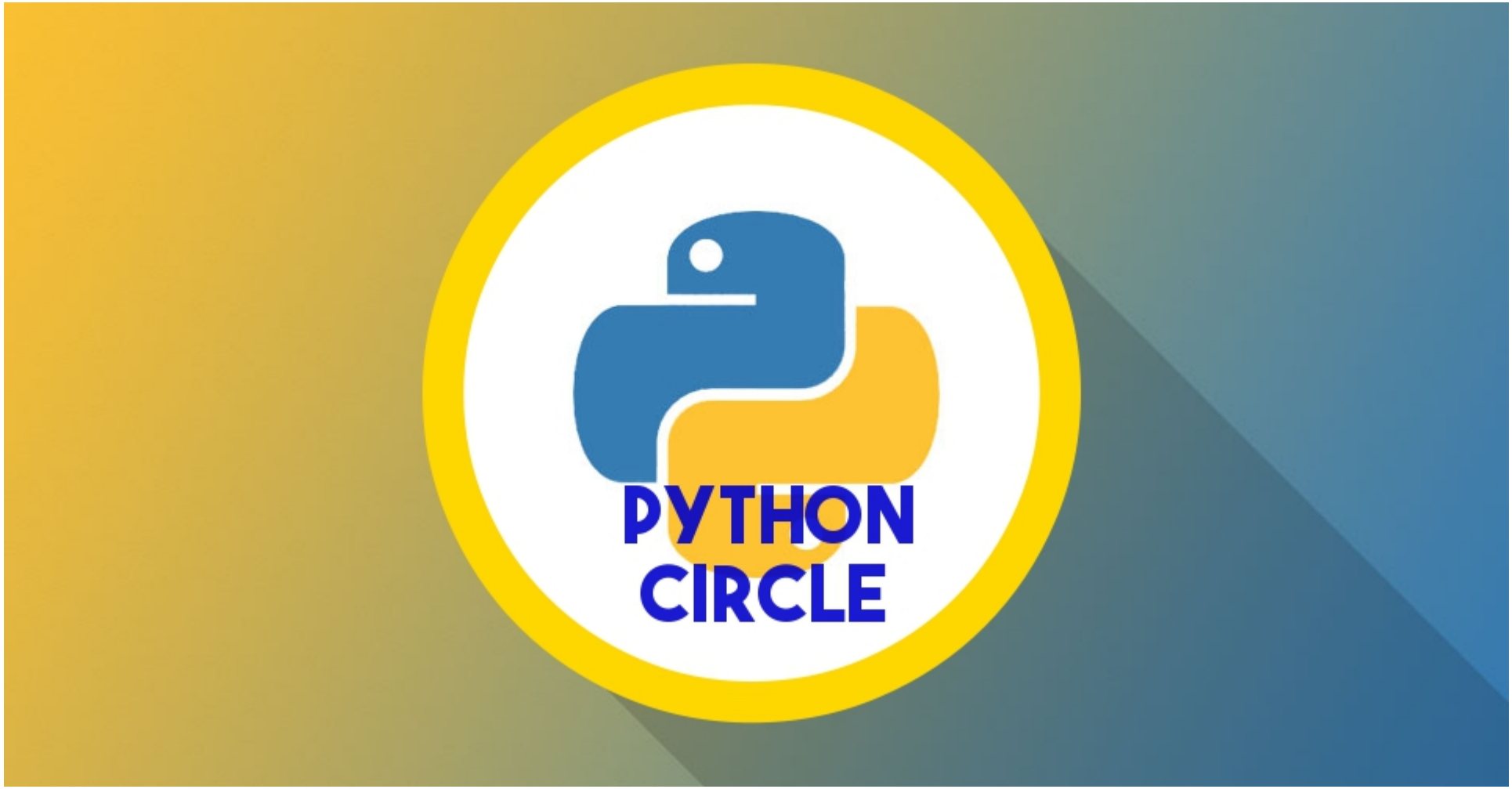
Python Script 2 : Crawling all emails from a website (/post/217/python-script-2-crawling-all-emails-from-a-website/)

scraping (/tag/scraping/) email (/tag/email/) 3 6850



This is the second article in the series of python scripts.

In this article we will see how to crawl all pages of a website and fetch all the emails.

Important: Please note that some sites may not want you to crawl their site. Please honour their robot.txt file. In some cases it may lead to legal action. This article is only for educational purpose. Readers are requested not to misuse it.

Instead of explaining the code separately, I have embedded the comments over the source code lines. I have tried to explain the code wherever I felt the requirement.

Please comment in case of any query. You might need to install some packages like `requests` and `BeautifulSoup` for this script to work. It is recommended that you create a virtual environment (https://www.pythoncircle.com/post/404/virtual-environment-in-python-a-pocket-guide/) and install packages in it.

```
import re
import requests
import requests.exceptions
from urllib.parse import urlsplit
from collections import deque
from bs4 import BeautifulSoup

# starting url. replace google with your own url.
starting_url = 'http://www.miet.ac.in'

# a queue of urls to be crawled
unprocessed_urls = deque([starting_url])

# set of already crawled urls for email
processed_urls = set()

# a set of fetched emails
emails = set()

# process urls one by one from unprocessed_url queue until queue is empty
while len(unprocessed_urls):

    # move next url from the queue to the set of processed urls
    url = unprocessed_urls.popleft()
    processed_urls.add(url)

    # extract base url to resolve relative links
    parts = urlsplit(url)
    base_url = "{0.scheme}://{0.netloc}".format(parts)
    path = url[:url.rfind('/')+1] if '/' in parts.path else url

    # get url's content
    print("Crawling URL %s" % url)
    try:
        response = requests.get(url)
    except (requests.exceptions.MissingSchema, requests.exceptions.ConnectionError):
```

```
        # ignore pages with errors and continue with next url
        continue

    # extract all email addresses and add them into the resulting set
    # You may edit the regular expression as per your requirement
    new_emails = set(re.findall(r"[a-z0-9\.\-+_]+@[a-z0-9\.\-+_]+\.[a-z]+", response.text, re.I))
    emails.update(new_emails)
    print(emails)
    # create a beutiful soup for the html document
    soup = BeautifulSoup(response.text, 'lxml')

    # Once this document is parsed and processed, now find and process all the anchors i.e. linked urls
    for anchor in soup.find_all("a"):
        # extract link url from the anchor
        link = anchor.attrs["href"] if "href" in anchor.attrs else ''
        # resolve relative links (starting with /)
        if link.startswith('/'):
            link = base_url + link
        elif not link.startswith('http'):
            link = path + link
        # add the new url to the queue if it was not in unprocessed list nor in processed list yet
        if not link in unprocessed_urls and not link in processed_urls:
            unprocessed_urls.append(link)
```

Constructive feedback is always welcomed.

scraping (/tag/scraping/) email (/tag/email/) 3 6850

Related Articles:

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

Collecting one million website links by scraping using requests and BeautifulSoup in Python. Python script to collect one million website urls, Using beautifulsoup to scrape data, Web scraping using python, web scraping using beautifulsoup, link collection using python beautifulsoup...

Read Full Article (/post/545/python-script-10-collecting-one-million-website-links/)


TRACKING EMAIL OPENS
IN DJANGO APP
PYTHONCIRCLE.COM

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

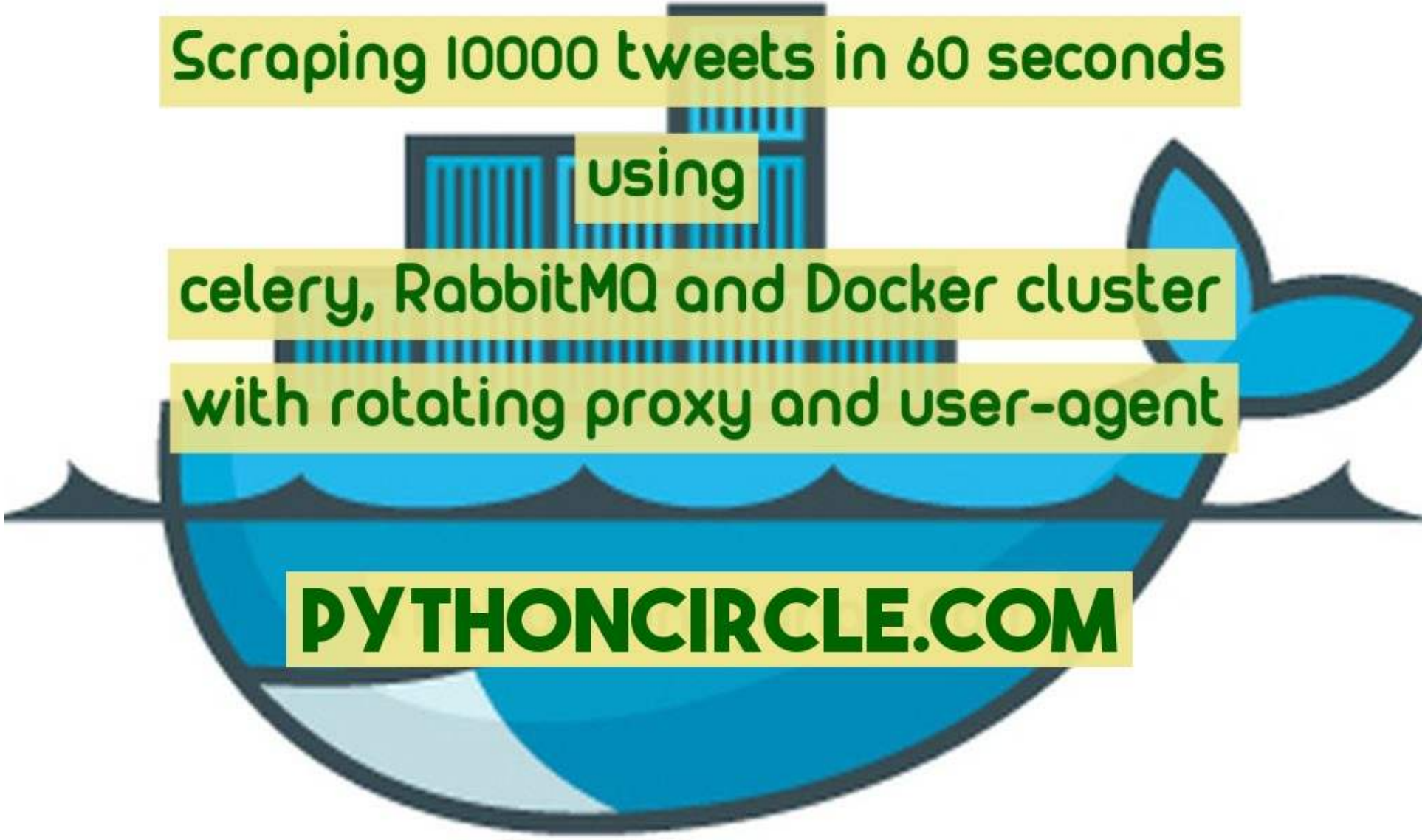
How to track email opens. Tracking email sent from django app. Finding the email open rate in Python Django. Email behaviour of users in Python Django. Finding when email is opened by user in python-django....

Read Full Article (/post/626/how-to-track-email-opens-sent-from-django-app/)

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

Python code to send free emails using Gmail credentials, Sending automated emails using python and Gmail, Using Google SMTP server to send emails using python. Python script to automate gmail sending emails, automating email sending using gmail...

Read Full Article (/post/628/sending-emails-using-python-and-gmail/)



Scraping 10000 tweets in 60 seconds using celery, RabbitMQ and Docker cluster with rotating proxy (/post/518/scraping-10000-tweets-in-60-seconds-using-celery-rabbitmq-and-docker-cluster-with-rotating-proxy/)

Scraping large amount of tweets within minutes using celery and python, RabbitMQ and docker cluster with Python, Scraping huge data quickly using docker cluster with TOR, using rotating proxy in python, using celery rabbitmq and docker cluster in python to scrape data, Using TOR with Python...

Read Full Article (/post/518/scraping-10000-tweets-in-60-seconds-using-celery-rabbitmq-and-docker-cluster-with-rotating-proxy/)

---

3 thoughts on 'Python Script 2 : Crawling All Emails From A Website'

Mrcee :
Hi there, How long does this script take to complete crawling? The crawl does not seem to be contained to the website?If you could advise it would be much appreciated.

> **Admin ⊘ :**
> yes, this script is not contained to one site only. it will crawl any other website as well if that is linked to current website. This will keep running it have no more links to process. If there is any specific issue, le us know.
>
> Reply

Faheem :
can you modify script that scraped emails it will export it in csv format alot of thanks in advance, appreciate your guidance.Regards,FaheemSkype mfaheem2009

> Reply

Mitchell :
What version of Python is required to run this code, I have version 3.7 and it doesent seem to work? I dont know what I am doing wrong

> **Admin ⊘ :**
> I used python 3.4. What is the error you are getting. please share pastebin

Reply

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/)