Exercise 1: Strings and Regular Express World

We covered basics on strings and regular expressions in this module's tutorials. Ir brain muscles to apply those concepts on a real world dataset.

The dataset we are using for this assignment comes from a set of Nigerian fraud e (https://www.kaggle.com/rtatman/fraudulent-email-corpus) For ease of access, the this notebook as well. We have also extracted a sample email from this dataset fo

This dataset consists of 2500 such fraudulent emails. Each email consists of secti receiver's email addresses, date, subject, etc.

Note: Certain cells have blanks to be filled, look replace them with your answers

---

Import Required Libraries

In [1]:
```
import re
```

Load Sample email

In [2]:
```
sample_email = open(r"Data/sample_email.txt", "r").read()
```

In [3]:
```
# print a few lines
for line in sample_email.split("\n")[:10]:
    print(line)
```

```
From r  Wed Oct 30 21:41:56 2002
Return-Path: <james_ngola2002@maktoob.com>
X-Sieve: cmu-sieve 2.0
Return-Path: <james_ngola2002@maktoob.com>
Message-Id: <200210310241.g9V2fNm6028281@cs.CU>
From: "MR. JAMES NGOLA." <james_ngola2002@maktoob.com>
Reply-To: james_ngola2002@maktoob.com
To: webmaster@aclweb.org
Date: Thu, 31 Oct 2002 02:38:20 +0000
Subject: URGENT BUSINESS ASSISTANCE AND PARTNERSHIP
```

Load Complete Corpus

In [4]:
```
email_corpus = open(r"Data/fradulent_emails.txt", "r",encoding='latin-1'
len(email_corpus)
```

```
17344435
```

In [5]:
```
# print(email_corpus)
print(email_corpus[0:100])
```

```
From r  Wed Oct 30 21:41:56 2002
Return-Path: <james_ngola2002@maktoob.com>
X-Sieve: cmu-sieve 2.0
R
```

Question 1: Prepare a list consisting of **Subjec** Email using String functions (1 point)

hint: use split() function

In [6]:
```
subject_lines = list()
for line in email_corpus.split("\n"):
    if "Subject:" in line:
        subject_lines.append(line)

subject_lines2 = list(map(lambda x: x.replace('Subject: ',''), subject_li
```

Question 2: Print number of Subject Lines capt question (1 point)

hint: Use string formatting and length utilities

In [7]:
```
print("Total number of subject lines captured {}".format(len(subject_line
```

```
Total number of subject lines captured 4268
```

Question 3: Transform Upper-Case subject line

point)

Print the first 5 values in the list

hint:use string functions

In [8]:
```
titlecase_subject_lines = [subj.title() for subj in subject_lines2]
print("Transformed Subject Lines:")
print(titlecase_subject_lines[:5])
```

```
Transformed Subject Lines:
['Urgent Business Assistance And Partnership', 'Urgent Assistance /Relationship (P)', 'Good
Need Your Assistance.']
```

In [9]:
```
for i in range(5):
    print(titlecase_subject_lines[i])
```

```
Urgent Business Assistance And Partnership
Urgent Assistance /Relationship (P)
Good Day To You
Good Day To You
I Need Your Assistance.
```

Question 4: Given a sample Subject line, print i
point)

hint:use negative indexing

In [10]:
```
sample_subject = "Subject: URGENT BUSINESS ASSISTANCE AND PARTNERSHIP"
print(sample_subject[::-1])
```

```
PIHSRENTRAP DNA ECNATSISSA SSENISUB TNEGRU :tcejbuS
```

Question 5: Merge two given sentences using '

*hint: string utility to **join** strings together*

In [11]:
```
sentence_1 = "YOU WERE INTRODUCED TO ME BY A RELIABLE FRIEND OF MINE WHO
sentence_2 = "AND ALSO A MEMBER OF CHAMBER OF COMMERCE AS A RELIABLE AND

merged_sentence = ', '.join([sentence_1, sentence_2])
print(merged_sentence)
```

```
YOU WERE INTRODUCED TO ME BY A RELIABLE FRIEND OF MINE WHO IS A TRAVELLER, AND ALSO A MEMBER
LE AND TRUSTWORTHY PERSON.
```

Question 6: Check if **$** symbols existing in Sam point)

*hint: use **in** keyword*

In [12]:
```
if '$' in sample_email:
    print("Yes, it exists")
```

```
Yes, it exists
```

Question 7: Check if a string is Alphanumeric o

hint: use string utility methods

In [13]:
```
sample_string = "From: MR. JAMES NGOLA. <james_ngola2002@maktoob.com>"
if sample_string.isalnum():
    print("Yes, this string is alpha-numeric")
else:
    print("No this string is not alpha-numeric")
```

```
No this string is not alpha-numeric
```

Question 8: Extract Time from Date-time field (

In [14]:
```
sample_date = "Fri, 1 Nov 2002 01:45:04 +0100"
```

In [15]:
```
## Separate date and time portions
import datetime
```

```
date_time_obj = datetime.datetime.strptime(sample_date, "%a, %d %b %Y %I
```

In [16]:

```
print(date_time_obj)
type(date_time_obj)
print(datetime.datetime.date(date_time_obj))
```

```
2002-11-01 01:45:04+01:00
2002-11-01
```

In [17]:

```
## print both segments
print("Date={}".format(datetime.datetime.date(date_time_obj)))
print("Time={}".format(datetime.datetime.time(date_time_obj)))
```

```
Date=2002-11-01
Time=01:45:04
```

Question 9: Extract the **From:....<....>** field of e point)

You should be able to extract the entire line starting with **From:** till end of the line

hint: use a regex utility to find all matches

In [18]:

```
iters = re.findall(r"^From:.*", email_corpus, re.MULTILINE)
for line in iters:
    print(line)
```

```
From: "MR. JAMES NGOLA." <james_ngola2002@maktoob.com>
From: "Mr. Ben Suleman" <bensul2004nng@spinfinder.com>
From: "PRINCE OBONG ELEME" <obong_715@epatra.com>
From: "PRINCE OBONG ELEME" <obong_715@epatra.com>
From: "Maryam Abacha" <m_abacha03@www.com>
From: Kuta David <davidkuta@postmark.net>
```

Question 10: Extract the To Email Addresses

***step 1**: find the lines with "**To**" information*

In [19]:
```
match = re.findall(r"^To:.*", email_corpus, re.MULTILINE)
for line in match:
    print(line)
```

```
To: webmaster@aclweb.org
To: R@M
To: webmaster@aclweb.org
To: webmaster@aclweb.org
To: R@M
To: davidkuta@yahoo.com
To: webmaster@aclweb.org
To: R@M
To: R@E
To: R@M
To: webmaster@aclweb.org
To: R@M
To: R@E
To: webmaster@aclweb.org
To: webmaster@aclweb.org
To: webmaster@aclweb.org
To: ntcir-listmem@newns.op.nii.ac.jp
To: ntcir-outgoing@nii.ac.jp
To: webmaster@aclweb.org
To: oshea@UM
To: webmaster@aclweb.org
To: webmaster@aclweb.org
To: webmaster@aclweb.org
To: R@M
```

***step 2**: iterate all matches to extract only email addresses*

***step 2**:replace ___ with a symbol most commonly found in email addresses*

In [20]:
```
for line in match:
    print(re.findall(r"To:\s*\b(.*.*)\b", line))
```

```
['webmaster@aclweb.org']
['R@M']
['webmaster@aclweb.org']
['webmaster@aclweb.org']
['R@M']
['davidkuta@yahoo.com']
['webmaster@aclweb.org']
['R@M']
['R@E']
['R@M']
['webmaster@aclweb.org']
['R@M']
['R@E']
['webmaster@aclweb.org']
['webmaster@aclweb.org']
['webmaster@aclweb.org']
['ntcir-listmem@newns.op.nii.ac.jp']
['ntcir-outgoing@nii.ac.jp']
['webmaster@aclweb.org']
['oshea@UM']
```

Explanation

Decoding the search pattern:

- a. To:\s* searches for lines beginning with To followed by : and optional space
- b. \b introduces word boundaries in the search pattern
- c. .* (wildcard) allows for multiple combinations of characters to be found

Question 11: Extract **Domains** from Email Add point)

Example: abc@**def.com** : here *def.com* is the domain

step 1*: iterate all matches*

step 2*: look for emails in the matched rows*

step 3*: break the email addresses using the "@" symbol using a utility from re*

hint: use the email search pattern from previous question

In [21]:
```
match = re.findall(r"^To:.*", email_corpus, re.MULTILINE)
for line in match:
    try:
        line2 = re.sub("To:", "", line)
        line2_list = line2.split("@")
        print("Username: {}\t\t\tDomain Name: {}".format(line2_list[0],
```

```
        except:
              print("Error [probably multiple-addresses] for={}".format(line
```

```
Username:  webmaster                  Domain Name: aclweb.org
Username:  R                      Domain Name: M
Username:  webmaster                  Domain Name: aclweb.org
Username:  webmaster                  Domain Name: aclweb.org
Username:  R                      Domain Name: M
Username:  davidkuta                  Domain Name: yahoo.com
Username:  webmaster                  Domain Name: aclweb.org
Username:  R                      Domain Name: M
Username:  R                      Domain Name: E
Username:  R                      Domain Name: M
Username:  webmaster                  Domain Name: aclweb.org
Username:  R                      Domain Name: M
Username:  R                      Domain Name: E
Username:  webmaster                  Domain Name: aclweb.org
Username:  webmaster                  Domain Name: aclweb.org
Username:  webmaster                  Domain Name: aclweb.org
Username:  ntcir-listmem                   Domain Name: newns.op.nii.ac.jp
Username:  ntcir-outgoing                  Domain Name: nii.ac.jp
Username:  webmaster                  Domain Name: aclweb.org
Username:  oshea                      Domain Name: UM
Username:  webmaster                  Domain Name: aclweb.org
Username:  webmaster                  Domain Name: aclweb.org
Username:  webmaster                  Domain Name: aclweb.org
```

In [22]:
```
#for line in match:
#    # hint: use the email search pattern
#    for email in re.findall(r"^To:.*", email_corpus, re.MULTILINE):
#        try:
#          username, domain_name = re.____(____, email)
#          print("{}, {}".format(____, ____))
#        except:
#          print("Error [probably multiple-addresses] for={}".format(lin
```

Question 12: Replace Field **Date** with **Sent On**

In [23]:
```
date_field = re.search("Date:.*", email_corpus)
original = date_field.group()
substitute = re.sub("Date", "Sent On", original)
print(original)
print(substitute)
```

```
Date: Thu, 31 Oct 2002 02:38:20 +0000
Sent On: Thu, 31 Oct 2002 02:38:20 +0000
```

Question 13: Find all Replacements (2 point)

In [24]:
```
sample_string = """
Please note that this transaction is 100% safe.
Please acknowledge reciept of this letter using this link.
Please get in touch with me via my private email address
"""

replacement_pattern = "Kindly"
```

In [25]:
```
replaced_text = re.subn("Please", "Kindly",
                        sample_string,
                        flags = re.IGNORECASE)[0]

print(replaced_text)
```

```
Kindly note that this transaction is 100% safe.
Kindly acknowledge reciept of this letter using this link.
Kindly get in touch with me via my private email address
```

Question 14: Find all Sentences that start with

hint: use anchors

In [26]:
```
iters = re.findall(r"^Dear:.*", email_corpus, re.MULTILINE)
for i in iters:
    print(i)
```

```
Dear: Good Day
Dear:
Dear:Sir/Ma,
Dear:Sir
```

In [27]:
```
for line in email_corpus.split("\n"):
    if re.findall(r"^Dear:.*", line):
```

```
            print(line)
```

```
Dear: Good Day
Dear:
Dear:Sir/Ma,
Dear:Sir
```

Question 15: Find Synonymns (2 point)

hint: use wordnet

In [28]:
```
import nltk
nltk.download('wordnet')

# load the Wordnet Corpus
from nltk.corpus import wordnet as wn
```

```
[nltk_data] Downloading package wordnet to /Users/fritz/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

In [29]:
```
word = 'email'

# get word synsets
word_synsets = wn.synsets(word)
word_synsets
```

```
[Synset('electronic_mail.n.01'), Synset('e-mail.v.01')]
```

In [30]:
```
word = 'offer'

# get word synsets
word_synsets = wn.synsets(word)
word_synsets
```

```
[Synset('offer.n.01'),
 Synset('offer.n.02'),
 Synset('crack.n.09'),
 Synset('offer.v.01'),
 Synset('offer.v.02'),
 Synset('volunteer.v.02'),
 Synset('offer.v.04'),
 Synset('offer.v.05'),
 Synset('offer.v.06')
```