

DR. ALVIN'S PUBLICATIONS

# KICKSTARTING PYTHON

---

DR. ALVIN ANG



# CONTENTS

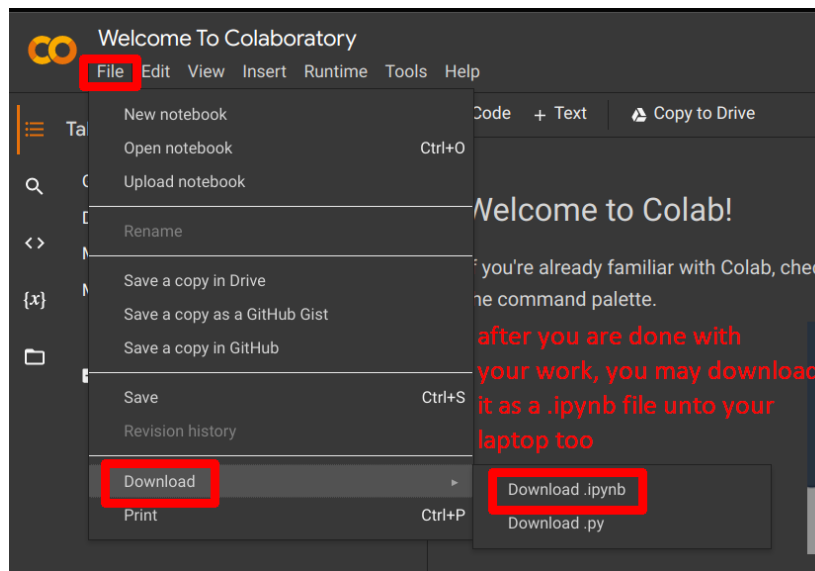
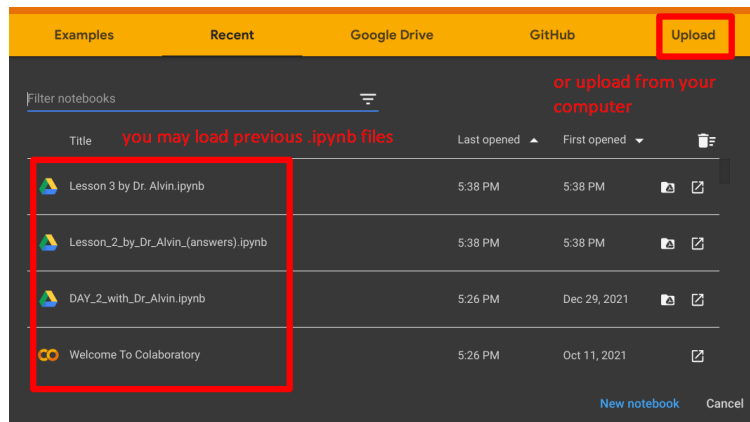
<b>I. Google Colab.....</b>	<b>3</b>
<b>A. How to Upload IPYNB .....</b>	<b>3</b>
<b>B. Dragging in CSVs to Colab.....</b>	<b>4</b>
<b>C. Importing CSV.....</b>	<b>5</b>
1. Using Colab Import Files .....	5
2. CSV file is on desktop .....	6
a) No Header.....	6
b) Read in Everything.....	6
c) Choosing the Columns .....	6
<b>D. Importing XLS .....</b>	<b>7</b>
1. XLS is on Desktop .....	7
a) No Header.....	7
b) if you want to select columns.....	7
2. CSV file is on the Internet.....	8
a) **note about Cloud Service (to store your CSV files).....	8
<b>E. Exporting CSV .....</b>	<b>9</b>
<b>F. Mounting Google Drive .....</b>	<b>10</b>
<b>G. Writing to a .TXT File.....</b>	<b>12</b>
<b>H. Computing BMI.....</b>	<b>13</b>
<b>I. Special Tricks in Colab!.....</b>	<b>14</b>
1. help().....	14
2. ?.....	14
3. Seeing Functions Pop Up After the .....	15
4. Leave the Cursor Inside the () .....	16
<b>II. IBM Skills Network Labs .....</b>	<b>17</b>
<b>III. Anaconda / Jupyter Notebook.....</b>	<b>19</b>
<b>A. Installing Jupyter Notebook On Linux Mint .....</b>	<b>21</b>
1. Install python 3 first .....	21
a) Open terminal:.....	21
b) If not upgraded to the latest python 3: .....	21
c) Install pip3: .....	21
2. Install Jupyter Notebook.....	21
a) To activate Jupyter Notebook: .....	21
b) If fail because file is in /.local.....	21
<b>About Dr. Alvin Ang .....</b>	<b>22</b>

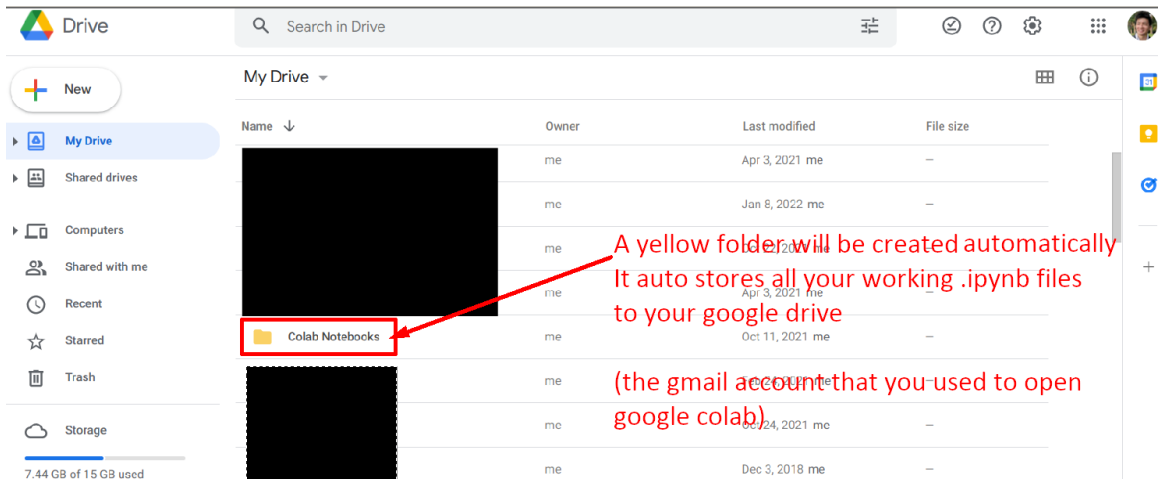
---

## I. GOOGLE COLAB

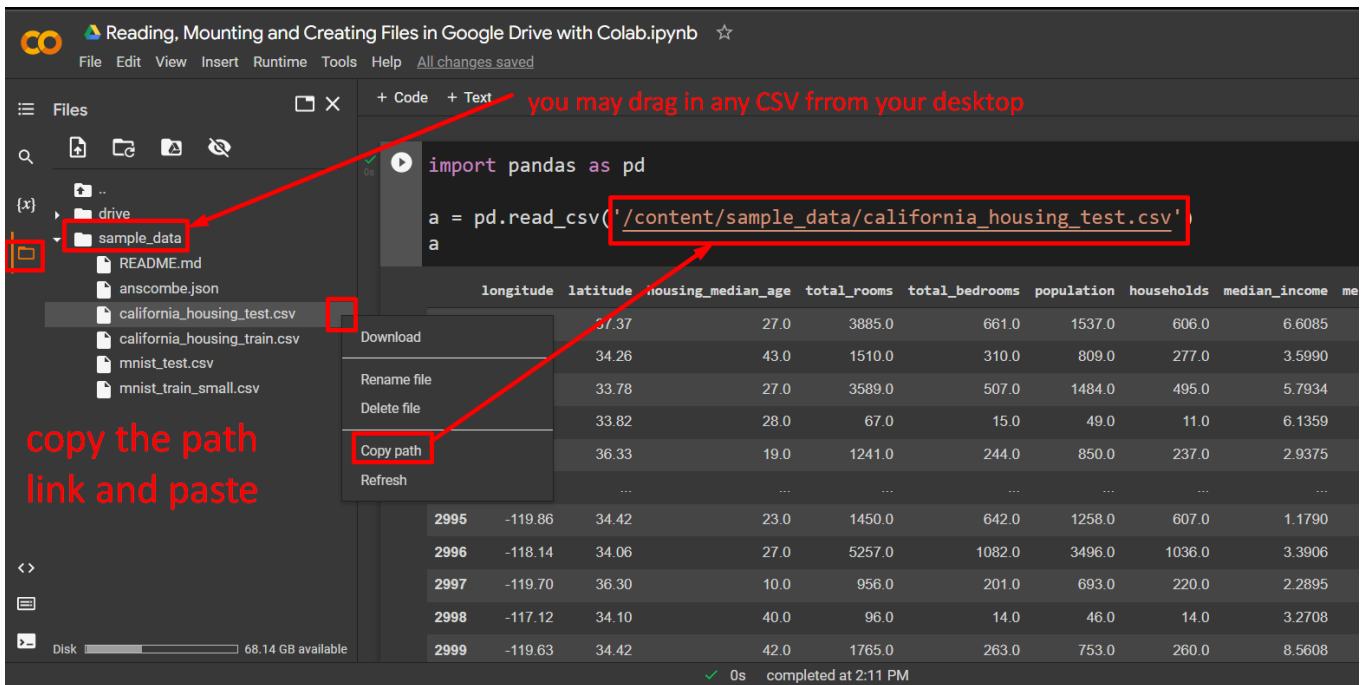
---

### A. HOW TO UPLOAD IPYNB





### B. DRAGGING IN CSVS TO COLAB



## C. IMPORTING CSV

### 1. USING COLAB IMPORT FILES

The screenshot shows the Google Colab interface with a code editor and a file browser. The file browser on the left shows a folder named 'sample\_data' containing a file named 'weather.csv'. A red box highlights this file, and a red arrow points to it with the text 'it will be uploaded here'. The code editor contains the following code:

```
[1] 1 from google.colab import files
    2 uploaded = files.upload()

[2] 1 uploaded

{'weather.csv':
b'Ozone,Solar.R,Wind,Temp,Month,Day\r\n41,190,7.4,67,5,1\r\n36,118,8,72,5,2\r\n12,149,12.6,74,5,3\r\n18,313,11.5,62,5,4\r\n,14.3,56,5,5\r\n28,,14.9,66,5,6\r\n23,299,3
```

A red box highlights the output of the first cell, which is a dictionary containing the uploaded file. A red arrow points to this output with the text 'it will be stored as a dictionary'. The code editor also shows the output of the second cell, which is the name of the uploaded file: 'sample\_data weather.csv'. The code editor also shows the output of the third cell, which is the first five rows of the CSV file:

```
[5] 1 import pandas as pd
    2
    3 df = pd.read_csv('weather.csv')
    4 df.head()
```

The output of the third cell is a DataFrame with the following columns: Ozone, Solar.R, Wind, Temp, Month, Day. The status bar at the bottom indicates that the code is completed at 12:04 AM.

## 2. CSV FILE IS ON DESKTOP

### a) *No Header*

```
import pandas as pd

other_path = r'C:\Users\Ron\Desktop\Clients.csv'
df = pd.read_csv(other_path, header=None)
```

### b) *Read in Everything*

```
import pandas as pd

df = pd.read_csv(r'C:\Users\Ron\Desktop\Clients.csv')
print(df)
```

### c) *Choosing the Columns*

```
import pandas as pd

data = pd.read_csv(r'C:\Users\Ron\Desktop\Clients.csv')
df = pd.DataFrame(data, columns= ['Client Name', 'Country'])
print(df)
```

## D. IMPORTING XLS

### 1. XLS IS ON DESKTOP

#### a) No Header

```
!pip install xlrd
!pip install openpyxl

import pandas as pd

other_path = r'/home/dralvin/Desktop/Book1.xlsx'
df = pd.read_excel(other_path, header=None)
```

#### b) if you want to select columns

```
import pandas as pd

data = pd.read_excel(r'C:\Users\Desktop\Product List.xlsx')
df = pd.DataFrame(data, columns= ['Product'])
print (df)
```

## 2. CSV FILE IS ON THE INTERNET

```
import pandas as pd

other_path =
"https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/DA0101EN/auto.csv"

df = pd.read_csv([other_path, header=None])
```

a) *\*\*note about Cloud Service (to store your CSV files)*

- <https://s3-api.us-geo.objectstorage.softlayer.net/...>
- Such links are free services to host your CSV files.
- You can sign up here at:
  - <https://cloud.ibm.com/catalog/services/cloud-object-storage>
  - Or
  - <https://aws.amazon.com/products/storage/>



## E. EXPORTING CSV

```
#Read in the .csv|
mtcars_sample = pd.read_csv('https://www.alvinang.sg/s/mtcars.csv',
                             index_col='car_names',
                             usecols=['car_names', 'mpg', 'hp', 'cyl', 'am'])
mtcars_sample

#You can Export out to .csv
mtcars_sample.to_csv('cars_sample.csv')

#You can Export out to .xls
mtcars_sample.to_excel('cars_sample.xlsx', sheet_name='cars', index=False)

#You can also Read in the .xls
mtcars_sample2 = pd.read_excel('cars_sample.xlsx', sheet_name='cars')
```

## F. MOUNTING GOOGLE DRIVE

File:

<https://www.alvinang.sg/s/Reading Mounting and Creating Files in Google Drive with Colab.ipynb>

```
Mounting Google Drive

[7] import os

[8] #Check the current Working Directory

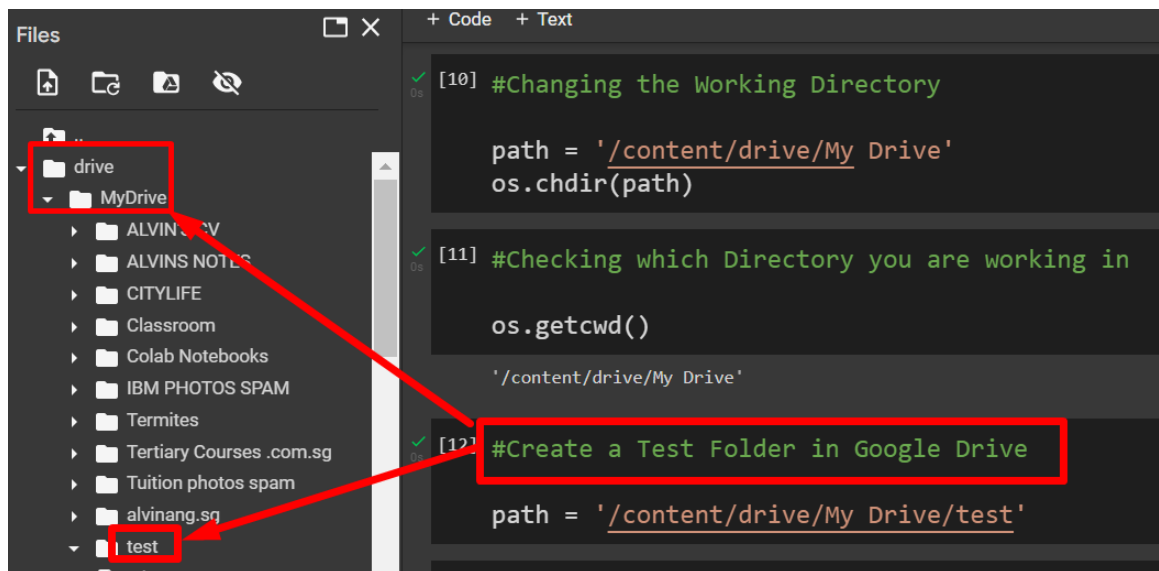
os.getcwd()

'/content/drive/MyDrive'

▶ #Mounting Google Drive

from google.colab import drive
drive.mount('/content/drive')

↳ Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```



```
Files + Code + Text

[10] #Changing the Working Directory

path = '/content/drive/My Drive'
os.chdir(path)

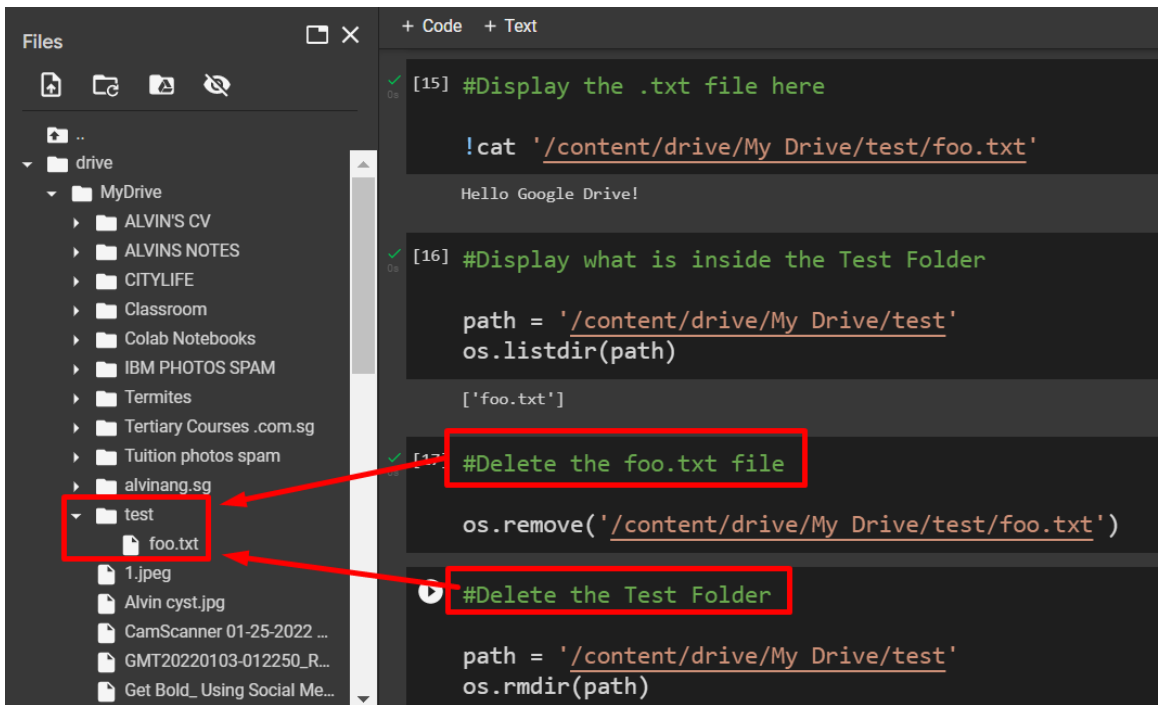
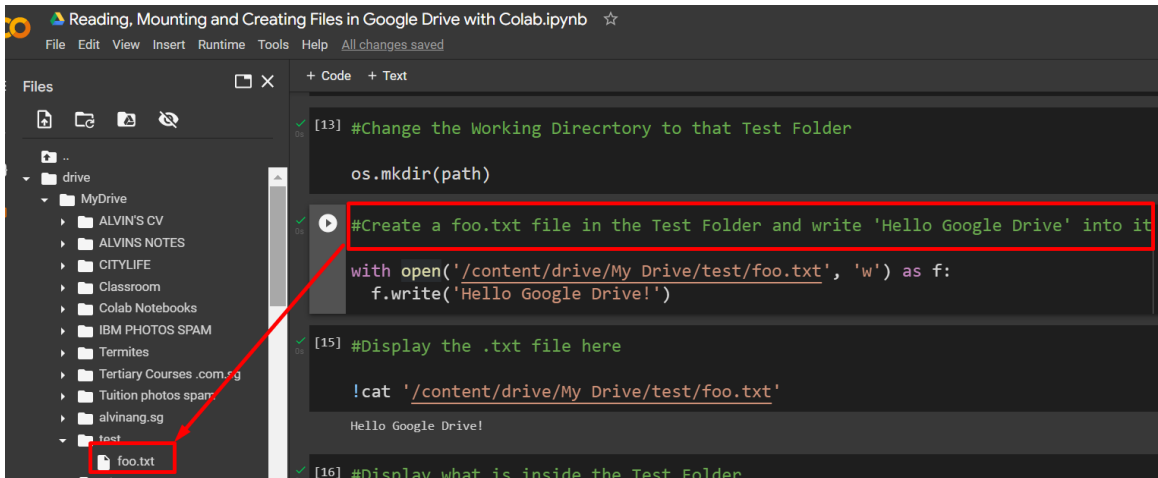
[11] #Checking which Directory you are working in

os.getcwd()

'/content/drive/My Drive'

[12] #Create a Test Folder in Google Drive

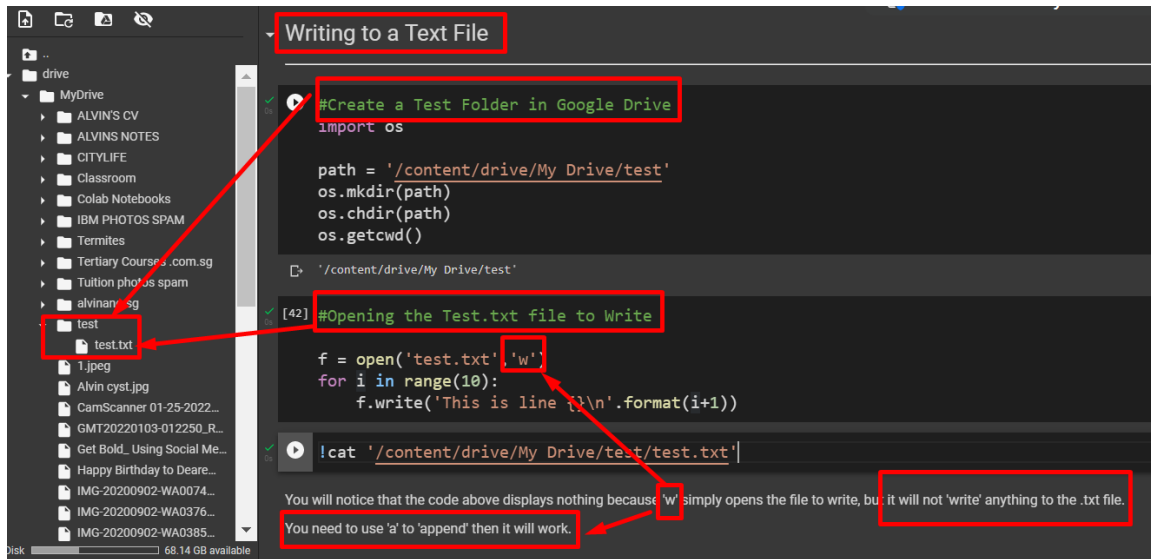
path = '/content/drive/My Drive/test'
```



## G. WRITING TO A .TXT FILE

File:

<https://www.alvinang.sg/s/Reading Mounting and Creating Files in Google Drive with Colab.ipynb>



The screenshot shows a Colab notebook titled "Writing to a Text File". The left sidebar displays a file explorer for "MyDrive" with a "test" folder containing a "test.txt" file. The notebook code is as follows:

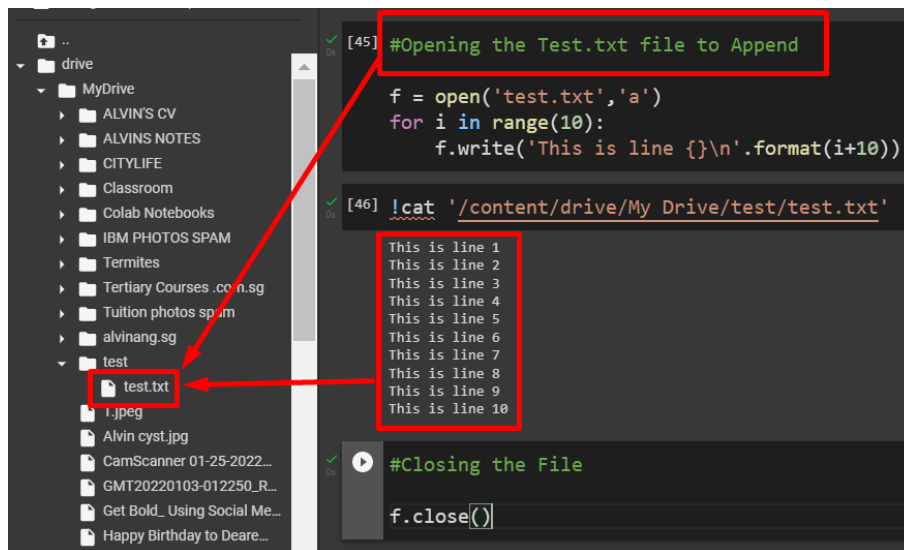
```
#Create a Test Folder in Google Drive
import os

path = '/content/drive/My Drive/test'
os.mkdir(path)
os.chdir(path)
os.getcwd()

[42] #Opening the Test.txt file to Write
f = open('test.txt', 'w')
for i in range(10):
    f.write('This is line {}'.format(i+1))

!cat '/content/drive/My Drive/test/test.txt'
```

Annotations include red boxes around the folder name "test", the file name "test.txt", the file opening code, and the terminal output. A text box explains: "You will notice that the code above displays nothing because 'w' simply opens the file to write, but it will not 'write' anything to the .txt file. You need to use 'a' to 'append' then it will work."



The screenshot shows the continuation of the Colab notebook. The code is as follows:

```
[45] #Opening the Test.txt file to Append
f = open('test.txt', 'a')
for i in range(10):
    f.write('This is line {}'.format(i+10))

[46] !cat '/content/drive/My Drive/test/test.txt'
```

The terminal output shows the contents of the file after appending:

```
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
This is line 6
This is line 7
This is line 8
This is line 9
This is line 10
```

The final code block shows the file being closed:

```
#Closing the File
f.close()
```

Annotations include red boxes around the file name "test.txt", the file opening code, the terminal output, and the closing code.

## H. COMPUTING BMI

File:

<https://www.alvinang.sg/s/Reading Mounting and Creating Files in Google Drive with Colab.ipynb>

The screenshot shows a Google Colab interface with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with 'test' and 'user.txt' files. The code editor contains the following code:

```
[47] T.close()

Prompt user to enter the name, weight and height, and compute the bmi
(=weight/height*height)

[48] from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)

[50] import os
path = '/content/drive/My Drive/test'
os.mkdir(path)
os.chdir(path)
os.getcwd()

/content/drive/My Drive/test

[51] f = open('user.txt', 'w')
f.write('name\theight\tweight\tbmi\n')
f.close()
```

The output shows the current directory path and the execution of the file creation code. A small window titled 'user.txt' is open, showing the following content:

```
1 name height weight bmi
2
```

The screenshot shows the same Google Colab interface. The code editor contains the following code:

```
[52] name = input('Enter name: ')
height = float(input('Enter height (m): '))
weight = float(input('Enter weight (kg): '))
with open('user.txt', 'a') as f:
    bmi = round(weight/(height*height),2)
    f.write("{}\t{}\t{}\t{}\n".format(name.lower(),height,weight,bmi))

Enter name: HAHABA
Enter height (m): 2.5
Enter weight (kg): 150

!cat /content/drive/My Drive/test/user.txt

name height weight bmi
hahaha 2.5 150.0 24.0
```

The output shows the user input and the execution of the file writing code. A small window titled 'user.txt' is open, showing the following content:

```
1 name height weight bmi
2 hahaha 2.5 150.0 24.0
3
```

## I. SPECIAL TRICKS IN COLAB!

### 1. HELP()

```
[9] help(print) use help() to find out anything about all functions
Help on built-in function print in module builtins:

print(...)
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file: a file-like object (stream); defaults to the current sys.stdout.
sep:  string inserted between values, default a space.
end:  string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
```

### 2. ?

```
print?
put a ? at the end of the function.. you will see this....

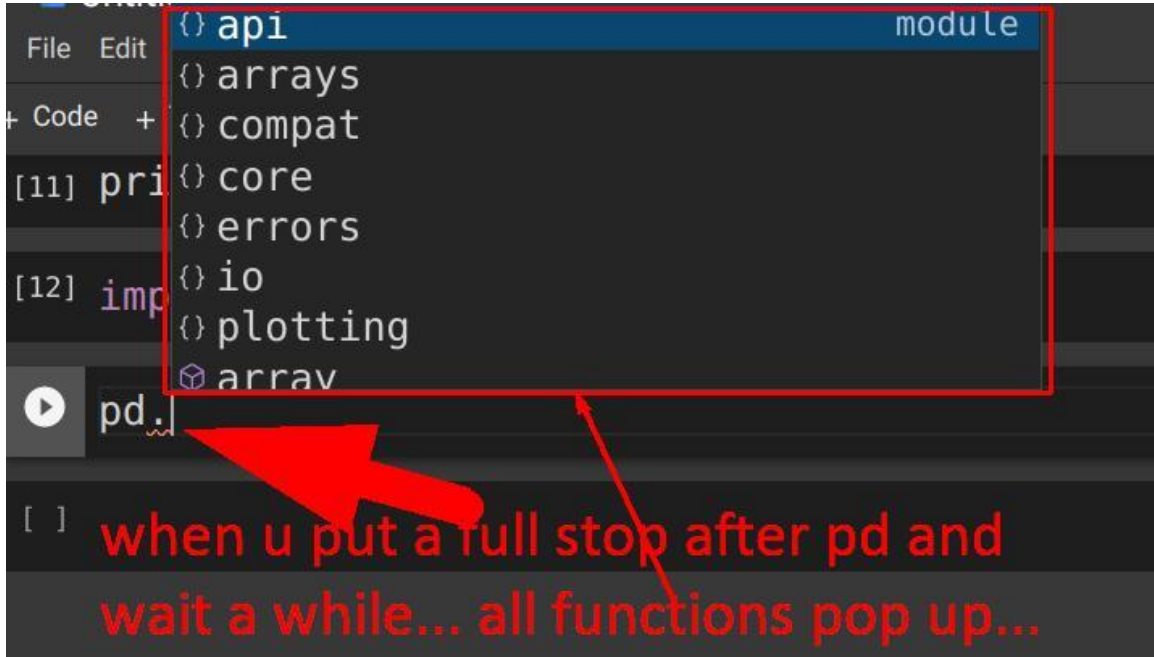
Docstring:
print(value, ..., sep=' ',
end='\n', file=sys.stdout,
flush=False)

Prints the values to a stream, or
to sys.stdout by default.
Optional keyword arguments:
file: a file-like object (stream);
defaults to the current sys.stdout.
sep: string inserted between
values, default a space.
end: string appended after the
last value, default a newline.
flush: whether to forcibly flush
```

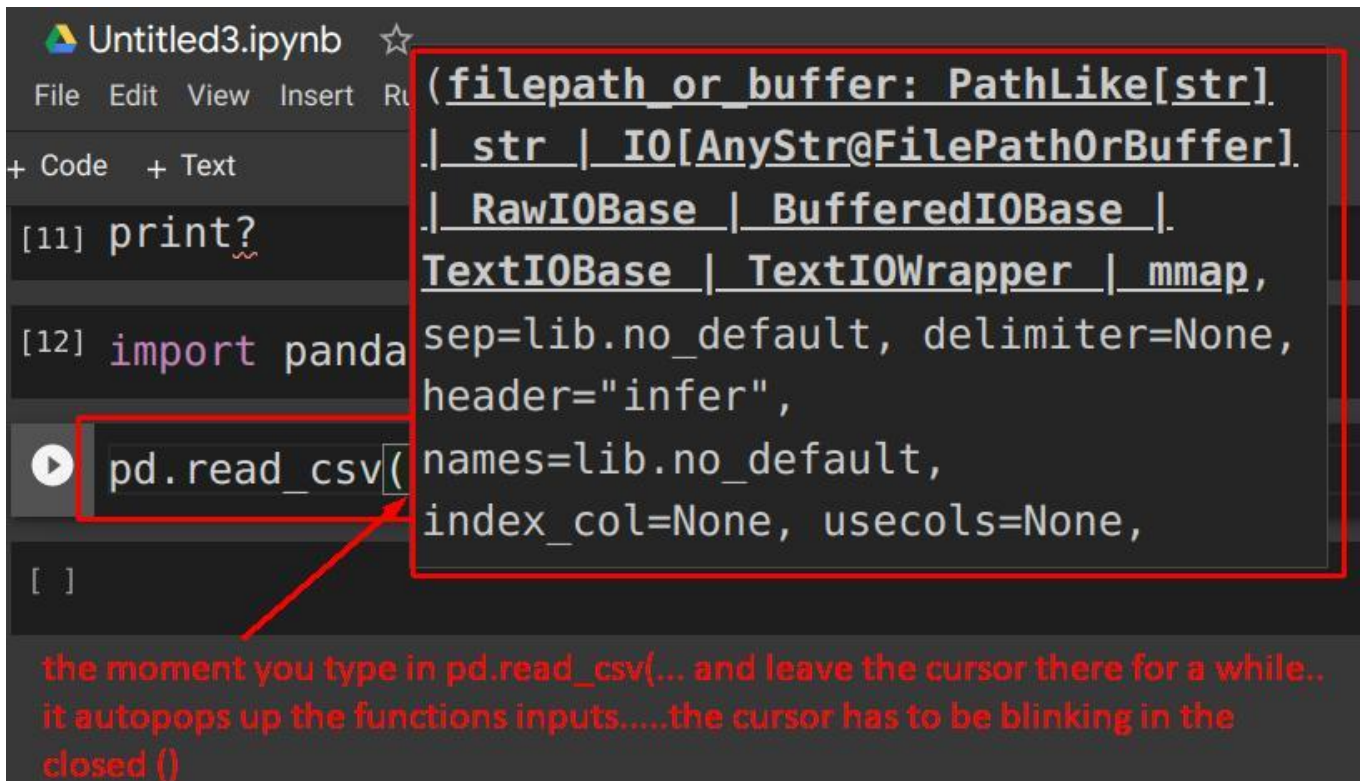
3. SEEING FUNCTIONS POP UP AFTER THE .

```
[12] import pandas as pd
```

Run this first”



4. LEAVE THE CURSOR INSIDE THE ()



Untitled3.ipynb ☆

File Edit View Insert Run

+ Code + Text

```
[11] print?
```

```
[12] import pandas
```

```
pd.read_csv(
```

[ ]

(filepath\_or\_buffer: PathLike[str] | str | IO[AnyStr@FilePathOrBuffer] | RawIOBase | BufferedIOBase | TextIOBase | TextIOWrapper | mmap, sep=lib.no\_default, delimiter=None, header="infer", names=lib.no\_default, index\_col=None, usecols=None,

the moment you type in `pd.read_csv(...` and leave the cursor there for a while.. it autopops up the functions inputs.....the cursor has to be blinking in the closed ()

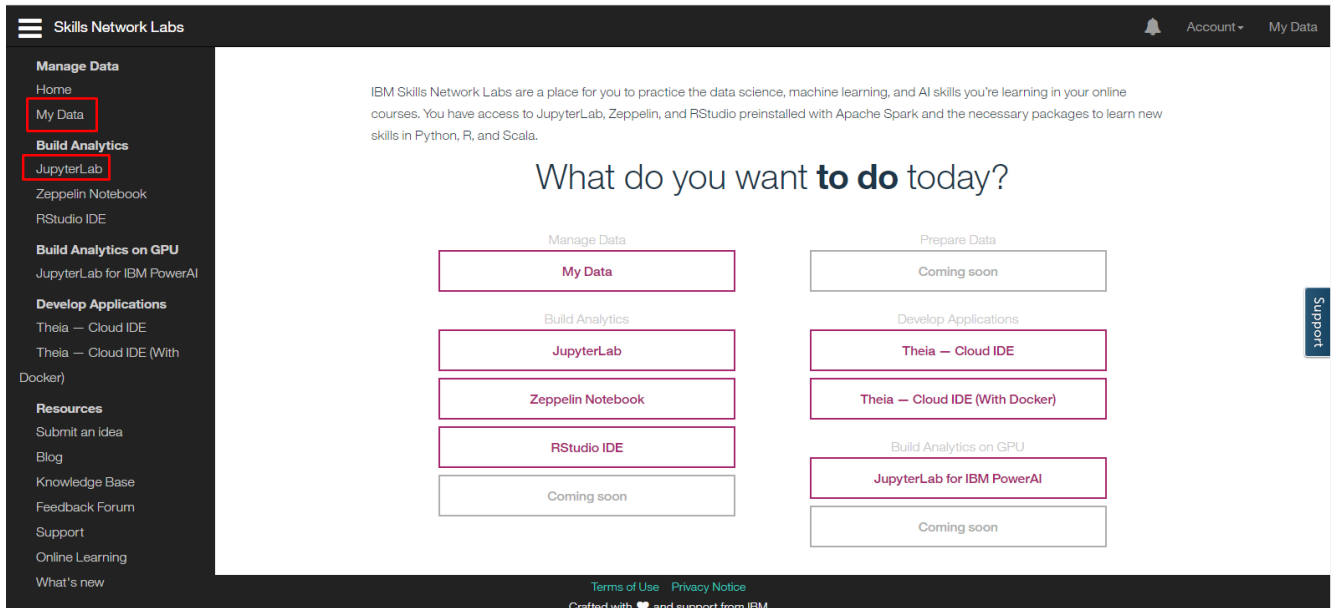


---

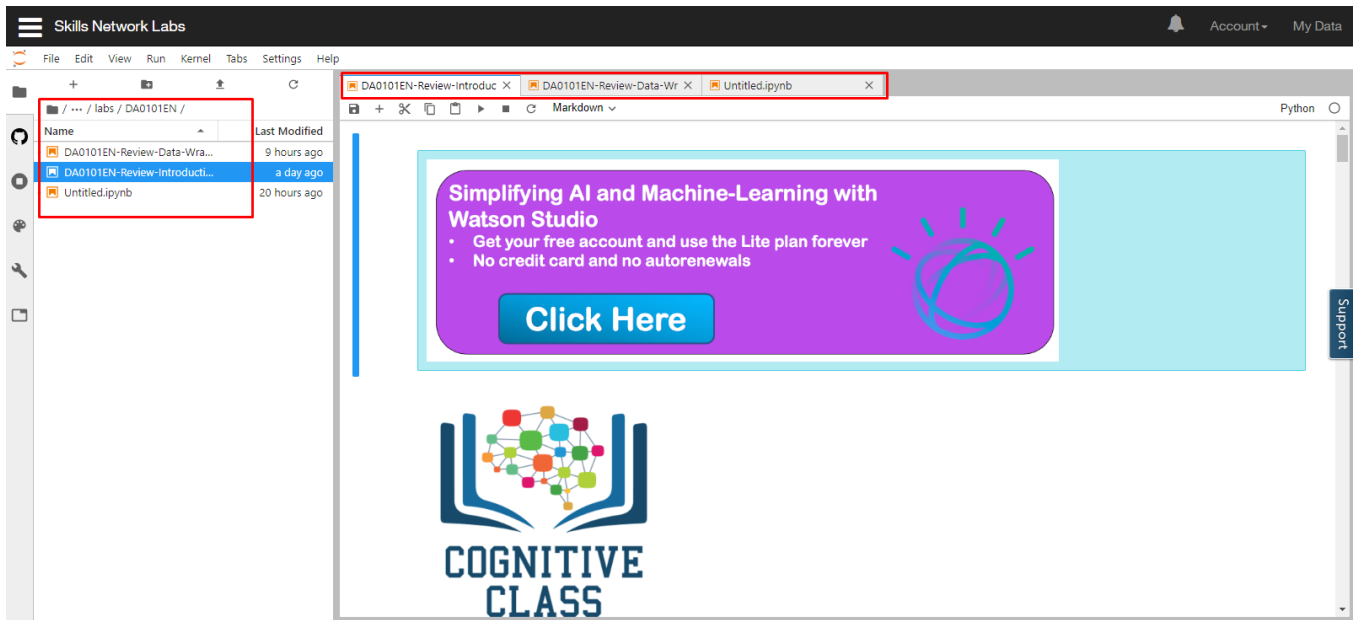
## II. IBM SKILLS NETWORK LABS

---

- Go to <https://labs.cognitiveclass.ai/>
- Create a username and password.
- You should come to this screen:



- If you already have taken up E-lessons on <http://cognitiveclass.ai/>, then those hands on lab will be stored in “My Data” and you can access those files.
- However, if you haven’t got any data files and you are just starting out, click on “Build Analytics → JupyterLab”



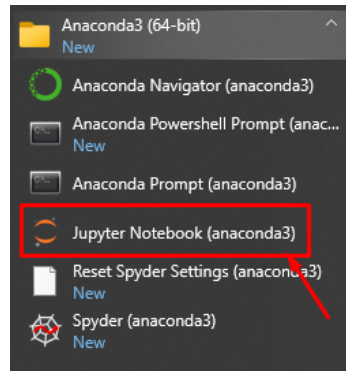
- You will come to this page shown above.
- Your Jupyter Notebook has been opened and all your .ipynb files can be now accessed.

---

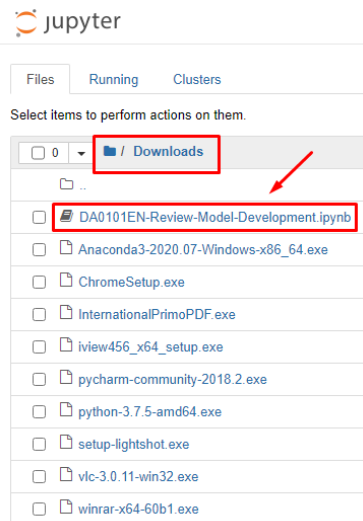
### III. ANACONDA / JUPYTER NOTEBOOK

---

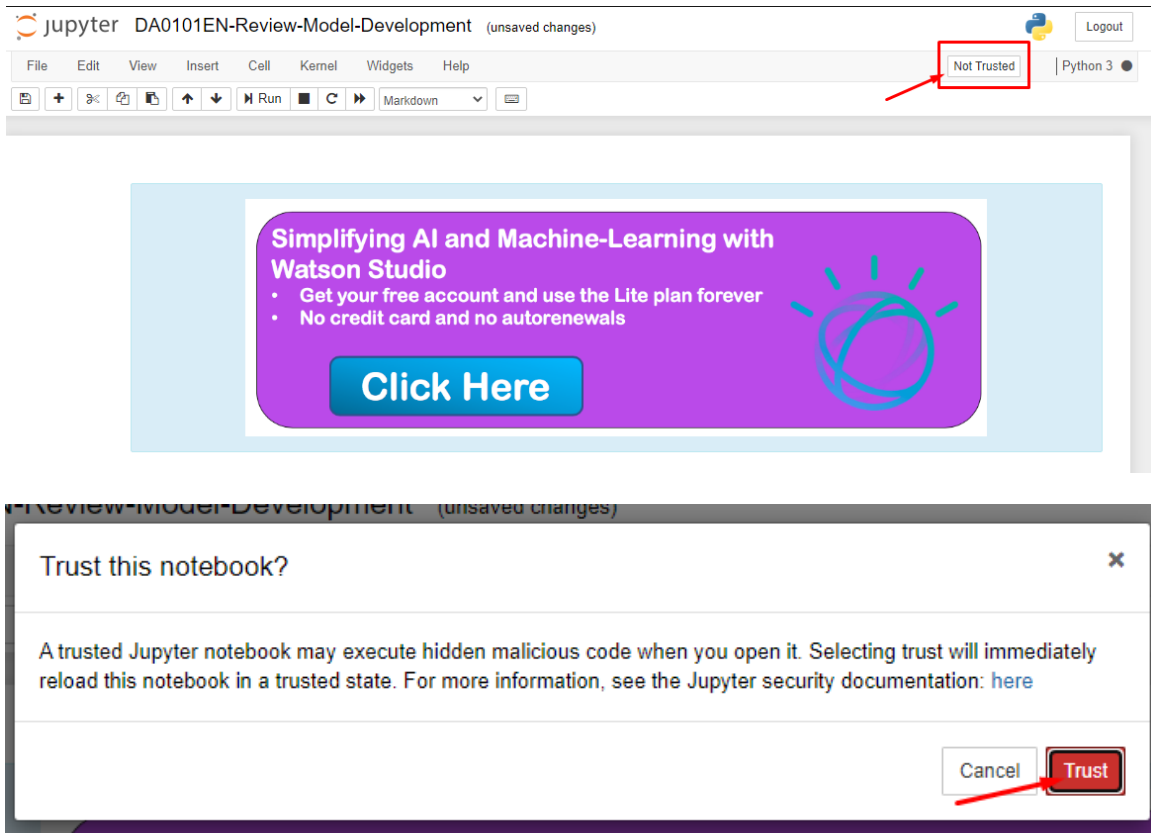
- Go to <https://www.anaconda.com/products/individual#windows>
- Install Anaconda on Windows.
- Installation usually takes a long time... be patient for entire process.
- Once Anaconda has been installed, you will see this in the start menu.



- Click on Jupyter Notebook (anaconda3).
- You will be directed to a web based browser...
- Within the browser, search for the folder where you installed the .ipynb.



- You may come across “Not Trusted”... just “Trust” it....



- That’s all for Anaconda!

## A. INSTALLING JUPYTER NOTEBOOK ON LINUX MINT

### 1. INSTALL PYTHON 3 FIRST

*a) Open terminal:*

- `sudo apt-get install python3-minimal`

*b) If not upgraded to the latest python 3:*

- `sudo apt install python3`

*c) Install pip3:*

- `sudo apt-get install python3-pip python-dev`

### 2. INSTALL JUPYTER NOTEBOOK

- `pip3 install jupyter`

*a) To activate Jupyter Notebook:*

- `jupyter notebook`

*b) If fail because file is in `./local`*

- Try:
- `~/local/bin/jupyter-notebook`

---

## ABOUT DR. ALVIN ANG

---



Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at [www.AlvinAng.sg](http://www.AlvinAng.sg).