

Forum:[Hack Forums](#)

Thread: [Storing strings to put in pastebin? \[EDIT: Possible Solution?\] \[EDIT: SOLVED\]](#)

antigrav - August 22, 2012, 4:42 pm

SOURCE CODE AT BOTTOM

Here's my problem. I am attempting to make a program that takes Newegg product URLs (like <http://www.newegg.com/Product/Product.aspx?Item=N82E16813157301>) and outputs info about the item. So if I give it the URL in the example, it returns with this:

ASRock H77 Pro4-M LGA 1155 Intel H77 HDMI SATA 6Gb/s USB 3.0 Micro ATX Intel Motherboard

\$89.99

<http://www.newegg.com/Product/Product.aspx?Item=N82E16813157301>

Here's my dilemma. I can do this with one item, no problem. After getting all the variables, it asks you to exit or paste to pastebin (only works if you are logged into pastebin on your default browser). When you paste to pastebin, the program starts over the process. Takes another url, gives more info, etc.

The problem with this is, once it does it more than once, if you want to paste to pastebin, it will make a new paste. I want to make it so it all goes into one paste.

Here's my source code. I'm not worried about people stealing it too much, since this is a personal project (but if you're going to use it in your own program to release to the public, please at least credit me first):

Code # REFORMATTED VERSION OF SOURCECODE FOR PROJECT
MADE PURELY TO INCREASE READABILITY

```
import urllib
import urllib2
import webbrowser
import json
import time
import sys
from bs4 import BeautifulSoup

while True:
    newegg_URL = raw_input('Enter the URL of the Newegg item: \n')
    not_product_code = 'http://www.newegg.com/Product/Product.aspx?Item='
    product_code = newegg_URL.replace(not_product_code, "")
    product_info_URL = 'http://www.ows.newegg.com/Products.egg/' + product_code

    soup = BeautifulSoup(urllib.urlopen(newegg_URL))
    name_with_site = soup.title.string
    name = name_with_site.replace('Newegg.com - ', '')
```

```
print '\n' + name
```

```
data = json.load(urllib2.urlopen(product_info_URL))
price = data[u'FinalPrice']
print '\n' + price
```

```
print '\n' + newegg_URL
```

```
product_info = name + '\n' + price + '\n' + newegg_URL
```

```
continue_or_export = raw_input("\n\nAre you done? ' +
'To quit the program, type 1. ' +
'To export to PasteBin, type 2. \n\n')
```

```
if continue_or_export == '1':
    sys.exit()
elif continue_or_export == '2':
    print 'Note: Exporting to pastebin only works if one is logged in on their default browser.
Attempting now.'
    pastebin_vars = {'api_dev_key':'57fe1369d02477a235057557cbeabaa1',
'api_option':'paste',
'api_paste_code':product_info}
    response = urllib.urlopen('http://pastebin.com/api/api_post.php', urllib.urlencode(pastebin_vars))
    pastebin_url = response.read()
    print pastebin_url
    webbrowser.open_new_tab(pastebin_url)
else:
    print 'Your input was not understood. Use "1" or "2" only.'
```

```
#product_category = 0  <- coming soon
```

```
# VARIABLE NAMES + DESCRIPTIONS
```

```
# newegg_URL - the url of the newegg product found in url
```

```
# soup - the beautiful soup html of the newegg product
```

```
# product_code - code of the product
```

```
# product_info_URL - url of the page that holds price and rating
```

```
# usock - urllib's product info url thing
```

```
# data - json interpretation of the page with details on it
```

```
# price - product price
```

```
# name_with_site - product name
```

```
# name - product name, but without the annoying "newegg.com - " part
```

```
# product_category - product category COMING SOON SO HARD TO DO D:
```

```
# todo: make it take more than one in one instance and make it export to SOMETHING
```

```
# (notepad? pastebin? SOMETHING?)
```

How should I go about solving this?

POSSIBLE SOLUTION

What if I use the old pastebin url, take all the info from it, store it in a new variable called old_product_info, and add that to product_info? If I do more than 2 recursions, I just keep adding to old_product_info.

Possible solution? Testing now.

SOLVED

Added a counter for how many times the program was ran, and if it was over one, it would go to the old info and add it to the new info.

Source code at end;

Code # REFORMATTED VERSION OF SOURCECODE FOR PROJECT
MADE PURELY TO INCREASE READABILITY

```
import urllib
import urllib2
import webbrowser
import json
import time
import sys
from bs4 import BeautifulSoup

count = 0
while True:
    newegg_URL = raw_input('Enter the URL of the Newegg item: \n')
    not_product_code = 'http://www.newegg.com/Product/Product.aspx?Item='
    product_code = newegg_URL.replace(not_product_code, "")
    product_info_URL = 'http://www.ows.newegg.com/Products.egg/' + product_code

    soup = BeautifulSoup(urllib.urlopen(newegg_URL))
    name_with_site = soup.title.string
    name = name_with_site.replace('Newegg.com - ', "")
    print '\n' + name

    data = json.load(urllib2.urlopen(product_info_URL))
    price = data[u'FinalPrice']
    print '\n' + price

    print '\n' + newegg_URL

    product_info = name + '\n' + price + '\n' + newegg_URL + '\n'
    final_product_info = product_info

    continue_or_export = raw_input("\n\nAre you done? ' +
    'To quit the program, type 1. ' +
    'To export to PasteBin, type 2. \n\n')
    count+=1
    if continue_or_export is '1':
        sys.exit()
    elif continue_or_export is '2':
        if count > 1:
            not_pastebin_url_id = 'http://pastebin.com/'
            pastebin_url_id = pastebin_url.replace(not_pastebin_url_id, "")
            pastebin_raw_url = 'http://pastebin.com/raw.php?i=' + pastebin_url_id
            usock = urllib2.urlopen(pastebin_raw_url)
            old_product_info = usock.read()
            usock.close()
```

```
final_product_info = old_product_info + product_info
print 'Note: Exporting to pastebin only works if one is logged in on their default browser.
Attempting now.'
```

```
pastebin_vars = {'api_dev_key':'57fe1369d02477a235057557cbeabaa1',
'api_option':'paste',
'api_paste_code':final_product_info}
```

```
response = urllib.urlopen('http://pastebin.com/api/api_post.php', urllib.urlencode(pastebin_vars))
pastebin_url = response.read()
print pastebin_url +'\n\n'
webbrowser.open_new_tab(pastebin_url)
```

```
else:
print 'Your input was not understood. Use "1" or "2" only.'
```

```
#product_category = 0  <- coming soon
```

```
# VARIABLE NAMES + DESCRIPTIONS
```

```
# newegg_URL - the url of the newegg product found in url
```

```
# soup - the beautiful soup html of the newegg product
```

```
# product_code - code of the product
```

```
# product_info_URL - url of the page that holds price and rating
```

```
# usock - urllib's product info url thing
```

```
# data - json interpretation of the page with details on it
```

```
# price - product price
```

```
# name_with_site - product name
```

```
# name - product name, but without the annoying "newegg.com - " part
```

```
# product_category - product category COMING SOON SO HARD TO DO D:
```

```
# todo: make it take more than one in one instance and make it export to SOMETHING
```

```
# (notepad? pastebin? SOMETHING?)
```

JrMn - August 22, 2012, 5:06 pm

Don't use is to compare integers, even they are small and are cached.

Use is for identity, == for equality.

dberube - August 22, 2012, 5:26 pm

May I suggest that you use txtuplib3 : <http://pastebin.com/3GLk1FCL>

I optimized it and it work perfectly and text-upload do not detect spam.

You code you need would look like that:

```
Code firstTime = True
```

```
session = txtUpload()
```

```
resp = session.openSession('aUsername','aPassword')
```

```
if firstTime:
```

```
    resp,ID = session.upload('content')
```

```
    firstTime = False
```

```
else:
```

```
    newcontent = session.read(ID)[1] + information
```

```
    session.edit(ID,newcontent,'aTitle','some,key,words')
```

antigrav - August 23, 2012, 12:58 pm

(Aug 22, 2012 05:26 PM)dberube Wrote: [\[CITE\]](#) May I suggest that you use txtuplib3 : <http://pastebin.com/3GLk1FCL>

I optimized it and it work perfectly and text-upload do not detect spam.

You code you need would look like that:

```
Code firstTime = True
```

```
session = txtUpload()
```

```
resp = session.openSession('aUsername','aPassword')
```

```
if firstTime:
```

```
    resp,ID = session.upload('content')
```

```
    firstTime = False
```

```
else:
```

```
    newcontent = session.read(ID)[1] + information
```

```
    session.edit(ID,newcontent,'aTitle','some,key,words')
```

Is there a way to do it without using another module? I would think just some reformatting (and possibly fixing some variable names) would be enough?

dbberube - August 23, 2012, 3:07 pm

There is always a workaround.

Textuplib have some advantages (no need to be logged, user friendly api) and you just need to copy past it in your code to use it (no need to import).

If you absolutely need to use pastebin, you will need to send an edit POST request since the pastbin api do not seem to support editing.

Here is the request:

Code POST /edit.php HTTP/1.1

Host: pastebin.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3

Accept-Encoding: gzip, deflate

Referer: <http://pastebin.com/edit.php?i=3GLk1FCL>

Cookie: some cookie

DNT: 1

Content-Length: 194

Content-Type: application/x-www-form-urlencoded

```
submit_hidden=submit_hidden&item_key=3GLk1FCL&paste_code=Text+to+upload+%28old+text+%2B+new+text%29&paste_format=Formatting&paste_expire_date=N&paste_private=0&paste_name=txtuplib3&submit=Submit
```

Please note that pastbin may refuse this request because the cookies are incorrect.