

# assignment1

January 3, 2021

## 1 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.

```
[41]: 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()
```

### 1.1 Part A

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

```
[54]: 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.""  
    names = []  
    for children_name in re.findall("([\w]*) (\ is)", simple_string):  
        names.append(children_name[0])  
  
    for parent1 in re.findall("[a-zA-Z]{1,100} (\ and)", simple_string):  
        names.append(parent1[0])  
  
    for parent2 in re.findall("[a-zA-Z]{1,100} (\, their)", simple_string):
```

```
names.append(parent2[0])

return names
raise NotImplementedError()
names()
```

[54]: ['Amy', 'Mary', 'Ruth', 'Peter']

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

## 1.2 Part B

The dataset file in `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.

```
[69]: import re
def grades():

    grades = open("assets/grades.txt", "r")
    dict_grades = {}
    student_names = []
    for line in grades.readlines():
        line = line.strip()
        name = line.split(": ")[0]
        gpa = line.split(": ")[1]
        dict_grades[name] = gpa

    for name, gpa in dict_grades.items():
        if gpa == 'B':
            student_names.append(name)

    return student_names

    raise NotImplementedError()
grades()
```

[69]: ['Bell Kassulke',  
'Simon Loidl',  
'Elias Jovanovic',  
'Hakim Botros',  
'Emilie Lorentsen',  
'Jake Wood',  
'Fatemeh Akhtar',  
'Kim Weston',  
'Yasmin Dar',  
'Viswamitra Upandhye',  
'Killian Kaufman',  
'Elwood Page',  
'Elodie Booker',

```
'Adnan Chen',  
'Hank Spinka',  
'Hannah Bayer']
```

```
[70]: assert len(grades()) == 16
```

### 1.3 Part C

Consider the standard web log file in `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"}
```

```
[98]: import re  
def logs():  
    answer = []  
    with open("assets/logdata.txt", "r") as file:  
        logdata = file.read()  
  
    pattern="""  
    (?P<host>.* )  
    (\ -\ )  
    (?P<user_name>.* )  
    (\ \[ )  
    (?P<time>.* )          # time: start from espace[  
    (\]\ \")  
    (?P<request>.*)(\")    # request: start from ] ", end with "  
    """"  
  
    for item in re.finditer(pattern, logdata, re.VERBOSE):  
        answer.append(item.groupdict())  
  
    return answer  
    raise NotImplementedError()  
  
#logs()
```

```
[99]: 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"
```

[ ]: