Colab Helper

May 1, 2020

0.1 1. Mount your drive

[1]:

```
# Mount the drive
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id
=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redire
ct_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&response_type=code&scope=email%20http
s%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.c
om%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.reado
nly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:
ůůůůůůůůůů
Mounted at /content/gdrive
```

[2]:

```
# Your Google Drive is now located at './gdrive/My Drive'
!ls
```

```
gdrive  sample_data
```

[3]:

```
%cd 'gdrive/My Drive'
```

```
/content/gdrive/My Drive
```

0.2 2. Import and connect your Github repository

[14]:

```
your_token = "653003a373b45d7e57bab11a7736128650490b47"
your_repository = "theomeb/gg-colab-helper"
!git clone https://theomeb:"$your_token"@github.com/"$your_repository".git
%cd gg-colab-helper
```

```
Cloning into 'gg-colab-helper'...
remote: Enumerating objects: 6, done.
```

```
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
```

[0]:

```
# Setting your github name (not your password, connexion handled by the token)
!git config user.email "theome.borck@student.ecp.fr"
!git config user.name "Théomé Borck"
```

[32]:

```
# Move your files in your repository (either on Google Drive or through the CLI)
# Commit your files (to avoid loosing them once the colab session is closed)
!git add .
!git commit -m "Add my files"
```

```
[master 7b5612e] Add my files
 1 file changed, 1 insertion(+), 1 deletion(-)
 rewrite Gestion projet (97%)
```

[38]:

```
# Push your commit
!git push
```

```
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 7.92 KiB | 1.98 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/theomeb/gg-colab-helper.git
   5e72191..e61e736  master -> master
```

[31]:

```
/content/gdrive/My Drive/gg-colab-helper
```

0.3 3. Manage the imports in your project

[27]:

```
# Google Colab runs python interactively, hence the empty string for the first path
import sys
print(sys.path)
```

```
['', '/env/python', '/usr/lib/python36.zip', '/usr/lib/python3.6',
'/usr/lib/python3.6/lib-dynload', '/usr/local/lib/python3.6/dist-packages',
'/usr/lib/python3/dist-packages', '/usr/local/lib/python3.6/dist-
packages/IPython/extensions', '/root/.ipython']
```

Let's say you have a python `project` and you want to import your `dumb_function` from `utils.py`. In order to make your script work, either you add the project folder to your sys paths, or you change your working directory: launched interactively, python will understand with the `''` path that it needs to look into your current working directory to find `utils`.

### a. Add your project folder to `sys.path`

[0]:
```
sys.path.append('/content/gdrive/My Drive/gg-colab-helper/project')
```

[41]:
```
# Import your dumb function
from utils import dumb_function
dumb_function()
```

```
Hey, what's up?
```

[42]:
```
sys.path.pop()
print(sys.path)
```

```
['', '/env/python', '/usr/lib/python36.zip', '/usr/lib/python3.6',
'/usr/lib/python3.6/lib-dynload', '/usr/local/lib/python3.6/dist-packages',
'/usr/lib/python3/dist-packages', '/usr/local/lib/python3.6/dist-
packages/IPython/extensions', '/root/.ipython']
```

0.3.1 b. Update the current working directory

[44]:
```
import os
print(os.getcwd())
os.chdir('/content/gdrive/My Drive/gg-colab-helper/project')
```

```
/content/gdrive/My Drive/gg-colab-helper
```

[45]:
```
# Import your dumb function
from utils import dumb_function
dumb_function()
```

```
Hey, what's up?
```

0.4 4. Upload files and zip directly from the url

[0]:
```
# Helper to upload directly file to your drive
import urllib.request
def download_file_from_url(url, file_name):
    rsp = urllib.request.urlopen(url)
    with open(file_name,'wb') as f:
        f.write(rsp.read())
```

[0]:

```
download_file_from_url('https://images.unsplash.com/
 ↪photo-1506973035872-a4ec16b8e8d9?ixlib=rb-1.2.
 ↪1&q=80&fm=jpg&crop=entropy&cs=tinysrgb&dl=dan-freeman-7Zb7kUyQg1E-unsplash.
 ↪jpg&w=640',
                       'sydney.jpg')
```

[0]:
```
# Helper to upload directly zip file to your drive
import requests, zipfile, io

def download_zip_from_url(url, file_location):
    r = requests.get(url)
    zip_ref = zipfile.ZipFile(io.BytesIO(r.content))
    zip_ref.extractall(file_location)
```

[0]:
```
download_zip_from_url('https://drive.google.com/u/0/uc?
 ↪id=19n0D5o330x6TqPWoGUGMVAfTF8bcDgJF&export=download',
                      '/content/gdrive/My Drive/gg-colab-helper/project/
 ↪zip_test')
```

0.5 5. Convert Jupyter Notebook on Drive to a PDF Latex file

[0]:
```
!apt-get -qq install texlive texlive-xetex texlive-latex-extra pandoc
!pip install pypandoc --quiet
```

[68]:
```
# In case your file is somewhere else in your drive, copy it to your actual␣
 ↪folder
!cp '/content/gdrive/My Drive/gg-colab-helper/Colab Helper.ipynb' ./
```

```
cp: '/content/gdrive/My Drive/gg-colab-helper/Colab Helper.ipynb' and './Colab
Helper.ipynb' are the same file
```

[81]:
```
!jupyter nbconvert --to PDF "Colab Helper.ipynb"
```

```
[NbConvertApp] Converting notebook Colab Helper.ipynb to PDF
[NbConvertApp] Writing 50392 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: [u'xelatex', u'./notebook.tex',
'-quiet']
[NbConvertApp] Running bibtex 1 time: [u'bibtex', u'./notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 47343 bytes to Colab Helper.pdf
```

0.6 Bonus : hack scripts

1. To **stay connected to the Colab cloud machine** when you have a long script to run, open the dev tools and launch in the console:

```
function ClickConnect(){
console.log("Working");
document.querySelector("colab-toolbar-button").click()
};
interval = setInterval(ClickConnect,300000);
```

This will create a section every 5 minutes, and prevent disconnection. To stop it:

```
clearInterval(interval)
```

2. To **get more RAM**, simply launch this:

   ```
   a = []
   while(1):
   a.append('1')
   ```

   Once your machine crashes, Google will raise a message asking you if you wanna switch to more RAM, just say yes and you will be rewarded with 25GB of RAM instead of 12GB. **Use it responsibly ;)**