Python Script 10: Collecting one million website links (/post/545/python-script-10-collecting-one-million-website-links/)

scraping (/tag/scraping/) beautifulsoup (/tag/beautifulsoup/) 0 2055

I needed a collection of different website links to experiment with Docker cluster. So I created this small script to collect one million website URLs.

Code is available on Github (https://github.com/anuragrana/Python-Scripts) too.

Running script:

Either create a new virtual environment using python3 or use existing one in your system.
Install the dependencies.

```
pip install requests, BeautifulSoup
```

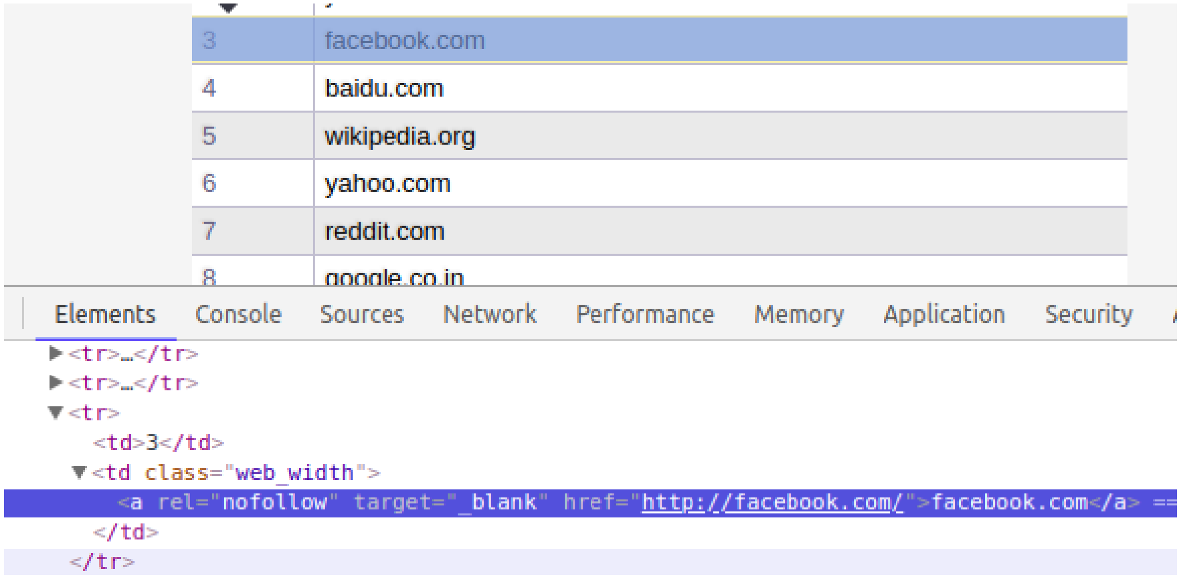
Activate the virtual environment and run the code.

```
python one_million_websites.py
```

Code:

```
import requests
from bs4 import BeautifulSoup
import sys
import time


headers = {
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8",
    "Accept-Language": "en-GB,en-US;q=0.9,en;q=0.8",
    "User-Agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromiu
}

site_link_count = 0

for i in range(1, 201):
    url = "http://websitelists.in/website-list-" + str(i) + ".html"
    response = requests.get(url, headers = headers)
    if response.status_code != 200:
        print(url + str(response.status_code))
        continue

    soup = BeautifulSoup(response.text, 'lxml')
    sites = soup.find_all("td",{"class": "web_width"})

    links = ""
    for site in sites:
        site = site.find("a")["href"]
        links += site + "\n"
        site_link_count += 1

    with open("one_million_websites.txt", "a") as f:
        f.write(links)

    print(str(site_link_count) + " links found")

    time.sleep(1)
```

We are scraping links from site `http://www.websitelists.in/` . If you inspect the webpage, you can see anchor tag inside `td` tag with class `web_width` .
We will convert the page response into BeautifulSoup object and get all such elements and extract the `HREF` value of them.



3	facebook.com
4	baidu.com
5	wikipedia.org
6	yahoo.com
7	reddit.com
8	google.co.in

```
▶<tr>…</tr>
▶<tr>…</tr>
▼<tr>
   <td>3</td>
 ▼<td class="web_width">
     <a rel="nofollow" target="_blank" href="http://facebook.com/">facebook.com</a> ==
   </td>
 </tr>
```

Although there is natural delay of more than 1 second between consecutive requests which is pretty slow but is good for server. I still introduced one second delay to avoid 429 HTTP status.

Scraped links will be dumped in text file in same directory.

Hosting Django App for free on PythonAnyWhere Server. (https://www.pythoncircle.com/post/18/how-to-host-django-app-on-pythonanywhere-for-free/)

Featured Image Source : http://ehacking.net/ (http://ehacking.net/)

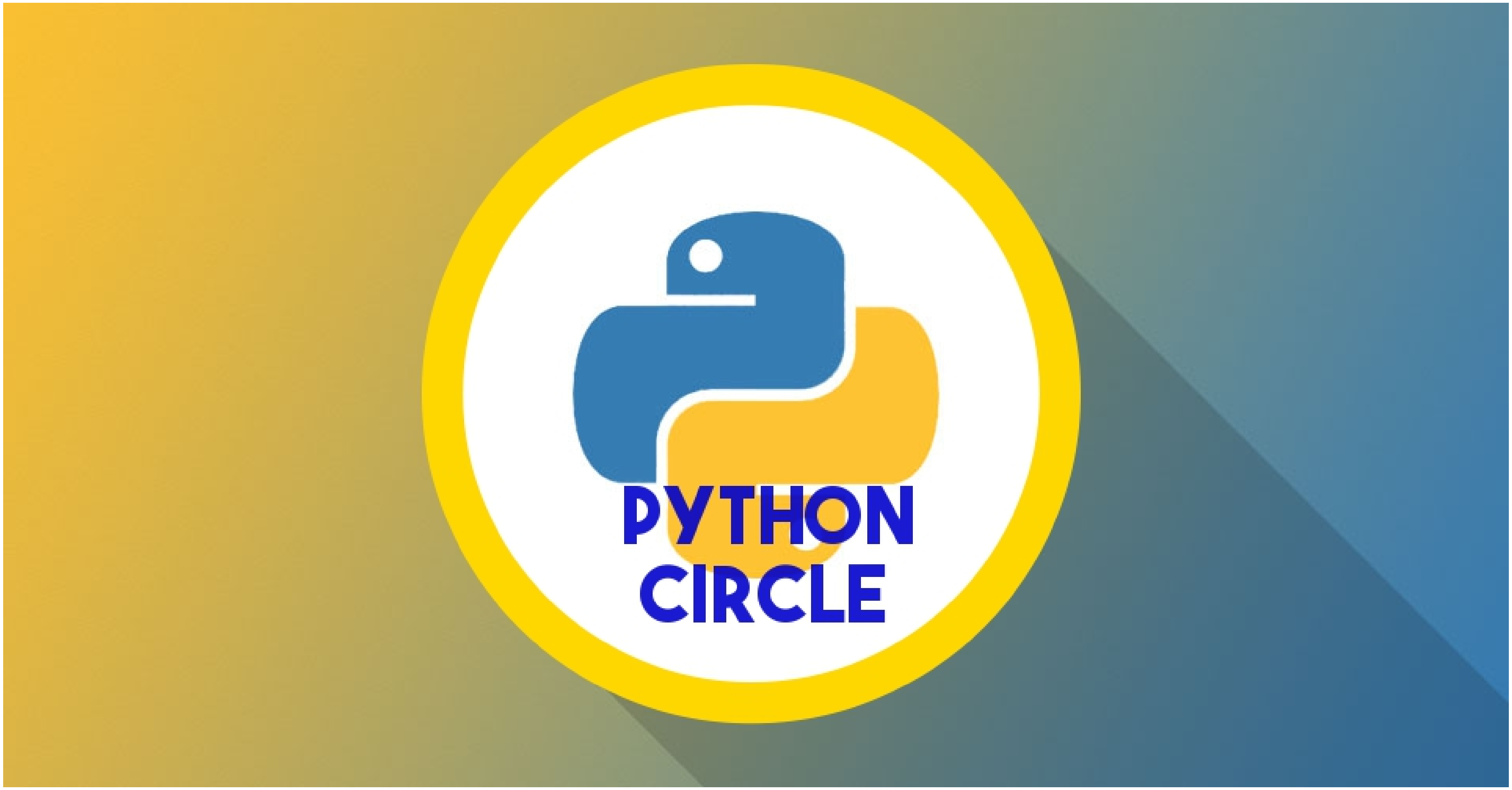
scraping (/tag/scraping/) beautifulsoup (/tag/beautifulsoup/) 0 2055

Related Articles:

Python Script 14: Scraping news headlines using python beautifulsoup (/post/678/python-script-14-scraping-news-headlines-using-python-beautifulsoup/)

Scraping news headlines using python beautifulsoup, web scraping using python, python script to scrape news, web scraping using beautifulsoup, news headlines scraping using python, python programm to get news headlines from web...

Read Full Article (/post/678/python-script-14-scraping-news-headlines-using-python-beautifulsoup/)



Python Script 2 : Crawling all emails from a website (/post/217/python-script-2-crawling-all-emails-from-a-website/)

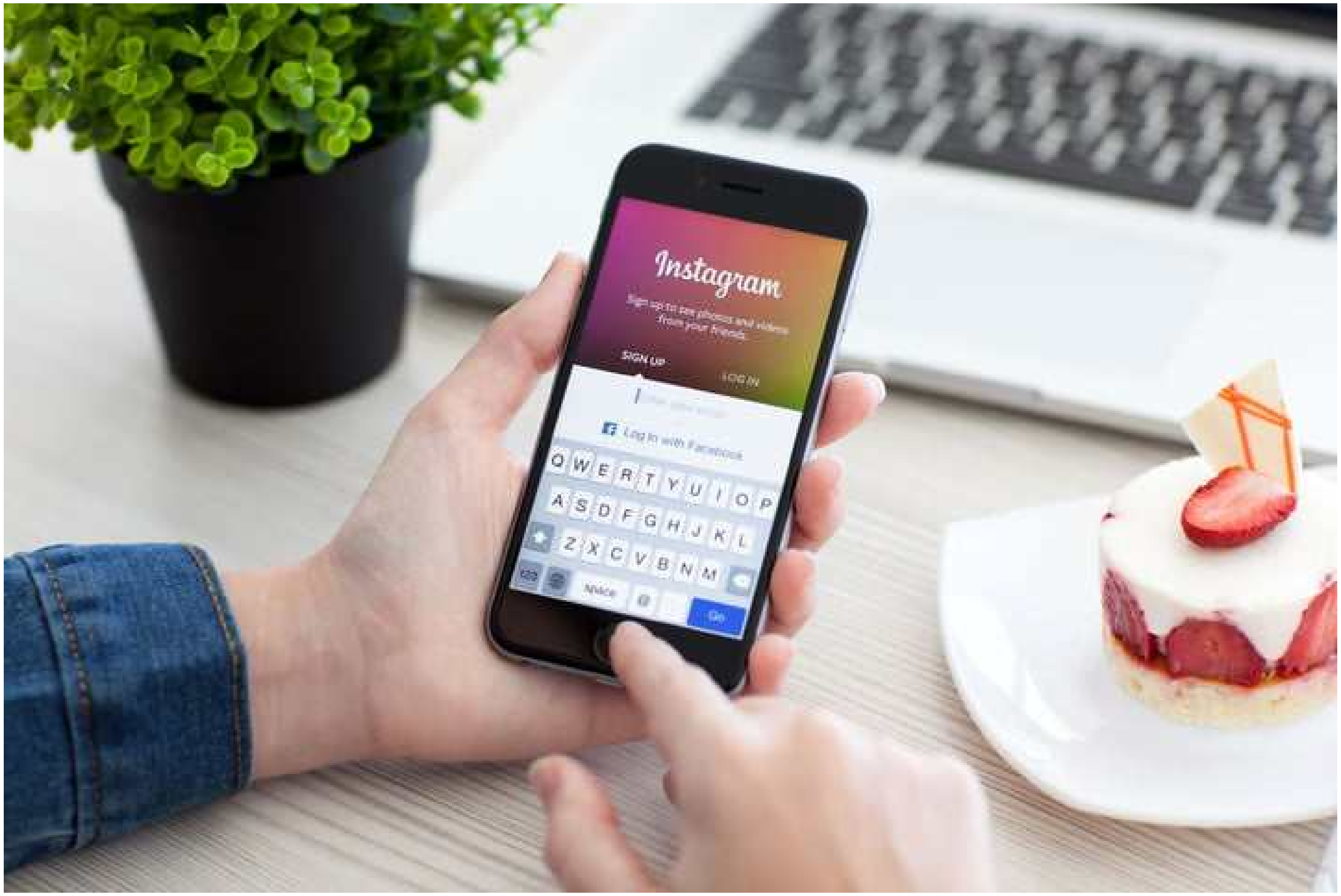
Website crawling for email address, web scraping for emails, data scraping and fetching email adress, python code to scrape all emails froma websites, automating the email id scraping using python script, collect emails using python script...

Read Full Article (/post/217/python-script-2-crawling-all-emails-from-a-website/)

How to create completely automated telegram channel with python (/post/265/how-to-create-completely-automated-telegram-channel-with-python/)

Creating a completely automated telegram channel to generate and post content using python code on regular basis. Automating the Telegram channel using python script...

Read Full Article (/post/265/how-to-create-completely-automated-telegram-channel-with-python/)



py_instagram_dl - The Python Package to Download All pictures of an Instagram User (/post/447/py_instagram_dl-the-python-package-to-download-all-pictures-of-an-instagram-user/)

Download all instagram images for any user using this python package....

Read Full Article (/post/447/py_instagram_dl-the-python-package-to-download-all-pictures-of-an-instagram-user/)

0 thoughts on 'Python Script 10: Collecting One Million Website Links'

Leave a comment:

Type your comment here. For code use pastebin link. Limit 500 chars.

Your email please.

Your Name

Submit

*All Fields are mandatory. **Email Id will not be published publicly.

SUBSCRIBE

Please subscribe to get the latest articles in your mailbox.

Your Email ID Please

Subscribe

Recent Posts:

- 5 lesser used Django template tags (/post/694/5-lesser-used-django-template-tags/)

- Solving Python Error- KeyError: 'key_name' (/post/693/solving-python-error-keyerror-key_name/)

- Intellectual property Law and Coding (/post/692/intellectual-property-law-and-coding/)

- Python Script 17: Setting bing image of the day as desktop wallpaper (/post/691/python-script-17-setting-bing-image-of-the-day-as-desktop-wallpaper/)

- Django Template Fiddle Launched !!!! (/post/690/django-template-fiddle-launched/)

- Python Script 16: Generating word cloud image of a text using python (/post/689/python-script-16-generating-word-cloud-image-of-a-text-using-python/)

- Preventing cross-site scripting attack on your Django website (/post/688/preventing-cross-site-scripting-attack-on-your-django-website/)

- How to generate ATOM/RSS feed for Django website (/post/687/how-to-generate-atomrss-feed-for-django-website/)

- How to create sitemap of Django website (/post/686/how-to-create-sitemap-of-django-website/)

- For loop in Django template (/post/685/for-loop-in-django-template/)

- How to add Favicon to Django websites (/post/684/how-to-add-favicon-to-django-websites/)

- Scraping data of 2019 Indian General Election using Python Request and BeautifulSoup and analyzing it (/post/683/scraping-data-of-2019-indian-general-election-using-python-request-and-beautifulsoup-and-analyzing-it/)

- How to display PDF in browser in Django instead of downloading it (/post/682/how-to-display-pdf-in-browser-in-django-instead-of-downloading-it/)

- Solving Python Error - UnboundLocalError: local variable 'x' referenced before assignment (/post/680/solving-python-error-unboundlocalerror-local-variable-x-referenced-before-assignment/)

- Python Script 15: Creating a port scanner in 8 lines of python (/post/679/python-script-15-creating-a-port-scanner-in-8-lines-of-python/)

- Python Script 14: Scraping news headlines using python beautifulsoup (/post/678/python-script-14-scraping-news-headlines-using-python-beautifulsoup/)

- How to download large csv files in Django (/post/677/how-to-download-large-csv-files-in-django/)

- Text based snake and ladder game in python (/post/676/text-based-snake-and-ladder-game-in-python/)

- Logging databases changes in Django Application (/post/675/logging-databases-changes-in-django-application/)
- Python Script 13: Generating ascii code from Image (/post/674/python-script-13-generating-ascii-code-from-image/)
- Python Frequently Asked Question 1: What can I do in Python? (/post/672/python-frequently-asked-question-1-what-can-i-do-in-python/)
- How to start with Python Programming - A beginner's guide (/post/671/how-to-start-with-python-programming-a-beginners-guide/)
- Python program to find whether a given year is leap year or not (/post/670/python-program-to-find-whether-a-given-year-is-leap-year-or-not/)
- Python Django Project Idea for beginners (/post/669/python-django-project-idea-for-beginners/)
- Uploading a file to FTP server using Python (/post/668/uploading-a-file-to-ftp-server-using-python/)
- Print statement in Python vs other programming languages (/post/667/print-statement-in-python-vs-other-programming-languages/)
- Automating Facebook page posts using python script (/post/666/automating-facebook-page-posts-using-python-script/)
- try .. except .. else .. in python with example (/post/664/try-except-else-in-python-with-example/)
- Python Script 12: Drawing Indian National Flag Tricolor using Python Turtle (/post/662/python-script-12-drawing-indian-national-flag-tricolor-using-python-turtle/)
- Python Script 11: Drawing Flag of United States of America using Python Turtle (/post/661/python-script-11-drawing-flag-of-united-states-of-america-using-python-turtle/)
- Solving Django Error: TemplateDoesNotExist at /app_name/ (/post/660/solving-django-error-templatedoesnotexist-at-app_name/)
- Adding Email Subscription Feature in Django Application (/post/657/adding-email-subscription-feature-in-django-application/)
- Using PostgreSQL Database with Python (/post/656/using-postgresql-database-with-python/)
- Programming On Raspberry Pi With Python: Activate LED and Buzzer on Motion Detection (/post/654/programming-on-raspberry-pi-with-python-activate-led-and-buzzer-on-motion-detection/)
- Programming on Raspberry Pi with Python: Controlling LED (/post/652/programming-on-raspberry-pi-with-python-controlling-led/)
- Programming on Raspberry Pi with Python: Sending IP address on Telegram channel on Raspberry Pi reboot (/post/651/programming-on-raspberry-pi-with-python-sending-ip-address-on-telegram-channel-on-raspberry-pi-reboot/)

- Programming on Raspberry Pi with Python: WIFI and SSH configuration (/post/650/programming-on-raspberry-pi-with-python-wifi-and-ssh-configuration/)

- Programming on Raspberry Pi with Python: Raspberry Pi Setup (/post/649/programming-on-raspberry-pi-with-python-raspberry-pi-setup/)

- Iterator and Generators in Python: Explained with example (/post/648/iterator-and-generators-in-python-explained-with-example/)

- Top 5 Python Books (/post/646/top-5-python-books/)

- Encryption-Decryption in Python Django (/post/641/encryption-decryption-in-python-django/)

- Sending Emails Using Python and Gmail (/post/628/sending-emails-using-python-and-gmail/)

- How to Track Email Opens Sent From Django App (/post/626/how-to-track-email-opens-sent-from-django-app/)

- Python Tip 1: Accessing localhost Django webserver over the Internet (/post/618/python-tip-1-accessing-localhost-django-webserver-over-the-internet/)

- 5 common mistakes made by beginner python programmers (/post/602/5-common-mistakes-made-by-beginner-python-programmers/)

- How to upload and process the Excel file in Django (/post/591/how-to-upload-and-process-the-excel-file-in-django/)

- Creating sitemap of Dynamic URLs in your Django Application (/post/584/creating-sitemap-of-dynamic-urls-in-your-django-application/)

- Adding Robots.txt file to Django Application (/post/578/adding-robotstxt-file-to-django-application/)

- Python Script 3: Validate, format and Beautify JSON string Using Python (/post/576/python-script-3-validate-format-and-beautify-json-string-using-python/)

- Displaying custom 404 error (page not found) page in Django 2.0 (/post/564/displaying-custom-404-error-page-not-found-page-in-django-20/)

Tags:

404 (/tag/404/) 500 (/tag/500/) AJAX (/tag/ajax/) API (/tag/api/) AUTHENTICATION (/tag/authentication/)
AUTOMATION (/tag/automation/) AWS (/tag/aws/) BACKUP (/tag/backup/) BEAUTIFULSOUP (/tag/beautifulsoup/) BOOK (/tag/book/)
BREADBOARD (/tag/breadboard/) CAPTCHA (/tag/captcha/) CELERY (/tag/celery/) COMMANDS (/tag/commands/) CSV (/tag/csv/)
DATA (/tag/data/) DATABASE (/tag/database/) DECRYPTION (/tag/decryption/) DJANGO (/tag/django/)
DJANGO APP (/tag/django%20app/) DJANGO ERROR (/tag/django%20error/) DOCKER (/tag/docker/) DOWNLOAD (/tag/download/)
EC2 (/tag/ec2/) ELASTIC SEARCH (/tag/elastic%20search/) EMAIL (/tag/email/) ENCRYPTION (/tag/encryption/) ERROR (/tag/error/)
ERROR PAGE (/tag/error%20page/) EXCEL (/tag/excel/) EXCEPTION (/tag/exception/) FACEBOOK (/tag/facebook/) FAQ (/tag/faq/)
FEEDS (/tag/feeds/) FOR LOOP (/tag/for%20loop/) FORM (/tag/form/) FREE EMAILS (/tag/free%20emails/) FTP (/tag/ftp/)
GAME (/tag/game/) GENERATOR (/tag/generator/) GMAIL (/tag/gmail/) GPIO (/tag/gpio/) GRAPH API (/tag/graph%20api/)
HARDWARE (/tag/hardware/) HOSTING (/tag/hosting/) IMAGE (/tag/image/) INSTAGRAM (/tag/instagram/)
IP ADDRESS (/tag/ip%20address/) IP LAW (/tag/ip%20law/) ITERATOR (/tag/iterator/) JSON (/tag/json/) KIBANA (/tag/kibana/)
LED (/tag/led/) LOGGING (/tag/logging/) MISTAKE (/tag/mistake/) MONGODB (/tag/mongodb/) PACKAGE (/tag/package/)
PAYMENT GATWAY (/tag/payment%20gatway/) PDF (/tag/pdf/) PITFALLS (/tag/pitfalls/) PROGRAM (/tag/program/)
PROJECT (/tag/project/) PROXY (/tag/proxy/) PYTHON BITES (/tag/python%20bites/) PYTHON TIPS (/tag/python%20tips/)
PYTHONANYWHERE (/tag/pythonanywhere/) RABBITMQ (/tag/rabbitmq/) RASPBERRYPI (/tag/raspberrypi/) REQUESTS (/tag/requests/)
RESPONSE (/tag/response/) SCRAPING (/tag/scraping/) SCRAPY (/tag/scrapy/) SCRIPT (/tag/script/) SECURITY (/tag/security/)
SENSOR (/tag/sensor/) SEO (/tag/seo/) SERVER (/tag/server/) SETUP (/tag/setup/) SIGNALS (/tag/signals/)
SITEMAP (/tag/sitemap/) SOCKET (/tag/socket/) SSH (/tag/ssh/) TAG (/tag/tag/) TELEGRAM (/tag/telegram/)

TEMPLATES (/tag/templates/) TIPS (/tag/tips/) TKINTER (/tag/tkinter/) TOR (/tag/tor/) TRACKING (/tag/tracking/)

TRY CATCH (/tag/try%20catch/) TURTLE (/tag/turtle/) UPLOAD (/tag/upload/) VIRTUAL ENV (/tag/virtual%20env/) WIFI (/tag/wifi/)

WORD CLOUD (/tag/word%20cloud/)

Contact Us (/contact/) Advertise with Us (/advertise/)