ПРОГРАММА ДЛЯ ЭВМ

Утилита для скачивания и анализа сообщений с форума velomania.ru

Фрагменты исходного текста программы

Листов _4_

Правообладатель-автор: Трофимов Владислав Александрович

г. Санкт-Петербург
2016 г.

```
import re
import csv

import requests
from bs4 import BeautifulSoup

__author__ = 'Stranger'

thread_url = 'http://forum.velomania.ru/showthread.php?t=74626'
image_label = '**IMAGE**'
video_label = '**VIDEO**'
output_file = 'thread1.tsv'

def extract_data_from_url(url):
    html = requests.get(url).content
    soup = BeautifulSoup(html, 'html.parser')
    posts = soup.findAll('div', {'class': 'postdetails'})

    extracted = []

    for post in posts:
        author = get_author(post)
        content = get_content(post)
        contains_video = video_label in content
        contains_images = image_label in content
        extracted.append((url, author, content, contains_images, contains_video))

    return extracted


def get_content(post):
    raw_content = post.find('div', {'class': 'content'}).prettify()
    pre_content = remove_tags(raw_content)
    pre_content = linearize_and_remove_trash(pre_content)
    pre_content = replace_images_and_video_with_label(pre_content)
    pre_content = linearize(pre_content)
    # pre_content = highlight_links(pre_content)
    content = remove_nonprintable_chars(pre_content)
    return content


def get_author(post):
    author_block = post.find('a', {'class': 'username'})
    author = author_block.find('strong').contents[0]
    if not isinstance(author, str):
```

```
        author = author.contents[0]
    return author


def remove_nonprintable_chars(pre_content):
    content = re.sub('[\u200e\ufffd\u0301\u0394\u221e]', ' ', pre_content)
    return content


def highlight_links(pre_content):
    content = re.sub('http://', '\nhttp://', pre_content)
    return content


def linearize(pre_content):
    pre_content = re.sub('\s+', ' ', pre_content)
    return pre_content


def replace_images_and_video_with_label(pre_content):
    pre_content = re.sub('<img alt=', ' ' + image_label + ' ', pre_content)
    pre_content = re.sub('http://video.yandex.ru.* ', ' ' + video_label + ' ',
pre_content)
    pre_content = re.sub('http://.*you.* ', ' ' + video_label + ' ', pre_content)
    pre_content = re.sub('http://.*\.jpg.* ', ' ' + image_label + ' ', pre_content)
    pre_content = re.sub('http://.*\.png.* ', ' ' + image_label + ' ', pre_content)
    pre_content = re.sub('http://.*\.JPG.* ', ' ' + image_label + ' ', pre_content)
    pre_content = re.sub('http://`.*\.PNG.* ', ' ' + image_label + ' ', pre_content)
    pre_content = re.sub('http://photofile.* ', ' ', pre_content)
    pre_content = re.sub('\w+\.png', ' ' + image_label + ' ', pre_content)
    return pre_content


def linearize_and_remove_trash(pre_content):
    pre_content = re.sub('\n', ' ', pre_content)
    pre_content = re.sub(('\n'
                          '|<!-->'
                          '|<embed width='
                          '|&gt;|&lt;'), ' ', pre_content)
    return pre_content


def remove_tags(raw_content):
    pre_content = re.sub(('<br/>|<br>'
                          '|<div.*>|</div>'
```

```
                    '|<i>|</i>'
                    '|<b>|</b>'
                    '|<ul>|</ul>'
                    '|<li.*>|</li>'
                    '|<ol.*>|</ol>'
                    '|<object.*>|</object>'
                    '|<legend>|</legend>'
                    '|<strike>|</strike>'
                    '|<u>|</u>'
                    '|<span.*>|</span>'
                    '|<strong>|</strong>'
                    '|<param.*>|</param>'
                    '|<font.*>|</font>'
                    '|<fieldset.*>|</fieldset>'
                    '|</blockquote>|<blockquote.*>'
                    '|<img alt=\"Цитата\".*>'
                    '|<a href=\"showthread.php.*>'
                    '|<a href=\"'
                    '|".*>'
                    '|http://.{29}\.\.\..{14}'
                    '|<img .*src=\"images.*>'
                    '|&lt;.*&gt;'
                    '|<!--.*-->'
                    '|</a>'
                    '|\[.*\]'), '', raw_content)
    return pre_content


def get_all_pages_url():
    html = requests.get(thread_url).content
    soup = BeautifulSoup(html, 'html.parser')
    last_page_url = soup.find('span', {'class': 'first_last'}).find('a')['href']
    matches = re.match('showthread\.php.*&page=(\d+).*', last_page_url)

    if matches:
        last_page_number = int(matches.group(1))
    else:
        raise Exception('can\'t parse last page number')

    pages = list()

    for current_page_number in range(1, last_page_number + 1):
        current_page_url = thread_url + '&page=' + str(current_page_number)
        pages.append(current_page_url)
```

```
    return pages


def print_to_csv(data):
    with open(output_file, 'w', newline='') as csvfile:
        csvwriter = csv.writer(csvfile, delimiter='\t')
        for row in data:
            csvwriter.writerow(list(row))


def main():
    data = []
    pages = get_all_pages_url()

    for url in pages:
        print('Parsing page ' + url + ' of ' + str(len(pages)), end='\r')
        data += extract_data_from_url(url)

    print_to_csv(data)


if __name__ == '__main__':
    main()
```

Всего пронумеровано и прошнуровано
_4_ листов фрагментов исходного текста программы

Правообладатель

_____ /____Трофимов В.А._
подпись