Setting Up a Custom Environment on Google Drive for Colab

Introduction

Google Colab instances come with only a minimal set of pre-installed packages. Even in simple projects (such as the ones of the university courses you will be attending) there will be instances where a specific Python package you wish to use is not pre-installed.

For example, let's say that you **need** to use a koala emoji in your notebook. You can do so by importing the library `emoji` and run the command `print(emoji.emojize(':koala:'))`. If you try to run this code, you will get an error `ModuleNotFoundError: No module named 'emoji'` because Python can't find the library.

```
In [ ]:    import emoji  # Not available on default Colab environment
           print(emoji.emojize(':koala:'))
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-2-0d329a46eb67> in <cell line: 1>()
----> 1 import emoji  # Not available on default Colab environment
      2 print(emoji.emojize(':koala:'))

ModuleNotFoundError: No module named 'emoji'

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.
---------------------------------------------------------------------------
```

You can install this library on the Colab instance using the command

```
In [ ]:    !pip install emoji
           import emoji
           print(emoji.emojize(':koala:'))
```

```
Collecting emoji
  Downloading emoji-2.8.0-py2.py3-none-any.whl (358 kB)
                                                    358.9/358.9 kB 5.5 MB/s eta 0:0
0:00
Installing collected packages: emoji
Successfully installed emoji-2.8.0
🐨
```

However, instances are temporary, get created on the fly when you open the notebook and destroyed when you close it or after some inactivity. So, next time you will open this notebook on another Colab instance (or if you get disconnected) you will need to install again the library.

Try this yourself. Navigate to the upper-right corner of your Colab window, where RAM and Disk usage metrics are displayed, click on the downward-facing arrow and select "Change runtime." You may choose any available option. This action will switch the computational engine (also known as the kernel) powering this notebook. In other words, you are changing the instance (the computer in the cloud) that is doing the computation "under the hood" for your notebook. Given that the new kernel starts with a default configuration that lacks the libraries you previously installed, attempts to display the koala emoji in this new instance will result in an error:

```
In [ ]:  import emoji
         print(emoji.emojize(':koala:'))
```

🐨

The quick and dirty solution is to reinstall the library at every use. However, usually we use multiple packages in a single notebook. Doing the installation of each of them every time you open a new instance is a huge waste of time that we can avoid using different strategies.

In this notebook we will see how to set up a Colab environment that stores **persistently** the libraries you need in your GDrive.

Step 1: Mount Google Drive

First, mount your Google Drive so that Google Colab can access it. A permission dialog will appear, asking you to log in into your google account (use your unipd credentials) to grant Colab access to read and write files on your Drive. Keep in mind that granting this permission gives Colab a lot of power over the content of your GDrive and that may result in security issues. Once you have granted access to Colab, it can read and write on your Gdrive, and a malicious script could wipe out everything It is important that you only run notebooks from sources you trust and that you always understand what the code is doing!. Given the risks involved, each time you mount your Gdrive you will be required to manually grant the permissions.

```
In [ ]:  from google.colab import drive
         drive.mount("/content/drive")
```

```
---------------------------------------------------------------------------
MessageError                              Traceback (most recent call last)
<ipython-input-5-bf1647b42bbd> in <cell line: 2>()
      1 from google.colab import drive
----> 2 drive.mount("/content/drive")

/usr/local/lib/python3.10/dist-packages/google/colab/drive.py in mount(mount
point, force_remount, timeout_ms, readonly)
    101 def mount(mountpoint, force_remount=False, timeout_ms=120000, readon
ly=False):
    102     """Mount your Google Drive at the specified mountpoint path."""
--> 103     return _mount(
    104         mountpoint,
    105         force_remount=force_remount,

/usr/local/lib/python3.10/dist-packages/google/colab/drive.py in _mount(moun
tpoint, force_remount, timeout_ms, ephemeral, readonly)
    130     )
    131     if ephemeral:
--> 132       _message.blocking_request(
    133           'request_auth', request={'authType': 'dfs_ephemeral'}, timeo
ut_sec=None
    134       )

/usr/local/lib/python3.10/dist-packages/google/colab/_message.py in blocking
_request(request_type, request, timeout_sec, parent)
    174         request_type, request, parent=parent, expect_reply=True
    175     )
--> 176     return read_reply_from_input(request_id, timeout_sec)

/usr/local/lib/python3.10/dist-packages/google/colab/_message.py in read_rep
ly_from_input(message_id, timeout_sec)
    101     ):
    102         if 'error' in reply:
--> 103             raise MessageError(reply['error'])
    104         return reply.get('data', None)
    105

MessageError: Error: credential propagation was unsuccessful
```

Once the drive is mounted, you can see its contents by clicking the folder icon on the left shoulder of this page. So, if you to store the notebooks of the course on Gdrive it is very easy accessing them.

Step 2: Install Virtual Environment Package

Install the `Venv` package to manage isolated Python environments. Remember that an environment specifies the interpreter used and defines its fundamental settings. Among these setting we are particularly interested at the location where the libraries are stored.

```
In [ ]:  !pip install virtualenv
```

Step 3: Create a Virtual Environment

Next, you need to specify the path in your Google Drive where you'd like to create the virtual environment (where to store the interpreter and the other settings). The notebooks of this course will use `"/content/drive/MyDrive/codepy"` as a default one. If you are changing this, you will need to update accordingly the other notebooks (my advice is to just keep this and the following defaults)

```
In [ ]:  venv_path = "/content/drive/MyDrive/codepy"
         !virtualenv $venv_path
```

Step 4: Activate the Virtual Environment

Activate the virtual environment you've just created. By doing so you are telling the notebook that this environment is the one to use.

```
In [ ]:  venvBin_path = "/content/drive/MyDrive/codepy/bin/activate"
         !source $venvBin_path
```

Step 5: Install Desired Packages

Identify the folder in your Google Drive where you'd like to store libraries, and then install the `emoji` package there by using the option `target`.

```
In [ ]:  import sys
         colabLib_path = "/content/drive/MyDrive/codepy/lib/python3.10/site-packages"
         !pip install --target=$colabLib_path emoji
```

Step 6: Update System Path

Add the library path to the system path so that Colab can find the newly installed packages. In this way the current notebook knows where to find the libraries.

```
In [ ]:  import sys
         sys.path.append(colabLib_path)
```

Step 7: Test the Setup

Now you can happily import and use the `emoji` library to print a koala emoji.

```
In [ ]: import emoji
        print(emoji.emojize(':koala:'))
```

Using the Custom Environment in Future Sessions

The procedure we followed stored the he virtual environment and libraries on your Google Drive. When you will start a new instance you will need to make them accessible to Colab by telling it that you whis to use them and where to find them. Specifically, you'll need to:

1. Mount your Google Drive to make it accessible
2. Activate the previously created virtual environment
3. Tell the environment to search for libraries in your Google Drive

---

```
In [1]: import sys
        from google.colab import drive
        drive.mount("/content/drive")
        venvBin_path = "/content/drive/MyDrive/codepy/bin/activate"
        !source $venvBin_path
        colabLib_path = "/content/drive/MyDrive/codepy/lib/python3.10/site-packages"
        sys.path.append(colabLib_path)
```

```
Mounted at /content/drive
```

At the beginning of each notebook using libraries you will find a cell that sets up the Gdrive environment (as the one above). This code will be stored in a markdown cell, because those using the VLEM or with a local installation of anaconda will not need it. But if you chose to use Colab, simply convert it to code and run it. Remember that this will only work if you did not change the default paths during this installation, otherwise you will need to change the paths also in the other notebooks..

Once that is set up, you will be ablo to import the libraries and **happily** print your koala emoji without re-installing the library

```
In [2]: import emoji
        print(emoji.emojize(':koala:'))
```

🐨

Create a custom environment with all the libraries for this course

To be sure to set things properly, start a new instance now. Navigate to the upper-right corner of your Colab window, where RAM and Disk usage metrics are displayed, click on

the downward-facing arrow and select "Change runtime." You may choose any available option.

Now run the code below to will install all the packages needed in the course.

```
In [1]: #mounting Gdrive
        from google.colab import drive
        drive.mount("/content/drive")
        #install virtualenv
        !pip install virtualenv
        #set path in Gdrive where to store the custom environment
        venv_path = "/content/drive/MyDrive/codepy"
        #create custom environment
        !virtualenv $venv_path
        #set virtual enviroment activation folder
        venvBin_path = "/content/drive/MyDrive/codepy/bin/activate"
        #activate environment
        !source $venvBin_path
        #set the path where the environment will store the libraries
        colabLib_path = "/content/drive/MyDrive/codepy/lib/python3.10/site-packages"
        #set the libraries to install
        libraries_to_install = ["arrow"]
        #install all the libraries in the custom path for libraries
        for lib in libraries_to_install:
            !pip install --target=$colabLib_path {lib}
        #add the library path to the ones where the instance will search for librari
        import sys
        sys.path.append(colabLib_path)
        #test that the installed libraries can be imported
        import pandas
        import matplotlib
        import seaborn
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[1], line 2
      1 #mounting Gdrive
----> 2 from google.colab import drive
      3 drive.mount("/content/drive")
      4 #install virtualenv

ModuleNotFoundError: No module named 'google'
```

That's it, done, you have set up your custom environment on Gdrive so that you can use it with Colab.

If you had problems in running the code of this setup in the right order, or you think you did some mistake in the installation, the simplest solution is to go to your drive, delete the custom environment folder (the default name is `"codepy"`), start a brand new instance in a new window and re-do the installation.

Notes

Book references:

Ch.1 of Python for Data Analysis

Preface of Python Data Science Handbook

Office Hours:

See on my website

Author: Duccio Gamannossi degl'Innocenti

Web: https://www.dgdi.me/

Mail: duccio.gamannossi@unipd.it