the name you gave the project) and select **New > Python File** (*not New > File or New > New Scratch File*)

- c. Name your file **hello.py** (or whatever name you prefer, but ending in a **.py** extension). *Make sure not to use spaces in any file or folder names! Use underscore instead.*

- d. Press **[Enter]** -- the main window on the right becomes blank, with a small tab above it. Look to the project view on the left -- the file appears under the project folder (click the tiny triangle to show and hide it).

The file should be inside the project folder (indented directly below and to the right of it), but please double check the location of your new file -- it should be below and to the right of it, not aligned with it. If the file is not located there, you can select and delete it, click on the project folder, then right click and choose **New > Python File** again.

- e. Compose your first Python script by entering the following Python code in the main window:

```
print('hello, world!')
```

...be careful to replicate the text shown here exactly.

- f. You can save your project with **Ctrl-S** (Windows) or **Cmd-S** (Mac) -- a good habit for any file you're writing -- although PyCharm saves projects (and the files within) automatically.

- g. If you see a message at the top of your program reading "No Python interpreter configured for the project", this means that PyCharm can't find Python. The needed steps are detailed below in the major section Select Python Interpreter, below.

4. Run the Python program for the first time. Again, there may be some activity at the bottom of the PyCharm window. Please wait for this activity to end before continuing.
 - a. Right click the file and select **Run 'hello'** (*do not select **Run file in Python console***).

If you don't see "Run 'hello'" in the menu when you right-click the file, or this option is greyed out, PyCharm may be indexing as described above. You'll know this is the case if you see swirly progress bars and other activity along with **Indexing...** or **Updating skeletons...** noted at the bottom of the PyCharm window. Simply wait several minutes and right-click the file again -- you should eventually see **"Run 'hello'"** in the menu. (Please do not select **Run file in Python console**.)

After clicking **Run 'hello'**, a new horizontal tabbed window should appear at the bottom of the screen with the result of running your code. In this new **Run** window at the bottom of the screen, you should see **hello, world!** followed by the words **Process finished with exit code 0**. This last line (**Process finished...**) indicates that your Python program has finished executing successfully. If you don't see this line, it means that your program has not yet finished executing.

- b. If you see **Traceback:** or **File [filename]** along with 4 or so lines of output (some of this text may be colored red) with the word **SyntaxError**, **IndentationError** or **NameError**, you've made your first error. Congrats! This is the first of many, :) and an important part of the learning process. Go back to your script and compare your text character-by-character with the program text: **print('hello, world!')**. It must match this program character-by-character to be correctly interpreted by Python. If you replicate this code source exactly, you should not see an error. If you see an error that you can't account for and prevents you from running your script, please contact me.
 - c. Again, please note that when PyCharm is used for the first time, it is sometimes slow to create projects or run programs (if this is the case you'll see a progress bar with **Indexing...** or **Updating...**). This delay should not occur after the first few runs of a program or creation of a project.
5. Edit, re-run, edit, re-run... Here's a tip: it can be very helpful to develop a keyboard-based workflow that will allow you to run through the development cycle repeatedly, without having to move to the mouse or trackpad, as shifting your hands between mouse and keyboard repeatedly is unnecessary effort.
 - a. Click in the edit window (the window with the code in it) and make an edit to your script (change **world!** to **weird!**).
 - b. Use a keyboard shortcut to execute the script: **Ctrl-Shift-R** Note the change in output in the Run window.
 - c. Continue making edits to the script. When done, use **Ctrl-Shift-R** to run the script again.
 - d. Attempt to make each "round trip" without moving your hand to the mouse. You'll save hours of this movement in the long run!

6. Running a new second script. You may notice a green triangle near the top of the PyCharm window and also on the lower left of the window. It's important to know that these are "Re-run" buttons which can only run the script that has been run last. To run a new script for the first time, right click the main window and select **Run 'newscrip'** (where *newscrip* is the name of your new script, not the actual name of the script). Or use, **Ctrl-Shift-R** to run this script from the keyboard.
7. Select an editor font.
 - a. Choose PyCharm > Preferences (Mac) or File > Settings (Windows). A Preferences / Settings window appears.
 - b. On the left side menu, click on **Editor**, and in the submenu below that, **Font**.
 - c. In the Size blank, enter a minimum size of **20.0**. Feel free to choose larger according to your comfort, but I can't recommend smaller – we must pay attention to every character we type and a small font makes this more difficult.
 - d. Click **OK**.

Download and Install the Course Lab Files

The lab files contain all of the inclass exercises and data files that we will use in this course.

1. Download the zip file.
 - a. Please visit the class website.
 - b. At the top of the home page find the link **download source data** and click it. The download should begin automatically, usually to your **Downloads** directory. The file is called **python_data_ipy.zip**.
 - c. You may leave the zip file where it is, or you may move it to a more familiar location, but you should be able to navigate to it in an upcoming step.
 - d. Unzip the file:
 - i. Mac: double-click the file; a folder named **python_data_ipy**
 - ii. Windows: right-click the **.zip** file and select **Extract All...** and allow Windows to suggest the folder name **python_data_ipy**. (Note that Windows may behave in varying ways when double-clicking, so I recommend this method.)

The folder should have the following the following structure:

```
python_data_ipy/  
  session_00_test_project/  
  session_01_objects_types/  
  ..etc..  
  session_10_classes/
```

Please make sure that your overall folder is named **python_data_ipy/** and that there are 11 folders within.

2. Open the class folder in PyCharm.

- a. If your earlier project is open, close the project with **File > Close Project**.
- b. Open the project folder. Choose **Open** and then navigate and highlight the new folder. Make sure you are not navigating to the .zip file, but instead the folder unzipped in the previous step.
- c. A “Trust and Open” dialog appears. You may trust this project ;)
- d. If the left side rectangular box is not showing, expand this project view on the left by clicking the folder icon or hitting **Ctrl-1 / Cmd-1**.
- e. In the project view on the left, see that the project folder is named **python_data_ipy** (you may need to open the project window on the left to see the folder); click the triangle next to this folder to see the folders numbered **session_00**, **session_01**, etc.

Anytime you’d like to start over, you can choose **File > Close Project** and start again.

3. Confirm/select the correct Python interpreter. At the extreme bottom right of the PyCharm window, look to see **Python 3.11** (or greater), or you may see a path that includes **miniconda3** -- both of these are good.

If you see a lower version of Python, or **Python 2.7** or **<No Interpreter>**, click it and select the version of Python you installed (which should indicate the **miniconda3** directory, or **Python 3.11** or greater).

If you don’t see either of these, please follow the directions for **Select Python Interpreter**, below. (Keep track of this step -- you’ll return here when you’ve selected the interpreter.)

When the project loads and the interpreter is selected, PyCharm will typically take a few minutes to index all the files in the project; this will be displayed as activity at the bottom of the window, including swirly progress bars and messages. You must wait for this activity to cease before running your Python file.

If you see “download pre-built indexes” you can try it, but I have not experimented with this feature.

4. Run a python file.

- a. In the project window, expand the **session_00_test_project** by clicking the tiny triangle next to it. Then, double-click **hello_zip.py**. PyCharm loads the code for this file in the main window on the right side.
- b. Anywhere in the code window on the right, right-click and select **Run 'hello_zip'**. You should see the **Run** window open at the bottom, displaying this program's output:

```
hello, unzipped file!  
  
your Python version:  
3.12.x (...etc.)  
  
your Python executable:  
/Users/homedir/miniconda3/bin/python3
```

If you see a message like 'Cannot run /usr/local/bin/python3.7', please return to Confirm/select the correct Python interpreter, above.

Please confirm your Python version - it should start with 3.8, 3.9, 3.10 or 3.11. The minor version after these does not matter. If you are seeing **2.7** or other message, please follow the instructions in Select Python Interpreter below, or contact me.

Select the Python Interpreter in PyCharm

This section is only necessary if after loading new project folder/project you see **<No Interpreter>** or a Python version below **3.11**, or if when you try to run the script you see an **Edit Configuration** dialog window (which you should immediately close). If you see a path to **miniconda** or **Python 3.11** or greater, your Python interpreter is set, and you're done.

1. At the far bottom right, click **<No Interpreter>** or whatever Python version **< 3.11** is displayed.
 - a. If you see a path including **miniconda** above the **Add New Interpreter** selection, or **Python 3.11** or greater, select it -- you're done!
 - b. If you do not see a path including **miniconda**, or **Python 3.11** or greater, above the **Add New Interpreter** selection, then click **Add New Interpreter** and then **Add Local Interpreter**. A dialog box titled **Add Python Interpreter** appears.
 - c. On the left of the **Add Python Interpreter** dialog, select **System Interpreter**.
 - d. To the far right of **Interpreter**, click the ellipsis (...). A smaller box titled **Select Python Interpreter** appears, with a path (usually the home directory, or **C:**) highlighted in blue.

Please retrieve the path to miniconda which we saved earlier, which should be:

Mac:

/Users/[homedir]/miniconda3/bin/python
= or =
/Users/[homedir]/opt/miniconda3/bin/python

Windows:

C:\Users\[homedir]\miniconda3\python

Please paste or type this path directly into the blank at the top of this box.

- e. If you can't find the original path or the above path doesn't work:
 - i. On Windows, return to your Anaconda prompt; on Mac, open the Terminal.
 - ii. On Windows, type **where python** in the Anaconda Prompt; on Mac type **which python** in the Terminal.

Again, if these commands don't show you one of the **miniconda3** paths shown above, please get in touch.)
- f. Copy this path by highlighting it and pressing *Ctrl-C* or *Cmd-C*. Make sure you are selecting the path to Python, not your home directory.
- g. Return to PyCharm and paste the **miniconda3** path to python into this blank by pressing *Ctrl-V* or *Cmd-V*. The path to Python should have replaced the blue highlighted path and be in the blank by itself. *If somehow the existing text got de-highlighted and is in the box alongside of the path, highlight the entire path and delete it; then paste the path to Python again.*
 - i. Click OK: you'll be returned to the **Add Python Interpreter** window.
 - ii. If you attempt to click OK but nothing happens, this means that that path is invalid. Check the path, or use the file tree below to navigate to the location of **python.exe** or **python** (i.e., the path to python you recorded earlier).
- h. Doublecheck that you have **System Interpreter** highlighted on the left. Make sure you have not highlighted **VirtualEnv Environment** or **Conda Environment** or any of the other **Environment** choices. If **System Interpreter** is not highlighted, click on it and repeat the instructions starting at **Select the Python interpreter** above.
- i. Click **OK** in the **Add Python Interpreter** window, then **OK** in the next window. PyCharm should go through some initialization steps that take a few minutes and show some moving, swirly progress bars at the bottom of the screen -- this delay should not be repeated.

You can now return to the step where you left off above, where you run a Python file from the open project.

Nice Job, Pythonista.

Hey congratulations, that was a lot of steps! You're now ready to start the class.

Going forward, you may want to refer back to these instructions for creating a new project. Please send questions to me at dbb212@nyu.edu, and good luck! I'll be with you all the way.