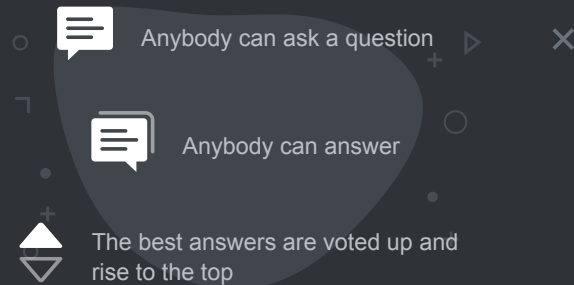


Code Review Stack Exchange is a question and answer site for peer programmer code reviews. It only takes a minute to sign up.

Sign up to join this community



CODE REVIEW

[Home](#)[Questions](#)[Tags](#)[Users](#)[Unanswered](#)

Simple Python Old Email Eraser From Inbox

[Ask Question](#)

Asked 3 years, 7 months ago Active 3 years, 7 months ago Viewed 939 times



1



*This is a old email eraser that deletes old unwanted email from your gmail or yahoo inbox, cause guys really we all had the nightmare of selecting and deleting 50 old emails at a time when their are thousands, well this one uses simple python imaplib library to do that, and even shows the email that its deleting with sender name And **Deletes up to a certain date that we specify***

```
try:
    import imaplib
    import argparse
    import email
    import webbrowser
except Exception as e:
    print e

def Connect_And_Login(uname, passwd, bedate):

    if uname.endswith('@gmail.com'):
        mail = imaplib.IMAP4_SSL('imap.gmail.com')
        inbox = "INBOX"
        url = 'https://www.google.com/settings/security/lesssecureapps'

    elif uname.endswith('@yahoo.com'):
        mail = imaplib.IMAP4_SSL('imap.mail.yahoo.com')
```

The Overflow Blog

Podcast 231: Make it So

The Overflow #19: Jokes on us

Featured on Meta

Improving the Review Queues - Project overview

Introducing the Moderator Council - and its first, pro-tempore, representatives

Related

4 Email obfuscator

7 Python Email Program

```

        inbox = "Inbox"
        url = 'https://login.yahoo.com/account/security#less-secure-apps'

    else:
        print 'Enter The Email Address Carefully'
        exit(0)

    try:
        mail.login(uname, passwd)

    except Exception as e:
        print "\n\nFailed to login Allow less secure apps from your email se
        webbrowser.open(url)
        exit(0)

    Mark_emails_as_Deleted(mail, bedate, inbox)
    #print mail.list()
    return

def Mark_emails_as_Deleted(mail, bedate, inbox):

```

python email

share improve this question follow

add a comment

asked Sep 9 '16 at 20:12



Shantanu Bedajna

177 1 7

1 Answer

Active Oldest Votes



1



One nice programming principle is *the principle of least astonishment*, which you violate with your code. Imagine someone using your module and calling `Connect_And_Login`. If it was me I would be **very** astonished that suddenly many of my emails are deleted! At least the function does not have today as a default `bedate`, otherwise it would be a disaster waiting to happen.



It would be a lot better if that function just returned the `mail` object, which the function `Mark_emails_as_Deleted` then takes as parameter.



```

mail, inbox = connect_and_login(args.uname, args.path)
mark_emails_as_deleted(mail, args.bedate, inbox)

```

(Before the `return` in `Connect_And_Login` was actually superfluous, because python automatically returns `None` at the end of a function body

3

Simple email validation script

6

Email redirection script in python

6

Quick email program using Python classes

6

Python program that obfuscates an email address

3

CliwPw - Yet Another Python3 Password Manager

2

Simple Email Subject Filtering (python)

Hot Network Questions



Simultaneous find and replace



Biggest successful secret project in history?



Mentor says I cannot be first author of my paper because I am an undergraduate



The Crosswordsearch



How to use match and blank



Is there a vaccine against the plague (Yersinia pestis)?



How can I prevent checkmate in this position?



Alternatives to batteries for "residential" power?



Removing rows whose elements appear less than a certain number



Can an Augmented triad be called a Quartal chord?



Why do they remove the rear view mirror in driving scenes?



My student published a preprint of our published paper without me as an author



"Putin enters a bar and proclaims, 'Next round's on the house!'" What's the point of this popular Russian joke?



Is it appropriate to say "Excel hell" on a LinkedIn profile?



Idiom for a fallacy "who are you to judge"?



Box a matrix row

And here we are at another thing. You should use a consistent naming scheme for your variables. Python actually has an official style guide, [PEP8](#), which programmers are encouraged to follow to improve readability. It recommends using `lower_case` for variables and functions (and `PascalCase` for classes).

It also recommends avoiding unnecessary white-space, such as between an `if` block and an `else` block, or between `try` and `except`.

Excepting an import error and then just printing the error does you no good. It will print the error, try to continue running the code, encounter wherever that module is actually needed and fail there with another cryptic error code. The output saying that it was actually an import error is buried above the stacktrace. It is better to just let it fail as an import error or at least terminate the program there.

The different email provider could be stored in a dictionary mapping to their data. This way it is more easily extendable to more providers:

```
supported_provider = {"gmail.com": {"mail": "imap.gmail.com",
                                     "inbox": "INBOX",
                                     "url": 'https://www.google.com/settings/'},
                     "yahoo.com": {...}}
provider = supported_provider[uname.split("@")[-1]]
mail = imaplib.IMAP4_SSL(provider['mail'])
inbox = provider['inbox']
url = provider['url']
```

It is in general faster, more readable and extendable to use `str.format` instead of string additions:

```
result, data = mail.search(None, '(BEFORE "{}")'.format(bedate))
```

Lastly, you could use the nice functionality of `argparse` to validate your input as a date as explained e.g. in [this](#) answer:

```
parser.add_argument('bedate', type=lambda s: datetime.datetime.strptime(s, '%b
```

`%b` is the abbreviated month name *in the current locale*. So in English it will be Jan, Feb, ..., Dec, while e.g. in German it will be Jan, Feb, ..., Dez. I guess email/yahoo can actually handle this because it is also localized for the user



- Did the Romans leave any technical instruction manuals behind?
- Meaning of 的 in 在蓝天白云的 衬托下，阳光的照射下
- Net production of CO2 in plants
- Why do (kilo)watt-hours double-count units per time?
- How do I apply this operation to a list?
- First College In English Speaking Country to Use Latin For Their Motto
- What can be causing my tremendous amount of FEC and CRC errors via my DSL connection?
- Are there economic studies surrounding the efficacy of gun control?

[Question feed](#)

gmail/yahoo can actually handle this because it is also localized for the user.

share improve this answer
follow

edited May 23 '17 at 12:41



answered Sep 9 '16 at 23:15



Graipher

37.3k 6 57 118

Thanks for always helping me out Graipher ,I see because of the python variable declaration standard stack exchange is thinking that my functions are classes and highlighting them blue –

[Shantanu Bedajna](#) Sep 10 '16 at 15:23

@Shantanu Yes, stackexchange Syntax highlighting uses PEP8 for Python. In the end it is nice to use the official style guide (because it makes it easier for everybody else to understand your code, like I see a capital letter name, it's gotta be a class...), But it is more important to be consistent. You use Capital_at_beginning_and_End, low_beginning_end_Capital, All_Capital and lowercase and lower_case (in library functions). This mix makes it hard to immediately know what type an object is. – [Graipher](#) Sep 11 '16 at 14:46

add a comment

Your Answer

B *I*

Sign up or [log in](#)

Sign up using Google

Sign up using Facebook

Sign up using Email and Password

Post as a guest

Name

Email

Required, but never shown

Post Your Answer

By clicking "Post Your Answer", you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#)

Not the answer you're looking for? Browse other questions tagged [python](#) [email](#) or [ask your own question](#).

CODE REVIEW

Tour
Help
Chat
Contact
Feedback
Mobile
Disable Responsiveness

COMPANY

Stack Overflow
Stack Overflow Business
Developer Jobs
About
Press
Legal
Privacy Policy

STACK EXCHANGE NETWORK

Technology >
Life / Arts >
Culture / Recreation >
Science >
Other >

[Blog](#) [Facebook](#) [Twitter](#) [LinkedIn](#)

site design / logo © 2020 Stack Exchange Inc; user contributions
licensed under [cc by-sa 4.0](#) with [attribution required](#).
rev 2020.5.1.36728