Assignment 1

For this assignment you are welcomed to use other regex resources such a regex "cheat sheets" you find on the web.

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 [1]:
```
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 comment 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 [2]:
```
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

    # return  all the names in the simple_string
    names = re.findall('[A-Z][\w]*', simple_string)

    return (names)

    raise NotImplementedError()
```

In [3]:
```
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

    names_with_B_grade = []

    # return student names with B grade
    for item in re.finditer('(?P<name>.*)(\: B)',grades):

        names_with_B_grade.append(item.group('name'))

    return names_with_B_grade
    raise NotImplementedError()
```

In [5]:
```
assert len(grades()) == 16
```

Part C

Consider the standard web log file in [assets/logdata.txt (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 [6]:

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

    # YOUR CODE HERE

    logs_dict = []

    # Regex
    pattern="""
    (?P<host>.*)                # host
    (\ -\ )
    (?P<user_name>.*)           # user name
    (\ \[)
    (?P<time>.*)                # time
    (\]\ \")
    (?P<request>.*)             # request
    (\")"""

    for item in re.finditer(pattern,logdata,re.VERBOSE):

        # We can get the dictionary returned for the item with .groupdict()
        logs_dict.append(item.groupdict())

    return (logs_dict)

    raise NotImplementedError()
```

In [7]:

```python
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"
```