



Sentiment analysis in social networks

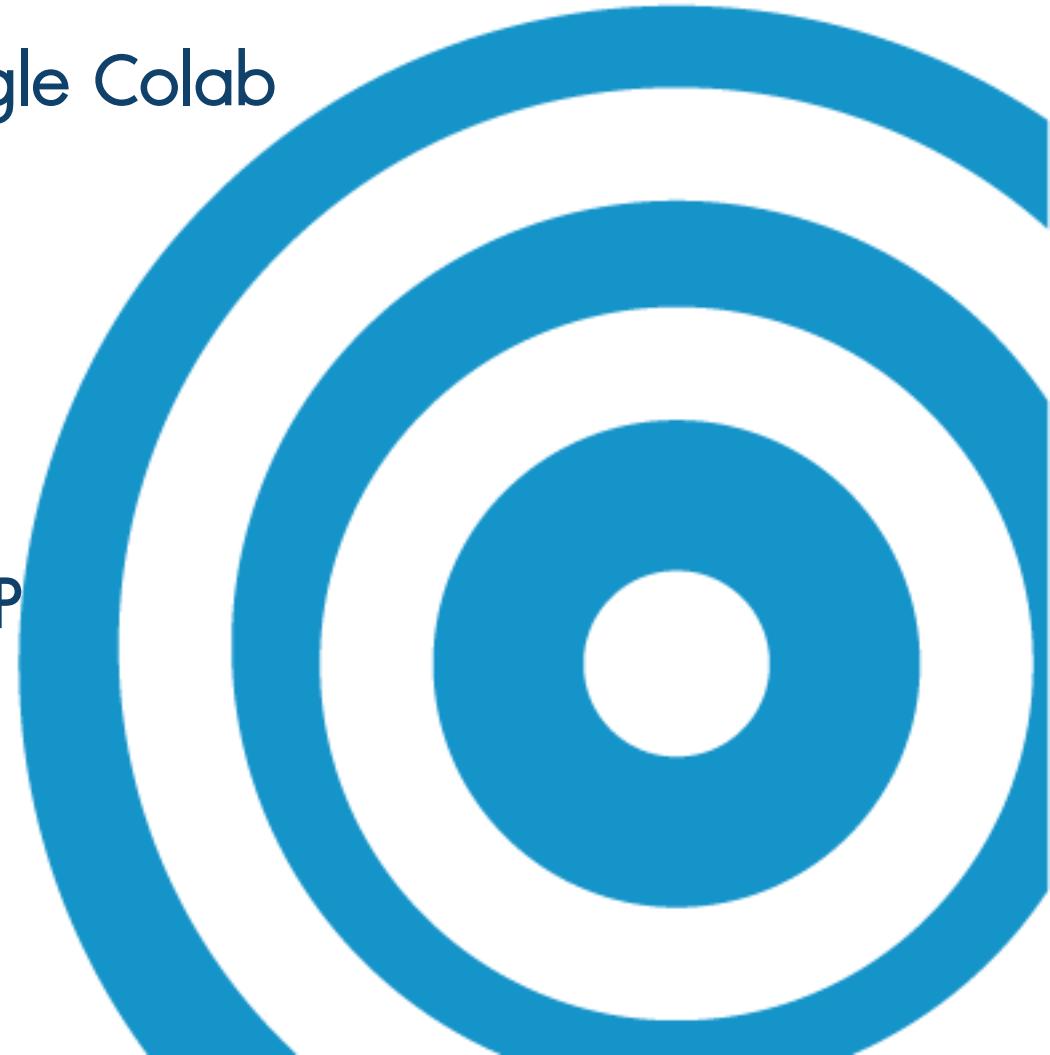
Chakrit Phain
Softnix Technology





Agenda Day 1

- การใช้งาน Python เป็องตันและการใช้ Google Colab
- การดึงข้อมูลจาก Facebook
- การดึงข้อมูลจาก Twitter
- การดึงข้อมูลจากเว็บ ด้วย Web Scraping
- การทำ Sentiment Analysis ด้วย PythaiNLP



Agenda Day 2

- หลักการ ทำ Sentiment Analysis
- การให้หัวข้อคำด้วย TF-IDF
- การทำ Sentiment Analysis ด้วย Naïve bayes + TF-IDF
- Overview Word2Vec สาเหตุและความสำคัญ
- การใช้ Word2Vec หาความใกล้เคียงของคำด้วย Projector Tensorflow
- ค้นหาคำสำคัญด้วย Cosine Similarity
- การใช้ Word2Vec กับการทำ Sentiment Analysis





การใช้งาน Google Colab





Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Table of contents X + Code + Text Copy to Drive RAM Disk Editing ^

What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
```

604800



krmonline / Sentiment-Analysis-in-Social-Networks

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Branch: master Sentiment-Analysis-in-Social-Networks / Getting_Started_With_Google_Colab.ipynb Find file Copy path

krmonline Created using Colaboratory 4819874 9 minutes ago

1 contributor

1349 lines (1349 sloc) | 111 KB Raw Blame History

[Open in Colab](#)

```
In [1]: 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=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0br4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3a0ob&response_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:
.....
Mounted at /content/gdrive

In [0]: !ls "/content/gdrive/My Drive/"

In [0]: !wget
#!unzip
!wget -cq https://s3.amazonaws.com/content.udacity-data.com/courses/nd188/flower_data.zip
!unzip -qq flower_data.zip

In [4]: !pip install python-twitter

Collecting python-twitter
  Downloading https://files.pythonhosted.org/packages/b3/a9/2eb36853d8ca49a70482e2332aa5082e09b3180391671101b1612e3aeaf1/python_twitter-3.5-py2.py3-none-any.whl (67kB)
    |████████| 71kB 2.1MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from python-twi
```

https://colab.research.google.com/github/krmonline/Sentiment-Analysis-in-Social-Networks/blob/master/Getting_Started_With_Google_Colab.ipynb



การใช้งาน Python เป็นอย่างตื้น





Python Crash Course in Colab

Crash Course.ipynb
PRO File Edit View Insert Runtime Tools Help Last edited on Sep 13, 2016 Share RAM Disk Editing

+ Code + Text Copy to Drive

↳ A Crash Course in Python for Scientists

Rick Muller, Sandia National Laboratories
version 0.6
This work is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).

Why Python?

Python is the programming language of choice for many scientists to a large degree because it offers a great deal of power to analyze and model scientific data with relatively little overhead in terms of learning, installation or development time. It is a language you can pick up in a weekend, and use for the rest of one's life.

The [Python Tutorial](#) is a great place to start getting a feel for the language. To complement this material, I taught a [Python Short Course](#) years ago to a group of computational chemists during a time that I was worried the field was moving too much in the direction of using canned software rather than developing one's own methods. I wanted to focus on what working scientists needed to be more productive: parsing output of other programs, building simple models, experimenting with object oriented programming, extending the language with C, and simple GUIs.

I'm trying to do something very similar here, to cut to the chase and focus on what scientists need. In the last year or so, the [IPython Project](#) has put together a notebook interface that I have found incredibly valuable. A large number of people have released very good IPython Notebooks that I have taken a huge amount of pleasure reading through. Some ones that I particularly like include:

- Rob Johansson's [excellent notebooks](#), including [Scientific Computing with Python](#) and [Computational Quantum Physics with QuTiP](#) lectures;
- [XKCD style graphs in matplotlib](#);
- [A collection of Notebooks for using IPython effectively](#)
- [A gallery of interesting IPython Notebooks](#)

I find IPython notebooks an easy way both to get important work done in my everyday job, as well as to communicate what I've done, how I've done it, and why it matters to my coworkers. I find myself endlessly sweeping the [IPython reddit](#) hoping someone will post a new notebook. In the interest of putting more notebooks out into the wild for other people to use and enjoy, I thought I would try to recreate some of what I was trying to get across in the original Python Short Course, updated by 15 years of Python, Numpy, Scipy, Matplotlib, and IPython development, as

<https://colab.research.google.com/github/rpmuller/PythonCrashCourse/blob/master/Crash%20Course.ipynb#scrollTo=pu0SrKPkyZa>



Python Cheat Sheet

Python Crash Course

Resources for Python Crash Course, from No Starch Press.

These are the resources for the first edition; the updated resources for the second edition are [here](#). I'd love to know what you think about Python Crash Course. Please consider taking a [brief survey](#). If you'd like to know when additional resources are available, you can sign up for [email notifications here](#).

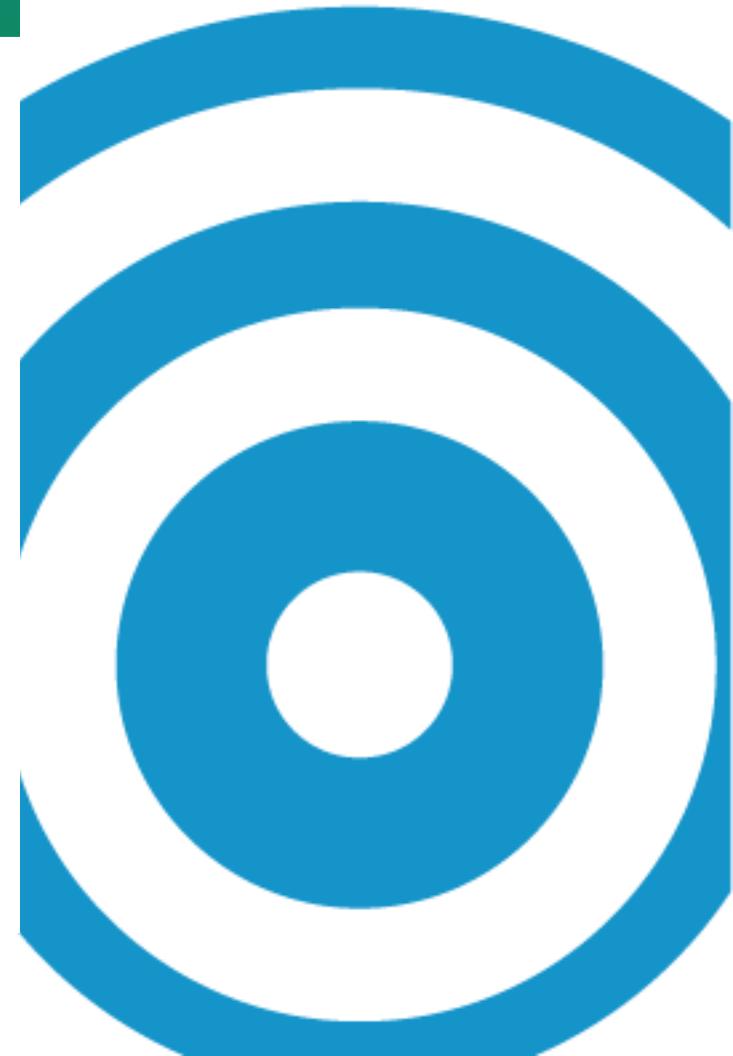
Python Crash Course - Cheat Sheets

Note: Updated cheat sheets [for the second edition are here](#). If you're working from the first edition of Python Crash Course, you should use the sheets described below. If you're working from the second edition, or any other Python resource, you should use the updated sheets.

A cheat sheet can be really helpful when you're trying a set of exercises related to a specific topic, or working on a project. Because you can only fit so much information on a single sheet of paper, most cheat sheets are a simple listing of syntax rules. This set of cheat sheets aims to remind you of syntax rules, but also remind you of important concepts as well.

You can download any individual cheat sheet, or download all the cheat sheets in [one document](#).

- [Beginner's Python Cheat Sheet](#)
 - Provides an overview of the basics of Python including variables, lists, dictionaries, functions, classes, and more.
- [Beginner's Python Cheat Sheet - Lists](#)
 - Focuses on lists: how to build and modify a list, access elements from a list, and loop through the values in a list. Also covers numerical lists, list comprehensions, tuples, and more.
- [Beginner's Python Cheat Sheet - Dictionaries](#)





- Beginner's Python
- Lists
- Dictionaries
- If Stagements and While Loops
- Files and Exceptions
- Matplotlib
- Pandas
- Numpy

Beginner's Python Cheat Sheet

Variables and Strings

Variables are used to store values. A string is a series of characters, surrounded by single or double quotes.

Hello world

```
print("Hello world!")
```

Hello world with a variable

```
msg = "Hello world!"  
print(msg)
```

Concatenation (combining strings)

```
first_name = 'albert'  
last_name = 'einstein'  
full_name = first_name + ' ' + last_name  
print(full_name)
```

Lists

A list stores a series of items in a particular order. You access items using an index, or within a loop.

Make a list

```
bikes = ['trek', 'redline', 'giant']
```

Get the first item in a list

```
first_bike = bikes[0]
```

Get the last item in a list

```
last_bike = bikes[-1]
```

Looping through a list

```
for bike in bikes:  
    print(bike)
```

Adding items to a list

```
bikes = []  
bikes.append('trek')  
bikes.append('redline')  
bikes.append('giant')
```

Making numerical lists

```
squares = []  
for x in range(1, 11):  
    squares.append(x**2)
```

Lists (cont.)

List comprehensions

```
squares = [x**2 for x in range(1, 11)]
```

Slicing a list

```
finishers = ['sam', 'bob', 'ada', 'bea']  
first_two = finishers[:2]
```

Copying a list

```
copy_of_bikes = bikes[:]
```

Tuples

Tuples are similar to lists, but the items in a tuple can't be modified.

Making a tuple

```
dimensions = (1920, 1080)
```

If statements

If statements are used to test for particular conditions and respond appropriately.

Conditional tests

equals	x == 42
not equal	x != 42
greater than	x > 42
or equal to	x >= 42
less than	x < 42
or equal to	x <= 42

Conditional test with lists

```
'trek' in bikes  
'surly' not in bikes
```

Assigning boolean values

```
game_active = True  
can_edit = False
```

A simple if test

```
if age >= 18:  
    print("You can vote!")
```

If-elif-else statements

```
if age < 4:  
    ticket_price = 0  
elif age < 18:  
    ticket_price = 10  
else:  
    ticket_price = 15
```

Dictionaries

Dictionaries store connections between pieces of information. Each item in a dictionary is a key-value pair.

A simple dictionary

```
alien = {'color': 'green', 'points': 5}
```

Accessing a value

```
print("The alien's color is " + alien['color'])
```

Adding a new key-value pair

```
alien['x_position'] = 0
```

Looping through all key-value pairs

```
fav_numbers = {'eric': 17, 'ever': 4}  
for name, number in fav_numbers.items():  
    print(name + ' loves ' + str(number))
```

Looping through all keys

```
fav_numbers = {'eric': 17, 'ever': 4}  
for name in fav_numbers.keys():  
    print(name + ' loves a number')
```

Looping through all the values

```
fav_numbers = {'eric': 17, 'ever': 4}  
for number in fav_numbers.values():  
    print(str(number) + ' is a favorite')
```

User input

Your programs can prompt the user for input. All input is stored as a string.

Prompting for a value

```
name = input("What's your name? ")  
print("Hello, " + name + "!")
```

Prompting for numerical input

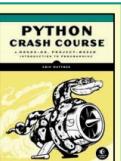
```
age = input("How old are you? ")  
age = int(age)
```

```
pi = input("What's the value of pi? ")  
pi = float(pi)
```

Python Crash Course

Covers Python 3 and Python 2

nostarchpress.com/pythoncrashcourse





- Beginner's Python
- Lists
- Dictionaries
- If Stagements and While Loops
- Files and Exceptions
- Matplotlib
- Pandas
- Numpy

Beginner's Python Cheat Sheet - Lists

What are lists?

A list stores a series of items in a particular order. Lists allow you to store sets of information in one place, whether you have just a few items or millions of items. Lists are one of Python's most powerful features readily accessible to new programmers, and they tie together many important concepts in programming.

Defining a list

Use square brackets to define a list, and use commas to separate individual items in the list. Use plural names for lists, to make your code easier to read.

Making a list

```
users = ['val', 'bob', 'mia', 'ron', 'ned']
```

Accessing elements

Individual elements in a list are accessed according to their position, called the index. The index of the first element is 0, the index of the second element is 1, and so forth. Negative indices refer to items at the end of the list. To get a particular element, write the name of the list and then the index of the element in square brackets.

Getting the first element

```
first_user = users[0]
```

Getting the second element

```
second_user = users[1]
```

Getting the last element

```
newest_user = users[-1]
```

Modifying individual items

Once you've defined a list, you can change individual elements in the list. You do this by referring to the index of the item you want to modify.

Changing an element

```
users[0] = 'valerie'  
users[-2] = 'ronald'
```

Adding elements

You can add elements to the end of a list, or you can insert them wherever you like in a list.

Adding an element to the end of the list

```
users.append('amy')
```

Starting with an empty list

```
users = []  
users.append('val')  
users.append('bob')  
users.append('mia')
```

Inserting elements at a particular position

```
users.insert(0, 'joe')  
users.insert(3, 'bea')
```

Removing elements

You can remove elements by their position in a list, or by the value of the item. If you remove an item by its value, Python removes only the first item that has that value.

Deleting an element by its position

```
del users[-1]
```

Removing an item by its value

```
users.remove('mia')
```

Popping elements

If you want to work with an element that you're removing from the list, you can "pop" the element. If you think of the list as a stack of items, pop() takes an item off the top of the stack. By default pop() returns the last element in the list, but you can also pop elements from any position in the list.

Pop the last item from a list

```
most_recent_user = users.pop()  
print(most_recent_user)
```

Pop the first item in a list

```
first_user = users.pop(0)  
print(first_user)
```

List length

The len() function returns the number of items in a list.

Find the length of a list

```
num_users = len(users)  
print("We have " + str(num_users) + " users.")
```

Sorting a list

The sort() method changes the order of a list permanently. The sorted() function returns a copy of the list, leaving the original list unchanged. You can sort the items in a list in alphabetical order, or reverse alphabetical order. You can also reverse the original order of the list. Keep in mind that lowercase and uppercase letters may affect the sort order.

Sorting a list permanently

```
users.sort()
```

Sorting a list permanently in reverse alphabetical order

```
users.sort(reverse=True)
```

Sorting a list temporarily

```
print(sorted(users))  
print(sorted(users, reverse=True))
```

Reversing the order of a list

```
users.reverse()
```

Looping through a list

Lists can contain millions of items, so Python provides an efficient way to loop through all the items in a list. When you set up a loop, Python pulls each item from the list one at a time and stores it in a temporary variable, which you provide a name for. This name should be the singular version of the list name.

The indented block of code makes up the body of the loop, where you can work with each individual item. Any lines that are not indented run after the loop is completed.

Printing all items in a list

```
for user in users:  
    print(user)
```

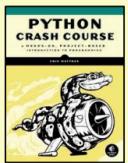
Printing a message for each item, and a separate message afterwards

```
for user in users:  
    print("Welcome, " + user + "!")  
  
print("Welcome, we're glad to see you all!")
```

Python Crash Course

Covers Python 3 and Python 2

nostarchpress.com/pythoncrashcourse





- Beginner's Python
- Lists
- Dictionaries
- If Statements and While Loops
- Files and Exceptions
- Matplotlib
- Pandas
- Numpy

Beginner's Python Cheat Sheet — Dictionaries

What are dictionaries?

Python's dictionaries allow you to connect pieces of related information. Each piece of information in a dictionary is stored as a key-value pair. When you provide a key, Python returns the value associated with that key. You can loop through all the key-value pairs, all the keys, or all the values.

Defining a dictionary

Use curly braces to define a dictionary. Use colons to connect keys and values, and use commas to separate individual key-value pairs.

Making a dictionary

```
alien_0 = {'color': 'green', 'points': 5}
```

Accessing values

To access the value associated with an individual key give the name of the dictionary and then place the key in a set of square brackets. If the key you're asking for is not in the dictionary, an error will occur.

You can also use the `get()` method, which returns `None` instead of an error if the key doesn't exist. You can also specify a default value to use if the key is not in the dictionary.

Getting the value associated with a key

```
alien_0 = {'color': 'green', 'points': 5}

print(alien_0['color'])
print(alien_0['points'])
```

Getting the value with `get()`

```
alien_0 = {'color': 'green'}

alien_color = alien_0.get('color')
alien_points = alien_0.get('points', 0)

print(alien_color)
print(alien_points)
```

Adding new key-value pairs

You can store as many key-value pairs as you want in a dictionary, until your computer runs out of memory. To add a new key-value pair to an existing dictionary give the name of the dictionary and the new key in square brackets, and set it equal to the new value.

This also allows you to start with an empty dictionary and add key-value pairs as they become relevant.

Adding a key-value pair

```
alien_0 = {'color': 'green', 'points': 5}

alien_0['x'] = 0
alien_0['y'] = 25
alien_0['speed'] = 1.5
```

Adding to an empty dictionary

```
alien_0 = {}
alien_0['color'] = 'green'
alien_0['points'] = 5
```

Modifying values

You can modify the value associated with any key in a dictionary. To do so give the name of the dictionary and enclose the key in square brackets, then provide the new value for that key.

Modifying values in a dictionary

```
alien_0 = {'color': 'green', 'points': 5}
print(alien_0)

# Change the alien's color and point value.
alien_0['color'] = 'yellow'
alien_0['points'] = 10
print(alien_0)
```

Removing key-value pairs

You can remove any key-value pair you want from a dictionary. To do so use the `del` keyword and the dictionary name, followed by the key in square brackets. This will delete the key and its associated value.

Deleting a key-value pair

```
alien_0 = {'color': 'green', 'points': 5}
print(alien_0)

del alien_0['points']
print(alien_0)
```

Visualizing dictionaries

Try running some of these examples on pythontutor.com.

Looping through a dictionary

You can loop through a dictionary in three ways: you can loop through all the key-value pairs, all the keys, or all the values.

A dictionary only tracks the connections between keys and values; it doesn't track the order of items in the dictionary. If you want to process the information in order, you can sort the keys in your loop.

Looping through all key-value pairs

```
# Store people's favorite languages.
fav_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
}
```

```
# Show each person's favorite language.
for name, language in fav_languages.items():
    print(name + ": " + language)
```

Looping through all the keys

```
# Show everyone who's taken the survey.
for name in fav_languages.keys():
    print(name)
```

Looping through all the values

```
# Show all the languages that have been chosen.
for language in fav_languages.values():
    print(language)
```

Looping through all the keys in order

```
# Show each person's favorite language,
# in order by the person's name.
for name in sorted(fav_languages.keys()):
    print(name + ": " + language)
```

Dictionary length

You can find the number of key-value pairs in a dictionary.

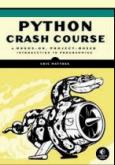
Finding a dictionary's length

```
num_responses = len(fav_languages)
```

Python Crash Course

Covers Python 3 and Python 2

nostarchpress.com/pythoncrashcourse





- Beginner's Python
- Lists
- Dictionaries
- If Stagements and While Loops
- Files and Exceptions
- Matplotlib
- Pandas
- Numpy

Beginner's Python Cheat Sheet — If Statements and While Loops

What are if statements? What are while loops?

If statements allow you to examine the current state of a program and respond appropriately to that state. You can write a simple if statement that checks one condition, or you can create a complex series of if statements that identify the exact conditions you're looking for.

While loops run as long as certain conditions remain true. You can use while loops to let your programs run as long as your users want them to.

Conditional Tests

A conditional test is an expression that can be evaluated as True or False. Python uses the values True and False to decide whether the code in an if statement should be executed.

Checking for equality

A single equal sign assigns a value to a variable. A double equal sign (==) checks whether two values are equal.

```
>>> car = 'bmw'  
>>> car == 'bmw'  
True  
>>> car = 'audi'  
>>> car == 'bmw'  
False
```

Ignoring case when making a comparison

```
>>> car = 'Audi'  
>>> car.lower() == 'audi'  
True
```

Checking for inequality

```
>>> topping = 'mushrooms'  
>>> topping != 'anchovies'  
True
```

Numerical comparisons

Testing numerical values is similar to testing string values.

Testing equality and inequality

```
>>> age = 18  
>>> age == 18  
True  
>>> age != 18  
False
```

Comparison operators

```
>>> age = 19  
>>> age < 21  
True  
>>> age <= 21  
True  
>>> age > 21  
False  
>>> age >= 21  
False
```

Checking multiple conditions

You can check multiple conditions at the same time. The and operator returns True if all the conditions listed are True. The or operator returns True if any condition is True.

Using and to check multiple conditions

```
>>> age_0 = 22  
>>> age_1 = 18  
>>> age_0 >= 21 and age_1 >= 21  
False  
>>> age_1 = 23  
>>> age_0 >= 21 and age_1 >= 21  
True
```

Using or to check multiple conditions

```
>>> age_0 = 22  
>>> age_1 = 18  
>>> age_0 >= 21 or age_1 >= 21  
True  
>>> age_0 = 18  
>>> age_0 >= 21 or age_1 >= 21  
False
```

Boolean values

A boolean value is either True or False. Variables with boolean values are often used to keep track of certain conditions within a program.

Simple boolean values

```
game_active = True  
can_edit = False
```

If statements

Several kinds of if statements exist. Your choice of which to use depends on the number of conditions you need to test. You can have as many elif blocks as you need, and the else block is always optional.

Simple if statement

```
age = 19  
  
if age >= 18:  
    print("You're old enough to vote!")
```

If-else statements

```
age = 17  
  
if age >= 18:  
    print("You're old enough to vote!")  
else:  
    print("You can't vote yet.")
```

The if-elif-else chain

```
age = 12  
  
if age < 4:  
    price = 0  
elif age < 18:  
    price = 5  
else:  
    price = 10  
  
print("Your cost is $" + str(price) + ".")
```

Conditional tests with lists

You can easily test whether a certain value is in a list. You can also test whether a list is empty before trying to loop through the list.

Testing if a value is in a list

```
>>> players = ['al', 'bea', 'cyn', 'dale']  
>>> 'al' in players  
True  
>>> 'eric' in players  
False
```

Python Crash Course

Covers Python 3 and Python 2

nostarchpress.com/pythoncrashcourse



- Beginner's Python
- Lists
- Dictionaries
- If Stagements and While Loops
- **Files and Exceptions**
- Matplotlib
- Pandas
- Numpy

Beginner's Python Cheat Sheet — Files and Exceptions

What are files? What are exceptions?

Your programs can read information from files, and they can write data to files. Reading from files allows you to work with a wide variety of information; writing to files allows users to pick up where they left off the next time they run your program. You can write text to files, and you can store Python structures such as lists in data files.

Exceptions are special objects that help your programs respond to errors in appropriate ways. For example if your program tries to open a file that doesn't exist, you can use exceptions to display an informative error message instead of having the program crash.

Reading from a file

To read from a file your program needs to open the file and then read the contents of the file. You can read the entire contents of the file at once, or read the file line by line. The with statement makes sure the file is closed properly when the program has finished accessing the file.

Reading an entire file at once

```
filename = 'siddhartha.txt'

with open(filename) as f_obj:
    contents = f_obj.read()

print(contents)
```

Reading line by line

Each line that's read from the file has a newline character at the end of the line, and the print function adds its own newline character. The rstrip() method gets rid of the extra blank lines this would result in when printing to the terminal.

```
filename = 'siddhartha.txt'

with open(filename) as f_obj:
    for line in f_obj:
        print(line.rstrip())
```

Reading from a file (cont.)

Storing the lines in a list

```
filename = 'siddhartha.txt'

with open(filename) as f_obj:
    lines = f_obj.readlines()

for line in lines:
    print(line.rstrip())
```

Writing to a file

Passing the 'w' argument to open() tells Python you want to write to the file. Be careful; this will erase the contents of the file if it already exists. Passing the 'a' argument tells Python you want to append to the end of an existing file.

Writing to an empty file

```
filename = 'programming.txt'

with open(filename, 'w') as f:
    f.write("I love programming!")
```

Writing multiple lines to an empty file

```
filename = 'programming.txt'

with open(filename, 'w') as f:
    f.write("I love programming!\n")
    f.write("I love creating new games.\n")
```

Appending to a file

```
filename = 'programming.txt'

with open(filename, 'a') as f:
    f.write("I also love working with data.\n")
    f.write("I love making apps as well.\n")
```

File paths

When Python runs the open() function, it looks for the file in the same directory where the program that's being executed is stored. You can open a file from a subfolder using a relative path. You can also use an absolute path to open any file on your system.

Opening a file from a subfolder

```
f_path = "text_files/alice.txt"

with open(f_path) as f_obj:
    lines = f_obj.readlines()

for line in lines:
    print(line.rstrip())
```

File paths (cont.)

Opening a file using an absolute path

```
f_path = "/home/ehmatthes/books/alice.txt"

with open(f_path) as f_obj:
    lines = f_obj.readlines()
```

Opening a file on Windows

Windows will sometimes interpret forward slashes incorrectly. If you run into this, use backslashes in your file paths.

```
f_path = "C:\Users\ehmatthes\books\alice.txt"
```

```
with open(f_path) as f_obj:
    lines = f_obj.readlines()
```

The try-except block

When you think an error may occur, you can write a try-except block to handle the exception that might be raised. The try block tells Python to try running some code, and the except block tells Python what to do if the code results in a particular kind of error.

Handling the ZeroDivisionError exception

```
try:
    print(5/0)
except ZeroDivisionError:
    print("You can't divide by zero!")
```

Handling the FileNotFoundError exception

```
f_name = 'siddhartha.txt'

try:
    with open(f_name) as f_obj:
        lines = f_obj.readlines()
except FileNotFoundError:
    msg = "Can't find file {}".format(f_name)
    print(msg)
```

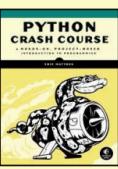
Knowing which exception to handle

It can be hard to know what kind of exception to handle when writing code. Try writing your code without a try block, and make it generate an error. The traceback will tell you what kind of exception your program needs to handle.

Python Crash Course

Covers Python 3 and Python 2

nostarchpress.com/pythoncrashcourse





- Beginner's Python
- Lists
- Dictionaries
- If Stagements and While Loops
- Files and Exceptions
- Matplotlib
- Pandas
- Numpy

Beginner's Python Cheat Sheet — matplotlib

What is matplotlib?

Data visualization involves exploring data through visual representations. The matplotlib package helps you make visually appealing representations of the data you're working with. matplotlib is extremely flexible; these examples will help you get started with a few simple visualizations.

Installing matplotlib

matplotlib runs on all systems, but setup is slightly different depending on your OS. If the minimal instructions here don't work for you, see the more detailed instructions at <http://ehmatthes.github.io/pcc/>. You should also consider installing the Anaconda distribution of Python from <https://continuum.io/downloads/>, which includes matplotlib.

matplotlib on Linux

```
$ sudo apt-get install python3-matplotlib
```

matplotlib on OS X

Start a terminal session and enter `import matplotlib` to see if it's already installed on your system. If not, try this command:

```
$ pip install --user matplotlib
```

matplotlib on Windows

You first need to install Visual Studio, which you can do from <https://dev.windows.com/>. The Community edition is free. Then go to <https://pypi.python.org/pypi/matplotlib/> or <http://www.ffd.uic.edu/~gohlke/pythonlibs/#matplotlib> and download an appropriate installer file.

Line graphs and scatter plots

Making a line graph

```
import matplotlib.pyplot as plt

x_values = [0, 1, 2, 3, 4, 5]
squares = [0, 1, 4, 9, 16, 25]
plt.plot(x_values, squares)
plt.show()
```

Line graphs and scatter plots (cont.)

Making a scatter plot

The `scatter()` function takes a list of x values and a list of y values, and a variety of optional arguments. The `s=10` argument controls the size of each point.

```
import matplotlib.pyplot as plt

x_values = list(range(1000))
squares = [x**2 for x in x_values]
plt.scatter(x_values, squares, s=10)
plt.show()
```

Customizing plots

Plots can be customized in a wide variety of ways. Just about any element of a plot can be customized.

Adding titles and labels, and scaling axes

```
import matplotlib.pyplot as plt

x_values = list(range(1000))
squares = [x**2 for x in x_values]
plt.scatter(x_values, squares, s=10)

plt.title("Square Numbers", fontsize=24)
plt.xlabel("Value", fontsize=18)
plt.ylabel("Square of Value", fontsize=18)
plt.tick_params(axis='both', which='major',
                labelsize=14)
plt.axis([0, 1100, 0, 1100000])

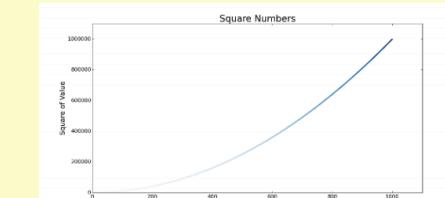
plt.show()
```

Using a colormap

A colormap varies the point colors from one shade to another, based on a certain value for each point. The value used to determine the color of each point is passed to the `c` argument, and the `cmap` argument specifies which colormap to use.

The `edgecolor='none'` argument removes the black outline from each point.

```
plt.scatter(x_values, squares, c=squares,
            cmap=plt.cm.Blues, edgecolor='none',
            s=10)
```



Customizing plots (cont.)

Emphasizing points

You can plot as much data as you want on one plot. Here we replot the first and last points larger to emphasize them.

```
import matplotlib.pyplot as plt

x_values = list(range(1000))
squares = [x**2 for x in x_values]
plt.scatter(x_values, squares, c=squares,
            cmap=plt.cm.Blues, edgecolor='none',
            s=10)

plt.scatter(x_values[0], squares[0], c='green',
            edgecolor='none', s=100)
plt.scatter(x_values[-1], squares[-1], c='red',
            edgecolor='none', s=100)

plt.title("Square Numbers", fontsize=24)
--snip--
```

Removing axes

You can customize or remove axes entirely. Here's how to access each axis, and hide it.

```
plt.axes().get_xaxis().set_visible(False)
plt.axes().get_yaxis().set_visible(False)
```

Setting a custom figure size

You can make your plot as big or small as you want. Before plotting your data, add the following code. The `dpi` argument is optional; if you don't know your system's resolution you can omit the argument and adjust the `figsize` argument accordingly.

```
plt.figure(dpi=128, figsize=(10, 6))
```

Saving a plot

The `matplotlib` viewer has an interactive save button, but you can also save your visualizations programmatically. To do so, replace `plt.show()` with `plt.savefig()`. The `bbox_inches='tight'` argument trims extra whitespace from the plot.

```
plt.savefig('squares.png', bbox_inches='tight')
```

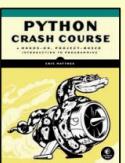
Online resources

The `matplotlib` gallery and documentation are at <http://matplotlib.org/>. Be sure to visit the examples, gallery, and pyplot links.

Python Crash Course

Covers Python 3 and Python 2

nostarchpress.com/pythoncrashcourse





Data Science Cheat Sheet

Pandas

- Beginner's Python
- Lists
- Dictionaries
- If Stagements and While Loops
- Files and Exceptions
- Matplotlib
- Pandas
- Numpy

KEY
We'll use shorthand in this cheat sheet
df - A pandas DataFrame object
s - A pandas Series object

IMPORTS
Import these to start
import pandas as pd
import numpy as np

IMPORTING DATA
pd.read_csv(filename) - From a CSV file
pd.read_table(filename) - From a delimited text file (like TSV)
pd.read_excel(filename) - From an Excel file
pd.read_sql(query, connection_object) - Reads from a SQL table/database
pd.read_json(json_string) - Reads from a JSON-formatted string, URL or file.
pd.read_html(url) - Parses an html URL, string or file and extracts tables to a list of dataframes
pd.read_clipboard() - Takes the contents of your clipboard and passes it to read_table()
pd.DataFrame(dict) - From a dict, keys for columns names, values for data as lists

EXPORTING DATA
df.to_csv(filename) - Writes to a CSV file
df.to_excel(filename) - Writes to an Excel file
df.to_sql(table_name, connection_object) - Writes to a SQL table
df.to_json(filename) - Writes to a file in JSON format
df.to_html(filename) - Saves as an HTML table
df.to_clipboard() - Writes to the clipboard

CREATE TEST OBJECTS
Useful for testing
pd.DataFrame(np.random.rand(20,5)) - 5 columns and 20 rows of random floats
pd.Series(my_list) - Creates a series from an iterable my_list
df.index = pd.date_range('1900/1/30', periods=df.shape[0]) - Adds a date index

VIEWING/INSPECTING DATA
df.head(n) - First n rows of the DataFrame
df.tail(n) - Last n rows of the DataFrame
df.shape - Number of rows and columns
df.info() - Index, Datatype and Memory information
df.describe() - Summary statistics for numerical columns
s.value_counts(dropna=False) - Views unique values and counts
df.apply(pd.Series.value_counts) - Unique values and counts for all columns

SELECTION
df[col] - Returns column with label col as Series
df[[col1, col2]] - Returns Columns as a new DataFrame
s.iloc[0] - Selection by position
s.loc[0] - Selection by index
df.iloc[0, :] - First row
df.iloc[0, 0] - First element of first column

DATA CLEANING
df.columns = ['a', 'b', 'c'] - Renames columns
pd.isnull() - Checks for null Values, Returns Boolean Array
pd.notnull() - Opposite of s.isnull()
df.dropna() - Drops all rows that contain null values
df.dropna(axis=1) - Drops all columns that contain null values
df.dropna(axis=1, thresh=n) - Drops all rows have less than n non null values
df.fillna(x) - Replaces all null values with x
s.fillna(s.mean()) - Replaces all null values with the mean (mean can be replaced with almost any function from the statistics section)
s.astype(float) - Converts the datatype of the series to float

s.replace(1, 'one') - Replaces all values equal to 1 with 'one'
s.replace([1,3], ['one', 'three']) - Replaces all 1 with 'one' and 3 with 'three'
df.rename(columns=lambda x: x + 1) - Mass renaming of columns
df.rename(columns={'old_name': 'new_name'}) - Selective renaming
df.set_index('column_one') - Changes the index
df.rename(index=lambda x: x + 1) - Mass renaming of index

FILTER, SORT, & GROUPBY
df[df[col] > 0.5] - Rows where the col column is greater than 0.5
df[(df[col] > 0.5) & (df[col] < 0.7)] - Rows where 0.7 > col > 0.5
df.sort_values(col1) - Sorts values by col1 in ascending order
df.sort_values(col2, ascending=False) - Sorts values by col2 in descending order
df.sort_values([col1, col2], ascending=[True, False]) - Sorts values by

col1 in ascending order then col2 in descending order
df.groupby(col) - Returns a groupby object for values from one column
df.groupby([col1, col2]) - Returns a groupby object values from multiple columns
df.groupby(col1)[col2].mean() - Returns the mean of the values in col2, grouped by the values in col1 (mean can be replaced with almost any function from the statistics section)
df.pivot_table(index=col1, values=[col2, col3], aggfunc='mean') - Creates a pivot table that groups by col1 and calculates the mean of col2 and col3
df.groupby(col1).agg(np.mean) - Finds the average across all columns for every unique column 1 group
df.apply(np.mean) - Applies a function across each column
df.apply(np.max, axis=1) - Applies a function across each row

JOIN/COMBINE
df1.append(df2) - Adds the rows in df1 to the end of df2 (columns should be identical)
pd.concat([df1, df2], axis=1) - Adds the columns in df1 to the end of df2 (rows should be identical)
df1.join(df2, on=col1, how='inner') - SQL-style joins the columns in df1 with the columns on df2 where the rows for col1 have identical values, how can be one of 'left', 'right', 'outer', 'inner'

STATISTICS
These can all be applied to a series as well.
df.describe() - Summary statistics for numerical columns
df.mean() - Returns the mean of all columns
df.corr() - Returns the correlation between columns in a DataFrame
df.count() - Returns the number of non-null values in each DataFrame column
df.max() - Returns the highest value in each column
df.min() - Returns the lowest value in each column
df.median() - Returns the median of each column
df.std() - Returns the standard deviation of each column



- Beginner's Python
- Lists
- Dictionaries
- If Stagements and While Loops
- Files and Exceptions
- Matplotlib
- Pandas
- Numpy

KEY

We'll use shorthand in this cheat sheet
`arr` - A numpy Array object

IMPORTS

Import these to start
`import numpy as np`

IMPORTING/EXPORTING

`np.loadtxt('file.txt')` - From a text file
`np.genfromtxt('file.csv', delimiter=',')` - From a CSV file
`np.savetxt('file.txt', arr, delimiter=' ')` - Writes to a text file
`np.savetxt('file.csv', arr, delimiter=',')` - Writes to a CSV file

CREATING ARRAYS

`np.array([1,2,3])` - One dimensional array
`np.array([(1,2,3),(4,5,6)])` - Two dimensional array
`np.zeros(3)` - 1D array of length 3 all values 0
`np.ones((3,4))` - 3x4 array with all values 1
`np.eye(5)` - 5x5 array of 0 with 1 on diagonal (identity matrix)
`np.linspace(0, 100, 6)` - Array of 6 evenly divided values from 0 to 100
`np.arange(0, 10, 3)` - Array of values from 0 to less than 10 with step 3 (eg [0,3,6,9])
`np.full((2,3), 8)` - 2x3 array with all values 8
`np.random.rand(4,5)` - 4x5 array of random floats between 0-1
`np.random.rand(5,7)*100` - 6x7 array of random floats between 0-100
`np.random.randint(5, size=(2,3))` - 2x3 array with random ints between 0-4

INSPECTING PROPERTIES

`arr.size` - Returns number of elements in arr
`arr.shape` - Returns dimensions of arr (rows, columns)
`arr.dtype` - Returns type of elements in arr
`arr.astype(dtype)` - Convert arr elements to type dtype
`arr.tolist()` - Convert arr to a Python list
`np.info(np.eye)` - View documentation for np.eye

COPYING/SORTING/RESHAPING

`np.copy(arr)` - Copies arr to new memory
`arr.view(dtype)` - Creates view of arr elements with type dtype
`arr.sort()` - Sorts arr
`arr.sort(axis=0)` - Sorts specific axis of arr
`two_d_arr.flatten()` - Flattens 2D array
`two_d_arr.to 1D`
`arr[arr<5]` - Returns array elements smaller than 5

`arr.T` - Transposes arr (rows become columns and vice versa)
`arr.reshape(3,4)` - Reshapes arr to 3 rows, 4 columns without changing data

ADDING/REMOVING ELEMENTS

`np.append(arr, values)` - Appends values to end of arr
`np.insert(arr, 2, values)` - Inserts values into arr before index 2
`np.delete(arr, 3, axis=0)` - Deletes row on index 3 of arr
`np.delete(arr, 4, axis=1)` - Deletes column on index 4 of arr

COMBINING/SPLITTING

`np.concatenate((arr1,arr2),axis=0)` - Adds arr2 as rows to the end of arr1
`np.concatenate((arr1,arr2),axis=1)` - Adds arr2 as columns to end of arr1
`np.split(arr, 3)` - Splits arr into 3 sub-arrays
`np.hsplit(arr, 5)` - Splits arr horizontally on the 5th index

INDEXING/SLICING/SUBSETTING

`arr[5]` - Returns the element at index 5
`arr[2,5]` - Returns the 2D array element on index [2][5]
`arr[1]=4` - Assigns array element on index 1 the value 4
`arr[1,3]=10` - Assigns array element on index [1][3] the value 10
`arr[0:3]` - Returns the elements at indices 0,1,2 (On a 2D array: returns rows 0,1,2 at column 4)
`arr[0:3,4]` - Returns the elements on rows 0,1,2 at column 4
`arr[:,2]` - Returns the elements at indices 0,1 (On a 2D array: returns rows 0,1)
`arr[:,1]` - Returns the elements at index 1 on all rows
`arr<5` - Returns an array with boolean values
`(arr1<3) & (arr2>5)` - Returns an array with boolean values
`~arr` - Inverts a boolean array
`arr[arr<5]` - Returns array elements smaller than 5

STATISTICS

`np.mean(arr, axis=0)` - Returns mean along specific axis
`arr.sum()` - Returns sum of arr
`arr.min()` - Returns minimum value of arr
`arr.max(axis=0)` - Returns maximum value of specific axis
`np.var(arr)` - Returns the variance of array
`np.std(arr, axis=1)` - Returns the standard deviation of specific axis
`arr.corrcoef()` - Returns correlation coefficient of array



การดึงข้อมูลจาก Facebook





Facebook Graph API Explorer

facebook for developers

Docs

Tools

Support

My Apps

Search developer documentation



Graph API Explorer

GET / v7.0 / me?fields=posts

Node: me

posts

+ Search for a field

+ Search for a field

```
{
  "posts": {
    "data": [
      {
        "message": "ร่วมบุญปล่อยปลา 100 กิโลกรัม จำนวน 50 บาท",
        "story": "Chakrit Phain added a new photo.",
        "created_time": "2020-05-13T00:41:21+0000",
        "id": "3249440901753063_3228541027176384"
      },
      {
        "message": "ร่วมบริจาคช่วยเหลือเด็ก 50 บาท",
        "story": "Chakrit Phain added a new photo.",
        "created_time": "2020-05-12T13:35:04+0000",
        "id": "3249440901753063_3227299530633867"
      },
      {
        "message": "ร่วมบุญช่วยเหลือแม่โค 100 บาท",
        "story": "Chakrit Phain added a new photo.",
        "created_time": "2020-05-12T00:51:20+0000",
        "id": "3249440901753063_3226009474096206"
      },
      {
        "message": "ร่วมบุญนำ pack 45 บาท วันวิสาขบูชา ส่งบุญให้พ่อเลิศ",
        "story": "Chakrit Phain added a new photo.",
        "created_time": "2020-05-06T14:32:26+0000",
        "id": "3249440901753063_3211539592209861"
      },
      {
        "message": "ช่วยเหลือโคกระนือ",
        "story": "Chakrit Phain added a new photo.",
        "created_time": "2020-05-05T05:10:10+0000"
      }
    ]
  }
}
```

Access Token

EAAHqDpf4q4ABAQyOiaGmXcSOiszgsjSUhHfJdNZC2wNgeiHwk

Generate Access Token

Facebook App

DemoPage - Test1

User or Page

User Token

Permissions

user_likes

user_friends

user_posts

user_gender

email

public_profile

Add a Permission

5 options selected



Facebook Graph API Explorer

CO PRO Facebook API.ipynb ★

File Edit View Insert Runtime Tools Help All changes saved

Comment Share ⚙️ 🌐

RAM Disk Editing

+ Code + Text

```
[2] !pip install facebook-sdk
Collecting facebook-sdk
  Downloading https://files.pythonhosted.org/packages/79/d7/4cb0f5930f79205cc45db22e17fa716396d813c3d6b8d5de77783a5fa59a/facebook_sdk-3.1.0-py2.py3-none-any.whl
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from facebook-sdk) (2.23.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.6/dist-packages (from requests->facebook-sdk) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packages (from requests->facebook-sdk) (2020.4.5.1)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests->facebook-sdk) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests->facebook-sdk) (2.9)
Installing collected packages: facebook-sdk
Successfully installed facebook-sdk-3.1.0
```

```
[3] import facebook
graph = facebook.GraphAPI(access_token="EAAHqDpf4q4ABAIQyOiaGmXcSOiszgsjSUhHfJdNZC2wNgeiHwkZAqa9jwnZBEhFiCpfKUgLZBHyHnL0IfbcbZBjfSDS8DFzrzlYpkILDgwzyf2zQzXd
graph.get_object(id='me', fields='posts')
```

```
[16] {'id': '3249440901753063',
      'posts': {'data': [{ 'created_time': '2020-05-13T00:41:21+0000',
                          'id': '3249440901753063_3228541027176384',
                          'message': 'ร่วมบุญปล่อยปลา 100 กิโลกรัม จำนวน 50 บาท',
                          'story': 'Chakrit Phain added a new photo.'},
                     { 'created_time': '2020-05-12T13:35:04+0000',
                          'id': '3249440901753063_3227299530633867',
                          'message': 'ร่วมบุญจ่ายเพื่อแคนดิ้นได้ 50 บาท',
                          'story': 'Chakrit Phain added a new photo.'},
                     { 'created_time': '2020-05-12T00:51:20+0000',
                          'id': '3249440901753063_3226009474096206',
                          'message': 'ร่วมบุญซื้อยาสีเฉลือแม่โค 100 บาท',
                          'story': 'Chakrit Phain added a new photo.'},
                     { 'created_time': '2020-05-06T14:32:26+0000',
                          'id': '3249440901753063_3211539592209861',
                          'message': 'ร่วมบุญผ้า pack 45 บาท วันวิสาขบูชา\ngส่งบุญให้พ่อเลิศ',
                          'story': 'Chakrit Phain added a new photo.'}]}]
```



Facebook Graph API Explorer

developers.facebook.com/apps/

facebook for developers Docs Tools Support My Apps Search developer documentation

Add a New App

DemoPage
App ID: 24347352357310
Status: In development

Add a Product

Chakrit 2
App ID: 134767023302872
Status: Live

✓ Facebook Login

✓ Analytics

Application1 1
App ID: 2206440516285624
Status: In development

Add a Product

The screenshot shows the Facebook Developers website at the URL `developers.facebook.com/apps/`. The main content area displays a list of existing apps and a button to add a new one. A red box highlights the 'Add a New App' button. Below it, there are two app cards: 'DemoPage' (status: In development) and 'Chakrit' (status: Live). Each card includes a 'Facebook Login' and 'Analytics' checkbox. At the bottom, there is another app card for 'Application1' (status: In development) with an 'Add a Product' button.



Facebook Graph API Explorer

facebook for developers

Docs

Tools

Support

My Apps

Search developer documentation



Search apps

DemoPage ID: 61047052057310

Chakrit 2 ID: 167023302872

Add a new app

Create a New App ID

Get started integrating Facebook into your app or website

Display Name

Contact Email

This email address is used to contact you about potential policy violations, app restrictions or steps to recover the app if it's been deleted or compromised.

By proceeding, you agree to the [Facebook Platform Policies](#)



Facebook Graph API Explorer

facebook for developers

APP ID: 284410219382115 In development

Docs Tools Support My Apps

SentimentAnalysisin... ▾

View Analytics Help

Dashboard

Settings

Roles

Alerts

App Review

PRODUCTS +

Add a Product

Facebook Login
The world's number one social login product.
[Read Docs](#) [Set Up](#)

Audience Network
Monetize your mobile app or website with native ads from 3 million Facebook advertisers.
[Read Docs](#) [Set Up](#)

Analytics
Understand how people engage with your business across apps, devices, platforms and websites.
[Read Docs](#) [Set Up](#)

Messenger
Customize the way you interact with people on Messenger.
[Read Docs](#) [Set Up](#)

Webhooks
Subscribe to changes and receive updates in real time without calling the API.
[Read Docs](#) [Set Up](#)

Instant Games
Create a cross platform HTML5 game hosted on Facebook
[Read Docs](#) [Set Up](#)

Marketing API

App Center

Web Payments



Facebook Graph API Explorer

App Dashboard APP ID: 284410219382115 In development

SentimentAnalysisIn... View Analytics Help

Dashboard

Settings

Basic Advanced

Roles

Alerts

App Review

PRODUCTS +

Facebook Login

Activity Log

Currently Ineligible for Submission

Your submission is missing data in the following fields:

- Privacy Policy URL

App ID: 284410219382115 App Secret: [REDACTED] Show

Display Name: SentimentAnalysisInSocialNetwork Namespace: [REDACTED]

App Domains: [REDACTED] Contact Email: krm.online@gmail.com

Privacy Policy URL: Privacy policy for Login dialog and App Details Terms of Service URL: Terms of Service for Login dialog and App Details

App Icon (1024 x 1024): Category: Education

Find out more information about app categories [here](#)

Business Use

This app uses Facebook tools or data to

Support my own business Provide services to other businesses

Discard **Save Changes**



Facebook Graph API Explorer

facebook for developers

APP ID: 284410219382115 In development

Add a Product

Facebook Login
The world's number one social login product.

Audience Network
Monetize your mobile app or website with native ads from 3 million Facebook advertisers.

Analytics
Understand how people engage with your business across apps, devices, platforms and websites.

Messenger
Customize the way you interact with people on Messenger.

Webhooks
Subscribe to changes and receive updates in real time without calling the API.

Instant Games
Create a cross platform HTML5 game hosted on Facebook

Create Test App



Facebook Graph API Explorer

facebook for developers

APP ID: 284410219382115 In development

[View Analytics](#) [Help](#)

SentimentAnalysisInSocialNet... SentimentAnalysisInSocialNet...

DemoPage - Test1

DemoPage

Chakrit

Chakrit - Test3

Application1 - Test1

[Create Test App](#) [Create New App](#)

Add a Product

Create Test App for SentimentAnalysisInSocialNetwork

Test Apps allow you to debug and test changes in development without affecting the behavior of your production app. [?]

Test App Name: SentimentAnalysisInSocial - Test1

[Cancel](#) [Create Test App](#)

[Read Docs](#) [Set Up](#) [Read Docs](#) [Set Up](#) [Read Docs](#) [Set Up](#)

Analytics

how people engage with your business across devices, platforms and websites.

Messenger

Customize the way you interact with people on Messenger.

[Read Docs](#) [Set Up](#)

Webhooks

Subscribe to changes and receive updates in real time without calling the API.

[Read Docs](#) [Set Up](#)

Instant Games

Create a cross platform HTML5 game hosted on Facebook

[Read Docs](#) [Set Up](#)

The screenshot shows the Facebook Graph API Explorer interface. A modal window titled 'Create Test App for SentimentAnalysisInSocialNetwork' is open, prompting the user to enter a 'Test App Name'. The name 'SentimentAnalysisInSocial - Test1' is typed into the input field. The modal includes a 'Cancel' button and a prominent blue 'Create Test App' button. This modal is highlighted with a thick red border. In the background, the main dashboard shows various app configurations and documentation links for Messenger, Webhooks, and Instant Games. The overall theme is light gray with blue accents from the Facebook branding.



Facebook Graph API Explorer

facebook for developers

APP ID: 3079606332097158 In development

[View Analytics](#) [Help](#)

[Dashboard](#) [Settings](#) [Roles](#)

PRODUCTS [+](#)

Activity Log

You are currently editing a test version of **SentimentAnalysisInSocialNetwork**

Application Rate Limit

SentimentAnalysisInSocia - Test1
App ID: 3079606332097158

0% of limit used [View Details](#)

100% Remaining

User Rate Limit

0 Users throttled

API Stats

Calls Errors Average Request Time

1

0 MAY 03 05 07 09 11 13 15 17 19 21 23 25 27

Calls



Facebook Graph API Explorer

developers.facebook.com/tools/explorer/

facebook for developers Docs Tools Support My Apps Search developer documentation

Graph API Explorer

GET / v7.0 / me?fields=id,name

Access Token

EAAHqDp4q4ABAJtVAwop56fnkCZBH1vF0fEWBQ2Ch7I02BeuelvdvBdtsHjjMB

Generate Access Token

Facebook App

DemoPage - Test1

Chakrit - Test1

Chakrit - Test2

Chakrit - Test3

DemoPage

DemoPage - Test1

SentimentAnalysisInSocial - Test1

SentimentAnalysisInSocialNetwork

5 options selected

Copy Debug Information </> Get Code Save Session

A screenshot of the Facebook Graph API Explorer. The URL in the address bar is 'developers.facebook.com/tools/explorer/'. The page title is 'facebook for developers'. The main navigation menu includes 'Docs', 'Tools', 'Support', 'My Apps', and a search bar for developer documentation. Below the menu, it says 'Graph API Explorer'. The main query bar shows 'GET / v7.0 / me?fields=id,name'. To the right, there's a sidebar titled 'Access Token' with an access token ID and a 'Generate Access Token' button. Under 'Facebook App', several apps are listed: 'DemoPage - Test1' (selected), 'Chakrit - Test1', 'Chakrit - Test2', 'Chakrit - Test3', 'DemoPage', 'DemoPage - Test1' (under DemoPage), and 'SentimentAnalysisInSocial - Test1' (highlighted with a red rectangle). At the bottom, there are buttons for 'Copy Debug Information', '</> Get Code', and 'Save Session'.



Facebook Graph API Explorer

facebook for developers

Docs Tools Support My Apps Search developer documentation

Graph API Explorer

Edge: me/
posts
+ Search for a field
+ Search for a field

GET / v7.0 / me/?fields=posts

Submit

{
 "posts": {
 "data": [
 {
 "message": "ร่วมบุญปล่อยปลา 100 กิโลกรัม จำนวน 50 บาท",
 "story": "Chakrit Phain added a new photo.",
 "created_time": "2020-05-13T00:41:21+0000",
 "id": "3272534462777040_3228541027176384"
 },
 {
 "message": "ร่วมบริจาคช่วยเหลือเด็กได้ 50 บาท",
 "story": "Chakrit Phain added a new photo.",
 "created_time": "2020-05-12T13:35:04+0000",
 "id": "3272534462777040_3227299530633867"
 },
 {
 "message": "ร่วมบุญซื้อเหลือแม็ค 100 บาท",
 "story": "Chakrit Phain added a new photo.",
 "created_time": "2020-05-12T00:51:20+0000",
 "id": "3272534462777040_3226009474096206"
 },
 {
 "message": "ร่วมบุญน้ำ pack 45 บาท วันวิสาขบูชา
ส่งบุญให้เพื่อนลีศ",
 "story": "Chakrit Phain added a new photo.",
 "created_time": "2020-05-06T14:32:26+0000",
 "id": "3272534462777040_3211539592209861"
 },
 {
 "message": "ช่วยเหลือโคคราฟฟ์",
 "story": "Chakrit Phain added a new photo.",
 "created_time": "2020-05-05T05:19:40+0000",
 "id": "3272534462777040_3208218215875332"
 },
 {
 "message": "ร่วมบุญปล่อยปลา",
 "story": "Chakrit Phain added a new photo.",
 "created_time": "2020-05-05T05:03:41+0000",
 "id": "3272534462777040_3208106165877507"
 }
]
 }
}

Access Token

EAArw4rl5AoYBAHBZABO7TswZBGcZCBa5bctkZBp519NDZAQqm2HJdv12jGre

Generate Access Token

Facebook App

SentimentAnalysisInSocia - Test1

User or Page

User Token

Permissions

user_likes
user_posts
pages_show_list
pages_read_user_content
pages_manage_engagement
public_profile

Add a Permission

7 options selected

Response received in 1059 ms

Copy Debug Information

Get Code

Save Session



Facebook Graph API Explorer

facebook for developers

Graph API Explorer

GET / v7.0 / me/posts?fields=message,story,created_time,id,message_tags,caption,comments,name

Edge: me/posts

- message
- story
- created_time
- id
- message_tags
- caption
- comments
- + Search for a field
- + Search for a field

```
{ "data": [ { "message": "ร่วมบุญปล่อยปลา 100 กิโลกรัม จำนวน 50 บาท", "story": "Chakrit Phain added a new photo.", "created_time": "2020-05-13T00:41:21+0000", "id": "3272534462777040_3228541027176384" }, { "message": "ร่วมบริจาคช่วยเหลือเด็กได้ 50 บาท", "story": "Chakrit Phain added a new photo.", "created_time": "2020-05-12T13:35:04+0000", "id": "3272534462777040_3227299530633867" }, { "message": "ร่วมบุญช่วยเหลือแม่โค 100 บาท", "story": "Chakrit Phain added a new photo.", "created_time": "2020-05-12T00:51:20+0000", "id": "3272534462777040_3226009474096206" }, { "message": "ร่วมบุญน้ำ pack 45 บาท วันวิสาขบูชา ส่งบุญให้เพื่อนลิสต์", "story": "Chakrit Phain added a new photo.", "created_time": "2020-05-06T14:32:26+0000", "id": "3272534462777040_3211539592209861" }, { "message": "ช่วยเหลือโคกระนี้อ", "story": "Chakrit Phain added a new photo.", "created_time": "2020-05-05T05:19:40+0000", "id": "3272534462777040_3208218215875332" }, { "message": "ร่วมบุญปล่อยปลา", "story": "Chakrit Phain added a new photo.", "created_time": "2020-05-05T05:03:41+0000", "id": "3272534462777040_3208196465877507" } ] }
```

Response received in 915 ms

Copy Debug Information

Get Code

Save Session

Access Token

EAAwr4rl5AoYBAHBZABO7TswZBGcZCBa5bctkZBp519NDZAQqm2HJdvt2jGre

Generate Access Token

Facebook App

SentimentAnalysisInSocia - Test1

User or Page

User Token

Permissions

user_likes

user_posts

pages_show_list

pages_read_user_content

pages_manage_engagement

public_profile

Add a Permission

7 options selected



Facebook Graph API Explorer

- me/?fields=posts{comments}
- me/posts
- me/posts?fields=message,story,created_time,id,comments
- me/likes
- [Page]/posts



Facebook API Reference

← → ⌂ facebook-sdk.readthedocs.io/en/latest/api.html ☆ ⌂

Facebook SDK for Python
latest

Search docs

Installation
Integrating the SDK with Other Frameworks
Support & Development

API Reference

class facebook.GraphAPI
Changelog

Private repos and priority support
Try Read the Docs for Business Today!

Sponsored • Ads served ethically

Read the Docs v latest

Docs » API Reference [Edit on GitHub](#)

API Reference

This page contains specific information on the SDK's classes, methods and functions.

class facebook.GraphAPI

A client for the Facebook Graph API. The Graph API is made up of the objects or nodes in Facebook (e.g., people, pages, events, photos) and the connections or edges between them (e.g., friends, photo tags, and event RSVPs). This client provides access to those primitive types in a generic way.

You can read more about [Facebook's Graph API here](#).

Parameters

- `access_token` - A `string` that identifies a user, app, or page and can be used by the app to make graph API calls. [Read more about access tokens here](#).
- `timeout` - A `float` describing (in seconds) how long the client will be waiting for a response from Facebook's servers. [See more here](#).
- `version` - A `string` describing the [version of Facebook's Graph API to use](#). The default version is the oldest current version. It is used if the version keyword argument is not provided.
- `proxies` - A `dict` with proxy-settings that Requests should use. [See Requests documentation](#).
- `session` - A [Requests Session object](#).

Example

```
import facebook
```

<https://facebook-sdk.readthedocs.io/en/latest/api.html>



Colab Example

krmonline / Sentiment-Analysis-in-Social-Networks

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

Branch: master Sentiment-Analysis-in-Social-Networks / Facebook_User_Access.ipynb Find file Copy path

krmonline Created using Colaboratory b6c6e6c 1 minute ago

1 contributor

796 lines (796 sloc) | 30.4 KB

Raw Blame History

Open in Colab

In [2]: `!pip install facebook-sdk`

```
Requirement already satisfied: facebook-sdk in /usr/local/lib/python3.6/dist-packages (3.1.0)
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from facebook-sdk) (2.23.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests>facebook-sdk) (2.9)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests>facebook-sdk) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packages (from requests>facebook-sdk) (2020.4.5.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.6/dist-packages (from requests>facebook-sdk) (1.24.3)
```

In [0]: `import facebook
import json
import pandas as pd`

In [0]: `graph = facebook.GraphAPI(access_token="EAAHgDpf4q4ABAJTIu4ZBy9rHnJZCPQSTmCZANbqW8E3UVV2svK0lkfItowy3If6FZAN4wdwl2j4mAeSeFBhw7ZADjhTLS3ZAJ7wSt7RDu6QkaXKkeiLeECU7k1irUnzAjqbVGUabJXGeSHhs9ED3iTpRdyik5Xs3gEqRI0fUHfcBW0ZCVyVduuPozNIFO0uXCeCHHLBpePsgQZDZD")`

In [134]: `loop_count = 0
df_comments = pd.DataFrame(columns=['id', 'from', 'message'])
df = pd.DataFrame(columns=['id', 'created_time', 'message', 'comments'])
js = graph.get_object(id='me/posts', fields='message,story,created_time,id,comments')
while('paging' in js):`

http://colab.research.google.com/github/krmonline/Sentiment-Analysis-in-Social-Networks/blob/master/Facebook_User_Access.ipynb



LAB#1



Use Colab + Python to Handle Facebook Page



LAB #1

1. Use Colab + Python to handle Facebook Page

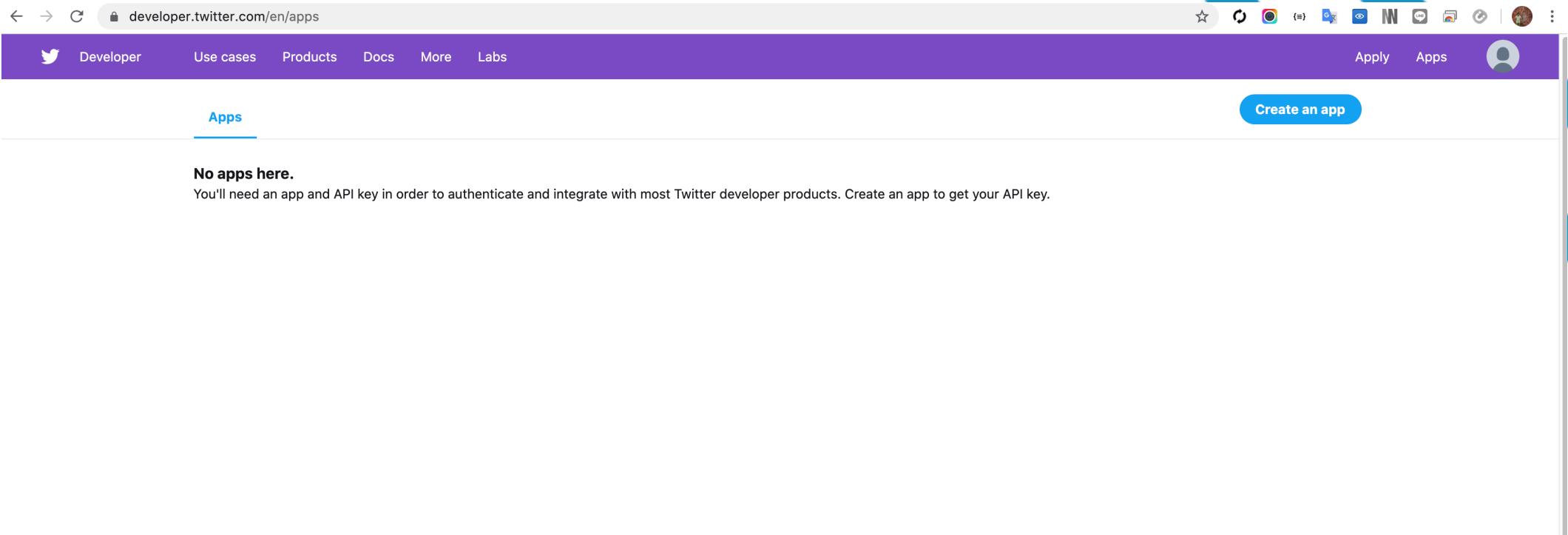




การดึงข้อมูลจาก Twitter



Register API



A screenshot of a web browser showing the Twitter Developer API registration page at developer.twitter.com/en/apps. The page has a purple header with navigation links: Developer, Use cases, Products, Docs, More, Labs, Apply, Apps, and a user profile icon. A blue button labeled "Create an app" is visible. The main content area displays a message: "No apps here." followed by the text "You'll need an app and API key in order to authenticate and integrate with most Twitter developer products. Create an app to get your API key." The background of the page features abstract blue and white curved shapes.

<https://developer.twitter.com/en/apps>



Register API

Twitter Developer Use cases Products Docs More Labs Apply Apps

#welcome

We're excited you want to use Twitter APIs and data!

As a developer platform, our first responsibility is to our users: to provide a place that supports the health of conversation on Twitter.

This application process helps us to:

1. Prevent abuse of the Twitter platform.
2. Better understand and serve our developer community.

Thank you for your time and thoughtful responses.
Applications are final once submitted and can't be edited.

What is your primary reason for using Twitter developer tools?

We'll help you on your path to getting the most out of Twitter APIs and data.

Professional	Hobbyist	Academic	Other
Building B2B products	Making a bot	Doing academic research	Embedding Tweets on a website
Building consumer products	Building tools for Twitter users	Teaching	Doing something else
Build customized solutions in-house	Exploring the API	Student	
Publishing ads programmatically			

Next



Register API

Twitter @username > Intended use > Review > Terms

Individual developer account
You are signing up for an individual developer account.

These are typically used for:

- Students**
- Individual researchers**
- Hobbyists**
- Bot creators**
- Makers**

If you think you may need to invite other people to your account in the future to share API access or apps, please [create a team developer account](#) instead.

CHAKRIT PHAIN
@ChakritPhain
[Switch @username](#)
[Create new @username](#)

This @username will be the login for your developer account.

Individual developer account You are signing up for an individual developer account. ⓘ
[Switch to a team developer account](#)

chakrit@softnix.co.th We'll send important communications about your account to this email. ⓘ
[Change email address](#)

What country do you live in? Thailand

What would you like us to call you? Ae|

Want updates about the Twitter API? Send me product updates & occasional promotional emails about the Twitter API

[Back](#) [Next](#)



Register API

Twitter @username > Intended use > Review > Terms

Developer Use cases Products Docs More Labs Apply Apps

Key things to keep in mind

This section of the application helps us ensure that users of our data are complying with [Twitter's Developer Policies](#).

This review process and our policies help us keep Twitter a safe and healthy space for public conversation.

Restricted uses

Some activities (like surveillance) are never allowed on Twitter. Take a look at our [restricted uses](#) page to ensure that your use case is policy-compliant before you submit an application.

Automation

Be sure to review the [automation rules](#) if you plan on enabling any sort of automated activity on the platform.

Be thorough

We need to completely understand your use case before we can approve it. So, please include as much detail as

How will you use the Twitter API or Twitter data?

All fields are required unless marked optional

In your words

In English, please describe how you plan to use Twitter data and/or APIs. The more detailed the response, the easier it is to review and approve.

I want to Test API for Academy for training and teaching and about sentiment analytic machine learning Artificial Intelligent Data mining Data Explorer Data Integration Data Quality Data Science and more

Response must be at least 200 characters

The specifics

Please answer each of the following with as much detail and accuracy as possible. Failure to do so could result in delays to your access to the Twitter developer platform or rejected applications.

Are you planning to analyze Twitter data? No Yes

Back **Next**



Register API

[Developer](#) [Use cases](#) [Products](#) [Docs](#) [More](#) [Labs](#)

[Apply](#) [Apps](#)



[Twitter @username](#) > [Intended use](#) > [Review](#) > [Terms](#)

Check your information
Please make sure your details are correct.

Your email will be used to contact you with important information regarding your account.

Is everything correct?

Primary use	Exploring the API
Account type	Personal
Twitter username	@ChakritPhain
Email	chakrit@softnix.co.th

In your words
I wan't to Test API for Academy for training and teaching and about sentiment analytic machine learning Artificial Intelligent Data mining Data Explorer Data Integration Data Quality Data Science and more

Analyze Twitter data	No
Tweet, Retweet or Like?	No
Show Tweets or Twitter information off Twitter	No

[Back](#)

[Looks good!](#)



Register API

Twitter @username > Intended use > Review > Terms

Please review and accept

Developer Agreement & Policy

We've carefully crafted our developer terms to help guide you in keeping Twitter a healthy and open platform for all.

We know it's long. Thanks for taking the time to read our terms.

Developer Agreement

Effective: March 10, 2020

This Twitter Developer Agreement ("Agreement") is made between you (either an individual or an entity, referred to herein as "you") and Twitter (as defined below) and governs your access to and use of the Licensed Material (as defined below). Your use of Twitter's websites, SMS, APIs, email notifications, applications, buttons, embeds, ads, and our other covered services is governed by our general Terms of Service and Privacy Policy.

PLEASE READ THE TERMS AND CONDITIONS OF THIS AGREEMENT CAREFULLY, INCLUDING ANY LINKED TERMS REFERENCED BELOW, WHICH ARE PART OF THIS LICENSE AGREEMENT. BY USING THE LICENSED MATERIAL, YOU ARE AGREEING THAT YOU HAVE READ, AND THAT YOU AGREE TO COMPLY WITH AND TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND ALL

By clicking on the box, you indicate that you have read and agree to this Developer Agreement and the Twitter Developer Policy, additionally as it relates to your display of any of the Content, the [Display Requirements](#); as it relates to your use and display of the Twitter Marks, the [Twitter Brand Assets and Guidelines](#); and as it relates to taking automated actions on your account, the [Automation Rules](#). These documents are available in hardcopy upon request to Twitter.

© 2020 TWITTER, INC

PRIVACY COOKIES TERMS OF SERVICE DEVELOPER POLICY & TERMS

By clicking **Submit Application** you are submitting your application for review. Applications are final and cannot be edited.

[Back](#) [Submit Application](#)



Register API

Twitter @username > Intended use > Review > Terms

Please review and accept

Developer Agreement & Policy

We've carefully crafted our developer terms to help guide you in keeping Twitter a healthy and open platform for all.

We know it's long. Thanks for taking the time to read our terms.

Developer Agreement

Effective: March 10, 2020

This Twitter Developer Agreement ("Agreement") is made between you (either an individual or an entity, referred to herein as "you") and Twitter (as defined below) and governs your access to and use of the Licensed Material (as defined below). Your use of Twitter's websites, SMS, APIs, email notifications, applications, buttons, embeds, ads, and our other covered services is governed by our general Terms of Service and Privacy Policy.

PLEASE READ THE TERMS AND CONDITIONS OF THIS AGREEMENT CAREFULLY, INCLUDING ANY LINKED TERMS REFERENCED BELOW, WHICH ARE PART OF THIS LICENSE AGREEMENT. BY USING THE LICENSED MATERIAL, YOU ARE AGREEING THAT YOU HAVE READ, AND THAT YOU AGREE TO COMPLY WITH AND TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND ALL

By clicking on the box, you indicate that you have read and agree to this Developer Agreement and the Twitter Developer Policy, additionally as it relates to your display of any of the Content, the [Display Requirements](#); as it relates to your use and display of the Twitter Marks, the [Twitter Brand Assets and Guidelines](#); and as it relates to taking automated actions on your account, the [Automation Rules](#). These documents are available in hardcopy upon request to Twitter.

© 2020 TWITTER, INC

PRIVACY COOKIES TERMS OF SERVICE DEVELOPER POLICY & TERMS

By clicking **Submit Application** you are submitting your application for review. Applications are final and cannot be edited.

[Back](#) [Submit Application](#)



Register API

Verify your Twitter Developer Account



o Twitter Developer Accounts <developer-accounts@twitter.com>

To: Chakrit Phain

Today at 03:51

⚠ To protect your privacy, some pictures in this message were not downloaded.

[Download pictures](#)

[Twitter Developers logo](#)

[Email verification](#)

Hi CHAKRIT PHAIN!

Thanks for applying for a Twitter Developer account.

Please confirm your email address to complete your application.

[Confirm your email](#)

Thanks!

The Twitter Dev team



Register API

Case# 0157265636 Twitter developer account application [ref:00DA000000...]



o developer-accounts <developer-accounts@twitter.com>

Today at 03:51

To: Chakrit Phain

To protect your privacy, some pictures in this message were not downloaded.

[Download pictures](#)

Hello,

We've received your application for a Twitter developer account, and we're starting our review.

Please keep an eye on this email address, and your spam folder, in case we reach out for more information. We know this application process might delay getting started with Twitter's APIs, but we use this information to offer the best support to our developers, while protecting our platform and serving the health of the public conversation.

You can read more about our [Terms of Service](#) and [Developer Terms](#), our [documentation](#), or explore our [community forums](#).

If no more details are needed, you'll get an email once we're done with our review. We appreciate your interest in developing on Twitter!

Thanks,

Twitter



Register API

developer.twitter.com/en/apps

Developer Use cases Products Docs More Labs Apps YRUCHakrit

Create an app

Apps

App Name	App ID	Actions
zabbix softnix	155876	Details
SoftnixLogAnalytic	14074015	Details
sna_course987	14175279	Details



Python-twitter

← → ⌂ python-twitter.readthedocs.io/en/latest/twitter.html#module-twitter.api ⌂

python-twitter
latest

Search docs

Installation & Testing

Getting Started

Contributing

Migration from v2 to v3

Changelog

Rate Limiting

Models

Searching

Using with Django

Modules Documentation

API

Models

Utilities

YOUR AD HERE

Reach over 7 million devs each month when you advertise with Read the Docs.

Sponsored · Ads served ethically

Read the Docs

Docs » Modules Documentation

Edit on GitHub

Modules Documentation

API

A library that provides a Python interface to the Twitter API

```
class twitter.api.Api(consumer_key=None, consumer_secret=None, access_token_key=None, access_token_secret=None, application_only_auth=False, input_encoding=None, request_headers=None, cache=<object object>, base_url=None, stream_url=None, upload_url=None, chunk_size=1048576, use_gzip_compression=False, debugHTTP=False, timeout=None, sleep_on_rate_limit=False, tweet_mode='compat', proxies=None) [source]
```

Bases: `object`

A python interface into the Twitter API

By default, the Api caches results for 1 minute.

Example usage:

To create an instance of the `twitter.Api` class, with no authentication:

```
>>> import twitter
>>> api = twitter.Api()
```

To fetch a single user's public status messages, where "user" is either a Twitter "short name" or their user id.

```
>>> statuses = api.GetUserTimeline(user)
>>> print([s.text for s in statuses])
```

<https://python-twitter.readthedocs.io/en/latest/twitter.html#module-twitter.api>



Search all documentation...

Basics

Accounts and users

Tweets

Post, retrieve, and engage with Tweets

Get Tweet timelines

Curate a collection of Tweets

Optimize Tweets with Cards

Search Tweets

Filter realtime Tweets

Sample realtime Tweets

Get batch historical Tweets

Rules and filtering

Data enrichments

Tweet objects

Tweet compliance

Tweet updates

Search Tweets

[Overview](#) [Quick start](#) [Guides](#) [FAQ](#) [API reference](#)

API reference contents ^

[Standard search API](#)

[Premium search APIs](#)

[Enterprise search APIs](#)

Standard search API

Returns a collection of relevant [Tweets](#) matching a specified query.

Please note that Twitter's search service and, by extension, the Search API is not meant to be an exhaustive source of Tweets. Not all Tweets will be indexed or made available via the search interface.

To learn how to use [Twitter Search](#) effectively, please see the [Standard search operators](#) page for a list of available filter operators. Also, see the [Working with Timelines](#) page to learn best practices for navigating results by `since_id` and `max_id`.

Resource URL

<https://api.twitter.com/1.1/search/tweets.json>

Resource Information



Twitter API in Colab

PRO CO **twitterAPI.ipynb** ★

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings

RAM Disk Editing

[79] !pip install python-twitter

Requirement already satisfied: python-twitter in /usr/local/lib/python3.6/dist-packages (3.5)
Requirement already satisfied: requests-oauthlib in /usr/local/lib/python3.6/dist-packages (from python-twitter) (1.3.0)
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from python-twitter) (2.23.0)
Requirement already satisfied: future in /usr/local/lib/python3.6/dist-packages (from python-twitter) (0.16.0)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.6/dist-packages (from requests->python-twitter) (3.1.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests->python-twitter) (2.9)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packages (from requests->python-twitter) (2020.4.5.1)
Requirement already satisfied: urllib3!=1.25.0,<1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.6/dist-packages (from requests->python-twitter) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests->python-twitter) (3.0.4)

[80] import twitter
import json
import pandas as pd

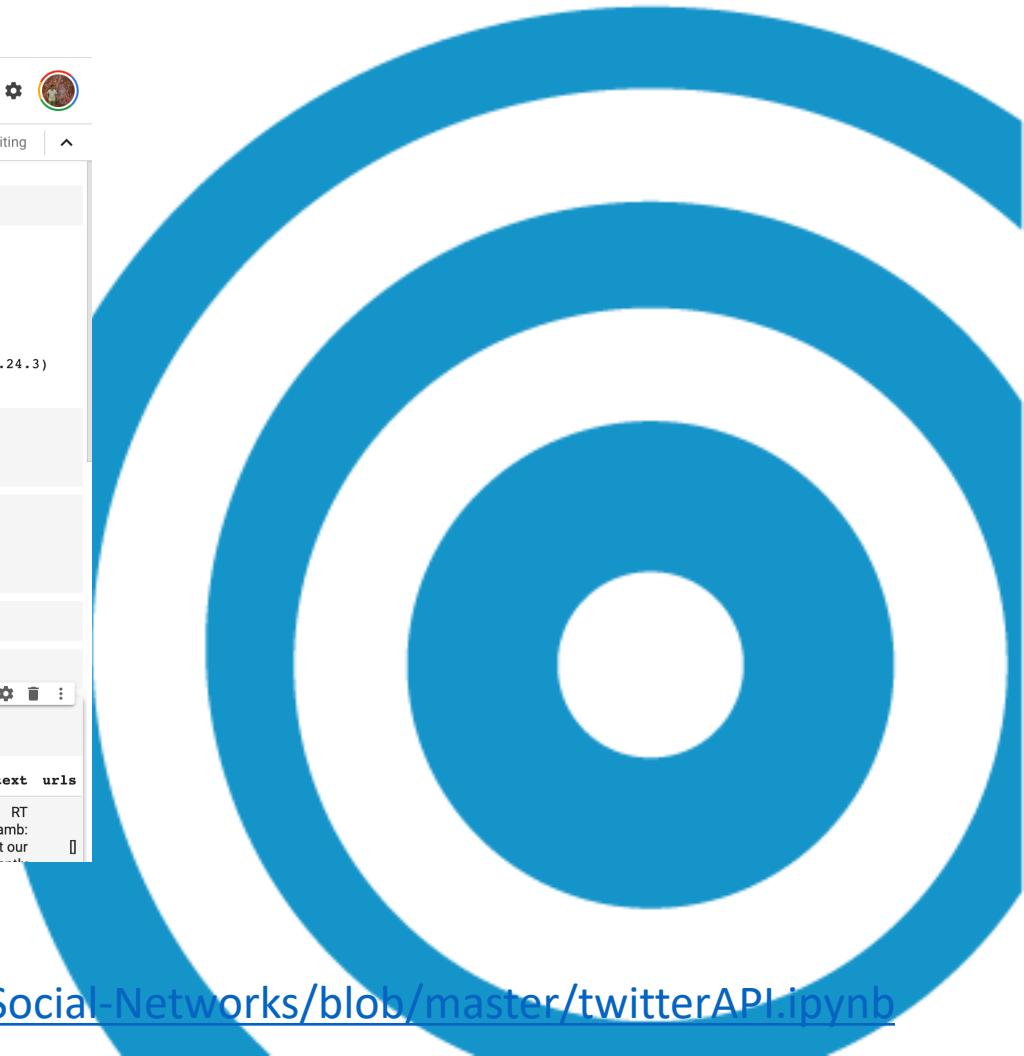
[81] api = twitter.Api(consumer_key='FGTvV91lFyIHlSIz1gC5XAwd',
 consumer_secret='HHmvEigcBlohiHm7EijA2eo0wV5MjUMauRsTQTNvPcqx8Wzglm',
 access_token_key='14649420-GsBjnvfcajjRD1AWtfDUOK8pl3b9oufh18ISoFFPA',
 access_token_secret='z5oh4oz5hTZ3s7Wc1L9g8TGKcQNui7VfUDs1OykHLj8Ue')

[82] arr_search = api.GetSearch('Anomaly Detection')

[83] a = [json.loads(i.AsJsonString()) for i in arr_search]

df = pd.DataFrame(a)
df.head(2)

	created_at	hashtags	id	id_str	lang	retweet_count	retweeted_status	source	text	urls
0	Tue May 26 07:38:01 +0000 2020	{'text': 'DeepLearning', 'text': 'DeepLearning'}	1265185407341408256	1265185407341408256	en	24.0	{'created_at': 'Mon May 25 09:14:32 +0000 2020', 'text': '@MaxChamb: Check out our RT', 'source': '', 'urls': '[]'}			



LAB #2

1. Merge facebook data and twitter data
2. Export data from 1. to CSV file





การดึงข้อมูลจากเว็บด้วย Web Scraping



- **Data extraction** is the act or process of retrieving data out of (usually unstructured or poorly structured) data sources for further data processing or data storage (data migration).



- HTTP programming
- DOM parsing
- Text pattern matching (Regular expression)
- Etc.





HTTP Request & Response

HTTP/1.1 200 OK

Date: Mon, 23 May 2005 22:38:34 GMT

Content-Type: text/html; charset=UTF-8

Content-Encoding: UTF-8

Content-Length: 138

Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT

Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)

ETag: "3f80f-1b6-3e1cb03b"

Accept-Ranges: bytes

Connection: close

<html>

<head>

<title>An Example Page</title>

</head>

<body> Hello World, this is a very simple HTML document. </body>

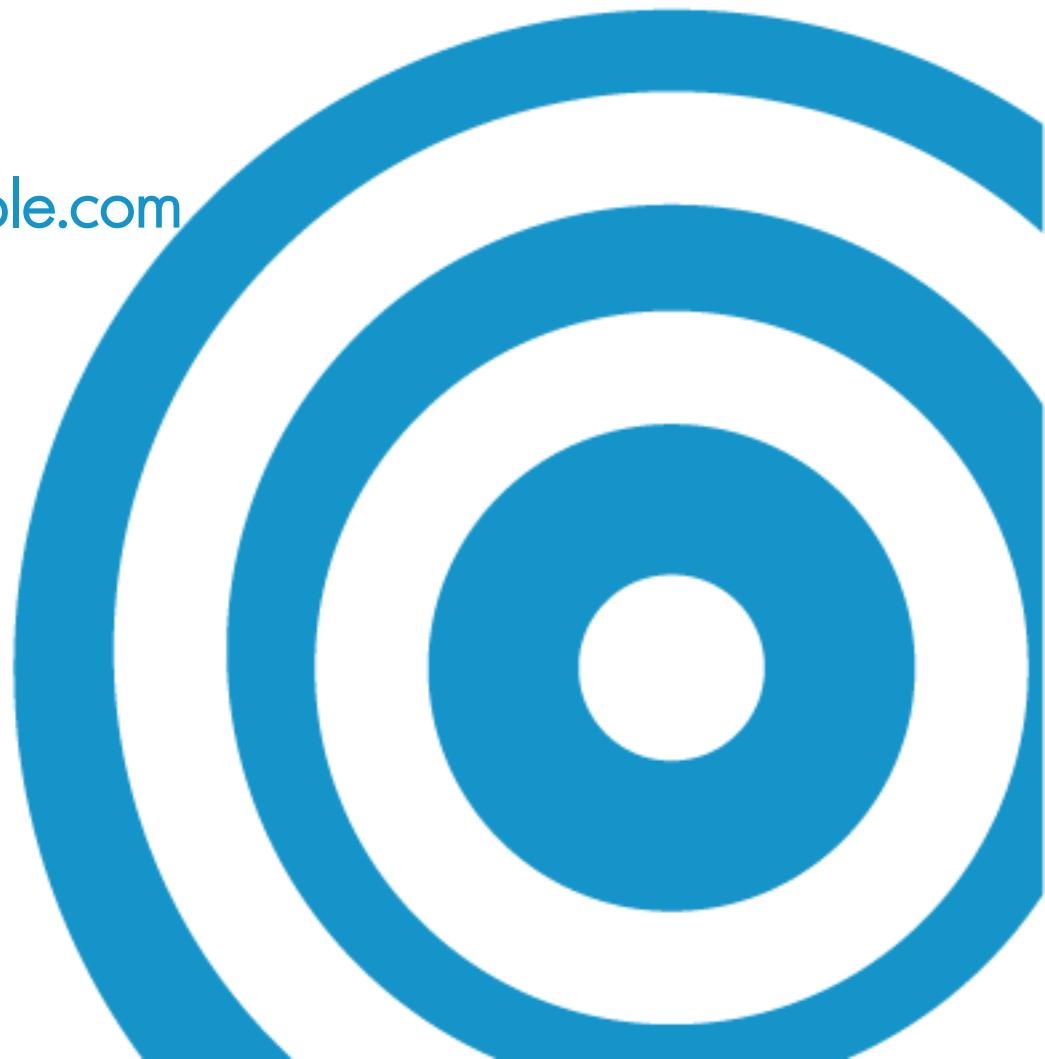
</html>

GET /index.html

HTTP/1.1

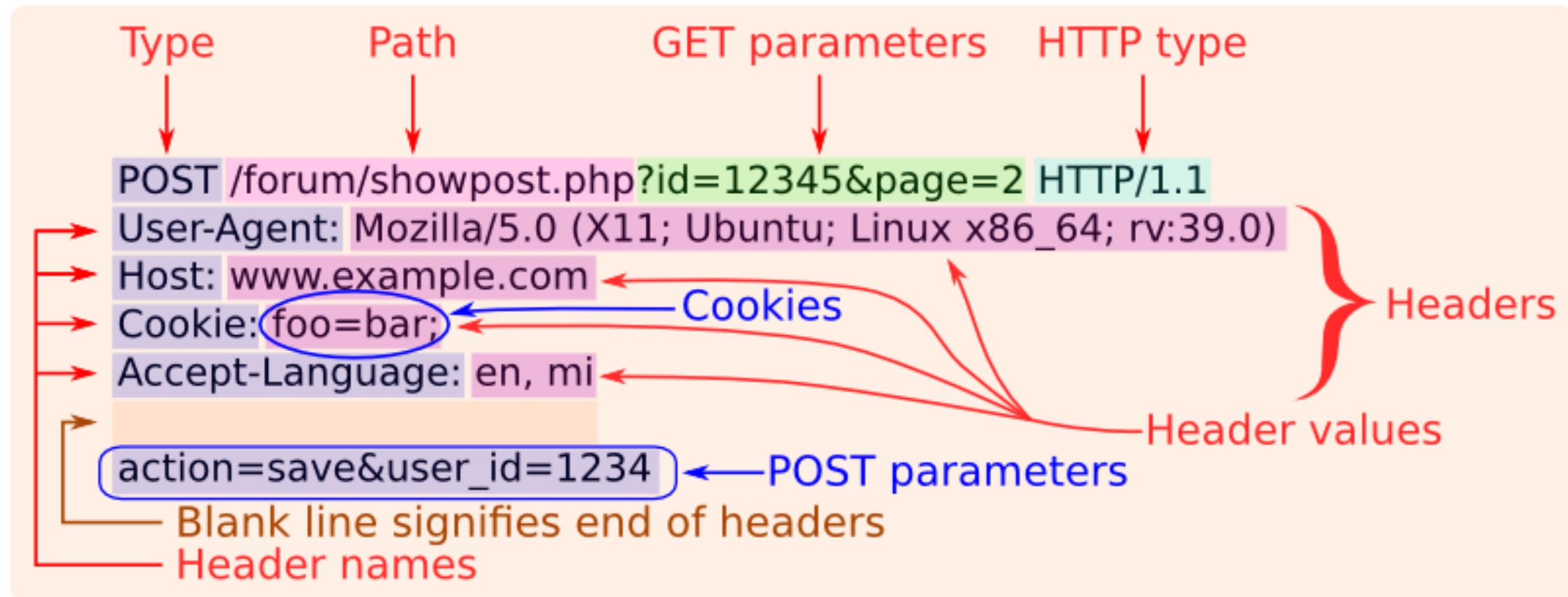
Host:

www.example.com



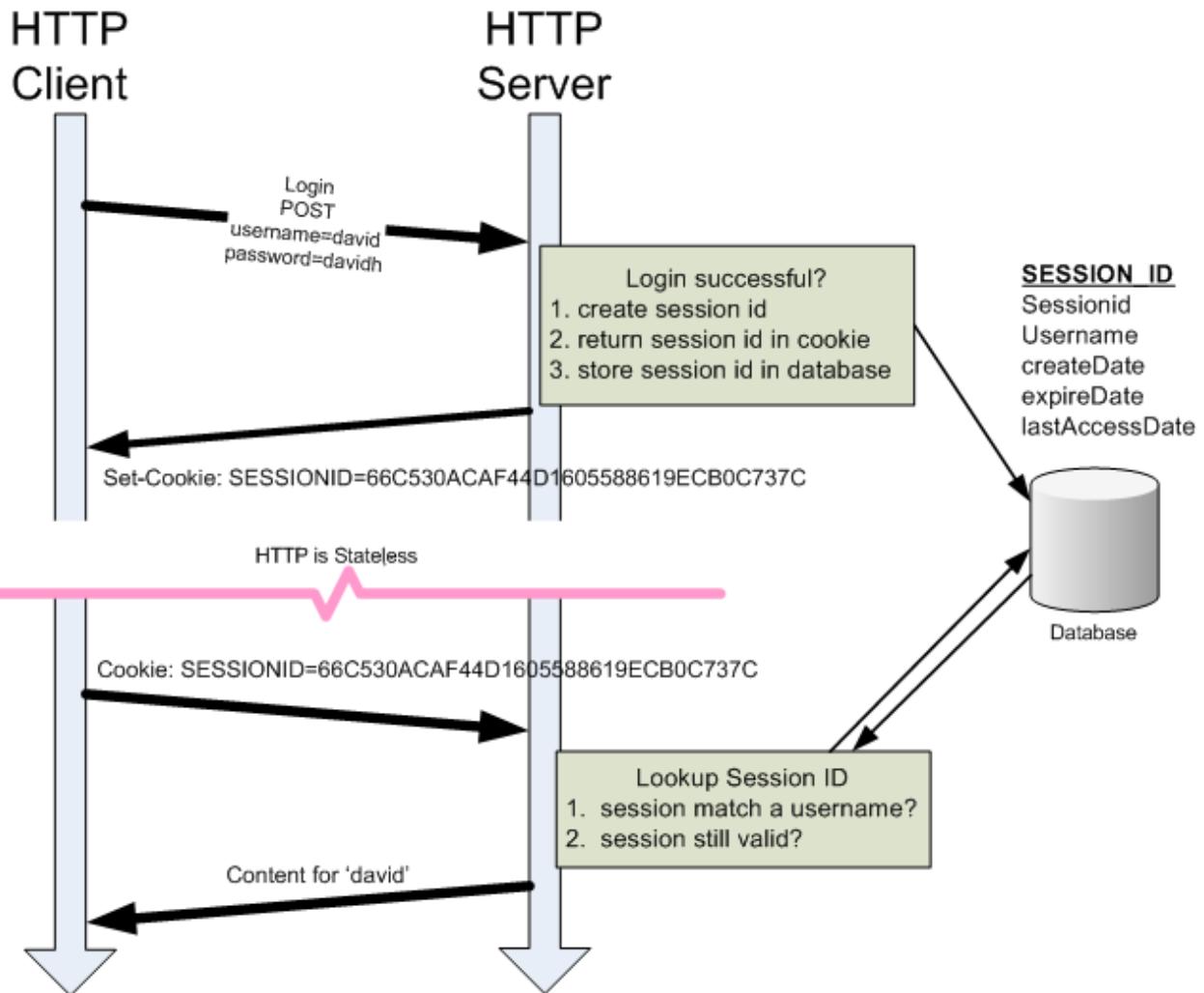


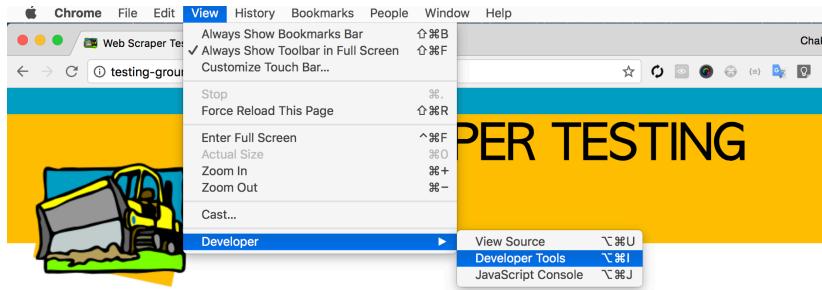
HTTP Component





Cookie & Session





LOGIN

Often in order to reach the desired information you need to be logged in to the website. Most of today's websites use so-called form-based authentication which implies sending user credentials using POST method, authenticating it on the server and storing user's session in a cookie.

This simple test shows scraper's ability to:

1. Send user credentials via POST method

The screenshot shows the Network tab of the Chrome DevTools developer tools. A single request is listed for the URL "http://testing-ground.scraping.pro/login?mode=welcome". The request method is "GET", status code is 200 OK, and the response time is 100 ms. The "General" section of the response details shows the request URL, method, status code, and headers like Connection, Content-Encoding, Content-Length, Content-Type, Date, Keep-Alive, Server, Vary, and X-Powered-By. The "Response Headers" section lists Accept, Accept-Encoding, Accept-Language, Cache-Control, Connection, and Cookie. The "Request Headers" section shows Accept, Accept-Encoding, Accept-Language, Cache-Control, Connection, and Cookie. The "Console" tab at the bottom shows the message "1 / 11 requests | 1.8 KB / 1.8 KB transferred ...".

Developer tools (Browser)

The screenshot shows the Postman application interface. The "Builder" tab is selected. A test script is displayed in the main pane, showing a sequence of steps for logging in. The script includes comments explaining the purpose of each step, such as sending user credentials via POST method, receiving a session cookie, and processing an HTTP redirect. It also includes instructions for how to test the process by entering credentials and pressing the login button. The "Params" and "Send" buttons are visible at the top right of the interface.

Postman

[Beautiful Soup 4.9.0 documentation »](#)

Table of Contents

Beautiful Soup Documentation

- [Getting help](#)
- [Quick Start](#)
- [Installing Beautiful Soup](#)
 - [Problems after installation](#)
 - [Installing a parser](#)
- [Making the soup](#)
- [Kinds of objects](#)
 - [`Tag`](#)
 - [Name](#)
 - [Attributes](#)
 - [Multi-valued attributes](#)
 - [`NavigableString`](#)
 - [`BeautifulSoup`](#)
 - [Comments and other special strings](#)
- [Navigating the tree](#)
 - [Going down](#)
 - [Navigating using tag names](#)
 - [.contents and .children](#)
 - [.descendants](#)
 - [.string](#)
 - [.strings and stripped_strings](#)
 - [Going up](#)
 - [.parent](#)
 - [.parents](#)
 - [Going sideways](#)
 - [.next_sibling and .previous_sibling](#)
 - [.next_siblings and .previous_siblings](#)
 - [Going back and forth](#)
 - [.next_element and .previous_element](#)
 - [.next_elements and .previous_elements](#)
- [Searching the tree](#)
 - [Kinds of filters](#)

Beautiful Soup Documentation

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.



These instructions illustrate all major features of Beautiful Soup 4, with examples. I show you what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.

This document covers Beautiful Soup version 4.9.1. The examples in this documentation should work the same way in Python 2.7 and Python 3.2.

You might be looking for the documentation for [Beautiful Soup 3](#). If so, you should know that Beautiful Soup 3 is no longer being developed and that support for it will be dropped on or after December 31, 2020. If you want to learn about the differences between Beautiful Soup 3 and Beautiful Soup 4, see [Porting code to BS4](#).

This documentation has been translated into other languages by Beautiful Soup users:

- 这篇文档当然还有中文版。
- このページは日本語で利用できます(外部リンク)
- 이 문서는 한국어 번역도 가능합니다.
- Este documento também está disponível em Português do Brasil.
- Эта документация доступна на русском языке.

Getting help

If you have questions about Beautiful Soup, or run into problems, send mail to the discussion group. If your problem involves parsing an HTML document, be sure to mention what the `diagnose()` function says about that document.

Quick Start



Web Scraping Demo

jupyter ราคาอ้อย scrap Last Checkpoint: 10/20/2019 (unsaved changes)

Logout Trusted Python 3

File Edit View Insert Cell Kernel Widgets Help

Run Code

```
import re
import collections
import numpy as np
from requests import get
from bs4 import BeautifulSoup
from urllib.parse import quote, unquote
import deepcut
import pickle
from keras.models import Sequential, Model
from keras.utils import np_utils
from keras.layers import Embedding, Reshape, Activation, Input, Dense, GRU, LSTM, Dropout
from pythainlp.tokenize import word_tokenize

Using TensorFlow backend.
```

In [3]:

```
header = []
for j in range(0,20,10):
    #print(j)
    url = "https://search.news.com/search/result?start="+str(j)+"&q=%E0%B8%A3%E0%B8%B2%E0%B8%84%E0%B8%B2%E0%B8%AD%E0%B9%80"
    r = get(url)
    content = BeautifulSoup(r.text).find_all('div', class_="detail")
    for i in content:
        header.append(i.find('a').text)
```

In [4]:

```
len(header)
```

Out[4]: 20

In [5]:

```
header
```

Out[5]: ['เศรษฐกิจ-ธุรกิจ',
'สังคม',
'ส่วนลดการบันทุณช้าไว้อ้อยราคាដื่งร่วงยาวยา',
'สังคม',
'จีเคาระราคาอ้อยสูนจ่ายค่าผลิตอ้วนชัวงปี้ใหญ่',
'ยืนดีสภาพกรณีไว้อ้อย แก้วิกฤตราชากด้า',
'ชาไว้เรือนรัชย์ราคาก้อยหัวงตันชั้น900บาท/ตัน',
'จีรุ๊คอดดสุดราคาก้อย',
'หมาดหนังสือพิพพ',
'ครม. กดปุ่มราคาก้อยฯ จากราช 881บาท',
'ราคาก้อย1,050บาท/ตัน',
'หมาดหนังสือพิพพ',
'นัดเคาะราคาก้อย พุ่งตันละ 1,000บาท',
'ชาไว้เรือนราคาก้อยที่ตัน 2559/60 พัน',
'ครม. อนุมัติซื้ออ้อยชั้น2ราคาแล้ว']



LAB #3

1. Scraping Wikipedia content





Sentiment Analysis with PythaiNLP





Installation pythainlp+sentiment in colab

!pip install pythainlp==1.7

```
!pip install pythainlp==1.7

Collecting pythainlp==1.7
  Downloading https://files.pythonhosted.org/packages/5a/6a/7475103451ea5c9f2bbfacc64a94bb769abce6323bae55558c1043f24520/pythainlp-1.7.0-py3-none-any.whl (10.3MB)
    |██████████| 10.3MB 2.9MB/s
Requirement already satisfied: tqdm in /usr/local/lib/python3.6/dist-packages (from pythainlp==1.7) (4.41.1)
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from pythainlp==1.7) (2.23.0)
Collecting marisa-trie
  Downloading https://files.pythonhosted.org/packages/20/95/d23071d0992dabcb61c948fb118a90683193befc88c23e745b050a29e7db/marisa-trie-0.7.5.tar.gz (270kB)
    |██████████| 276kB 44.1MB/s
Requirement already satisfied: dill in /usr/local/lib/python3.6/dist-packages (from pythainlp==1.7) (0.3.1.1)
Collecting conllu
  Downloading https://files.pythonhosted.org/packages/66/0b/a8863b5c14aee200a13a0f8c28550fd0132e947ae88441c9f517eb84613b/conllu-3.0-py2.py3-none-any.whl
Requirement already satisfied: future>=0.16.0 in /usr/local/lib/python3.6/dist-packages (from pythainlp==1.7) (0.16.0)
Requirement already satisfied: pytz in /usr/local/lib/python3.6/dist-packages (from pythainlp==1.7) (2018.9)
Requirement already satisfied: tinydb in /usr/local/lib/python3.6/dist-packages (from pythainlp==1.7) (4.1.1)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from pythainlp==1.7) (1.12.0)
Requirement already satisfied: nltk>=3.2.2 in /usr/local/lib/python3.6/dist-packages (from pythainlp==1.7) (3.5)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests->pythainlp==1.7) (2.9)
Requirement already satisfied: urllib3!=1.25.0,>=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.6/dist-packages (from requests->pythainlp==1.7) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests->pythainlp==1.7) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packages (from requests->pythainlp==1.7) (2020.4.5.1)
Requirement already satisfied: click in /usr/local/lib/python3.6/dist-packages (from nltk>=3.2.2->pythainlp==1.7) (7.1.2)
Requirement already satisfied: regex in /usr/local/lib/python3.6/dist-packages (from nltk>=3.2.2->pythainlp==1.7) (2019.12.20)
Requirement already satisfied: joblib in /usr/local/lib/python3.6/dist-packages (from nltk>=3.2.2->pythainlp==1.7) (0.15.1)
Building wheels for collected packages: marisa-trie
  Building wheel for marisa-trie (setup.py) ... done
  Created wheel for marisa-trie: filename=marisa_trie-0.7.5-cp36-cp36m-linux_x86_64.whl size=862414 sha256=e5ab254884a5eef405a9d57a7a68e4471382ee6c2830ab265c0ab030dbcec77c
  Stored in directory: /root/.cache/pip/wheels/45/24/79/022624fc914f0e559fe8a1141aaff1f9df810905a13fc75d57
Successfully built marisa-trie
Installing collected packages: marisa-trie, conllu, pythainlp
Successfully installed conllu-3.0 marisa-trie-0.7.5 pythainlp-1.7.0
```



Call sentiment function

pythainlp.sentiment("วันนี้อากาศดีจัง")

```
[92] pythainlp.sentiment("วันนี้อากาศดีจัง")
    ↵ 'pos'

[105] print(df.text[2])
    pythainlp.sentiment(df.text[2])
    ↵ พิพิธภัณฑ์น้ำ산ใจทั่วโลก พร้อมใจกันเปิดให้เราเข้าเที่ยวชมแล้วในแบบออนไลน์ (virtual tours) ใครอยากรู้เพิ่มเติมต้องไปที่... https://t.co/SIPuuGnaNk
    'pos'

▶ print(df.text[23])
    print(pythainlp.sentiment(df.text[23]))
    ↵ @JSongpromthip ถ้าหนีเที่ยวเจนนี่ไม่ผิดตัวแน่ https://t.co/UoXjsiesX5
    neg
```



LAB #4

1. Sentiment all content in your CSV file
2. Export data from 1. to CSV file





End with day 1st

Thank you