Quiz4 & Activity: Learning about Libraries

15-110 – Wednesday 03/30

Quiz4

Announcements

- Hw5 due **Monday**

Learning Goals

- Install **external modules** with the **pip** command

- Read **documentation** to learn how to use a new module

Today's Goal: How to edit images in Python?

We've now covered the core elements of programming. But that won't get us 100% of the way towards solving every problem we encounter.

Coding is usually easier if you build off of what has been done before instead of writing everything from scratch. For example, if we want to write a simple program to modify a set of images, we *could* write a function from scratch to interpret image files and modify their data systematically. But it's a lot easier to start from a **library** that already has image manipulation functions implemented!

That's our goal for today: learn how to install and use a library that will let us manipulate images.

Python Modules

Python Modules

The Python programming language comes with a large set of built-in functions that cover a range of different purposes. However, it would take too long to load all these functions every time we want to run a program.

Python organizes its different functions into **modules**. When you run Python, it loads only a small set of functions from the built-in module. To use any other functions, you must **import** them.

Built-in Modules

We've already used a few core modules for homework assignments - mainly `math` and `tkinter`.

For a full list of python libraries, look here:
https://docs.python.org/3/library/index.html

External Modules

There are many other libraries that have been built by developers outside of the core Python team to add additional functionality to the language. These modules don't come as part of the Python language, but can be added in. We call these **external modules**.

In order to use an external module, you must first **install** it on your machine. To install, you'll need to download the files from the internet to your computer, then integrate them with the main Python library so that the language knows where the module is located.

Finding Useful Modules

One of the main strengths of Python as a language is that there are thousands of external modules available, which means that you can start many projects based on work others have done instead of starting from scratch.

You can find a list of popular modules here: wiki.python.org/moin/UsefulModules

And a more complete list of pip-installable modules here: pypi.org

`pip`

It is usually possible to install modules manually, but this process can be a major pain. Luckily, Python also gives us a streamlined approach for installing modules – the **`pip` module**! This feature can locate modules that are indexed in the Python Package Index (a list of commonly-used modules), download them, and attempt to install them.

Traditionally, programmers run `pip` from the **terminal**. This is a command interface that lets you make changes directly to your computer. But in this class, we'll just run `pip` in Pyzo.

Running `pip`

To run `pip` in Pyzo, use this command in the interpreter

```
pip install module-name
```

This will identify the module and your version of Python and start the download and installation process. It may run into a **dependency error** if the module needs a second module to already be installed – in general, installing that module and then running `pip` again will fix the problem.

Note: you will not be able to run `pip` on CMU cluster machines, as these have restricted permissions. You may need to log into your main account on personal machines to run it.

Install an Image Library

Recall our primary goal today – to install a library that lets us manipulate images.

Pillow is a lightweight and easy-to-install module that lets you manipulate images beyond .gif files. It lets you modify images, or use different types of images in Tkinter.

You do: Try installing it now!

```
pip install pillow
```

Using an Installed Module

Once you've successfully installed a module, you should be able to put

```
import module-name
```

at the top of a Python file, and it will load the module the same way it would load a built-in library. For the Pillow library, the import name is slightly different from the install name:

```
import PIL
```

Note: this may fail if you have multiple versions of Python installed on your machine and you install in the terminal. Make sure to use the `pip` associated with the version of Python you're using in your editor. You can check your editor's version in Pyzo with Shell > Edit Shell Configurations (check the value in exe), then call `pip` using

```
pythonversion-number -m pip install module-name
```

Learning how a Module Works

Once a new module is installed, you're still left with one major question: how do you use it?

This varies by module, but the best answer is to **read the documentation**. Most external modules have official documentation or APIs that describe which functions exist and how to use the module.

It can also be helpful to search online for other projects that have used the same module, to find examples of how to set it up. Many people have written helpful tutorials online for this exact purpose.

Two standard resources for finding help are **StackOverflow**, a site where people can ask questions about code and get answers from other developers, and **GitHub**, a site where people post open-source projects for others to use and contribute to.

Reminder: always cite others' work!

You'll sometimes find a useful bit of code in a StackOverflow post or a GitHub project that you'll want to use in your own project.

Whenever you copy code from online, make sure to **cite it** the same way you would cite a paragraph of text in an essay. You can do this by putting a comment above the copied code that includes a link to the URL you got the code from.

This serves two purposes. First- it gives credit to the individual who originally wrote the code. Second- if you run into a problem with the code later on, you'll be able to look back to the original source to find a solution.

Note: policies around copying code change when you're working on a commercial product. Read the fine print if you're planning to sell your code!

Learning the Pillow Library

To learn how to use the Pillow library, let's start from the documentation!

https://pillow.readthedocs.io/en/stable/index.html

It often helps to start with a tutorial or 'start here' page. Pillow's docs have one:

https://pillow.readthedocs.io/en/stable/handbook/tutorial.html

Pillow Images

Pillow makes it very easy to open image files, using the `Image.open` function. You can even display those images with `image.show`, or save them with `image.save`.

```
from PIL import Image
img = Image.open("stella.jpg")
img.show()
img.save("new-stella.jpg")
```

Pillow Image Functions

You can provide Pillow images to the Tkinter `create_image` function, but you can also manipulate them directly!

There are functions that let you crop and resize pictures within the program, and much more! However, some of these functions require that you use lists to hold the dimensions of the picture.

```
newImg = img.crop([200, 200, 3500, 2500])
newImg.save("new-stella.jpg")

newImg2 = img.rotate(180)
newImg2.save("new-stella-2.jpg")

newImg3 = img.resize([1000, 1000])
newImg3.save("new-stella-3.jpg")
```

Lots of Libraries

There are thousands of useful libraries out there that you can install and use!

As a starting place, check out the bonus slides on the course website for some popular libraries that CMU students tend to like to use.

Learning Goals

- Install **external modules** with the **pip** command
- Read **documentation** to learn how to use a new module

Feedback: https://bit.ly/110-s22-feedback