

Lab 1, CSci 127, Introduction to Computer Science, Hunter College, CUNY, Fall 2024

 huntercsci127.github.io/f24/lab1.html

Lab 1

CSCI 127: Introduction to Computer Science

Hunter College, City University of New York

Fall 2024

Learning Objectives:

- Students will install and run Python on their Computer
- Students will **write and run a "Hello, World" program using Python**
- Students will **submit a program** to Gradescope
- Students will write programs that **use the Turtle library to draw simple graphics**
- Students will **explore several Turtle commands**

Software tools needed: web browser and Python programming environment.

Installing Python on Your Computer

The Python programming language and IDLE environment are freely available for many platforms from python.org. For this class, we are using **Python 3**.

Many features of the language (including the syntax of `print` statements) changed between the second and third version, so, you must use the Python 3 for submitting programs.

Linux:

If you have a Linux machine (Ubuntu or Debian based), at a terminal window type the following commands:

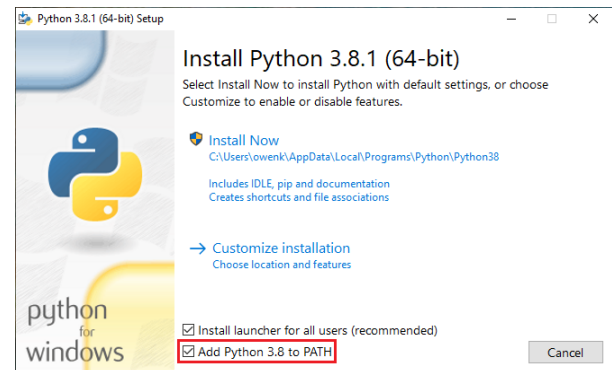
```
sudo apt update
sudo apt -y upgrade
sudo apt install python3 python3-pip idle3
pip3 install numpy pandas matplotlib scipy folium image
sudo apt install spyder3
```

Windows:

If you have a Windows computer, you can install python3 with the following instructions:

1. Go to the following [link](#).
2. Click latest Python 3 release.
3. Scroll down to the files section.
4. Click Windows x86-64 executable installer and save the file.
5. Once it is finished downloading, run the installer.
6. Make sure you click Add Python to PATH (This makes it a lot easier to install Python packages).
7. Click install now.
8. If asked do you want to allow this app to make changes to your device, click yes.
9. Wait until you see the setup was successful screen and then you can hit close.

You have successfully installed Python on your computer. To open IDLE, search IDLE in the Windows search bar.



macOS:

If you have a Mac computer, you can install python3 with the following instructions:

1. Go to the following [link](#).
2. Click latest Python 3 release.
3. Scroll down to the files section.
4. Click macOS 64-bit installer and save the file.
5. Once it is finished downloading, run the installer.
6. Hit continue, continue again, continue again, and then agree.
7. Click install.
8. Enter your password or use TouchID if prompted.
9. Wait until you see the installation was completed successfully screen and then you can hit close.
10. You can keep the install or move it to the trash.

You have successfully installed Python on your computer. To open IDLE, search IDLE in Spotlight search or find it in Launchpad. To open IDLE from terminal, type `idle3`.

Python Packages:

You can install the Python packages used in this course with the following instructions:

1. If on Windows, search cmd in the Windows search bar and open it. If on macOS, search terminal in Spotlight search and open it.

2. Install the Python packages by typing the following command:

```
pip3 install numpy pandas folium image matplotlib scipy
```

These installation instructions were written by Owen Kunhardt. You can find the full guides at the following links: [Windows](#) [macOS](#)

Using Python

We will be using the `IDLE` programming environment for Python, since it is very simple and comes with all distributions of Python. However, feel free to use any text editor or development environment that you prefer.

To launch `IDLE`:

- If you are running Windows, search IDLE in the Windows search bar.
- If you are running a Unix-like operating system such as a Linux distribution or MacOS, locate the **terminal** on your machine (you can search Spotlight for 'terminal' on a mac).
- In the terminal window, type:

```
idle
```

(followed by an enter/return).

- A new window will launch for IDLE.
- To see that it works, type at the IDLE prompt:

```
print("Hello, World!")
```

Write and submit your first program.:

Instead of using the shell window (where we can try things immediately), let's use a text window, where we can save our program for later and submit it to Gradescope (this is the basis of the first program).

1. First, open up a text window: on the menu bar, choose "File" and from that menu, choose "New File".
2. In that window, type:

```
#Name: ...your name here...
#Email: ...your email here...
#Date: Septmeber 3, 2024
#This program prints: Hello, World!
```

```
print("Hello, World!")
```

3. Save the program (using the "Save" under the "File" menu). When you save it, name it something that you will remember for the future and end it in `.py`. For example, `program1.py`.
4. Run your program (using the "Run Module" from the "Run" menu).
5. If it prints "Hello, World!" to the screen, then log into Gradescope (see notes below):
 - On the left hand menu, choose "Assignments".
 - From the list, choose "1. Hello, World!". In the file upload, drag over the `.py` file you just created and ran, and
 - click "Submit".

Using Gradescope

This course will use the online [Gradescope](#) system for submitting work electronically.

- Open your email and click the links to set up your account.

- Once you have your account, click on the "Assignments" menu (left hand side of window), and then choose the assignment "1. Hello, World."
- Drag and drop your program onto the submit box.
- Click on confirmation button to submit your program. You will see a pop-up box confirming that the program was uploaded.
- In 30 to 60 seconds, the Gradescope page will have your results (remember to put your name on the top of the file for full credit!).
- Each program can be submitted multiple times up to the deadline.

More Python: Turtles

Now that you have just submitted your first program, let's try some other Python commands. Here's a quick demo (click the triangle to run the program):

A quick overview of the above program:

- Lines that begin with # are ignored by Python-- they are comments for you to remember what you did and others to follow what's going on.
- The line `import turtle` loads in the built-in turtle graphics drawing package. It's part of all versions of Python but, to keep programs small, is not included unless you explicitly import it. In addition to built-in packages, there are many others that have been written to make Python more useful. We will use both kinds as the semester progresses.
- The line `thomasH = turtle.Turtle()` creates a turtle object called `thomasH` (you can call your turtles almost any combination of letters (and underscores and numbers)- - we used a name starting with "t" since turtle starts with "t").
- The next line: `for i in range(4) :` is the first part of a for-loop that will repeat the commands indented beneath it 4 times.
- The turtle class has many functions that you can use for your turtle. The next two lines demonstrate two of them:
 - `thomasH.forward(150)` moves `thomasH` forward 150 steps.
 - `thomasH.right(90)` turns `thomasH` to the right 90 degrees.

Let's write the same program in IDLE:

1. Open up a new file window in IDLE ("File > New File").
2. Type (or copy) into your window:

```
import turtle
thomasH = turtle.Turtle()
for i in range(4):
    thomasH.forward(150)
    thomasH.left(90)
```

3. Note: if you are working with IDLE, you may want to setup the screen to exit on click, so the graphics window does not stall:

```
import turtle
wn = turtle.Screen()

thomasH = turtle.Turtle()
for i in range(4):
    thomasH.forward(150)
    thomasH.left(90)

wn.exitonclick()
```

4. Save your program ("File > Save").

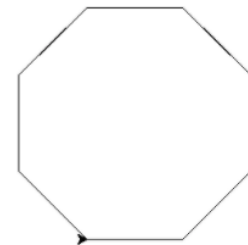
IMPORTANT: Choose a name for your file that is **not** `turtle.py`. When executing the "import turtle" statement, the computer first looks in the folder where the file is saved for the Turtle library and then in the libraries (and other places on the path). So, it thinks the module is itself, causing all kinds of errors. To avoid this, name your program something like "myTurtle.py" or "program2.py".

5. Run your program (using the "Run Module" from the "Run" menu).

6. **Modify your program so that it draws an octagon.**

7. Test your program and modify until you have a octagon (**Hint:** you need to modify the number of repetitions -- number in parenthesis after `range`-- and the angle -- number in parenthesis after `left`--). Play around with these numbers, run your modified program, observe and adjust until it draws a octagon. When it does, add comments at the top of your program:

```
#Name: ...your name here...
#Email: ...your email here...
#Date: September 3, 2024
#This program draws an octagon.
```



Run your program after editing to make sure you do not have any typos.

8. Log into Gradescope (see notes above). On the left hand menu, choose "Assignments". From the list, choose "2. Octagon". In the file upload, drag the `.py` file you just created and ran, and click "Submit".

More Turtle Commands

To review, we introduced the turtle commands:

- `forward()`

- `left()`
- `right()`

as well as importing the turtle package (`import turtle`) and creating ("instantiating") a turtle: `thomasH = turtle.Turtle()`.

There are many more turtles commands. Over the next couple of classes, we will use those in the [turtle chapter](#) from the textbook. In addition to the ones that control movement, you can also change the color of the turtle, change the size of the drawing pen, and change the background color (by making a window or screen object, and changing its color):

A [complete list of turtle commands](#) is part of the Python 3 documentation.

What's Next?

You can start working on this week's programming assignments. The [Programming Problem List](#) has problem descriptions, suggested reading, and due dates next to each problem.