UMBC

CMSC 201 Section 40
Spring 2020
Lab 10 - The Great Python Beyond: PyPi, pip and virtual environments

Value: 10 points no prelab quiz

This week’s lab will expose you to advanced features of Python.

(Having concepts explained in a new and different way can often lead to a better understanding, so make sure to pay attention as your TA explains.)

Nothing in this lab can be used on other CMSC 201 assignments.

Part 1: Introduction to Libraries

In this lab, you will be learning how to import libraries written by other developers into your workspaces on the GL server.

A *library* is a project that is written for the use of other programmers.[1] Typically a (programming) library supplies functions (and objects, but you'll learn more about that in 202) that extend the capabilities of a particular programming language to accomplish a wider range of tasks. Libraries can be used for networking, web development, robotics, artificial intelligence -- pretty much everything under the sun!

PyPi (https://pypi.org/), the Python Package Index, is a central location where powerful and useful Python libraries[2] live. When you install Python on a computer, you will be able to use the **`pip`** program. **`pip`**[3] is the command that you can use to download libraries from PyPi and use them in your code.

Libraries installed using **`pip`** go into your Python install. This can be a problem, because sometimes you as a user don't have control of the Python installation. This is the case when you are using the GL system.

To deal with this, Python has *virtual environments*. These are your own installs of Python that are copied from the system install. When you are using a virtual environment, you can install all the libraries that you want without (1) having permission from the system admin and (2) without messing up your own Python install[4].

A word of caution: Virtual environments can be rather large, so be careful how many you set up on your GL account. You may run out of disk space! (You can check your disk space by running fs quota)

[1] It is also a large building or room filled with books.

[2] Not a physical building filled with Python books. Or Pythons. Then that would be a zoo.

[3] Also not to be confused with Phillip Pirip, central protagonist of Charles Dickens's *Great Expectations*, prequel to Dr. Johnson's *Greater Disappointments*

[4] Some operating systems use their Python installation to run critical system processes. Putting a library in there for rendering cats taking naps is probably not ideal.

In this week's lab, you will be installing the great and powerful requests library: https://requests.readthedocs.io/en/master/ This library allows you to get data off of the internet as a string in your Python program. Some people find that the Internet has between three to five useful things on it.

Part 2: Exercise

In this lab, you'll be setting up a virtual environment on your GL account, installing requests into it, and then writing a program to download the UMBC homepage -- the most exciting place on the world wide webs.

Tasks

Starting:

□ Follow the ***INSTRUCTIONS CLOSELY*** to create a virtual environment
□ CONTINUE YOUR ***ULTRA FASTIDIOUS*** READING and install the requests library

Programming:

☐ Import the requests library
☐ Go to the requests library website and figure out how to download the UMBC homepage into a string.
□ Print the UMBC website's HTML
□ Compute how many links are on the UMBC homepage

General:

☐ Run and test your code as needed
☐ Submit your lab using the command submit cmsc201 LAB10 <filename>

Part 3A: Setting up a virtual environment[5]

First, create the `lab11` folder using the `mkdir` command – the folder needs to be inside your `Labs` folder as well.

#0 Run this command: `virtualenv venv`

[5] It's like the holodeck! But it's not and it's also kinda lame and annoying.

You will see the following:

```
Using base prefix '/usr'
New python executable in
/afs/umbc.edu/users/b/e/benj1/home/src/lab11/venv/bin/pyth
on3
Also creating executable in
/afs/umbc.edu/users/b/e/benj1/home/src/lab11/venv/bin/pyth
on
Installing setuptools, pip, wheel...done.
```

This will create a folder called venv in your lab11 folder. This is where the virtual environment will live! Cozy!

NOW WE ENTER THE ENVIRONMENT![6]

#1 First, run the command: **`bash`**

This command is NOT necessary outside of GL. GL is just weird and we have to live with it. Your prompt should now read: **`bash-5.0$`**

#2 Entering the environment is done with: **`source venv/bin/activate`**

If your prompt is now: **`(venv) bash-5.0$`** You have successfully entered the virtual environment. No, you cannot now dodge bullets.

#3 Now let's install the requests library using this command: **`pip install requests`**

You should see a bunch of text fly by, ending with this:

```
Successfully installed certifi-2019.11.28 chardet-3.0.4
idna-2.8 requests-2.22.0 urllib3-1.25.7
```

[6] There will be no fun techno music or bulky goggles. Yeah... It's just more unix commands : (

If you have already set up the virtual environment in the past and installed everything you need, you will only need to do step (1)[7] and step (2).

Now witness the firepower of this fully armed and operational battle station...[8]

Part 3B: Welcome to the Internets

Create a file called **`get_page.py`** At the top of the file, after the header comment, write the line **`import requests.`** Then, do the usual main program stuff: **`if___name__== "__main__":`**

Now you're going to write a ton of sophistical TCP/IP and HTTP request logic to contact a webserver and request a network transmission of HTML without headers and --

Just kidding.

Write these lines:

```
grit_page = requests.get("http://www.umbc.edu").text
print(grit_page)
```

Run your file. You should see a huge explosion of text. You did it! You tamed the Internet![9]

Lastly, you will compute how many links are present on the webpage. Here are some hints:

- You can reliably detect a link by finding the substring "**`</a>`**" in the HTML
- .**`split()`** is your friend.

[7] Again, step 1 is only necessary on GL. If you do this on another system it won't be needed.

[8] Yeah, it's a Star Wars quote. It's the last lab! If you thought I wasn't going to have fun with it, you were sorely mistaken.

[9] You did not tame the Internet. Actually, it's often a pretty horrible place.

Part 4: Completing Your Lab

Once your lab is complete, you can submit it using submit cmsc201 LAB10 <filename>
You have until Thursday night, April 30 at 11:59:59 pm. In other words, before May Day.

Tasks

Starting:

□ Follow the ***NSTRUCTIONS CLOSELY*** to create a virtual environment
□ CONTINUE YOUR ***ULTRA FASTIDIOUS*** READING and install the requests library

Programming:

☐ Import the requests library
☐ Go to the requests library website and figure out how to download the UMBC homepage into a string.
□ Print the UMBC website's HTML
□ Compute how many links are on the UMBC homepage.

General:

☐ Run and test your code as needed
☐ Submit the lab on GL