```
 1 # -*- coding: utf-8 -*-
 2
 3 # Sergei Bugrov
 4 # 7-9-17
 5 #
 6 # Downloads all available books in English language in
    .txt format from http://www.gutenberg.org,
 7 # unpacks them from .zip archives, saves them to ../
   books/ folder, and deletes .zip files.
 8 #
 9 # usage : python gutenberg.py
10 #
11 # python version : 3.6.1
12
13 import requests, bs4, os, errno, zipfile, glob
14 from urllib.request import urlretrieve
15
16
17 def main():
18
19     if not os.path.exists('books/'):
20         try:
21             os.makedirs('books/')
22         except OSError as e:
23             if e.errno != errno.EEXIST:
24                 raise
25
26     # STEP 1. BUILD A LIST OF URLS
27
28     urls_to_books = []
29
30     if not os.path.exists('urls_to_books.txt'):
31
32         page_w_books_url = 'http://www.gutenberg.org/
   robot/harvest?filetypes[]=txt&langs[]=en'
33
34         while 1 == 1:
35
```

```
36              is_last_page = False
37
38              print('Reading page: ' + page_w_books_url)
39
40              page_w_books = requests.get(
   page_w_books_url, timeout=20.0)
41
42              if page_w_books:
43                  page_w_books = bs4.BeautifulSoup(
   page_w_books.text, "lxml")
44                  urls = [el.get('href') for el in
   page_w_books.select('body > p > a[href^="http://aleph.
   gutenberg.org/"]')]
45                  url_to_next_page = page_w_books.
   find_all('a', string='Next Page')
46
47                  if len(urls) > 0:
48                      urls_to_books.append(urls)
49
50                      if url_to_next_page[0]:
51                          page_w_books_url = "http://www.
   gutenberg.org/robot/" + url_to_next_page[0].get('href')
52                  else:
53                      is_last_page = True
54
55              if is_last_page:
56                  break
57
58          urls_to_books = [item for sublist in
   urls_to_books for item in sublist]
59
60          # Backing up the list of URLs
61          with open('urls_to_books.txt', 'w') as output:
62              for u in urls_to_books:
63                  output.write('%s\n' % u)
64
65      # STEP 2. DOWNLOAD BOOKS
66
```

```
67      # If, at some point, Step 2 is interrupted due to unforeseen
68      # circumstances (power outage, lost of internet connection), replace the number
69      # (value of the variable url_num) below with the one you will find in the logfile.log
70      # Example
71      #       logfile.log : Unzipping file #99 books/10020.zip
72      #       the number  : 99
73      url_num = 0
74
75      if os.path.exists('urls_to_books.txt') and len(urls_to_books) == 0:
76          with open('urls_to_books.txt', 'r') as f:
77              urls_to_books = f.read().splitlines()
78
79      for url in urls_to_books[url_num:]:
80
81          dst = 'books/' + url.split('/')[-1].split('.')[0].split('-')[0]
82
83          with open('logfile.log', 'w') as f:
84              f.write('Unzipping file #' + str(url_num) + ' ' + dst + '.zip' + '\n')
85
86          if len(glob.glob(dst + '*')) == 0:
87              urlretrieve(url, dst + '.zip')
88
89              with zipfile.ZipFile(dst + '.zip', "r") as zip_ref:
90                  try:
91                      zip_ref.extractall("books/")
92                      print(str(url_num) + ' ' + dst + '.zip ' + 'unzipped successfully!')
93                  except NotImplementedError:
94                      print(str(url_num) + ' Cannot unzip file:', dst)
```

```
 95
 96             os.remove(dst + '.zip')
 97
 98         url_num += 1
 99
100
101 if __name__ == '__main__':
102     """
103     The main function is called when gutenberg.py is
    run from the command line
104     """
105
106     main()
```