

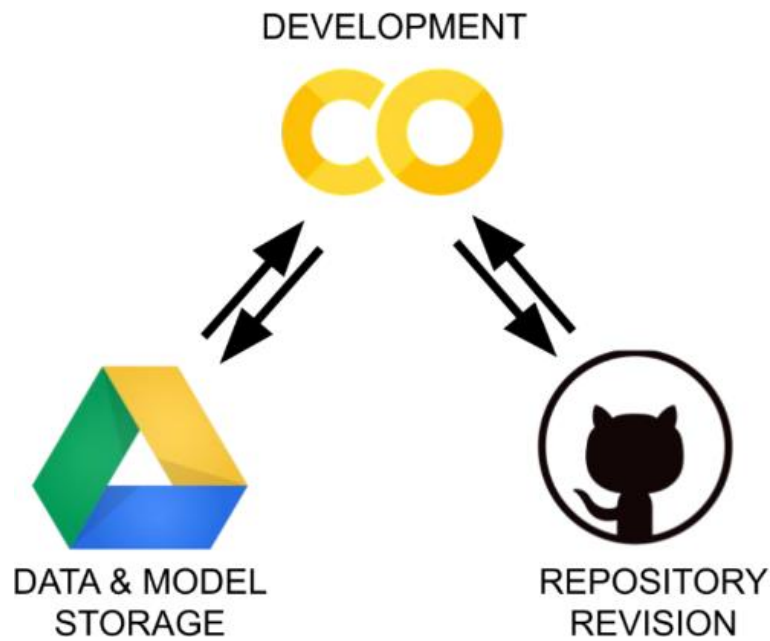
Handout – week I

Getting started with Python, Google Colab, and GitHub

Python is an interpreted, high-level, general-purpose programming language.

Google Colab is a python development environment that runs in the browser using Google cloud based on Jupyter Notebooks. It is a virtual machine providing free GPU (Graphics processing unit) and TPU (Tensor processing unit) and supports various libraries such as PyTorch, TensorFlow, Keras, and OpenCV. Notebooks in Colab can be created, uploaded, stored, shared, and mounted via Google drive, as well as, uploaded directly from GitHub and Kaggle files.

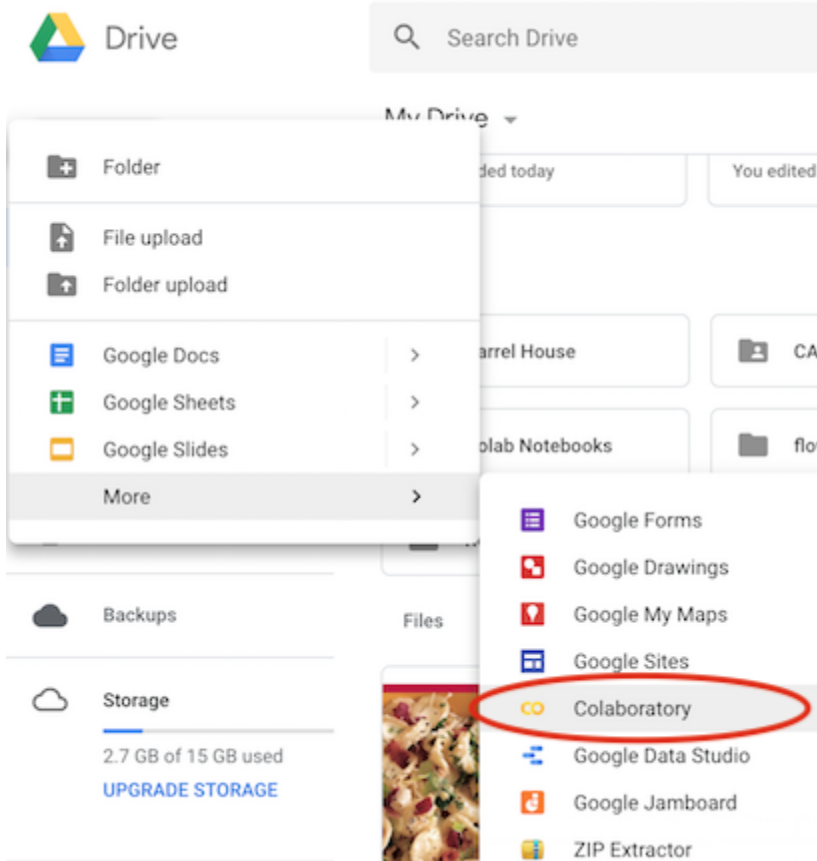
GitHub is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.



<https://medium.com/analytics-vidhya/how-to-use-google-colab-with-github-via-google-drive-68efb23a42d>

Setting up your drive

1. Create a folder for your notebooks
 - Go to Google Drive
 - Create a new Colab notebook by clicking “New” => “More” => “Colaboratory”



- Otherwise, go directly to Google Colab

2. Set up your free GPU

- Go to “runtime” => “change runtime type” => “GPU”

Notebook settings

Runtime type
Python 3 ▼

Hardware accelerator
GPU ▼

☐ Omit code cell output when saving this notebook

CANCEL SAVE

3. Get Coding

- Compile the code by pressing Shift + Enter

Example:

1. Hello world

```
# Never mind this statement, for compatibility reasons
from __future__ import absolute_import, division, print_function, unicode_literals

print("Hello World")
```

2. Function, Conditionals, and Iteration

```
def HelloWorldXY(x, y):
    if (x < 10):
        print("Hello World, x was < 10")
    elif (x < 20):
        print("Hello World, x was >= 10 but < 20")
    else:
        print("Hello World, x was >= 20")
    return x + y

for i in range(8, 25, 5): # i=8, 13, 18, 23 (start, stop, step)
    print("--- Now running with i: {}".format(i))
    r = HelloWorldXY(i,i)
    print("Result from HelloWorld: {}".format(r))
```

```
print(HelloWorldXY(1,2))
```

3. Various kinds of loops starting at 0 to 2

```
print("Iterate over the items. `range(2)` is like a list [0,1].")
for i in range(2):
    print(i)

print("Iterate over an actual list.")
for i in [0,1]:
    print(i)

print("While works")
i = 0
while i < 2:
    print(i)
    i += 1
```

4. Numpy and lists

Numpy as np is the fundamental package for scientific computing with Python.

Lists (Array) are data structure in Python that is a mutable, or changeable, ordered sequence of elements. Each element or value that is inside of a list is called an item. Just as strings are defined as characters between quotes, lists are defined by having values between square brackets [].

```
sea_creatures = ['shark', 'cuttlefish', 'squid', 'mantis shrimp', 'anemone']
```

```
sea_creatures[0] = 'shark'
sea_creatures[2] = 'squid'
sea_creatures[4] = 'anemone'
```

```
import numpy as np # Make numpy available using np.

# Create a numpy array, and append an element
a = np.array(["Hello", "World"])
a = np.append(a, "!")
print("Current array: {}".format(a))
print("Printing each element")
for i in a:
    print(i)

print("\nPrinting each element and their index")
for i,e in enumerate(a):
    print("Index: {}, was: {}".format(i, e))
```

```
print("\nShowing some basic math on arrays")
b = np.array([0,1,4,3,2])
print("Max: {}".format(np.max(b)))
print("Average: {}".format(np.average(b)))
print("Max index: {}".format(np.argmax(b)))
```

```
print("\nYou can print the type of anything")
print("Type of b: {}, type of b[0]: {}".format(type(b), type(b[0])))
```

```
print("\nUse numpy to create a [3,3] dimension array with random number")
c = np.random.rand(3, 3)
print(c)
```

```
print("\nShowing some basic math on arrays")
b = np.array([0,1,4,3,2])
print("Max: {}".format(np.max(b)))
print("Average: {}".format(np.average(b)))
print("Max index: {}".format(np.argmax(b)))
```

```
print("\nYou can print the dimensions of arrays")
print("Shape of a: {}".format(a.shape))
print("Shape of b: {}".format(b.shape))
print("Shape of c: {}".format(c.shape))
```

5. Colab Specifics: to run commands at the VM's terminal, prefix the line with an exclamation point (!)

```
print("\nDoing $ls on filesystem")
!ls -l
!pwd
```

```
print("Install numpy")    # Just for test, numpy is actually preinstalled in  
                           all Colab instances  
  
!pip install numpy
```

Exercise

1. **Colab:** Create a code cell underneath this text cell and add code to:

- List the path of the current directory (`pwd`)
- Go to `/` (`cd`) and list the content (`ls-l`)

2. **Python lists without Numpy:** Print the second item in the fruits list:

```
fruits = ["apple", "banana", "cherry"]
```

3. Change the value from “apple” to “kiwi”, in the fruits list:

4. Use the append method to add “orange” to the fruits list:

5. Use the insert method to add “lemon” as the second item in the fruits list:

6. Use the remove method to delete “banana” from the fruits list:

7. **Python If...Else:** Print “Hello World” if a is greater than b:

```
a = 50    c = 20
b = 10    d = 20
```

8. Print “Hello World” if a is not equal to b:

9. Print “Yes” if a is equal to b, otherwise print “No”:

10. Print “1” if a is equal to b, print “2” if a is greater than b, otherwise print “3”:

11. Print “Hello” if a is equal to b, and c is equal to d:

12. Print “Hello” if either a is equal to b, or c is equal to d:

13. **Python Loops:** Print i as long as i is less than 6:

14. From #13 stop the loop if i is 3:

Reference:

1. <https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c>
2. <https://classroom.udacity.com/courses/ud187>
3. <https://www.w3schools.com/python/>