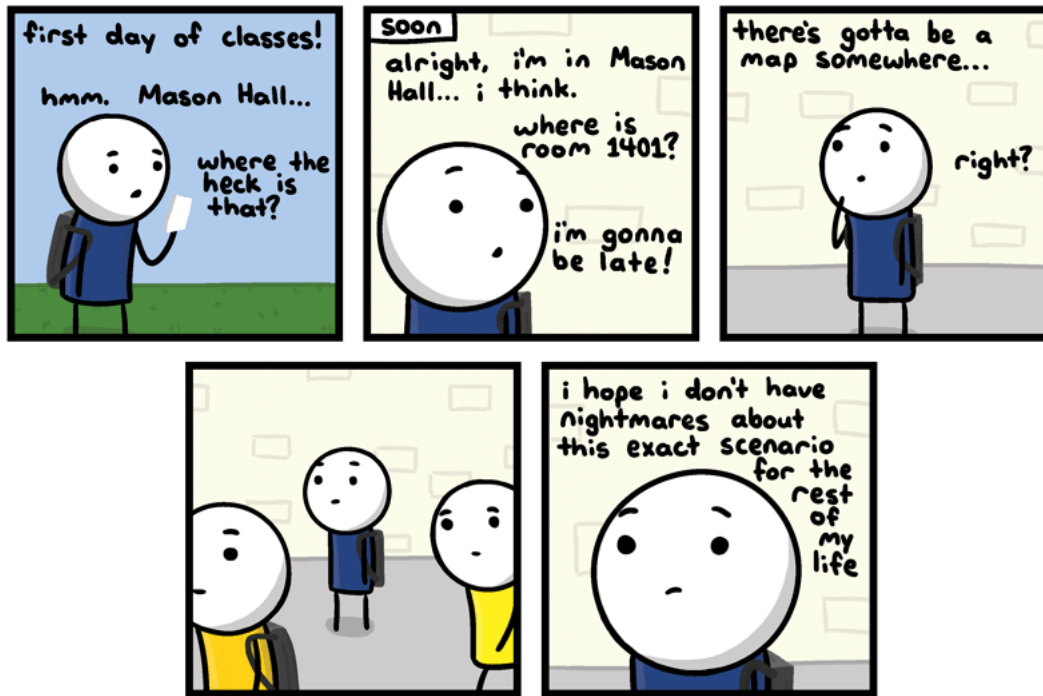# CS51A - Lab 1
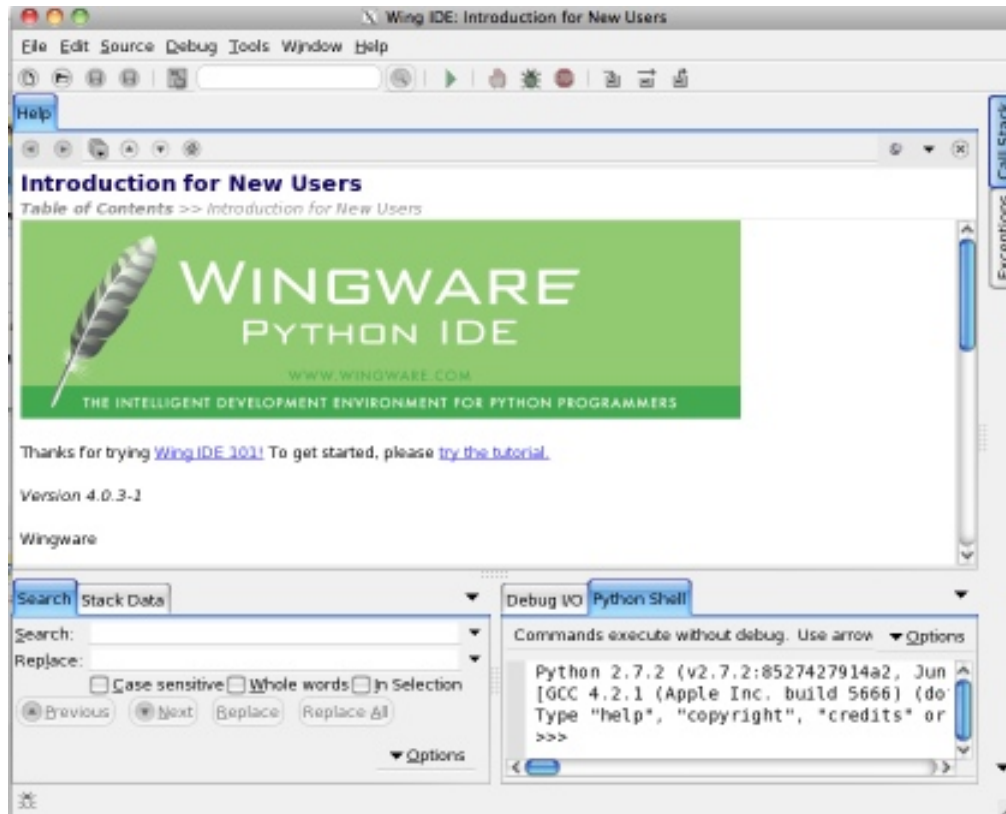
This lab will walk you through the basics of using WingIDE, Python and the online submission mechanism.

1 Accounts

You should have received an e-mail with you CS account information containing your username and password. Note that this CS account is different from your other campus accounts. Files that you store on these computers will not be accessible when not in the CS labs and vice versa. Come talk to me if you ever want to transfer files from your CS account (though, e-mailing them to yourself or uploading them to a cloud file service is also an easy solution). If you want to work on the lab machines, this is the account that you will use to login.

2 Some Housekeeping

- You'll be spending a lot of time looking at the course web page, so go ahead and bookmark it in your browser at:

  http://www.cs.pomona.edu/classes/cs51a/

- Create a new directory on your desktop called **cs51a** (or somewhere where you will remember and can store all your work for the class)

- Create a new directory within the **cs51a** folder called **lab1**

- Create a shortcut for WingIDE on the Dock at the bottom of the screen

  If you're working on the lab computers:
  Mac includes shortcuts at the bottom of the screen called the "Dock" that allows you to launch applications, folders, etc. quickly. Below is a picture of my Dock:

  We're going to add WingIDE to the dock for easy access (what does IDE stand for again? :).)

  To add an application to the Dock, all we need to do is find it and then drag it over the Dock where we want it. The easiest way to do this is to *search* for the application we want. Click the magnifying glass at the top right corner (or type `command + space_bar`) and then type "Wing101". As you start to type you should see "Wing101.app". Click and drag this into your Dock at the bottom of the screen to add it.

- You can now click on the logo in the Dock (it looks like a feather) and it starts the WingIDE. Go ahead and do this.

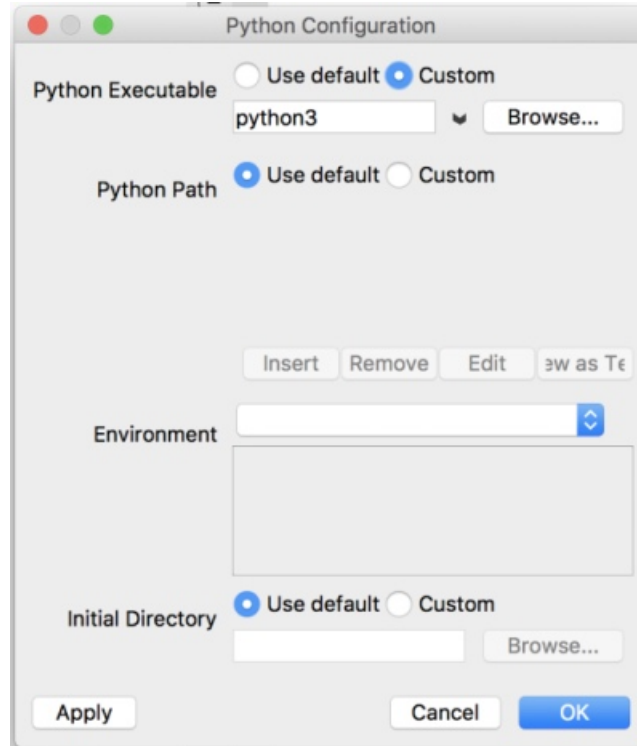 When you start WindIDE is should look something like:

If this screen comes then your Dock link for Wing worked!

- *If you're working on the lab computers:*

  The default on the lab machines is to use Python 2, but we'll be using Python 3 for this class, so we need to configure Wing to use this. In Wing, select "Edit → Configure Python...". In this new window, under "Python Executable" (right at the top), select "Custom" and then enter "python3" (no spaces) in the text box. I.e., it should look like this:

Python Configuration

Python Executable ○ Use default ● Custom
python3 ⌄ Browse...

Python Path ● Use default ○ Custom

Insert Remove Edit ew as Te

Environment

Initial Directory ● Use default ○ Custom
Browse...

Apply Cancel OK

Then click "OK". Wing will ask you if you want to restart. Select restart and then you should see the Python shell window in Wind restart and display 3.7.0. If it does, you're all set. If not, come talk to either the Professor or one of the TAs and we'll sort it out!

3 Playing with the Python Interpreter

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

Try a few commands in the interactive shell.

```
>>>
```

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.

4 Your first Python program

So far, we've only been interacting with the Python shell. This is good for some situations, but eventually, you're going to want to write longer programs that you can edit easily and persist when you quit Wing.

- Create a new file in WingIDE

  Click `File->New` (or the new file button)

- Save the file in the folder you created earlier (`lab1`) as
    [first_name][last_name]-lab1.py
  (without the square brackets). For example, my lab would be:
    davidkauchak-lab1.py.

  Click `File->Save` (or the save button or type `command + s`)

  This will bring up a prompt for the location of the file.

  Your desktop can be found in `/home/yourusername/Desktop/`.

  Underneath the buttons in the "save" window, you'll see displayed the current directory. Double-clicking on things in the left side of the window (e.g. "Desktop") will move into that folder. Navigate until you're in your `cs51a` folder. We will use the file extension ".py" for all of our Python programs, which is the standard way to indicate that the file contains Python code. Make sure you include this, otherwise Wing won't know it's a Python program and won't do all the nice highlighting and indenting for you.

- Put a few statements from above in the file, save it and then run your program

  Each line you enter in the file will be executed in the shell as if you typed it. Save the file again (this time you shouldn't have to enter the file location).

  You can run your file by clicking the green arrow .

  When you do this, you'll notice that the Python shell in the bottom right of Wing restarts and then your program will execute. Most likely, your program won't show anything in the shell window. *Make sure you understand why your program displays what it does!* Discuss the output with your neighbor.

- Put some `print` function calls in your program

  When you are running a program (vs. interacting with the shell) the intermediary results are not shown. Instead, if you want things to be displayed you need to use the `print` statement. You can print anything that represents a value, for example:

```
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
```

  Add a few print statements to your program and run it again. You should now see some results printed out.

5   Testing assignment submission

We'll be using an online mechanism for submitting assignments. Go to the course website and follow the link for "Assignment submission". You'll need to enter your *college* login information

(not your CS). Enroll in the CS51A course and then submit your .py file under the link for "lab1". Once you've done this, ask the professor to double-check that your submission was received. If so, move on to the assignment!