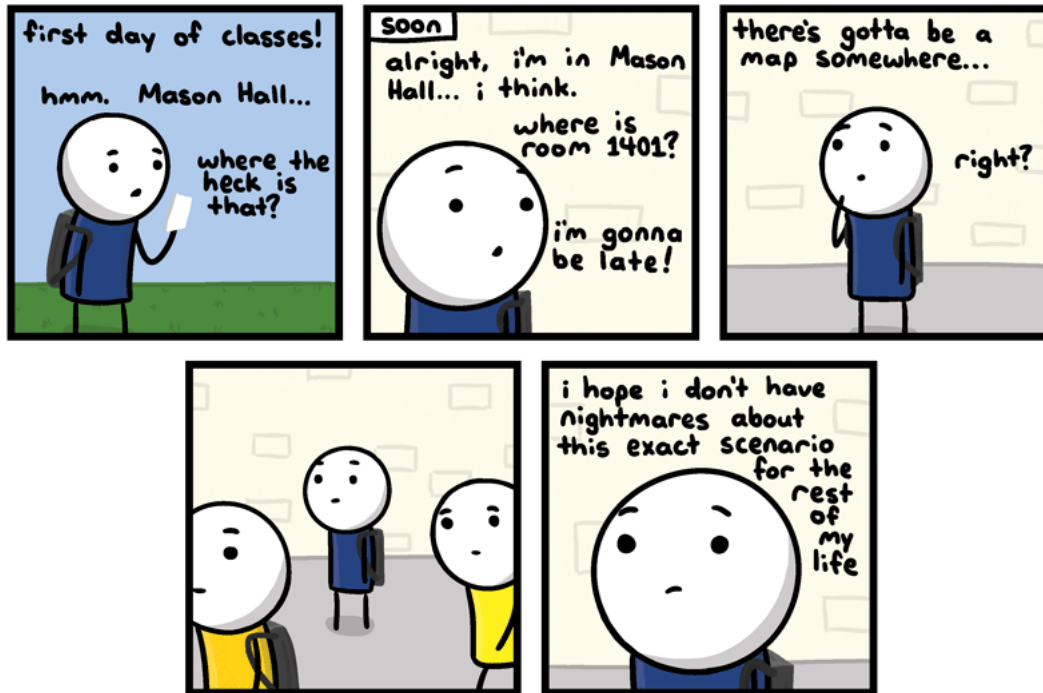


CS51A—Lab 1

PyCharm, Python, and GradeScope



invisiblebread.com

<http://joyreactor.com/post/954711>

This lab will walk you through the basics of using PyCharm, Python, and GradeScope.

1 Installing Python and PyCharm

In this semester, the first two weeks will be conducted online, so you are welcome to do these projects entirely on your own machine. To do so, you will need to download and install Python 3.10 and PyCharm.

The latest Python interpreter can be downloaded from <https://www.python.org/downloads> (with packages for Windows and Mac OS X). You should select the latest distribution of Python 3.10.1, and install it after it is downloaded.

PyCharm can be downloaded by going to <https://www.jetbrains.com/pycharm/> and selecting your platform (Windows, macOS or Linux). **Make sure you download the (free, open-source) Community Edition.**

It might be best if you attempt to do this before coming to the lab, so that we can help you more quickly to resolve any problems.

2 Starting PyCharm

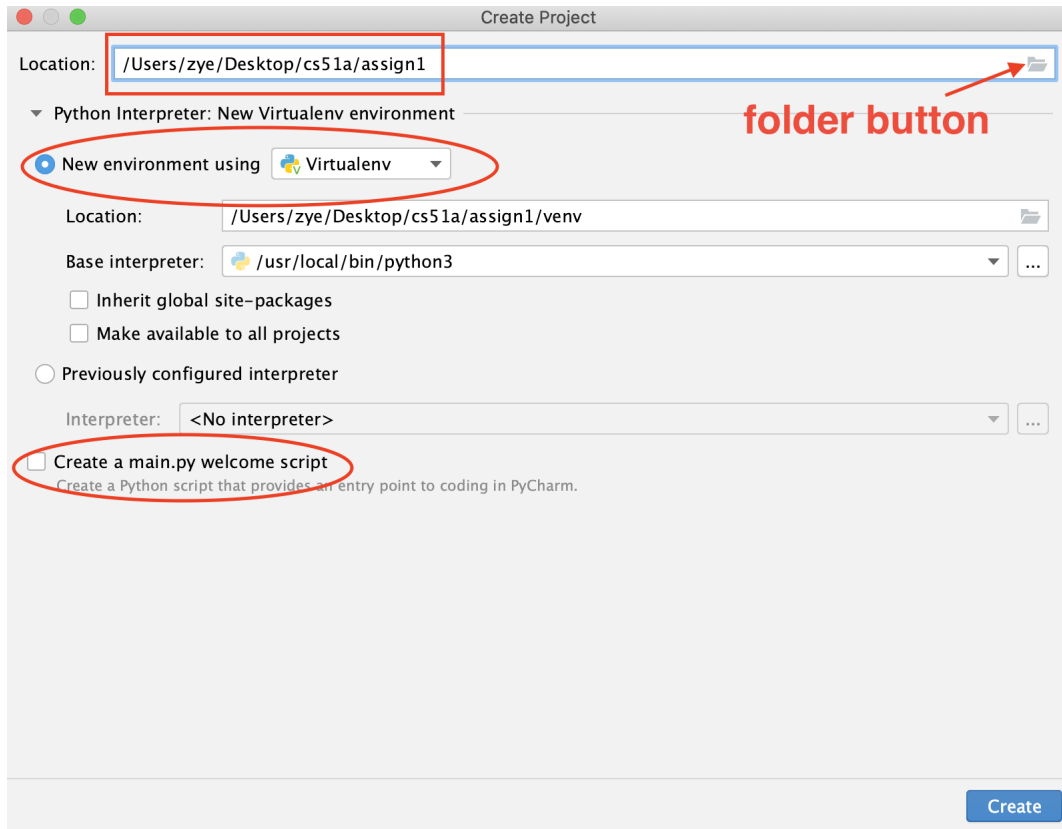
PyCharm will, by default, try to maintain all of your projects under a single folder. Before starting PyCharm for the first time you should create a new workspace folder. We suggest that you create a new folder on your **Desktop** named **cs51a**.

On a Mac, you can start PyCharm by clicking the **PC** icon (as shown below) on your menu-bar. On a Windows system, you can click **Start** and search the name of PyCharm to find it, and you can drag it to generate a shortcut or pin it to the start menu.



3 Creating a new project

After starting PyCharm you can accept the defaults and, eventually, should get to a screen that invites you to select **Create New Project**. Go ahead and create the project, making sure that the location of the project is a new sub-folder (e.g. **assign1**) in the **cs51a** you just created on your **Desktop**. In order to achieve this, you need to click on the **folder** button as shown below to navigate to your **Desktop**, and then select the **cs51a** to open. After that, in the **Location** fillable space, you need to manually type in the subfolder name **/assign1** to create this new subfolder



WARNINGS for new project creation

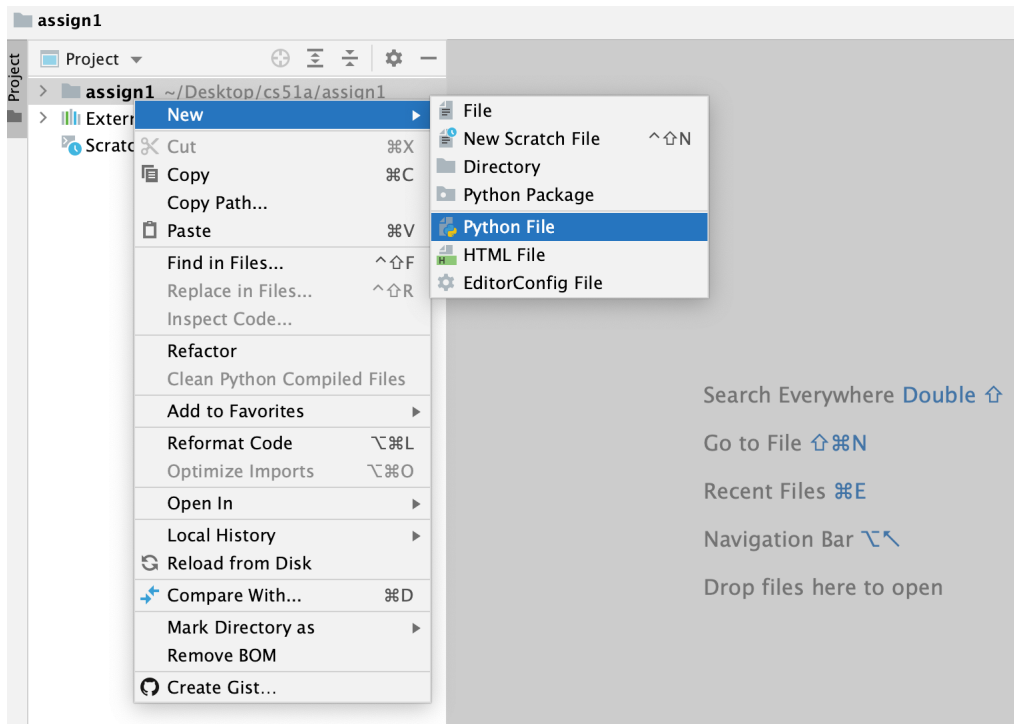
- make sure you put the new project in the `cs51a` folder you just created. Not in the default location, which will be very difficult for you to find.
- check the option `New environment using Virtualenv` to create a virtual environment, as highlighted in the above picture.
- uncheck the option `Create a main.py welcome script`, as highlighted in the above picture.

After these configurations, you can click on the `Create` button on the bottom right to create a new project.

Important Note: In this class it's probably best to use one project per assignment. Make sure it's inside of your `cs51a` folder and then go ahead and call it `assign1`, `assign2`, `assign3`,

4 Editing and running a Python program

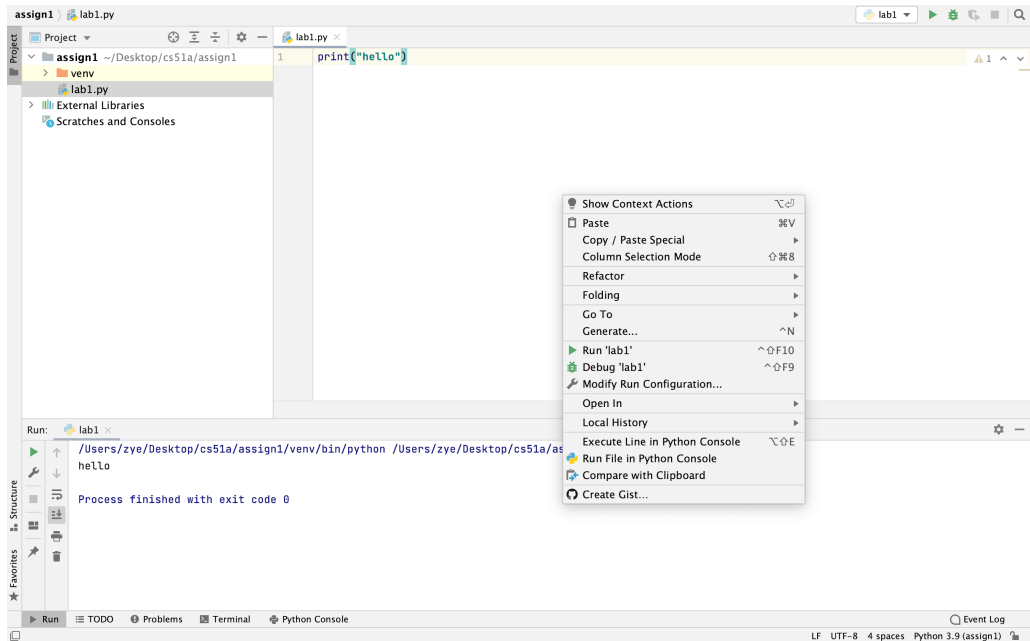
Once a new project is created, you can right click the project name (e.g., `assign1`) on the left panel, choose `New` and navigate to `Python File` to create a new python file in this project. An example is shown below.



You can give the name to the python file, e.g., `lab1` or `assign1`, and then enter. After that, you should be looking at the main screen with the following components:

- a menu bar at the top
- a (nearly empty at this point) list of projects and files on the left
- a main window on the right
- a list of screen selections on the bottom (note that a screen that displays the program output will be shown after you run a program)

You can edit your python program in the main window. For example, you can add a print statement in the file. Then, you can save the file by selecting `File/Save all` from the menu, or by typing `command + s` on Mac or `Ctrl + s` on Windows.



The easiest way to Run a program that you are editing is to place the cursor in the file editing window and right-click. As shown in the above figure, one of the options will be Run (and the name of the currently selected file). If you select this option, the Python interpreter will execute the selected program and display the output (in a Run window) at the bottom of the screen.

Another way to run your python program is to use the green triangle button on the top right of the window. You can choose which python program you want to run with a selector and buttons near the top right of the screen:

- a drop-down list contains the names of each previously run file.
- a triangular run button will re-run the selected program.

Important Note: at the bottom right of the screen, you will be able to see the python interpreter that you are using for this project. Make sure that it is Python 3.x, e.g., Python 3.10

After running your first python program, you can perform another practice by adding a few more print statements. Add the following print statements to your program and run it again. You should now see some results printed out.

```
print(10) # printing a number
print(22/7) # printing another number
pi = 22/7 # assign 22/7 to pi
print(pi) # print out what is stored in pi
print("Hello computer user") # printing a string
```

5 Playing with the Python Interpreter

PyCharm has three different working sections. We'll mostly use the top right one, which is where we will edit files and the bottom one, which gives us an interactive shell or shows the results of running our programs. You can adjust these internal window sizes to make them bigger or smaller, move where they are, etc.

Besides, we can use an interactive shell, which is the `Python Console`. Try a few commands in the interactive shell (from the `Tools` menu, pick `Python Console...` or hit `Command, Shift, and A` at the same time and type “`Python Console`” and hit return).

```
>>>
```

is the prompt, meaning Python is ready for a command,

```
...
```

means it's a line continuation (i.e. Python is waiting for you to finish the statement) and a line without anything in front of it is generally the response from the interpreter.

- Try a few simple mathematical equations (e.g. “`1+1`”, “`2**3`”, “`(100/20)+45*7`”). Notice that Python makes for a pretty easy to use calculator.
- `22/7` is an approximation for Pi. Type this in.
- Remember that we can use variables to store intermediary values. Assign `22/7` to a variable called `pi`. Use that variable to calculate the area of a circle with radius 15 (remember, the area of a circle is π times the radius squared).
- In 2011, Pomona built a new parking structure. For a variety of reasons, the college decided to put a field on top of the parking structure (pretty cool, right?!).



Since we're in southern California, we don't need any covering for this field. However, let's pretend that we anticipate bad weather more regularly and we decide we need to cover it

with a dome (global warming, maybe?). If we make the approximating assumption that the field is a circle of radius 100 feet, what is the surface area of the dome that would cover it?

Hint: the surface area of a *sphere* is $4\pi r^2$.

In the Python shell, you can use the up and down arrow keys to revisit commands you typed previously. Use the up arrow key to get your previous statement and then edit it to get the surface area of the dome assuming that the radius is 200 feet. Discuss your answers with your neighbor and make sure you agree.