

DMSN TALKS #2



Andria Arisal, MEDC

Pusat Penelitian Informatika, LIPI

LEARN PYTHON

Basics programming for beginner and migrater

5TH OCTOBER 2018 @ 1 PM
ROOM 2303 SBM ITB

limited for 30 seats.

rsvp at <http://bit.ly/DMSNTALKS2>



LAPTOP
ALERT!

Taming the Python For Text Crunching

for Beginner and Migrater

Andria Arisal

andria.arisal@gmail.com

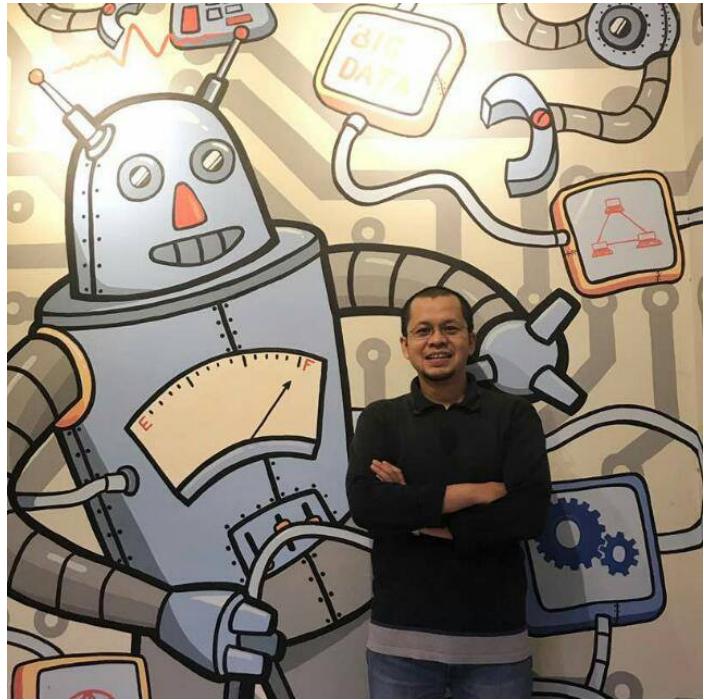


DMSN Talks #2
Bandung, 2018.10.05



Hello... I am Andria

- Andria Arisal
- ITB (2002), UniMelb (2008)
- Research Center for Informatics - LIPI
- HPC, Database, Data Science
- Java, Python

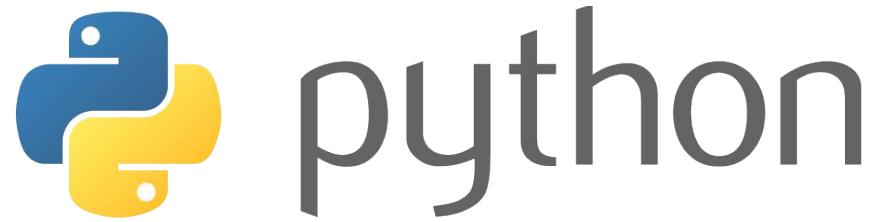


Outlines



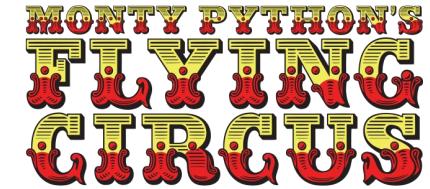
- Basic Python Programming
 - Type, Control Sequence, Conditional, Input/Output, Modules/Libraries
 - Tools
- Text Processing
 - Collect, Extract, Analyze, Integrate, Utilize

Python



Very short history of Python

- Guido van Rossum
- Monty Python
- ABC language
- 1991, first release
- 2000, Python 2 (EOL 2020)
- 2008, Python 3



...In December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus).

Features

- Highly readable
- Interpreter
- Multi-paradigm programming language
- Dynamic typing
- Modularity
- Automatic memory management

There should be one—and preferably only one—obvious way to do it

*Beautiful is better than ugly
Explicit is better than implicit
Simple is better than complex
Complex is better than complicated
Readability counts*

-- *Zen of Python*

<https://www.python.org/dev/peps/pep-0020/>



Installation

- Usually comes with *nix
 - Ask your system administrator
- Should be installed on Windows
 - Install from the installer packages
- Additional libraries installation
 - pip | conda
- Installer varieties:
 - Python
 - Anaconda
 - Miniconda
- IDEs:
 - Terminal
 - IDLE
 - PyDev (Eclipse)
 - PyCharm
 - Jupyter



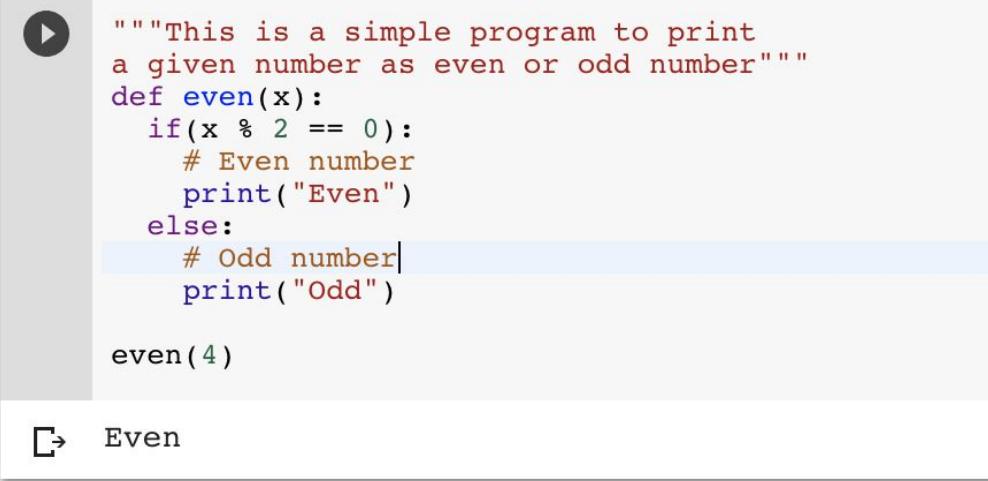
Programming Rules

- Syntax (Indentation, Comments, DocStrings)
- Keywords (`False, None, True, and, as, assert, break, class, def, del, elif, else, except, finally, for, from, global, if, import, in, is lambda, nonlocal, not, or, pass, raise, return, try, while, with, yield`)
- Data structures (base types, composite, ...)
- Literals (strings, numbers)
- Containers (list, tuples, set, dictionaries)
- Operators, Methods, Functions



Syntax

- Indentation
 - is important !!
 - Spaces or Tabs
 - Block of code
- Comments
- DocStrings



```
"""This is a simple program to print
a given number as even or odd number"""
def even(x):
    if(x % 2 == 0):
        # Even number
        print("Even")
    else:
        # Odd number|
        print("Odd")

even(4)
```

even(4) → Even



Variables

- No specific commands
- It is created upon assignment
- Type is not static, can be changed
- Traits:
 - Should be started with a letter or underscore
 - Case sensitive
 - Can only contain alphanumeric, underscore
 - CONSTANT

```
[23] PI = 3.14
```

```
[24] anumber = 42
      print(type(anumber))
```

⇨ <type 'int'>

▶ anumber = "42"
 print(type(anumber))

⇨ <type 'str'>



Type: Number

- Integer
 - Float
 - Complex
 - Can be very large or small
 - Operators: *, /, //, %, +, -
 - Automatic casting*

```
[12] anInteger = 42
      aFloat = 3.14
      aComplex = 0.5 + 14.315i
```

```
print(type(anInteger))  
print(type(aFloat))  
print(type(aComplex))
```

```
↳ <type 'int'>
    <type 'float'>
    <type 'complex'>
```

```
[13] print(anInteger + aComplex)
```

$\Gamma \rightarrow (42.5 + 14.315j)$

```
aVeryLargeInteger = 9999999999999999999999999999999999999999999999999999999  
aVerySmallFloat = 0.00000000000000000000000000000001  
  
print(aVeryLargeInteger + aVerySmallFloat).
```

→ 1e+35



Type: Number

- Literal
 - Binary
 - Decimal
 - Octal
 - Hexadecimal



```
binaryLiteral = 0b00101010
decimalLiteral = 42
octalLiteral = 0o52
hexadecimalLiteral = 0x2A

print(binaryLiteral)
print(oct(decimalLiteral))
print(hex(octalLiteral).)
print(bin(hexadecimalLiteral))
```



```
42
052
0x2a
0b101010
```

Type: Boolean

- True | False
- Operators: and, or, not, ==,
!=, <, >

```
def trueTest(x):  
    if (x == True):  
        return True  
    else:  
        return False  
  
print(trueTest(not False))
```

→ True



Type: Strings

- Immutable
- Addressable with index
- Multiline strings
- Raw strings

```
[ ] a_string = "This is a beautiful world"  
print("you are " + a_string[10:19.])
```

⇒ you are beautiful

```
[ ] m_string = """Chaos reigns within.  
Reflect, repent, and reboot.  
Order shall return."""  
print(m_string).
```

⇒ Chaos reigns within.
Reflect, repent, and reboot.
Order shall return.



```
r_string = r"Großer Bruder\t1984"  
a_string = "Großer Bruder\t1984"  
print(r_string)  
print(a_string).
```

⇒ Großer Bruder\t1984
Großer Bruder 1984

Type: List

- Mutable
- Addressable with index
- Functions: append, remove, pop, index, +

```
[52] a_list = [42, 3.14, "Hello World", "python"]
      print(a_list[1:-1])
```

```
⇒ [3.14, 'Hello World']
```

```
[64] c_list.insert(2, "itb").
      print(c_list)
```

```
⇒ [42, 'Hello World', 'itb', 'python', 'java', 2, 5]
```

```
▶ c_list[1] = "Hello universe"
      print(c_list.)
```

```
⇒ [42, 'Hello universe', 'itb', 'python', 'java', 2, 5]
```



Type: Tuple

- Immutable
- Faster than List
- Can hold mutable item

```
[70] a_tuple = (42, 3.14, "Hello World", "python", ["java", "c"])
      print(a_tuple[1:-1])
```

```
↪ (3.14, 'Hello World', 'python')
```

```
[71] a_tuple[1] = 2.5
```

```
↪
TypeErrorTraceback (most recent call last)
<ipython-input-71-c3ea225acf07> in <module>()
      ----> 1 a_tuple[1] = 2.5
```

```
TypeError: 'tuple' object does not support item assignment
```



Type: Set

- Mutable
- No duplication
- Unordered
- Functions: add, discard, difference, union, intersection, isdisjoint, issubset, issuperset, pop, clear

```
[ ] a_set = {42, 3.14, "Hello World", 2, 5}
print(a_set)
```

```
▶ a_set.add(3.14)
print(a_set)
```

```
⇨ set([42, 'Hello World', 5, 2, 3.14])
```

```
[ ] a_set.add(3)
print(a_set)
```

```
⇨ set([2, 3, 5, 3.14, 42, 'Hello World'])
```



Type: Dictionary

- Mutable
- key-value pair
- Keys are unique
- Functions: items, keys, values

```
▶ a_dict = { 40131 : "Cipaganti",
    40132 : "Taman Sari"}
print(a_dict)

⇒ {40131: 'Cipaganti', 40132: 'Taman Sari'}
```

```
[3] a_dict.keys(.)
```

```
⇒ [40131, 40132]
```

```
[4] a_dict[40135] = "Sadang Serang"
print(a_dict)
```

```
⇒ {40131: 'Cipaganti', 40132: 'Taman Sari', 40135: 'Sadang Serang'}
```



Iteration

- For
- While

```
[14] for i in range(1, 5):  
      print(i)
```



```
i = 1  
while(i < 5):  
    print(i)  
    i += 1
```



```
1  
2  
3  
4
```



Condition

- If - else

```
if(x < 0):
    print("negative")
elif(x == 0):
    print("zero")
else:
    print("positive")
```

Functions

- Method
 - Do something
- Function
 - Return something

```
def greet_args(greet, *args, **names):  
    for name in names:  
        print(greet),  
        for arg in args:  
            print(arg),  
        print(names[name])
```

```
[28] greetArgs("Selamat", "Pagi", "Siang", "Sore", "Malam", name = "Dunia")
```

```
[6] def fGreet(x):  
    return "Hello " + x
```

```
greet_str = fGreet("World")  
print(greet_str.)
```

```
↪ Hello World
```



Input/Output

- Read from File

```
[ ] with open("hamlet-gutenberg.txt") as f:  
    first_four = f.read(4)  
    print(first_four)
```

```
⇒ THE
```

```
▶ with open("hamlet-gutenberg.txt") as f:  
    lines = f.readlines()  
  
    print(lines).
```

```
⇒ [ 'THE TRAGEDY OF HAMLET, PRINCE OF DENMARK\r\n', '\r\n',
```

- Write to File

```
▶ with open("test.txt", "w") as wf:  
    wf.write("Hello world")
```



Module

- Created program can be executed as module

```
>>> cat fibo.py
def fib(n):      # write Fibonacci series up to n
    a, b = 0, 1
    while a < n:
        print(a, end = " ")
        a, b = b, a+b
    print()

if __name__ == "__main__":
    import sys
    fib(int(sys.argv[1]))
```

```
>>> python3 fibo.py 10
0 1 1 2 3 5 8
```



Libraries

- Developed modules from other
- Ready to use
- Read the documentation

```
[ ] !pip install nltk
[ ] Requirement already satisfied: nltk in /usr/local/lib/python3.6/dist-packages (3.2.1)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from nltk)

[ ] import nltk
nltk.download("all")
[ ] [nltk_data]      Unzipping corpora/shakespeare.zip.
[ ] [nltk_data]      Downloading package sinica_treebank to
[ ] [nltk_data]          /root/nltk_data...
[ ] [nltk_data]              Unzipping corpora/sinica_treebank.zip.
[ ] [nltk_data]      Downloading package smultron to /root/nltk_data...
[ ] [nltk_data]          Unzipping corpora/smultron.zip.
[ ] [nltk_data]      Downloading package state_union to /root/nltk_data...
[ ] [nltk_data]          Unzipping corpora/state_union.zip.
[ ] [nltk_data]      Downloading package stopwords to /root/nltk_data...
[ ] [nltk_data]          Unzipping corpora/stopwords.zip.
[ ] [nltk_data]      Downloading package subjectivity to
[ ] [nltk_data]          /root/nltk_data...
[ ] [nltk_data]              Unzipping corpora/subjectivity.zip.
[ ] [nltk_data]      Downloading package swadesh to /root/nltk_data...
[ ] [nltk_data]          Unzipping corpora/swadesh.zip.
```



Help

```
help(dict).
```

↳ Help on class dict in module __builtin__:

```
class dict(object)
| dict() -> new empty dictionary
| dict(mapping) -> new dictionary initialized from a mapping object's
|   (key, value) pairs
| dict(iterable) -> new dictionary initialized as if via:
|     d = {}
|     for k, v in iterable:
|       d[k] = v
| dict(**kwargs) -> new dictionary initialized with the name=value pairs
|   in the keyword argument list.  For example:  dict(one=1, two=2)

Methods defined here:

__cmp__(...)
  x.__cmp__(y) <==> cmp(x,y)
```

```
dir(dict).
```

```
['__class__',
 '__cmp__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
```



Virtual Environment

- A directory into which some binaries and shell scripts are installed
 - The binaries include python for executing scripts and pip for installing other modules within the environment.
 - Manage dependencies between projects

```
[aarisl ~] ⚡/tmp
>>> virtualenv -p python3 sbmvenv

Running virtualenv with interpreter /usr/local/bin/python3
Using base prefix '/usr/local/Cellar/python/3.6.5_1/Frameworks/Python.framework/Versions/3.6'
New python executable in /private/tmp/sbmvenv/bin/python3.6
Also creating executable in /private/tmp/sbmvenv/bin/python
Installing setuptools, pip, wheel...done.

[aarisl ~] ⚡/tmp
>>> source sbmvenv/bin/activate

[aarisl ~] ⚡/tmp ✘ sbmvenv
>>> pip install nltk
Collecting nltk
  Collecting six (from nltk)
    Using cached https://files.pythonhosted.org/packages/67/4b/141a581104b1f6397bfa78ac9d43d8ad29a7ca43ea90a2d863fe3056e86
Installing collected packages: six, nltk
Successfully installed nltk-3.3 six-1.11.0

[aarisl ~] ⚡/tmp ✘ sbmvenv
>>> python
Python 3.6.5 (default, Jun 17 2018, 12:13:06)
[GCC 4.2.1 Compatible Apple LLVM 9.1.0 (clang-902.0.39.2)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>>

[aarisl ~] ⚡/tmp ✘ sbmvenv
>>> ls sbmvenv
bin           include        lib          pip-selfcheck.json

[aarisl ~] ⚡/tmp ✘ sbmvenv
```

Text Processing



Motivation

- Text processing is part of Natural Language Processing
- Various applications
 - Summarization
 - Information Extraction
 - Language Translation
 - Sentiment/Emotion Analysis
 - Text Generation



Text Processing Pipeline

- Collecting
 - Data / Document
- Extracting & Pre-processing
 - Normalization, Data Cleaning, Case conversion, Tokenization, Punctuation/Stopwords removal
- Analyzing
 - Morphological analysis, Syntactic analysis, Semantic analysis
 - Discourse analysis (Classification model, Predictive analysis, Term Frequency/Inverse Document Frequency, Conditional probabilities, Topic model), Error evaluation
- *Integrating*
 - *Link to data structure, Model deployment*
- Utilizing



Text Files

- Open
- Read
- Write
- Close

```
▶ file_name = "hamlet-gutenberg-20.txt"  
with (open(file_name)) as f:  
    lines = f.readlines()  
    for line in lines:  
        print(type(line))
```

```
[13] file_name = "hamlet-gutenberg-20.txt"  
with (open(file_name)) as f:  
    content = f.read(100)  
    print(type(content))  
    print(content)
```

```
↳ <class 'str'>  
THE TRAGEDY OF HAMLET, PRINCE OF DENMARK
```

by William Shakespeare

Contents

ACT I Scene I. Elsinore.

```
▶ read_file_name = "hamlet-gutenberg-20.txt"  
write_file_name = "hamlet-gutenberg-20_100.txt"  
with (open(read_file_name, "r")). as rf:  
    content = rf.read(100)  
try:  
    wf = open(write_file_name, "w")  
    wf.write(content)  
finally:  
    wf.close()
```



Text Collection

- Vary
- Depend on where data is collected
- Use target API



```
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener

#consumer key, consumer secret, access token, access secret.
consumer_key="..."
consumer_secret="..."
access_token="..."
access_token_secret="..."

class Listener(StreamListener):
    def on_data(self, data):
        print(data)
        return(True)

    def on_error(self, status):
        print(status)

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_tokensecret)

twitterStream = Stream(auth, Listener())
twitterStream.sample()
```



Text Pre-processing

- Tokenizing
 - Word
 - Sentence

```
[19] import nltk  
  
file_name = "hamlet-gutenberg-20.txt"  
with (open(file_name)) as f:  
    content = f.read()  
    nltk_tokens = nltk.word_tokenize(content)  
    print(nltk_tokens)  
    print("\n")  
    print("Number of Words: " , len(nltk_tokens))
```



```
file_name = "hamlet-gutenberg-20.txt"  
with (open(file_name)) as f:  
    content = f.read()  
    print(content.split())  
    print("\n")  
    print("Number of Words: " , len(content.split()))
```

Text Pre-processing (normalization)

- Lower case



```
file_name = "hamlet-gutenberg-20.txt"
with (open(file_name)) as f:
    content = f.read()
    content_lower = content.lower()
    nltk_tokens = nltk.word_tokenize(content_lower)
    tokens_set = set(nltk_tokens)
    tokens_list = list(tokens_set)
    tokens_list.sort()
    print(tokens_list)
    print("\n")
    print("Number of Words: " , len(tokens_list))
```

```
⇒ [',', '.', 'a', 'act', 'another', 'before', 'by', 'castle', 'contents',
```

```
Number of Words: 39
```



Text Pre-processing (normalization)

- Non Alphanumeric

```
import re

text1 = """RT @TheRealC_Castle: My cousin think it's normal to be " working from
home" and still get FULLY dressed 😂😂😂😂😂😂 https://t.co/UztWETx18w"""

new_text = re.compile(r'\W+', re.UNICODE).split(text1)
print(new_text)

⇒ ['RT', 'TheRealC_Castle', 'My', 'cousin', 'think', 'it', 's', 'normal', 'to',
```

- Regular expression pattern test
 - <https://regex101.com/>



Text Pre-processing (Cleanup)

- Stopwords
- Punctuation

```
▶ from nltk.corpus import stopwords  
stop_words = stopwords.words('indonesian')  
print(stop_words)  
⇒ ['ada', 'adalah', 'adanya', 'adapun', 'agak', 'aq
```

```
▶ from nltk.corpus import stopwords  
from string import punctuation  
  
stop_words = stopwords.words('english')  
  
token_filtered = []  
with (open(file_name)) as f:  
    content = f.read()  
    nltk_tokens = nltk.word_tokenize(content)  
    for token in nltk_tokens:  
        if token not in stop_words:  
            if token not in punctuation:  
                token_filtered.append(token)  
  
print(token_filtered)  
⇒ ['THE', 'TRAGEDY', 'OF', 'HAMLET', 'PRINCE', 'OF', 'DENMARK', 'William',
```



Text Pre-processing (Stemming)

- Stem

```
▶ from nltk.stem import PorterStemmer
  from nltk.tokenize import sent_tokenize, word_tokenize

  words = ["play", "players", "played", "playing"]
  ps = PorterStemmer()

  for word in words:
    print(ps.stem(word))
```

⇨ play
player
play
play

Text Processing

● Word Count

```
import operator

file_name = "hamlet-gutenberg-20.txt"
token_count = {}
with (open(file_name)) as f:
    content = f.read()
    nltk_tokens = nltk.word_tokenize(content)
for token in nltk_tokens:
    if(token in token_count):
        token_count[token] += 1
    else:
        token_count[token] = 1

sorted_token_count = sorted(token_count.items(), key=operator.itemgetter(1), reverse = True)

for k, v in sorted_token_count:
    print(k + ":" + str(v))
```

```
↳ .:32
Scene:17
in:13
A:12
the:12
Castle:12
room:11
II:5
```



Text Processing

- Synonim



```
from nltk.corpus import wordnet as wn
```

```
[27] from nltk.corpus import wordnet as wn

for synset in wn.synsets('car'):
    print(synset.lemma_names())
```



```
['car', 'auto', 'automobile', 'machine', 'motorcar']
['car', 'railcar', 'railway_car', 'railroad_car']
['car', 'gondola']
['car', 'elevator_car']
['cable_car', 'car']
```



Text Analysis

● POS Tag

- NN
- NNP
- PRP
- VB
- IN
- TO
- CD
- ...



```
from nltk.corpus import state_union
from nltk.tokenize import PunktSentenceTokenizer
file_name = "frankenstein-100.txt"
data = []
with (open(file_name)) as f:
    content = f.read()
    sentences = nltk.sent_tokenize(content)
    for sent in sentences:
        data = data + nltk.pos_tag(nltk.word_tokenize(sent))

for word in data:
    print(word)

→ ( '— , 'CD' )
( '.', '.' )
('You', 'PRP')
('will', 'MD')
('rejoice', 'VB')
('to', 'TO')
('hear', 'VB')
('that', 'IN')
```



Text Analysis

- NE Tag
 - ORG
 - PER
 - LOC
 - DATE
 - TIME
 - MON
 - PER
 - FAC
 - GPE



```
file_name = "frankenstein-100.txt"
data = []
with (open(file_name)) as f:
    content = f.read()
    sentences = nltk.sent_tokenize(content)
    for sent in sentences:
        data = data + nltk.pos_tag(nltk.word_tokenize(sent))

    ne_data = nltk.ne_chunk(data)

for word in ne_data:
    print(word)

    ↵ ('nortn', 'NN')
    ('of', 'IN')
    (GPE London/NNP)
    ('', '', '')
    ('and', 'CC')
    ('as', 'IN')
    ('I', 'PRP')
    ('walk', 'VBP')
    ('in', 'IN')
    ('the', 'DT')
    ('streets', 'NNS')
    ('of', 'IN')
    (GPE Pittsburgh/NNP)
```



Text Analysis

- TF/IDF
 - TF
 - IDF

```
from nltk.text import TextCollection

text1 = ''
text2 = ''

with (open('hamlet-gutenberg.txt')) as f1:
    text1 = f1.read()
with (open('frankenstein.txt')) as f2:
    text2 = f2.read()

texts = TextCollection([text1, text2])

print(texts.tf('HAMLET', text1))
print(texts.idf('HAMLET'))
print(texts.tf_idf('HAMLET', text1))
```

```
→ 0.0020433321512403533
0.6931471805599453
0.0014163299195797386
```



Text Analysis

● Sentiment Analysis

```
file_name = 'tweets_sample.json'  
sid = SentimentIntensityAnalyzer()  
  
tweets_text_sentiment = {}  
with (open(file_name, 'r')) as f:  
    for line in f:  
        try:  
            tweet = json.loads(line)  
            if('delete' not in tweet.keys()):  
                if(tweet['lang'] == 'en'):  
                    text = tweet['text']  
                    print(text + " : " + str(sid.polarity_scores(text)[ 'compound']))  
        except:  
            continue
```

RT @SinfulCaffeine: Intober day 3: Jellyfish.
I went a little overboard with this design haha 🌟
Her name is Fanna <https://t.co/r31k6ea6FU> : 0.4588
RT @Asyrafhoyeh: Short sad story:

I found a love♥ but it's not for me...💔

The End. : -0.2617

@KillSection not weird just different. it's a lot better if u don't try 2 comp

RT @GetFlexSeal: 🚫 STOP LEAKS FAST WITH THE FLEX SEAL® FAMILY OF PRODUCTS! [ht](#)

RT @aiellexx: I love how Donny wants Kisses to experience to enjoy and live in

RT @NoHoesChriss: " how's your October so far ? " 🍁

Me: <https://t.co/Jrody6ueAg> : 0.0

Instead of just quoting this guy in ya bio maybe read the words a

Text Utilization

```
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
from nltk.sentiment.vader import SentimentIntensityAnalyzer

#consumer key, consumer secret, access token, access secret.
consumer_key="..."
consumer_secret="..."
access_token="..."
access_token_secret="..."

sid = SentimentIntensityAnalyzer()

class Listener(StreamListener):
    def on_data(self, data):
        tweet = json.loads(data)
        if('delete' not in tweet.keys()):
            if(tweet['lang'] == 'en'):
                text = tweet['text']

                sentiment_score = sid.polarity_scores(text)['compound']
                if(sentiment_score < 0):
                    sentiment = 'NEGATIVE'
                elif(sentiment_score > 0):
                    sentiment = 'POSITIVE'
                else:
                    sentiment = 'NEUTRAL'

                print(sentiment + " : " + text)
                print('=====') 
            return(True)

    def on_error(self, status):
        print(status)

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_tokensecret)

twitterStream = Stream(auth, Listener())
twitterStream.sample()
```

▶ POSITIVE : that period in my life i was watching stranger things was the best
=====

POSITIVE : RT @BTS_twt: thx CHI 😊😊 <https://t.co/z96bB2CajT>
=====

NEUTRAL : RT @BobBrownPGH: The moment your teammates find out you attended tl
=====

NEUTRAL : RT @zackfox: Y'all I just airdropped a pic of trumps dad to this l:
=====

NEUTRAL : We've been using the term #rightsizing in all our work with client
=====

NEGATIVE : @nytimes @secularcitizen2 That way it looks like he talked a bunc
=====

POSITIVE : RT @SquishieKsoo: I have a feeling that this comeback will be str
=====

POSITIVE : RT @dbongino: Obama: I used illegal drugs in my teens and I got i:
=====

Media butt kissers: We love you Barack, we have thrills up our...
=====

POSITIVE : @stuj88 True though
=====

POSITIVE : RT @RichardBurgon: Solidarity with workers at McDonald's, TGI Frie
=====

NEGATIVE : I added a video to a @YouTube playlist <https://t.co/b116YU5X4d> Go:
=====

POSITIVE : @Independent Nope, he was mocking her, then called her and others



Text Utilization

```
>>> virtualenv -p python3 dmsn2
Running virtualenv with interpreter /usr/local/bin/python3
Using base prefix '/usr/local/Cellar/python/3.6.5_1/Frameworks/Python.framework/Versions/3.6'
New python executable in /private/tmp/dmsn2/bin/python3.6
Also creating executable in /private/tmp/dmsn2/bin/python
Installing setuptools, pip, wheel...done.

[+] aarisa1 ~ /tmp
>>> mkdir twitter_sa_dmsn2
[+] aarisa1 ~ /tmp
>>> cd twitter_sa_dmsn2
[+] aarisa1 ~ /tmp/twitter_sa_dmsn2
>>> source ../dmsn2/bin/activate
[+] aarisa1 ~ /tmp/twitter_sa_dmsn2 dmsn2
>>> pip install nltk
Collecting nltk
  Downloading https://files.pythonhosted.org/packages/67/4b/141a581104b1f6397bfa78ac9d43d8ad29a7ca43ea90a2d863fe3056e86a/six-1.11.0-py3-none-any.whl
Collecting six (from nltk)
  Using cached https://files.pythonhosted.org/packages/67/4b/141a581104b1f6397bfa78ac9d43d8ad29a7ca43ea90a2d863fe3056e86a/six-1.11.0-py3-none-any.whl
Installing collected packages: six, nltk
Successfully installed nltk-3.3 six-1.11.0

[+] aarisa1 ~ /tmp/twitter_sa_dmsn2 dmsn2
>>> pip install tweepy
Collecting tweepy
  Using cached https://files.pythonhosted.org/packages/85/f1/2e8c7b202dd04117a378ac0c55cc7dafa80280ebd7f692f1fa8f27fd6288/tweepy-3.1.0-py3-none-any.whl
Requirement already satisfied: six<=1.10.0 in /private/tmp/dmsn2/lib/python3.6/site-packages (from tweepy) (1.11.0)
Collecting requests-oauthlib<0.7.0 (from tweepy)
  Using cached https://files.pythonhosted.org/packages/94/e7/c250d122992e1561690d9c0f7856dad679d61fd4bdd0e598087dce607f6c/requests-oauthlib-0.7.0-py3-none-any.whl
Collecting requests<2.11.1 (from tweepy)
  Using cached https://files.pythonhosted.org/packages/65/47/7e02164a2a3db506d68a6ab1dd6d0b69c4c3df57a284257925dfc17bda/requests-2.11.1-py3-none-any.whl
Collecting PySocks<1.5.7 (from tweepy)
  Using cached https://files.pythonhosted.org/packages/e6/d1/dd9cfea3e736399b97ded5c2dd62d1322ade4fa72d816f1ed1049d6a179/oauthlib-0.6.2-py3-none-any.whl
Collecting requests-oauthlib<0.7.0->tweepy
  Using cached https://files.pythonhosted.org/packages/e6/01/ddd9cfea3e736399b97ded5c2dd62d1322ade4fa72d816f1ed1049d6a179/oauthlib-0.6.2-py3-none-any.whl
Collecting urllib3<1.24,>=1.21.1 (from requests-oauthlib<0.7.0->tweepy)
  Using cached https://files.pythonhosted.org/packages/bd/c9/6f6dd990019071a4a32a5e7cb78a1d92c53851ef4f56f62a3486e6a7d8ffbf/urllib3-1.24.1-py3-none-any.whl
Collecting chardet<1.0,>=3.0.2 (from requests<2.11.1->tweepy)
  Using cached https://files.pythonhosted.org/packages/bc/a9/01febfb562e42746487b4bb1dde7ca55ec7510b22e4c51f14098443b/chardet-3.0.2-py3-none-any.whl
Collecting idna<2.8,>=2.5 (from requests<2.11.1->tweepy)
  Using cached https://files.pythonhosted.org/packages/4b/2a/0276479a4b3cae8a8c1af2f8e4355746a97fab05a372e4a2c6a6b876165/idna-2.7.0-py3-none-any.whl
Collecting certifi<2017.4.1,>=2.11.1-tweepy
  Using cached https://files.pythonhosted.org/packages/df/f7/04fee6ac349e915b2171f8e23cee63644d3663b34c539f7a09a0ed18f9e/certifi-2017.4.1-py3-none-any.whl
  Downloading https://files.pythonhosted.org/packages/df/f7/04fee6ac349e915b2171f8e23cee63644d3663b34c539f7a09a0ed18f9e/certifi-2017.4.1-py3-none-any.whl (153KB 1.9MB/s)
Installing collected packages: urllib3, chardet, idna, certifi, requests, oauthlib, requests-oauthlib, PySocks, tweepy
Successfully installed PySocks<1.6.8 certifi<2018.8.24 chardet<3.0.4 idna<2.7 oauthlib<2.1.0 requests<2.19.1 requests-oauthlib<1.0.0

[+] aarisa1 ~ /tmp/twitter_sa_dmsn2 dmsn2
>>> python
Python 3.6.5 (default, Jun 17 2018, 12:13:06)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-902.0.39.0)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> nltk.download('vader_lexicon')
[nltk_data] Downloading package vader_lexicon ...
[nltk_data]   Users/aarisa1/nltk_data...
True
>>>
```

```
[+] aarisa1 ~ /tmp/twitter_sa_dmsn2 dmsn2
>>> cat twitter_sa.py
import json
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
from nltk.sentiment.vader import SentimentIntensityAnalyzer

#consumer key, consumer secret, access token, access secret.
consumer_key="IKhgGzD"
consumer_secret="o1x9
access_token="7951749
access_token_secret="

sid = SentimentIntensityAnalyzer()

class Listener(StreamListener):
    def on_data(self, data):
        tweet = json.loads(data)
        if 'delete' not in tweet.keys():
            if(tweet['lang'] == 'en'):
                text = tweet['text']

                sentiment_score = sid.polarity_scores(text)['compound']
                if(sentiment_score < 0):
                    sentiment = 'NEGATIVE'
                elif(sentiment_score > 0):
                    sentiment = 'POSITIVE'
                else:
                    sentiment = 'NEUTRAL'

                print(sentiment + " : " + text)
                print("====")
            return(True)

    def on_error(self, status):
        print(status)

def main():
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)

    twitterStream = Stream(auth, Listener())
    twitterStream.sample()

if __name__ == "__main__":
    main()
```



Text Utilization

```
aarisl  ~/tmp/twitter_sa_dmsn2  dmsn2
>>> vi twitter_sa.py

aarisl  ~/tmp/twitter_sa_dmsn2  dmsn2
>>> python twitter_sa.py
/private/tmp/dmsn2/lib/python3.6/site-packages/nltk/twitter/_init__.py:20: UserWarning: The twython library has not been installed. Some functionality from the tw
  warnings.warn("The twython library has not been installed."
POSITIVE : RT @ChrisEvans: Officially wrapped on Avengers 4. It was an emotional day to say the least. Playing this role over the last 8 years has bee...
=====
POSITIVE : RT @KOKIEHOLY: @BTSTwt I VOTE FOR @BTStwt AS FAVORITE SOCIAL ARTIST #AMAs https://t.co/XAiGBg7X2I
=====
NEGATIVE : RT @june196: @SocialistVoice @comefollowath Nothing sadder & more pathetic than a rich old man thinking his threats of leaving country, ma...
=====
POSITIVE : RT @Hanhan07830939: Super Junior
The Last Man Standing
#TheGroup #SuperJunior #PCAs @SJofficial https://t.co/gauCq4QMgw
=====
POSITIVE : RT @DarkieCastle: And the lion deadass thought this was funny af 😂😂 look at the face he makes https://t.co/Mw0gdS7i3i
=====
POSITIVE : RT @TaylorSPolls: I vote for @taylorswift13 as the Favorite Female Artist Pop/Rock at the #AMAs https://t.co/1mq9K5nRCD
=====
POSITIVE : RT @luluramadan: Retired U.S. Supreme Court Justice John Paul Stevens, a lifelong Republican, told a small crowd in Boca Raton that Judge B...
=====
POSITIVE : RT @Target: 5 love languages
1. Acts of Service: I'm going to Target. Do you want anything?
2. Quality Time: Want to go to Target with me?...
=====
POSITIVE : TGIF 🌟
=====
NEGATIVE : Yesterday's grades

FX USD/JPY 02wins and 1 losses. L0 S2
October FX009wins 01 loss
2018 FX944wins 52 losses
=====
NEGATIVE : RT @MikeJohnson777: If a judge cannot be trusted, he should not be a judge. https://t.co/bpypE0il3d
=====
POSITIVE : RT @Blizzard_Ent: Mike Morhaime is stepping down from Blizzard, to be succeeded as President of Blizzard Entertainment by J. Allen Brack.
=====
NEUTRAL : RT @NCTsmtown_127: 'Regular Dream'

#NCT127-Regular_Irregular
#Regular_Irregular
#NCT127-Regular
#NCT127 #NCT
#NCT127_10월 12일 컴백 https://t...
=====
POSITIVE : RT @w_terrence: Brett Kavanaugh Protesters are having Hissy Fits because the FBI didn't find anything. The only time Democrats create job...
=====
NEGATIVE : @sincerelyvh WHAT THE FUCK WHY ISKOID IM GLAD IM NOT ONE OF THOSE LAKE STEALERS.
```



Thank you

terima kasih

