Assignment 1

For this assignment you are welcomed to use other regex resources such a regex "cheat sheets" you find on the web. Feel free to share good resources with your peers in slack!

Before start working on the problems, here is a small example to help you understand how to write your own answers. In short, the solution should be written within the function body given, and the final result should be returned. Then the autograder will try to call the function and validate your returned result accordingly.

In [ ]:

```
def example_word_count():
    # This example question requires counting words in the example_string below.
    example_string = "Amy is 5 years old"

    # YOUR CODE HERE.
    # You should write your solution here, and return your result, you can comme
nt out or delete the
    # NotImplementedError below.
    result = example_string.split(" ")
    return len(result)

    #raise NotImplementedError()
```

Part A

Find a list of all of the names in the following string using regex.

In [5]:

```
import re
def names():
    simple_string = """Amy is 5 years old, and her sister Mary is 2 years old.
    Ruth and Peter, their parents, have 3 kids."""

    # YOUR CODE HERE
     #What I did: [a-z]* represents any possible character, so I assert a capita
l A-Z and 1 or more lowercase chars.
    #The name end is detected when there is a space in the character or when a c
omma is detected.
    result = re.findall('([A-Z][a-z]+)(?=\s|[,])',simple_string)
    return result
    raise NotImplementedError()
len(names())
```

Out[5]:

```
4
```

In [ ]:

```
assert len(names()) == 4, "There are four names in the simple_string"
```

Part B

The dataset file in assets/grades.txt (assets/grades.txt) contains a line separated list of people with their grade in a class. Create a regex to generate a list of just those students who received a B in the course.

In [4]:

```
import re
def grades():
    with open ("assets/grades.txt", "r") as file:
        grades = file.read()

    # YOUR CODE HERE
    result = re.findall('([A-Z][a-z]+\s[A-Z][a-z]+)(?=[:]\s[B])', grades) # Firs
tname (space) Lastname
    return result
    raise NotImplementedError()
len(grades())
```

Out[4]:

```
16
```

In [ ]:

```
assert len(grades()) == 16
```

Part C

Consider the standard web log file in assets/logdata.txt (assets/logdata.txt). This file records the access a user makes when visiting a web page (like this one!). Each line of the log has the following items:

- a host (e.g., '146.204.224.152')
- a user_name (e.g., 'feest6811' **note: sometimes the user name is missing! In this case, use '-' as the value for the username.**)
- the time a request was made (e.g., '21/Jun/2019:15:45:24 -0700')
- the post request type (e.g., 'POST /incentivize HTTP/1.1' **note: not everything is a POST!**)

Your task is to convert this into a list of dictionaries, where each dictionary looks like the following:

```
example_dict = {"host":"146.204.224.152",
                "user_name":"feest6811",
                "time":"21/Jun/2019:15:45:24 -0700",
                "request":"POST /incentivize HTTP/1.1"}
```

In [1]:

```
import re
def logs():
    with open("assets/logdata.txt", "r") as file:
        logdata = file.read()
    pattern = """#multiline string
    (?P<host>[0-9]+[.][0-9]+[.][0-9]+[.][0-9]+)#host key and value
    (\ -\ )#junk characters in between host and username
    (?P<user_name>[a-z]*[0-9]*|-)#username key and value
    (\s\[)#junk characters in between username and timestamp
    (?P<time>\d+[\/]\w+[\/]\d+[:]\d+[:]\d+[:]\d+\s[-]\d+)#timestamp key and valu
e
    (\]\s["])#junk characters
    (?P<request>[A-Z]+\ \/.*\/\d[.]\d)#request key and value
    """
    result = list()
    for item in re.finditer(pattern,logdata, re.VERBOSE):
        result.append(item.groupdict())
    return result
    # YOUR CODE HERE
    raise NotImplementedError()
len(logs())
```

Out[1]:

```
979
```

In [ ]:

```
assert len(logs()) == 979

one_item={'host': '146.204.224.152',
  'user_name': 'feest6811',
  'time': '21/Jun/2019:15:45:24 -0700',
  'request': 'POST /incentivize HTTP/1.1'}
assert one_item in logs(), "Sorry, this item should be in the log results, check
your formating"
```