We'll be starting shortly!

To help us run the workshop smoothly, please kindly:

- Switch off screen sharing and mute your microphone
- Submit all questions using the Q&A function
- If you have an urgent request, please use the "Raise Hand" function

Thank you!

{<oding:lab}

Shopee CODE LEAGUE 2020

{<oding:lab}

Data Science 101

Session 4

About Coding Lab

- Founded by an MIT Graduate who worked in Silicon Valley
- Global Tech advisory team based in New York, Japan and Singapore
- We have Campuses in Japan, Australia and Singapore
- We offer coding classes starting from age 4 to adulthood

{<oding:lab}

Features and Partners

PSA The World's Port of Call

CISCO

NUS National University of Singapore

Zendesk

UBISOFT

{<oding:lab}

Shopee CODE LEAGUE 2020

Features and Partners

Shopee

Ministry of Education SINGAPORE

THE STRAITS TIMES

zaobao.sg 早晚全新

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

企業経営の新潮流を読み解く Bizコンパス

YOUNG PARENTS

mediacorp

BRAND'S

CGTN

Sassy Mama

parentsworld

And many more...

{<oding:lab}

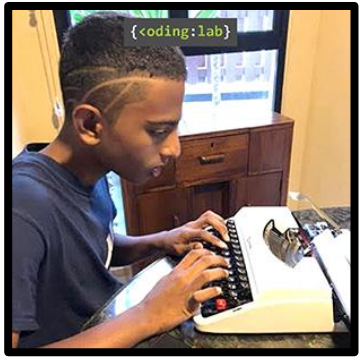
Shopee CODE LEAGUE 2020

Meet our Students

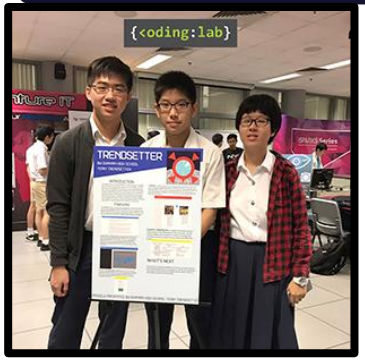
Sarah, 18
Honourable Mention, NOI 2018

Team ajdisjd
1st Place, iCode 2019

Elijah, 14
Youngest Medalist, NOI 2019



Surya, 14
Created a stock rating algorithm



Team Trendsetter
Best Presentation Award

More schools we have taught at:

EtonHouse International School • Pre-School

Modern Montessori International

eunoia JUNIOR COLLEGE

{<oding:lab}

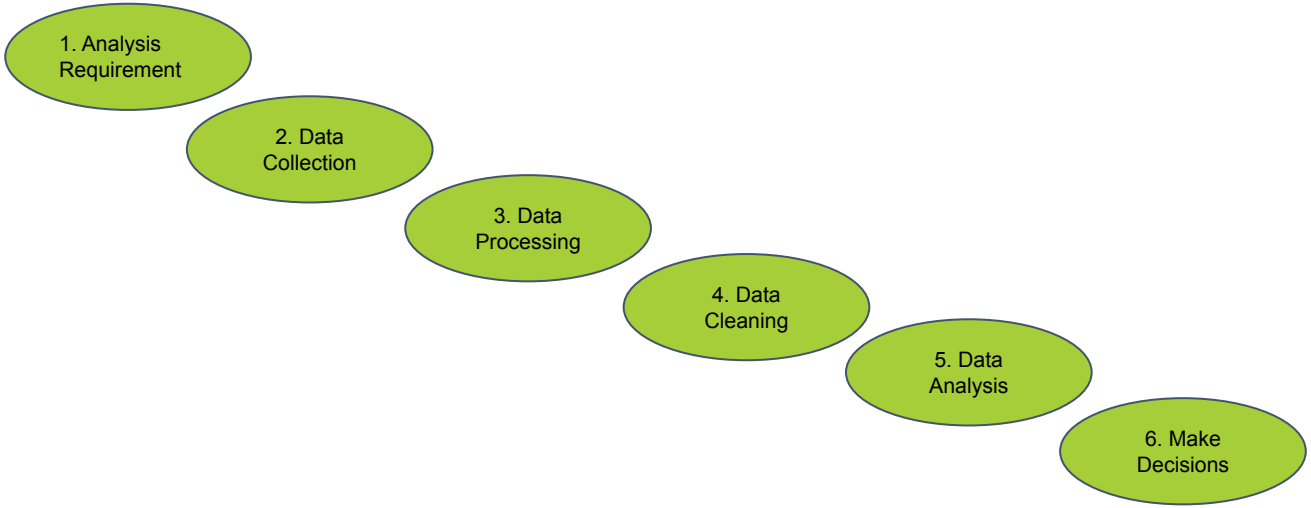
Shopee CODE LEAGUE 2020

Session Overview

- Playing with real world data
  - Data.gov
  - Kaggle
  - News sites
  - Wikipedia

{<oding:lab}

Data Analytics Process Overview



{<oding:lab}

Shopee CODE LEAGUE 2020

Introduction to Data Extraction

{<oding:lab}

What is Data Extraction

- Process of retrieving data from various sources
  - Perform analysis on collected datas
- Most of it is poorly organised and unstructured
- Data extraction makes it possible to consolidate, process, and refine necessary information

{<oding:lab}

Shopee CODE LEAGUE 2020

Why is Data Extraction important

- Collecting necessary data for analysis
  - Eg, conducting research regarding marketing techniques
- Improving accuracy
  - Gaining new insights from multiple resources become coming to conclusion

{<oding:lab}

How is data extraction applicable to us?

- Gain insights into real world problems
- Draw conclusions regarding the relationship between two datasets/variables
- We will be focusing on data available from two sources for this course
  - Data.gov.sg
  - Kaggle.com

{<oding:lab}

Request function

- Before we extract data from websites, we need to make use of the built in method request
- As the name suggests, the method abstracts the complexities of making request behind a simple API
  - Allowing users to simply access the data for usage
- The method uses basic HTTP(Hypertext Transfer Protocol) to enable communications between clients and servers
- We are going to focus on ‘get’

{<oding:lab}

Request.get

- We first need to install request on Google Colab before importing it

```
1   pip install requests
2
3   import requests
```

- The most common HTTP method is GET, which tells your program you would like to retrieve data from a specific source
- E.g. Try typing the following

```
1   requests.get('https://api.github.com')
```

{<oding:lab}

Response Object

- Response inspects the result of the request which contains a lot of information about the source
- The first bit of information you can get is the status code
  - A 200 status indicates that the request was successful
  - A 404 status indicated an error where the url was not found
- The code below should return a 200 status code

```
1   response = requests.get('https://api.github.com')
2
3   response.status_code
```

{<oding:lab}

Response.content

- The response of a request contains much more useful data and information beneficial for analysis
- To access the information, we simply make use of .content

```
1  response = requests.get('https://api.github.com')
2
3  response.content
```

{<oding:lab}

Response.json()

- To access every value in the object by keys, we can store the data in a dictionary
- The return output of response.json() is a dictionary, which allows easier extraction of data

Response Function (Demo/Practice - 1)

- There are many interesting API online
- Search one up from the website and retrieve the status code and content of the request

Checkpoint 1

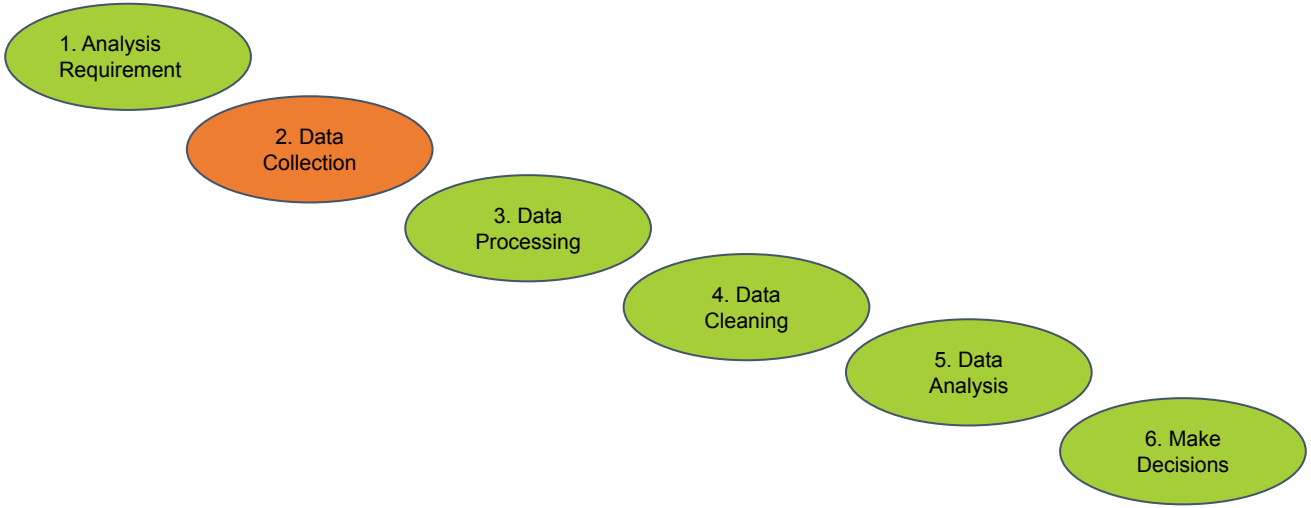
- Every student must be able to:
  - Understand what requests method does
  - Able to print out the content of the requested API

- For students who are waiting, try the following:
  - What other information can you gather from the response?
  - Search up and share your findings with the class

{<oding:lab}

Shopee CODE LEAGUE 2020

Data.gov.sg - Data Collection



{<oding:lab}

API (Application Programming Interface)

- Allow applications or website to communicate with one another
- Extract data from a particular website
- An API is not a database
  - It is an access point to an app that can access a database
- In order to get the information we want
  - We send API request detailing the information we want

{<oding:lab}

What is Data.gov.sg

- A government portal which contains publicly available datasets
- Relevant datasets can be extracted for analysis and research
- Charts and articles are actively used to facilitate easier understanding
- We are going to access the real time API on the website
  - Traffic congestion
  - Weather

{<oding:lab}

Shopee CODE LEAGUE 2020

Real time API

- Real time datasets such as traffic condition and weather can be easily accessed
- Knowing how to write the function is crucial to extract data from anywhere on the web
- You will be learning how to write the function for extraction of information from data.gov.sg
    - We are going to extract the data in JSON format first then store them into a Dataframe

{<oding:lab}

Shopee CODE LEAGUE 2020

Function to extract data (1/2)

- Start by writing a function which takes in the api URL available on the website
- Then we would request to get the apiURL and verify that it is true
  - For security purpose to ensure sensitive information sent across the Internet is encrypted
  - Eg, if you look at the left end of URL bar, it should be a secured connection or strangers are able to access the information

{<oding:lab}

Function to extract data (2/2)

- .raise_for_status() helps check whether a request is successful
  - If the request fails, it returns an error

```
1  def getJSONData (apiURL):
2      response = requests.get(
3          apiURL,
4          verify = True,  # Verify SSL certificate
5      )
6      response.raise_for_status() # optional but good practice in case the call fails
7      return(response.json())
```

{<oding:lab}

Shopee CODE LEAGUE 2020

Main function

- Once the extraction is done, the main codes would be fairly easy
- We simply require the apiURL to be accepted by our function and print out the result

```
1   apiURL = "https://api.data.gov.sg/v1/environment/air-temperature/"
2   dataFromDataGov = getJSONData(apiURL)
3
4   print(dataFromDataGov)
```

{<oding:lab}

Shopee CODE LEAGUE 2020

Extraction of Data (Demo/Practice - 2)

- Make use of what we have learnt, extract one of Real-time APIs available on data.gov.sg
- Observe how the data is stored
  - Dictionaries or lists?
- Try putting the data into a Dataframe, is it possible?

{<oding:lab}

Shopee CODE LEAGUE 2020

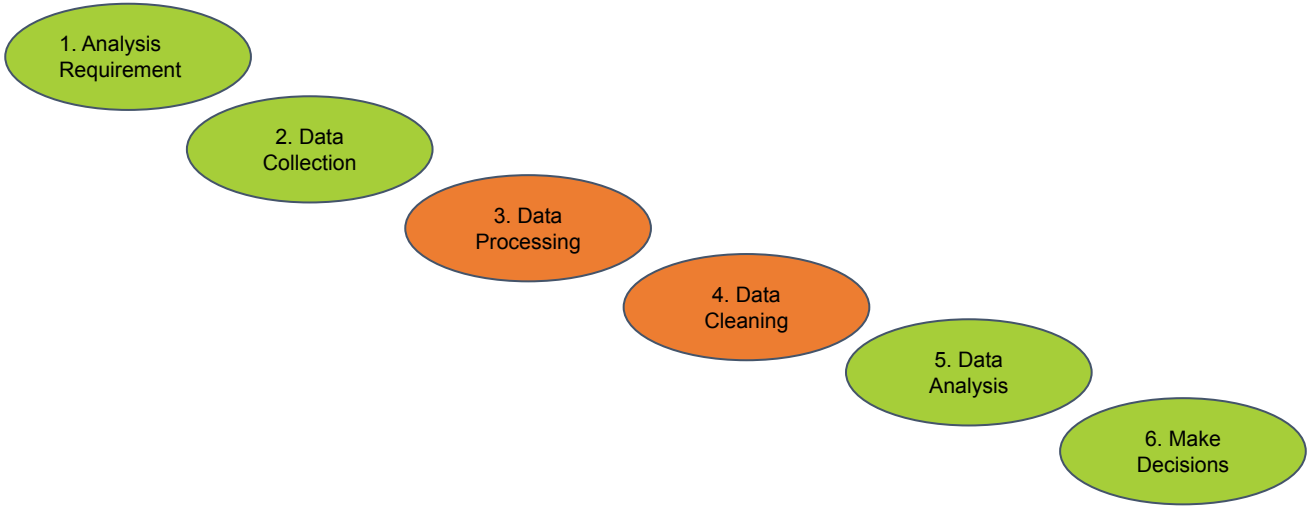
Checkpoint 2

- Every student must be able to:
  - Extract data from Data.gov.sg
  - Store the data in JSON format
- For students who are waiting, try the following:
  - Try finding other API available on kaggle.com and extract the data

{<oding:lab}

Shopee CODE LEAGUE 2020

Data.gov.sg - Data Filtering



{<oding:lab}

Unorganised Data

- Observe the output of our code
- We have successfully extracted the data from the website but the data is a mess!
  - It contains multiple nested dictionaries and lists
  - There is no way to store the data in a Dataframe
  - Hence we need to extract the data step by step

Nested dictionaries

- How do we access the data in a nested dictionary?

```
1   nested_dict = { 'dictA': {'key_1': 'value_1'},
2                   'dictB': {'key_2': 'value_2'}}
3
4   for k, v in nested_dict.items():
5     if k == 'dictA':
6       print(v)
```

- Recall there are keys and values in a dictionary

{<oding:lab}

First dictionary

- Accessing the first dictionary only returns you 3 keys
  - ‘Metadata’ , ‘items’ and ‘api info’

```
for k, v in dataFromDataGov.items(): #metadata, items,  api info
  print(k)
```

- The metadata contains the most relevant information and to extract the dictionary ‘metadata’,

```
for k, v in dataFromDataGov.items(): #metadata, items,  api info
  if k == 'metadata':
    print(v)
```

{<oding:lab}

Shopee CODE LEAGUE 2020

Second dictionary

- The second dictionary contains even more information and we follow the same codes to extract the desired values based on our interested key
- Observe that currently we are accessing a list instead of a dictionary, thus the extraction process changes slightly

```
for key , value in v.items():
    if key =='stations':
        print(value[0])
```

{<oding:lab}

Storing data in Dataframe

- After much filtering and getting our desired data, we now want to store it into a dataframe
  - First normalize the json data into a flat table
  - Use Dataframe.from_dict to convert the data into a dataframe
  - The orient simply means the orientation of the table

```
df = pd.DataFrame.from_dict(json_normalize(value), orient='columns')
df
```

{<oding:lab}

Shopee CODE LEAGUE 2020

Extracting data and storing in Dataframe(Demo/Practice 3)

- Access the apiURL- Taxi Availability
- Extract the data and store the coordinates of the taxis in a dataframe
    - Make sure to extract the information properly as we will be using the data later for analysis

{<oding:lab}

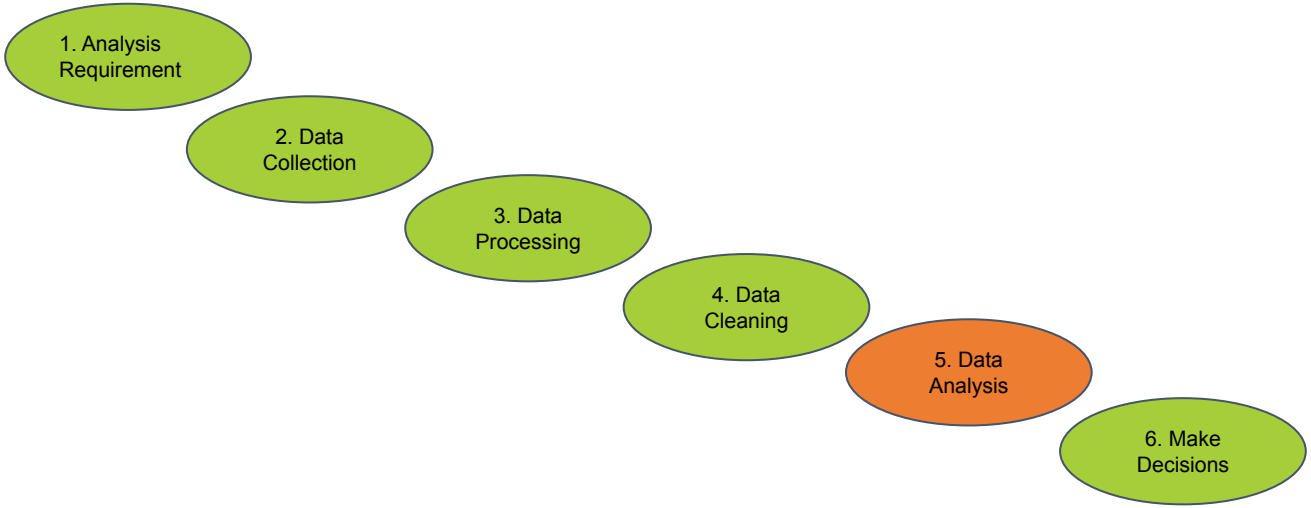
Checkpoint 3

- Every student must be able to:
  - Extract specific data from a massive dataset
  - Understand how to extract information from a list and dictionary
- For students who are waiting, try the following:
  - Access other apiURL from the website and extract specific data

{<oding:lab}

Shopee CODE LEAGUE 2020

Data.gov.sg - Data Analysis



{<oding:lab}

What do we do after extraction?

- We have extracted the coordinates of taxis for our demo practice earlier, what's next?
- Data is still considered useless if we do not know understand what is the purpose of extraction
    - Thus we use data visualisation tools to bring purpose to our extracted data

{<oding:lab}

Shopee CODE LEAGUE 2020

Taxi Availability in Singapore

- The data shows the coordinates of taxis all around Singapore, with such a massive amount of numbers, it is difficult to gauge the quantity
- We can plot the data in a scatter plot with plotly
  - Recall in Session 1 how we plot graphs on plotly
  - Plotting a scatter is almost the same as plotting any other graphs

{<oding:lab}

Shopee CODE LEAGUE 2020

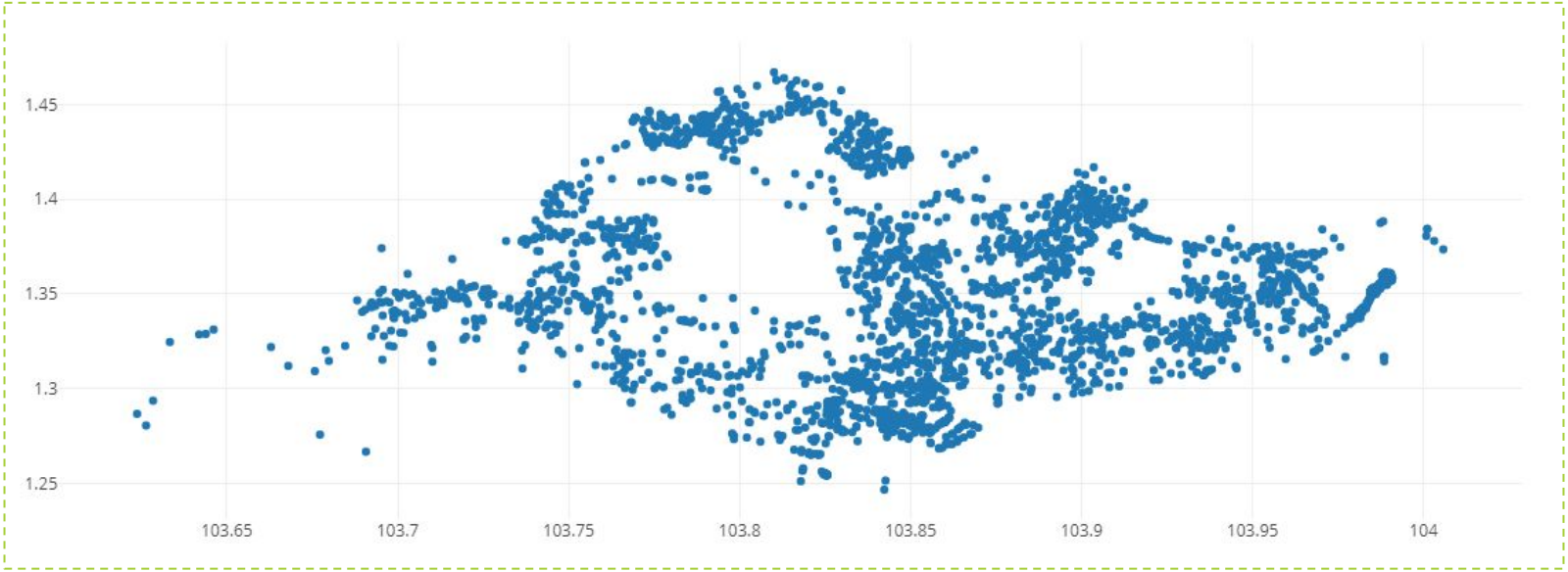
Scatter plot of data

- The x axis would be the longitude while the y axis is the latitude
- We can then plot the diagram out

```
27  import plotly.express as px
28  plot = px.scatter(df,'longitude','latitude',)
29  plot
```

{<oding:lab}

Data visualisation output



{<oding:lab}

Shopee CODE LEAGUE 2020

Drawing conclusions

- Observe the scatter plot
  - What can you conclude from this scatter plot?
    - Is taxi availability an issue in Singapore?
    - Which area has a shortage or taxis?
- Know what your data means is useful when it comes to drawing certain conclusion and thus finding solution to solve the issue

{<oding:lab}

Data Visualisation (Demo/Practice - 4)

- The file ‘laptops.csv’ is a dataset for 1300 laptop models found on kaggle.com
- Find out which company earned the most from sales of laptop
- Follow the guidelines if you are unsure what to do
  - Extract the file and store it in a dataframe
  - Which plot should we use for a clearer visualisation of data?

{<oding:lab}

Shopee CODE LEAGUE 2020

Checkpoint 4

- Every student must be able to:
  - Plot extracted data into graph using plotly

- For students who are waiting, try the following:
  - Explore other datasets from kaggle.com and plot data out using plotly

{<oding:lab}

Shopee CODE LEAGUE 2020

Kaggle.com

{<oding:lab}

Shopee CODE LEAGUE 2020

Kaggle Overview

- A platform for data science enthusiasts to interact and compete in solving real-life problems
- Public data platform - many datasets available
- Subsidiary of Google LLC

{<oding:lab}

Shopee CODE LEAGUE 2020

Interesting Datasets on Kaggle

- Spotify dataset
  - https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks
- Coursera courses
  - https://www.kaggle.com/siddharthm1698/coursera-course-dataset
- Covid-19
  - https://www.kaggle.com/imdevskp/corona-virus-report

{<oding:lab}

Kaggle Walkthrough

- Using the Covid-19 dataset from the link in the previous slide
  - Sign in to Kaggle and download dataset
- 3 files inside
  - covid_19_clean_complete.csv
  - usa_county_wise.csv
  - worldometer_data.csv

{<oding:lab}

Shopee CODE LEAGUE 2020

Scenario

- Say we want to write a program that will plot the number of confirmed infections by country according to a date selected by the user
- We will be using covid_19_clean_complete.csv

Read csv with pandas

- Read csv with pandas
  - Rename ‘Country/Region’ to ‘Country’
    - So we can call df.Country later. Special characters are excluded from this

```
1  import pandas as pd
2
3  data = pd.read_csv('covid_19_clean_complete.csv')
4  df = pd.DataFrame(data)
5  df = df.rename(columns={'Country/Region': 'Country'})
6  df
```

{<oding:lab}

Shopee CODE LEAGUE 2020

Pandas DataFrame

	Province/State	Country	Lat	Long	Date	Confirmed	Deaths	Recovered	WHO Region
0	NaN	Afghanistan	33.000000	65.000000	1/22/20	0	0	0	emro
1	NaN	Albania	41.153300	20.168300	1/22/20	0	0	0	euro
2	NaN	Algeria	28.033900	1.659600	1/22/20	0	0	0	afro
3	NaN	Andorra	42.506300	1.521800	1/22/20	0	0	0	euro
4	NaN	Angola	-11.202700	17.873900	1/22/20	0	0	0	afro
...	...	...	...	...	...	...	...	...	...
36300	NaN	Sao Tome and Principe	0.186360	6.613081	6/6/20	499	12	68	afro
36301	NaN	Yemen	15.552727	48.516388	6/6/20	482	111	0	emro
36302	NaN	Comoros	-11.645500	43.333300	6/6/20	141	2	67	afro
36303	NaN	Tajikistan	38.861034	71.276093	6/6/20	4453	48	0	euro
36304	NaN	Lesotho	-29.609988	28.233608	6/6/20	4	0	2	afro

36305 rows × 9 columns

{<oding:lab}

Shopee CODE LEAGUE 2020

Filter by Date and Country

- Write a simple program that filters the dataframe by date and

```
1  date = input("Select a date in mm/dd/yy format. \nE.g. 3/20/20 for 20th March or 3/3/20 for 3rd March: ")
2  date = [date]
3  countries = input("Enter a list of countries of interest, separated by commas: ")
4  country_list = countries.split(',')
5  country_and_date_df = df[df.Country.isin(country_list) & df.Date.isin(date)]
6  country_and_date_df
```

{<oding:lab}

Shopee CODE LEAGUE 2020

Sample Output

```
Select a date in mm/dd/yy format.
E.g. 3/20/20 for 20th March or 3/3/20 for 3rd March: 6/6/20
Enter a list of countries of interest, separated by commas: Italy,Malaysia,Singapore,Spain,United Kingdom
```

	Province/State	Country	Lat	Long	Date	Confirmed	Deaths	Recovered	WHO Region
36177	NaN	Italy	43.0000	12.0000	2020-06-06	234801	33846	165078	euro
36193	NaN	Malaysia	2.5000	112.5000	2020-06-06	8303	117	6635	wpro
36236	NaN	Singapore	1.2833	103.8333	2020-06-06	37527	25	24559	wpro
36241	NaN	Spain	40.0000	-4.0000	2020-06-06	241310	27135	150376	euro
36257	Bermuda	United Kingdom	32.3078	-64.7505	2020-06-06	141	9	114	euro
36258	Cayman Islands	United Kingdom	19.3133	-81.2546	2020-06-06	164	1	93	euro
36259	Channel Islands	United Kingdom	49.3723	-2.3644	2020-06-06	563	46	512	euro
36260	Gibraltar	United Kingdom	36.1408	-5.3536	2020-06-06	175	0	155	euro
36261	Isle of Man	United Kingdom	54.2361	-4.5481	2020-06-06	336	24	312	euro
36262	Montserrat	United Kingdom	16.7425	-62.1874	2020-06-06	11	1	10	euro
36263	NaN	United Kingdom	55.3781	-3.4360	2020-06-06	284868	40465	0	euro
36288	Anguilla	United Kingdom	18.2206	-63.0686	2020-06-06	3	0	3	euro
36289	British Virgin Islands	United Kingdom	18.4207	-64.6400	2020-06-06	8	1	7	euro
36290	Turks and Caicos Islands	United Kingdom	21.6940	-71.7979	2020-06-06	12	1	11	euro
36296	Falkland Islands (Malvinas)	United Kingdom	-51.7963	-59.5236	2020-06-06	13	0	13	euro

{<oding:lab}

Shopee CODE LEAGUE 2020

Plotting a Graph

- Use plotly to plot bargraphs

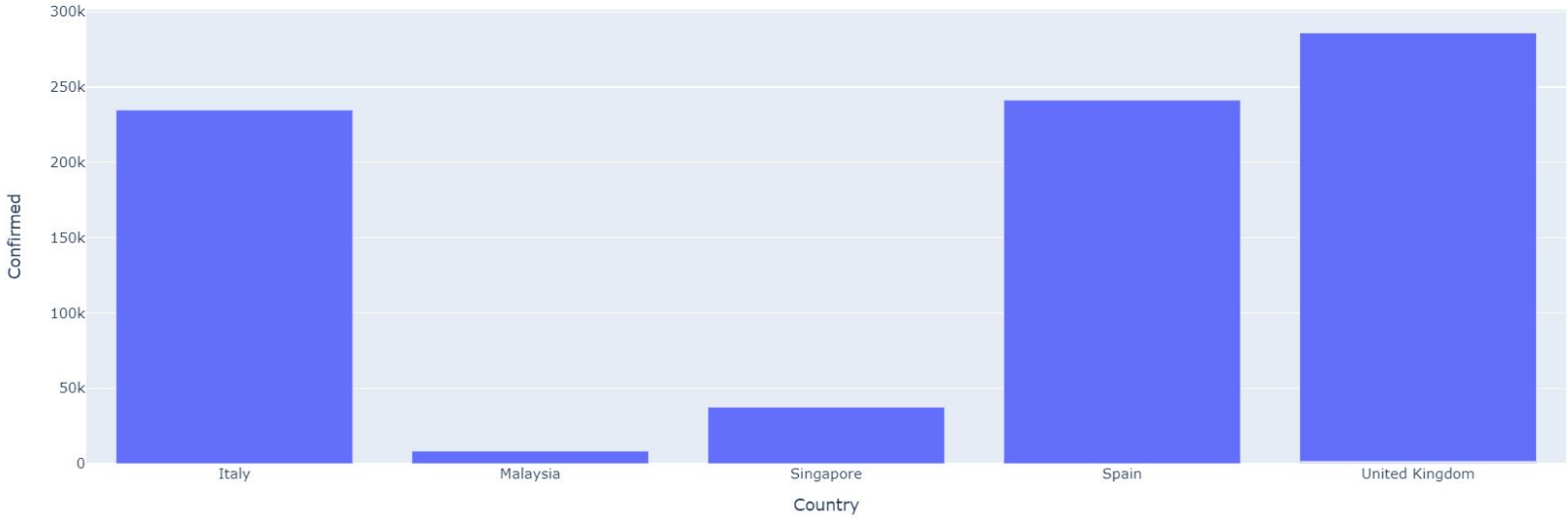
```
1  import plotly.express as px
2
3  px.bar(country_and_date_df, x = "Country", y = "Confirmed")
```

```
1  px.bar(country_and_date_df, x = "Country", y = "Deaths")
```

{<oding:lab}

Shopee CODE LEAGUE 2020

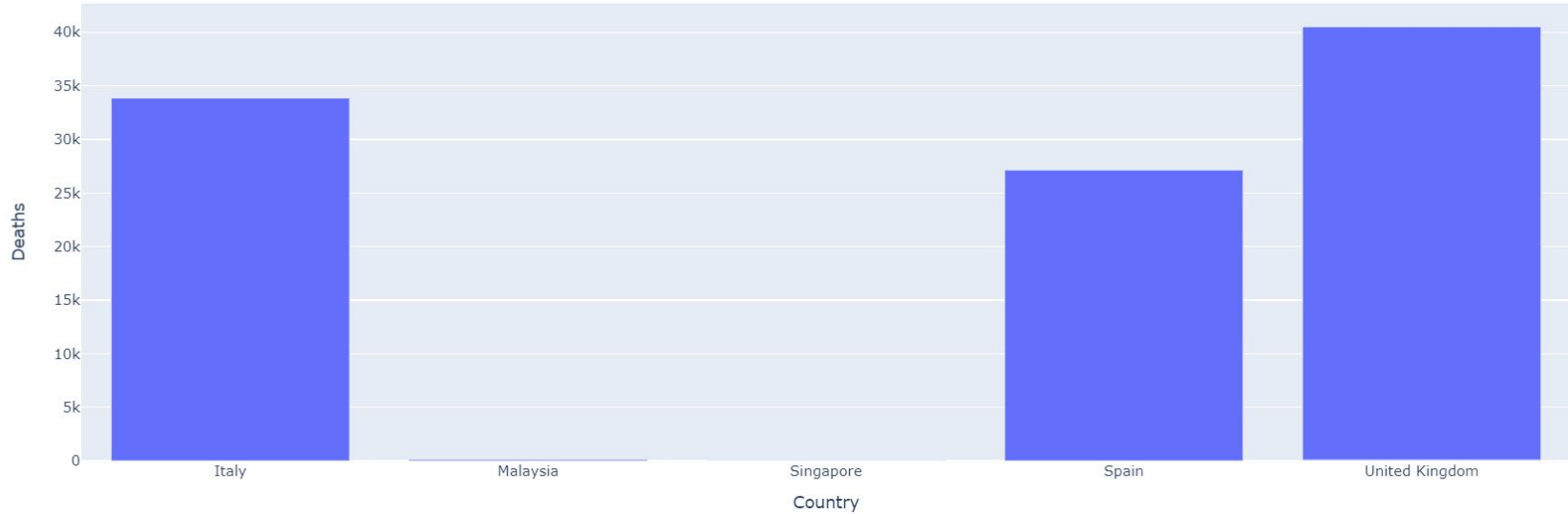
Graph of Confirmed Cases by Country



{<oding:lab}

Shopee CODE LEAGUE 2020

Graph of Death Count by Country



{<oding:lab}

Shopee CODE LEAGUE 2020

Kaggle Data (Demo/Practice - 5)

- Download covid_19_clean_complete.csv from Kaggle
  - https://www.kaggle.com/imdevskp/corona-virus-report
- Write a program that will plot the number of confirmed infections according to a date selected by the user
- Create a suitable data frame and plot out the respective graphs showing the number of confirmed cases and deaths

{<oding:lab}

Shopee CODE LEAGUE 2020

Checkpoint 5

- Every student must be able to:
  - Plot extracted data into graph using plotly

- For students who are waiting, try the following:
  - Create a program that allows the user to view covid data from a specific country over a range of dates

{<oding:lab}

Viewing Covid Data from one Country

- Say now we want to view data from one country
  - But over a range of dates to see how the country is coping with the virus
- We can create a program that asks the user for the date range, and the country of interest

{<oding:lab}

Shopee CODE LEAGUE 2020

Country by Date Range: Sample Code

- Using pd.to_datetime to convert a column to date type
  - For easy filtering of dates

```
start_date = input("Select a start date in mm/dd/yy format: ")
end_date = input("Select a start date in mm/dd/yy format: ")
country = input("Select a country: ")
df['Date'] = pd.to_datetime(df['Date'])
date_range_df = df[(df['Date'] >= start_date) & (df['Date'] <= end_date)]
country_date_range_df = date_range_df[date_range_df['Country'] == country]
country_date_range_df
```

{<oding:lab}

Shopee CODE LEAGUE 2020

Country by Date Range: Sample Output

```
Select a start date in mm/dd/yy format: 3/1/20
Select a start date in mm/dd/yy format: 6/1/20
Select a country: Singapore
```

	Province/State	Country	Lat	Long	Date	Confirmed	Deaths	Recovered	WHO Region
10531	NaN	Singapore	1.2833	103.8333	2020-03-01	106	0	72	wpro
10796	NaN	Singapore	1.2833	103.8333	2020-03-02	108	0	78	wpro
11061	NaN	Singapore	1.2833	103.8333	2020-03-03	110	0	78	wpro
11326	NaN	Singapore	1.2833	103.8333	2020-03-04	110	0	78	wpro
11591	NaN	Singapore	1.2833	103.8333	2020-03-05	117	0	78	wpro
...	...	...	...	...	...	...	...	...	...
33851	NaN	Singapore	1.2833	103.8333	2020-05-28	33249	23	18294	wpro
34116	NaN	Singapore	1.2833	103.8333	2020-05-29	33860	23	19631	wpro
34381	NaN	Singapore	1.2833	103.8333	2020-05-30	34366	23	20727	wpro
34646	NaN	Singapore	1.2833	103.8333	2020-05-31	34884	23	21699	wpro
34911	NaN	Singapore	1.2833	103.8333	2020-06-01	35292	24	22466	wpro

93 rows × 9 columns

{<oding:lab}

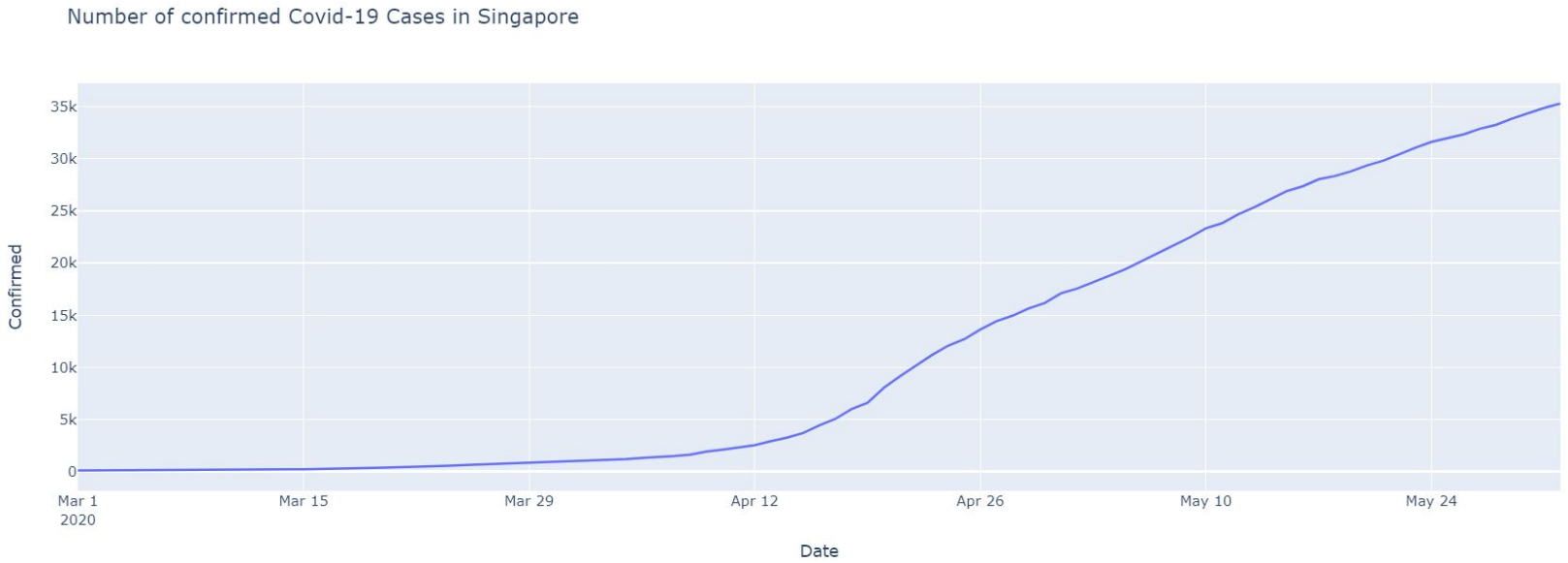
Shopee CODE LEAGUE 2020

Line Graph: Sample Code

- Plot a line graph showing the number of cases over time

```
1  import plotly.express as px
2  graph = px.line(country_date_range_df, x = "Date", y = "Confirmed")
3  graph.show()
```

Line Graph: Sample Output



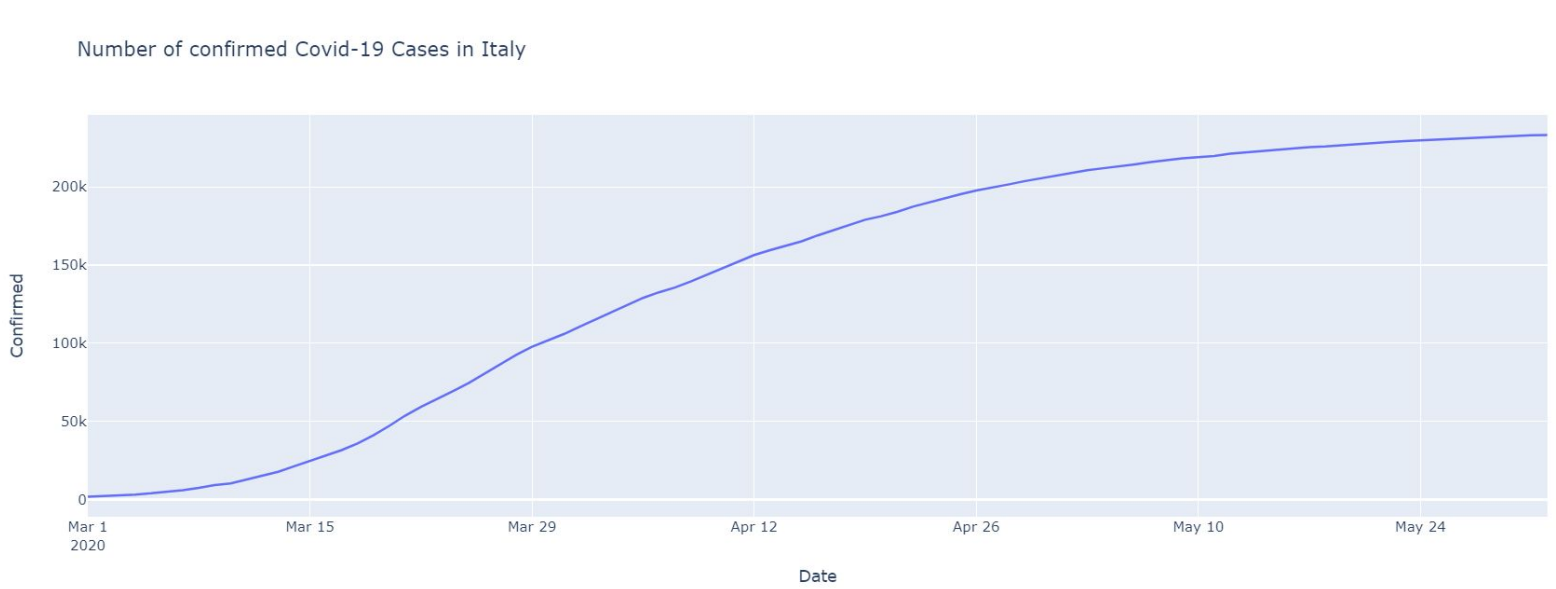
{<oding:lab}

Shopee CODE LEAGUE 2020

Country by Date Range (Demo/Practice - 6)

- Write a program that asks the user for the date range, and the country of interest
- Extract the relevant data frame and plot a suitable graph

Demo/Practice - 6: Sample Output



{<oding:lab}

Shopee CODE LEAGUE 2020

Checkpoint 6

- Every student must be able to:
  - Write a program to extract data based on a date range
  - Plot a line graph using the extracted data

- For students who are waiting, try the following:
  - Create a program that compares data from different countries over a date range in one graph

{<oding:lab}

Shopee CODE LEAGUE 2020

Multiple Countries in one Graph: Input Code

- Sample Code for input program

```
start_date = input("Select a start date in mm/dd/yy format: ")
end_date = input("Select a start date in mm/dd/yy format: ")
country = input("Select countries, separated by commas: ")
country_list = country.split(',')
df['Date'] = pd.to_datetime(df['Date'])
date_range_df = df[(df['Date'] >= start_date) & (df['Date'] <= end_date)]
country_date_range_df = date_range_df[date_range_df.Country.isin(country_list)]
country_date_range_df
```

{<oding:lab}

Shopee CODE LEAGUE 2020

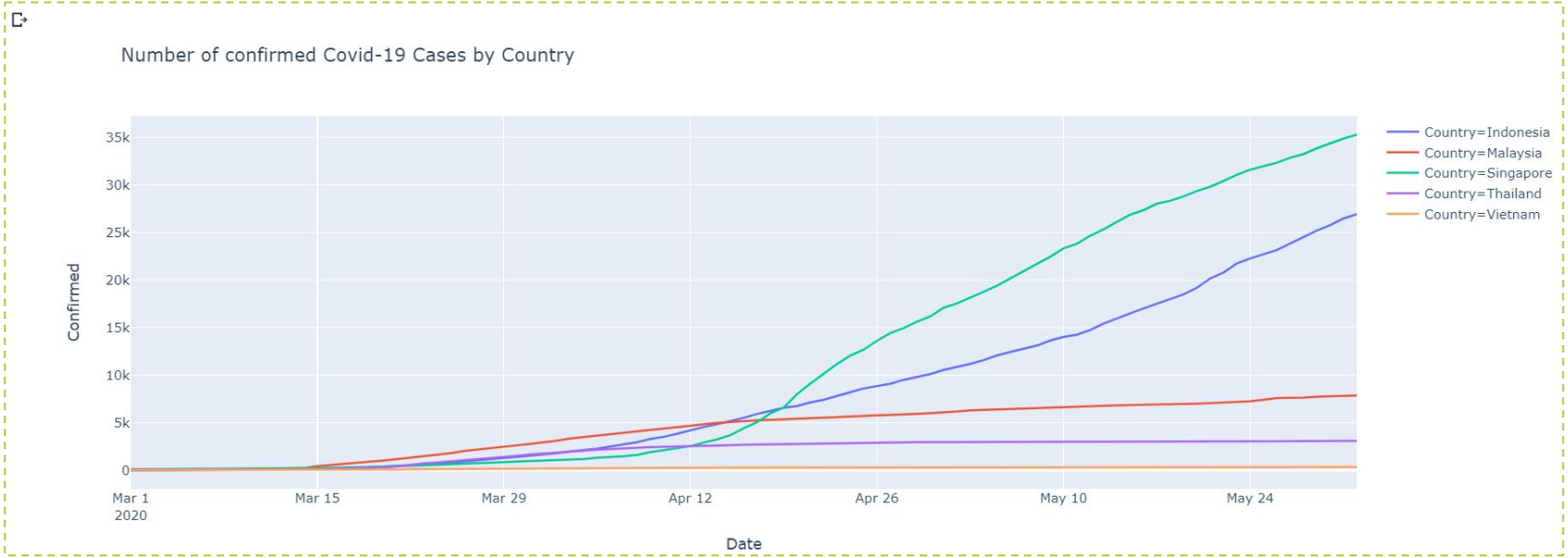
Multiple Countries in one Graph: Graph Code

- Sample Code for line graph

```
1  import plotly.express as px
2
3  graph = px.line(country_date_range_df, title = "Number of confirmed Covid-19 Cases by Country",
4  x = "Date", y = "Confirmed", color='Country')
5
6  graph.show()
```

{<oding:lab}

Multiple Countries in one Graph: Output



{<oding:lab}

Shopee CODE LEAGUE 2020

Some Interesting APIs

{<oding:lab}

Wikipedia API in Python

- Wikipedia is an external library that we can install
  - Implementation of Wikipedia API in Python to fetch information from a Wikipedia article
- To install Wikipedia in Python (Google Colab)

```
1   !pip install wikipedia
```

{<oding:lab}

Querying Wikipedia

- Now, let's fetch data from Wikipedia library in Python!
- We begin by importing the wikipedia library
- We then do a Wikipedia search for query
  - wikipedia.search('query', results=)

```
1  import wikipedia
2
3  resultList = wikipedia.search("Harry Potter", results = 15)
4
5  for item in resultList:
6      print(item)
```

{<oding:lab}

Getting a Wikipedia Page

- wikipedia.page()
  - Get a WikipediaPage object for the page with given title
  - Do you remember how to store the data into a dataframe?
    - How can we store all the information in a dataframe

```
1  import wikipedia
2
3  wikiPage = 'Harry Potter'
4  print(wikipedia.WikipediaPage(wikiPage).categories)
```

{<oding:lab}

Intelligent Guess

- wikipedia.suggest()
  - makes an intelligent guess based on what you are searching and returns result

```
1   import wikipedia
2   print(wikipedia.suggest('facebook'))
```

Summary of Page

- wikipedia.summary()
  - Returns summary of the page

```
1  import wikipedia
2  print(wikipedia.summary('google'))
```

- What should you include if:
  - you only want 2 sentences from the summary?
  - 200 characters of the summary?

{<oding:lab}

Search and Summary (Demo/Practice - 7)

- Making use of the functions we have learnt, write a program which returns you the summary of the page based on your input
  - Get your program to make an intelligent guess!

Checkpoint 7

- Every students must be able to:
  - Understand and utilise wikipedia built in functions
    - Wikipedia.search
    - Wikipedia.page
    - Wikipedia.suggest
    - Wikipedia.summary
- For students who are waiting, try the following:
  - Search up what else can wikipedia API do

{<oding:lab}

Random Pages

- wikipedia.random()
  - Get a list of random wikipedia pages
- Sample program containing a randomKnowledge() function that returns 10 sentences about a random topic on wikipedia

```
1   import wikipedia
2
3   # Function that returns 10 sentences about a random topic on wikipedia
4   def randomKnowledge():
5       page = wikipedia.random(pages = 1)
6       print(wikipedia.summary(page, sentences = 10))
7
8   print(randomKnowledge())
```

{<oding:lab}

Wikipedia Languages

- wikipedia.set_lang()
  - Set or change the language of an article
  - In this example, the website is translated to French
  - Search up the different language code

```
1  import wikipedia
2
3  wikipedia.set_lang('fr')
4  print(wikipedia.summary('google',sentences=1))
```

{<oding:lab}

Wikipedia URL

- Getting URL for the page

```
1   import wikipedia
2
3   complete_url = wikipedia.page('facebook')
4   print(complete_url.url)
```

Extraction of section

- Instead of printing the entire page, sometimes we are only interested in certain sections
  - In this case, we are interested in the plot of the movie

```
import wikipedia
print(wikipedia.WikipediaPage(title = 'Harry Potter and the Philosopher\'s Stone (film)').summary)

# get the section of a page. In this case the Plot description of Metropolis
section = wikipedia.WikipediaPage('Harry Potter and the Philosopher\'s Stone (film)').section('Plot')

# that will return fairly clean text, but the next line of code
# will help clean that up.
section = section.replace('\n','').replace("\'","")
```

{<oding:lab}

Return of section (Demo/Practice - 8)

- Write a program that allows the user to input a search term and return a section of the search term accordingly
- Thinking Process
  - First ask the user for an input
  - Conduct a search over wikipedia which returns a page related to the input
  - Return a section of the page

Checkpoint 8

- Every student must be able to :
  - Understand and utilise the functions of Wikipedia Library in Python
  - How to extract a certain section/part of the page

- For those who are waiting, try the following:
  - Read up the other available functions in Wikipedia Library
  - Did you find anything interesting?

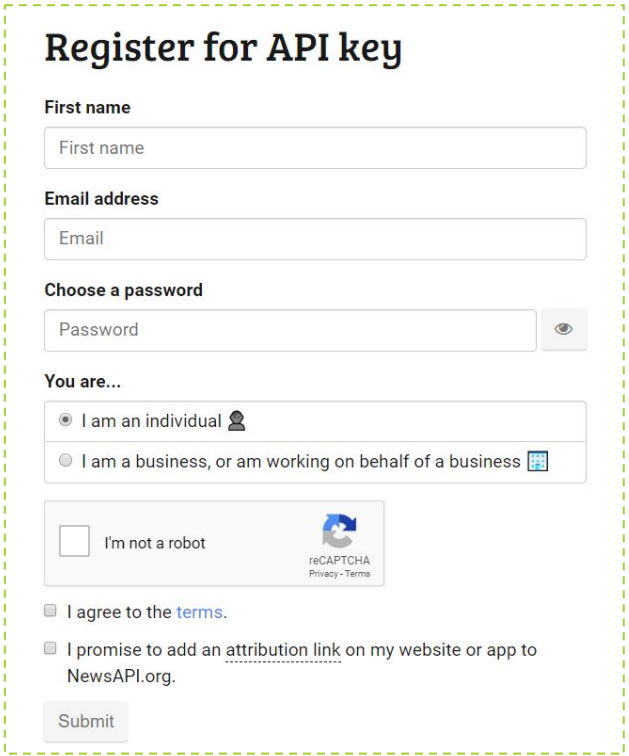
{<oding:lab}

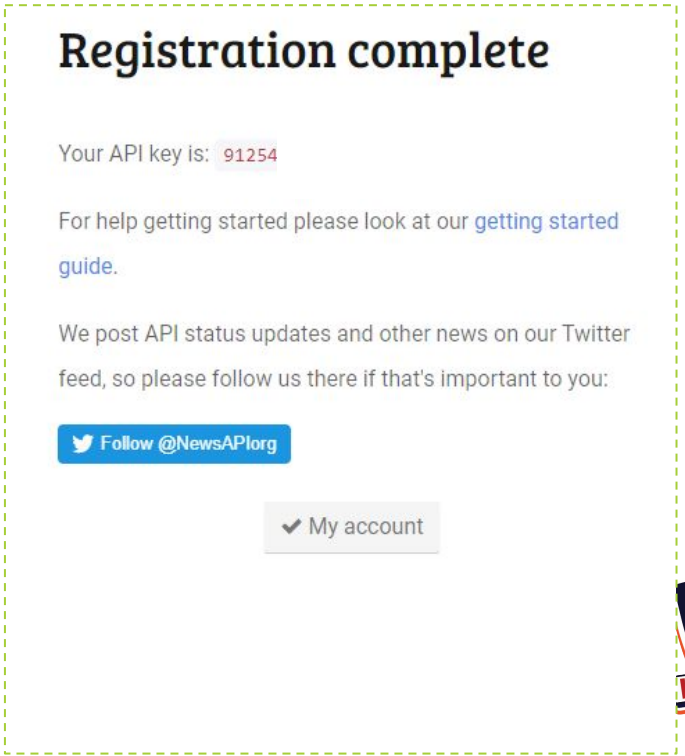
NewsAPI in Python

- A popular API used to search and fetch news articles from any website
- This API anyone can fetch top 10 heading line of news from any website
- Before we can make import this API, we have to request for access
  - https://newsapi.org/register

{<oding:lab}

Requesting for API key





Finding Your API Key

- After registering at NewsAPI.org, you can find your API key at: https://newsapi.org/docs/authentication

Find Your API Key (Demo/Practice - 9)

- Register for newsapi and get your api key
  - https://newsapi.org/register

Checkpoint 9

- Every student must be able to:
  - Successfully register for newsapi and get API key

- For students who are waiting, try the following:
  - Explore the API documentation for newsapi

{<oding:lab}

Installing NewsAPI on Google Colab

- Install the newsapi package using command prompt
  - “pip install newsapi”

{<oding:lab}

NewsAPI Endpoints

- NewsAPI offers three endpoints
  - Get_everything
    - for all the news articles from over 30,000 sources
  - Get_top-headlines
    - for the most important headlines per country and category
  - Get_sources
    - for information on the various sources

{<oding:lab}

NewsAPI First Walkthrough (1/2)

- The first thing we need to do is to import the newsapi library

```
1  # Import library
2  from newsapi import NewsApiClient
```

  - Then we set our API keys

```
4  # set api key
5  api = NewsApiClient(api_key='a6a26f3d119444f6910c2b208954d168')
```

  - Get all the headlines of the top articles from “bbc-news”

```
7  # Getting top articles from sources
8  top_articles = api.get_top_headlines(sources='bbc-news')
```

{<oding:lab}

NewsAPI First Walkthrough (2/2)

- Now that we have all the top articles, we can go through each of them and print the respective details

```
10  # Printing every single title in the list of articles
11  for i in top_articles['articles']:
12      print(i['title'])
13      print(i['description'])
14      print(i['url'])
15      print(i['publishedAt'])
16      print("\n\n\n")
```

{<oding:lab}

NewsAPI First Walkthrough (Demo/Practice - 10)

- Try out the code to print the top headlines from “bbc-news”
  - Look through the API and see what other sources you can print headlines from
  - Documentation for sources: https://newsapi.org/docs/endpoints/sources

{<oding:lab}

Checkpoint 10

- Every student must be able to:
  - Successfully get API keys
  - Print top headlines from “bbc-news” source

- For students who are waiting, try the following:
  - Find out the height of all your classmates and calculate the mean, median and mode!

{<oding:lab}

newsapi.get_everything()

- Search through articles from over 30,000 large and small news sources and blogs
  - This includes breaking news as well as lesser known articles

- Useful for discovering and analyzing articles
  - Can also be used to retrieve articles for display

{<oding:lab}

Bitcoin Headlines Example: Retrieving Articles

- We want to search various news sources for articles on bitcoin
  - Note that the “from_param” has a limit of 1 month for free accounts

Bitcoin Headlines Example: Retrieving Articles Sample Code

```
from newsapi import NewsApiClient

api = NewsApiClient(api_key='a6a26f3d119444f6910c2b208954d168')

# One week
all_articles = api.get_everything(q='bitcoin',
                                  sources='bbc-news,the-verge,cnn,reuters',
                                  from_param='2019-05-13',
                                  to='2019-06-13',
                                  language='en',
                                  sort_by='relevancy',
                                  page=2)

```

{<oding:lab}

Bitcoin Headlines Example: Printing Article Titles

- We then print all the article titles

```
14  for i in all_articles['articles']:
15      print(i['title'])
```

{<oding:lab}

newsapi.get_everything() (Demo/Practice - 11)

- Try out the code to print the article titles from a query of your choice
  - Try changing the sources or adding more sources
  - Vary the date range to see if it affects your results
    - This is done by changing “from_param” and “to”

Checkpoint 11

- Every student must be able to:
  - Print article titles related to a search term by using .get_everything()

- For students who are waiting, try the following:
  - Edit your program to allow the user to select the search term
  - Try to print other parts of the article apart from the ‘title’

{<oding:lab}

Summary

- Extracting data using API
- Filtering extracted data
- Real world examples
  - Data.gov.sg
  - Kaggle.com
- newsapi
- wikipedia

Your Feedback Matters!

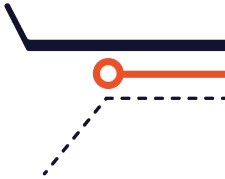
LIKE and follow us for more resources and tips!

@codinglabasia

{<oding:lab}

Shopee CODE LEAGUE 2020

Appendix



{<oding:lab}

Movie description

- Singaporeans watch movies regularly, but sometimes we might not know what the movie is about and who the casts are.
- Develop a program which prompts the user for a movie title
  - Returns the summary of the movie
  - Returns the casts of the movie
    - Add in more functions where your program decides if the movie is worth watch based on the overall rating

{<oding:lab}

Movie description?: Hints

- Before you start developing your program, here are some questions you can ask yourself
  - How do I perform a search?
  - Which wikipedia method should I use?
    - What kind of output will I get?