National University of Singapore
School of Computing
CS1010X: Programming Methodology
Semester II, 2024/2025

**Mission 0
Setting Up Python**

Release date: 15 December 2024
Due: 16 January 2025, 23:59

Required Files

- mission00-template.py

The objective of this mission is to guide you in the installation and setting up of the programming tool, called IDLE, that you will be using for the rest of the semester. Also, we include a simple exercise to help you familiarize yourself with the basics of Python, and also learn how to submit homework through Coursemology system.

This mission consists of only **one** task.

Part 1: Installing Python 3.12.x and required packages

Before you start on your quest to master the Python programming language, you have to install the necessary tools. Please follow the following instructions to set up your programming environment for the class.

PIL is required to render images that will be used in the later missions. We will be using PILLOW, which is a modern replacement for PIL. Scipy/Numpy packages are also required in the later missions.

Note: The highest priority package to install is PILLOW, as we will be using it over the next few weeks.

Please follow the following instructions carefully. Note that if you have previously installed some older version of Python, you may want to uninstall it first before you install our stated version. If you have a newer version, we will not be sure it will work with all our exercises as newer versions Python may deprecate some libraries that we rely upon in our exercises. So, on the safe side, it is good to stick to our stated version.

Troubleshooting

You can contact the teaching staff or check the Coursemology forum if you face any difficulty during the setup/installation process.

Please provide the following information to help us quickly identify your problem:

1. The operating system and version that you are using. MacOS, Windows 10, 32-bit or 64-bit, etc.

2. Which step in the instructions given below did it fail or produce an error?

3. Take a screen shot of the failure or error.

Download Python

You may download Python 3.12.x or newer version from `https://www.python.org/downloads`. Download the appropriate installer for the version of your Operating System.

Note that coursemology autograder is currently up to Python 3.12.x and there is no guarantee that newer version codes can always work under autograding. However, this concern is not of any issue as we should not be using any sophisticated syntax (that differs among different Python versions) in our learning here. Still, on the safe side, if you have not yet installed your Python, you should just download Python 3.12.x .

Windows Users:

Step 1: Run the Installer

Download the **executable installer** for easier installation. Upon running it, you should see something similar to this:
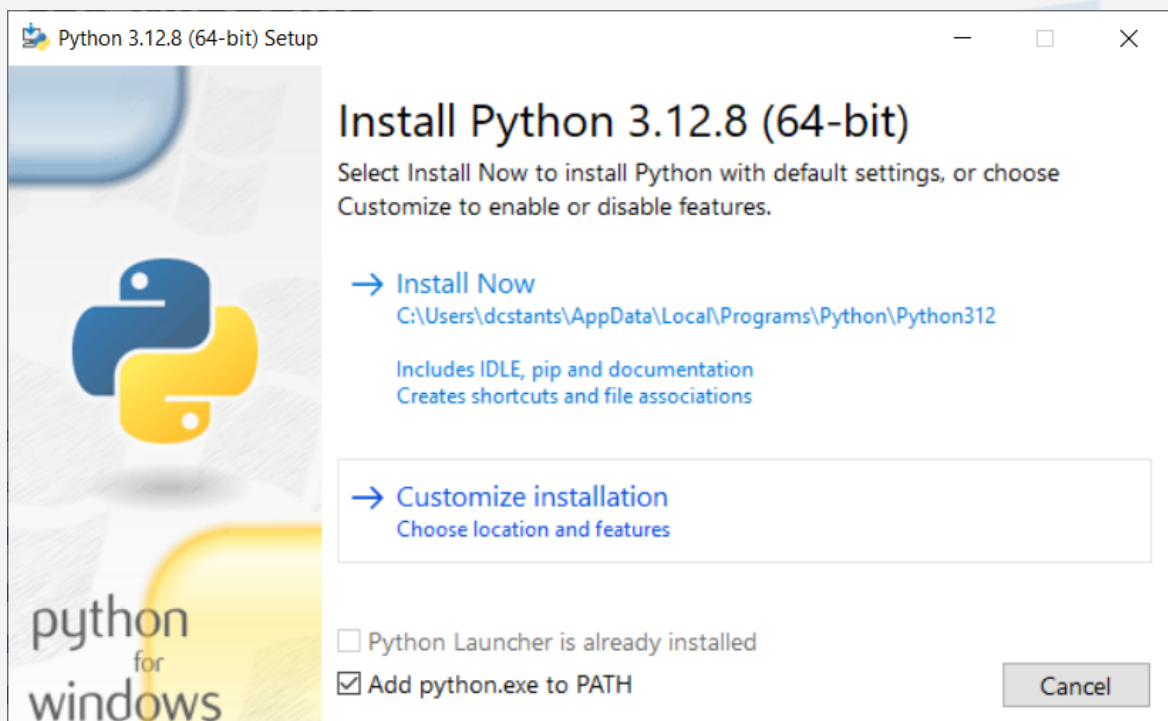
> **Important:** You want to be sure to check the box that says **Add Python 3.x to PATH** as shown to ensure that the interpreter will be placed in your execution path.
>
> You do not need to install for all users if you have no admin access.

Click **Install Now** and it will complete in a few minutes. You should see a new program **IDLE** in your programs menu.

Step 2: Installing the packages

Open the **Command Prompt** from your Start menu.

Enter the following command as one line and press enter:

`pip3 install Pillow numpy scipy matplotlib olefile cycler pytz python-dateutil pyparsing`

If successful, you should see a line saying "Successfully installed..." like so:

Otherwise, if you see a bunch of red text, the packages might not be installed properly. Take a screenshot and check the Coursemology forums for assistance.

In the event that you did not successfully install anyone of the mentioned package, you can first uninstall it and then try installing it again as follows (e.g. with matplotlib):

```
pip3 uninstall matplotlib
```

```
pip3 install matplotlib
```

Mac Users:

Step 1: You may download and run the pkg installer. Once installation is completed, you should see that **IDLE** is available from your finder. It is assumed that Tcl/Tk were installed in your machine. Otherwise, you may need to install Tcl/Tk to run IDLE.

Step 2: Setting up Command Line Tools for Mac

Install the Command Line Tools (required for PILLOW installation). Follow the instructions depending on your OS X version. You can determine the version by clicking on the Apple Icon (in the menu bar) > About This Mac.

Run the following commands in your terminal. (You can find the terminal by clicking on Finder on the dock, Go > Utilities > Terminal.)

```
xcode-select --install
```

A popup will appear, asking if you wish to install the command line developer tools. Click on Install to begin the installation.

Step 3: Setting up Homebrew and dependencies

After installing the Command Line tools, Run the following commands in your terminal. (You can find the terminal by clicking on Finder on the dock, Go > Utilities > Terminal.) You will **need to be connected to the Internet** as the installer will download the required files.

First, we will install Homebrew, a package manager for MacOS.

Visit `https://brew.sh/` and follow the instruction on that page. Basically you should copy and paste the given command in your Terminal prompt.

After which, enter the next two commands into the Terminal as separate lines, i.e., press Enter after entering each line.

```
brew install freetype
brew install pkg-config
```

Step 4: Setting up PILLOW, Scipy/Numpy, Matplotlib

In the same terminal, run the following commands. This will install pip — a package manager for Python, with the following command.

```
curl https://bootstrap.pypa.io/get-pip.py | sudo python3
```

Follow up with the next 3 commands:

```
sudo pip3.12 install PILLOW
sudo pip3.12 install scipy numpy
sudo pip3.12 install matplotlib
```

Note: You may be prompted to enter your Mac password when you run the above commands. **It is normal for nothing (not even *) to appear in the Terminal as you key in your password. Just type it in and press Enter.**

Editing Python Files

The default behaviour of double clicking on the Python file **executes** the content of the Python file. You should see the a command line window briefly opens, and close when Python has finish executing the file.

In order to make changes to the Python file, you will need to **edit** the file using the IDLE program.

- Windows Users: Right click on the Python file > **Edit with IDLE**

- Mac Users: Right click on the Python file > **Open With** > **IDLE**

The content of the Python file should now appear in the IDLE program. You can then make changes to the file, and execute it.

To execute the Python file, go to **Run** > **Run Module**. The output of your Python file should then appear in the Shell Window.

Part 2: The Task

Consider the following Python expressions. Your job is to predict the output when each expression is evaluated in IDLE.

Before checking your answers with IDLE, write down briefly your guess of what the interpreter would **display** when the expressions are evaluated **sequentially** as listed. If you do not expect any output, you may write "no output" and if you expect an error, you may write "error".

Now, run the code by removing the # in the front of the respective lines in the template file. You should comment out error-causing definitions by adding # to the front of such definitions to allow IDLE to skip processing them. (Or by choosing the area that you would like to comment out and then pressing IDLE's hot-key alt+3 for windows, control+3 for macOS)

After you have checked your answers, if any expression has evaluated differently from your expected answer, write down what it evaluated to **below** your expected answer. **Please note that you will be graded only on your <u>final</u> answer (If you have no corrections to make, your initial answer will be your final answer).**

However, if any expression would generate an error message after running, please **specify the type of error** (such as TypeError) together with the **error message** in your final answer. You do not need to include the full error output. The required answer is usually only the *last line* of the error output.

Please use the template file provided in **mission00-template.py** instead of copying from this PDF file.

Reminder: Lines that begin with a # are comments (text that do not affect the Python execution). To execute a particular expression, ensure it does not begin with a #.

```python
print(42)
# expected answer:
# final  answer:

print(0000)
# expected answer:
# final  answer:

print("the force!")
# expected answer:
# final  answer:

print("Hello World")
# expected answer:
# final  answer:

print "Hello World"
# expected answer:
# final  answer:

print(6 * 9)
# expected answer:
```

```python
# final answer:

print(2 + 3)
# expected answer:
# final answer:

print(2 ** 4)
# expected answer:
# final answer:

print(2.1**2.0)
# expected answer:
# final answer:

print(15 > 9.7)
# expected answer:
# final answer:

print((5 + 3) ** (5 - 3))
# expected answer:
# final answer:

print(--4)
# expected answer:
# final answer:

print(1 / 2)
# expected answer:
# final answer:

print(1 / 3)
# expected answer:
# final answer:

print(1 / 0)
# expected answer:
# final answer:

print(7 / 3 == 7 / 3.0)
# expected answer:
# final answer:

print(3 * 6 == 6.0 * 3.0)
# expected answer:
# final answer:

print(11 % 3)
# expected answer:
# final answer:
```

```python
print(2 > 5 or (1 < 2 and 9 >= 11))
# expected answer:
# final answer:

print(3 > 4 or (2 < 3 and 9 > 10))
# expected answer:
# final answer:

print("2" + "3")
# expected answer:
# final answer:

print("2" + "3" == "5")
# expected answer:
# final answer:

print("2" <= "5")
# expected answer:
# final answer:

print("2 + 3")
# expected answer:
# final answer:

print("May the force" + " be " + "with you")
# expected answer:
# final answer:

print("force"*3)
# expected answer:
# final answer:

print('daw' in 'padawan')
# expected answer:
# final answer:

a, b = 3, 4

print(a)
# expected answer:
# final answer:

print(b)
# expected answer:
# final answer:

a, b = b, a

print(a)
# expected answer:
```

```python
# final answer:

print(b)
# expected answer:
# final answer:

print(red == 44)
# expected answer:
# final answer:

red, green = 44, 43

print(red == 44)
# expected answer:
# final answer:

print(red = 44)
# expected answer:
# final answer:

print("red is 1") if red == 1 else print("red is not 1")
# expected answer:
# final answer:

print(red - green)
# expected answer:
# final answer:

purple = red + green

print("purple")
# expected answer:
# final answer:

print(red + green != purple + purple / purple - red % green)
# expected answer:
# final answer:

print(green > red)
# expected answer:
# final answer:

print("green bigger") if green > red else print("red equal or bigger")
# expected answer:
# final answer:

print(green + 5)
# expected answer:
# final answer:
```

```python
print(round(1.8))
# expected answer:
# final answer:

print(int(1.8))
# expected answer:
# final answer:

# The following questions are to ensure that you have installed
# PILLOW, matplotlib, scipy, and numpy correctly.
# Do not worry about the syntax - just run the line and observe the output
# (if any)

from PIL import *
# expected answer:
# final answer:

# Please check with your tutor if you still encounter errors for the
# following statements. This is only required for Side Quest 8.1

from matplotlib import *
# expected answer:
# final answer:

from scipy import *
# expected answer:
# final answer:

from numpy import *
# expected answer:
# final answer:
```

To submit your work to Coursemology, complete **mission00-template.py** and copy the contents from the file into the box that says "Your answer" on the mission page, and click "Save." You can continue to make changes to your submission if you wish.

Once you are satisfied with your solution, click "Finalize Submission." **Note that once your submission is finalized, it is considered to be submitted and cannot be changed.** If you need to undo this action, you will have to email your tutor or the lecturer. You will not be allowed to undo your *finalized* submission, or a penalty might be imposed. Please do not finalize your submission until you are sure that you want to submit your solutions for grading.