```
Refactoring Activity
Paul ReFalo 11/19/17

    For this activity I choose chapter 7's assignment for the ISS viewing tim
when completed it the first time that I was reusing code to in two places to
API succeeded or failed.  I refactored this code to using one method instead
code in two places.
    I also choose to refactor the way I was checking for the API success/fail
had originally wrote called the API with get and then I checked to see if I g
in text form from that response.  If so, this was judged a successful API cal
this, I used the demonstration code the instructor gave which makes use of th
rather than checking for response.text.  This seems much cleaner than the way
originally done it.

==================    New / Refactored Code ======================

# CS 519 assignment 7 - ISS.py "When can I see the ISS" by Paul ReFalo
# View | Tool Windows | Terminal
# python3 ISSrefactored.py "Portland, OR" 3

import sys
import requests
import json
from pprint import pprint
import time

# Global variables to hold lat and long
lat = 0
long = 0

address = sys.argv[1]   # address of location from user input
n = int(sys.argv[2])    # number of desired results

# get Request function takes url, headers, query and returns API response if
def getRequest(requestURL, requestHeaders, requestQueryString):
    error_count = 0
    success = False

    while not success and error_count < 3:
      #response = requests.request("GET", requestURL)
      getResponse = requests.get(requestURL, headers=requestHeaders, params=r
      if getResponse.status_code < 400:
        success = True
      else:
        print("===== API failed.  Retrying it now...standby. =====")
        error_count += 1
```

```
47
48      if success:
49          return getResponse
50      else:
51          print("Too many errors, giving up")
52          sys.exit()
53
54  # use input from user to get Lat and Long of address
55  googleUrl = "https://maps.googleapis.com/maps/api/geocode/json"     # set up
56  googleQueryString = {"address":address}                             # config
57  googleHeaders = { }                                                 # no head
58
59  googleResponse = getRequest(googleUrl, googleHeaders, googleQueryString)
60  googleResults = json.loads(googleResponse.text)  # use loads to get json text
61
62
63  for gr in googleResults["results"]:                       # loop and extract la
64      lat = gr["geometry"]["location"]["lat"]
65      long = gr["geometry"]["location"]["lng"]
66
67  # use lat and long to get ISS viewing times
68  issUrl = "http://api.open-notify.org/iss-pass.json"     # set up ISS url
69  issHeaders = { }                                        # no headers needed
70  issQueryString = {"lat":lat, "lon":long, "n":n}         # config queryString
71
72  issResponse = getRequest(issUrl, issHeaders, issQueryString)
73  issResults = json.loads(issResponse.text)  # use loads to get json text
74
75  issResultsArray = []                     # set up array to hold ISS results
76  for issR in issResults["response"]:      # loop to get individual results
77      duration = issR["duration"]          # extract duration in seconds
78      risetime = issR["risetime"]          # extract risetime in seconds since e
79      risetime = time.strftime('%a %b %d %H:%M:%S %Y', time.localtime(risetime)
80
81      issResultsArray.append((duration, risetime)) # append to results array
82
83  print("From " + address + " you will be able to see the ISS on:")
84  for idx, e in enumerate(issResultsArray):       # loop to display results in
85      if idx >= n:                                # break if length(issResultsA
86          break                                   # this shouldn't happen or be
87      print(str(e[1]) + " for " + str(e[0]) + " seconds")     # print results
88
89
90  '''
91  Output demo:
92  MacBook-Pro:week-7-files paulrefalo$ python ISS.py "Santa Cruz, CA" 3
```

```
From Salem, OR you will be able to see the ISS on:
Sun Nov 19 19:57:16 2017 for 352 seconds
Sun Nov 19 21:30:39 2017 for 622 seconds
Sun Nov 19 23:07:01 2017 for 633 seconds

if API fails you might get:
===== API failed.  Retrying it now...standby. =====
===== API failed.  Retrying it now...standby. =====
From Salem, OR you will be able to see the ISS on:
Sun Nov 19 19:57:16 2017 for 352 seconds
Sun Nov 19 21:30:39 2017 for 622 seconds
Sun Nov 19 23:07:01 2017 for 633 seconds

==================    Original Code ======================

# CS 519 Refactoring assignment - ISSrefactored.py "When can I see the ISS" b

import sys
import requests
import json
from pprint import pprint
import time

# Global variables to hold lat and long
lat = 0
long = 0

address = sys.argv[1]   # address of location from user input
n = int(sys.argv[2])    # number of desired results

# use input from user to get Lat and Long of address
googleUrl = "https://maps.googleapis.com/maps/api/geocode/json"     # set up
googleQueryString = {"address":address}                             # config
googleHeaders = { }                                                 # no head

googleAPI = False
for i in range(5):      # give API a few chances to succeed.  Abort script in
    if googleAPI:
        break
    for attempt in range(5):
        try:
            response = requests.get(googleUrl, headers=googleHeaders, params=
            googleResults = json.loads(response.text)  # use loads to get jso

            if googleResults["results"]:
                googleAPI = True
```

```
                break
        except ConnectionError:
            continue
        else:
            print("===== Google API failed.  Retrying it now...standby. =====
            break
    else:
        print("Google geolocation failed.  Try script again or check connecti
        sys.exit()

#print(response.text)


for gr in googleResults["results"]:                    # loop and extract la
    lat = gr["geometry"]["location"]["lat"]
    long = gr["geometry"]["location"]["lng"]

# use lat and long to get ISS viewing times
issUrl = "http://api.open-notify.org/iss-pass.json"    # set up ISS url
issHeaders = { }                                       # no headers needed
issQueryString = {"lat":lat, "lon":long, "n":n}        # config queryString

issAPI = False
for i in range(5):      # give API a few chances to succeed.  Abort script in
    if issAPI:
        break
    for attempt in range(5):
        try:
            issResponse = requests.get(issUrl, headers=issHeaders, params=iss
            issResults = json.loads(issResponse.text)  # use loads to get jso

            if issResults["response"]:
                issAPI = True
                break
        except ConnectionError:
            continue
        else:
            print("===== ISS API failed.  Retrying it now...standby.  =====")
            break
    else:
        print("ISS Api failed.  Try script again or check connection.")
        sys.exit()

#print(issResponse.text)

issResultsArray = []                   # set up array to hold ISS results
```

```
for issR in issResults["response"]:     # loop to get individual results
    duration = issR["duration"]         # extract duration in seconds
    risetime = issR["risetime"]         # extract risetime in seconds since e
    risetime = time.strftime('%a %b %d %H:%M:%S %Y', time.localtime(risetime)

    issResultsArray.append((duration, risetime)) # append to results array

print("You will be able to see the ISS on:")
for idx, e in enumerate(issResultsArray):       # loop to display results in
    if idx >= n:                                # break if length(issResultsA
        break                                   # this shouldn't happen or be
    print(str(e[1]) + " for " + str(e[0]) + " seconds")     # print results



'''
Output demo:
MacBook-Pro:week-7-files paulrefalo$ python ISS.py "Santa Cruz, CA" 3
You will be able to see the ISS on:
Sat Nov 04 10:52:18 2017 for 625 seconds
Sat Nov 04 12:28:58 2017 for 582 seconds
Sun Nov 05 01:59:17 2017 for 103 seconds

if API fails you might get:
===== Google API failed.  Retrying it now...standby. =====
===== Google API failed.  Retrying it now...standby. =====
You will be able to see the ISS on:
Sat Nov 04 10:52:18 2017 for 625 seconds
Sat Nov 04 12:28:58 2017 for 582 seconds
Sun Nov 05 01:59:17 2017 for 103 seconds

'''

'''
```