In [13]:
```
import pandas as pd
from requests import get
from time import sleep
from random import randint
import numpy as np
from bs4 import BeautifulSoup
import requests
from urllib.request import urlopen
```

In [20]:
```
url_main = "https://www.amazon.com/s?i=stripbooks&bbn=283155&rh=n%3A283155&s=review-count-rank&dc&Adv-Srch-Books-Submit.x=30&Adv-Srch-Books-Submit.y=21&qid=1602339864&unfiltered=1&ref=sr_ex_n_1%27"
r = requests.get(url_main)
soup_a = BeautifulSoup(r.content, "html.parser")
type (soup_a)
```

Out[20]:
```
bs4.BeautifulSoup
```

In [37]:
```
title_a = soup_a.title.content
print (title_a)
```

```
None
```

In [38]:
```
print (r.status_code)
```

```
503
```

In [54]:
```
import urllib.request
with urllib.request.urlopen('https://www.amazon.com/s?i=stripbooks&bbn=283155&rh=n%3A283155&s=review-count-rank&dc&Adv-Srch-Books-Submit.x=30&Adv-Srch-Books-Submit.y=21&qid=1602339864&unfiltered=1&ref=sr_ex_n_1%27') as response:
   html = response.read()
   print (html)
```

```
----------------------------------------------------------------------
```

```
----
HTTPError                                 Traceback (most recent call l
ast)
<ipython-input-54-e984fa118340> in <module>
      1 import urllib.request
----> 2 with urllib.request.urlopen('https://www.amazon.com/s?i=stripbo
oks&bbn=283155&rh=n%3A283155&s=review-count-rank&dc&Adv-Srch-Books-Subm
it.x=30&Adv-Srch-Books-Submit.y=21&qid=1602339864&unfiltered=1&ref=sr_e
x_n_1%27') as response:
      3     html = response.read()
      4     print (html)

~\anaconda3\lib\urllib\request.py in urlopen(url, data, timeout, cafil
e, capath, cadefault, context)
    220     else:
    221         opener = _opener
--> 222     return opener.open(url, data, timeout)
    223
    224 def install_opener(opener):

~\anaconda3\lib\urllib\request.py in open(self, fullurl, data, timeout)
    529         for processor in self.process_response.get(protocol, []
):
    530             meth = getattr(processor, meth_name)
--> 531             response = meth(req, response)
    532
    533         return response

~\anaconda3\lib\urllib\request.py in http_response(self, request, respo
nse)
    639         if not (200 <= code < 300):
    640             response = self.parent.error(
--> 641                 'http', request, response, code, msg, hdrs)
    642
    643         return response

~\anaconda3\lib\urllib\request.py in error(self, proto, *args)
    567         if http_err:
    568             args = (dict, 'default', 'http_error_default') + or
```

```
ig_args
--> 569             return self._call_chain(*args)
    570
    571 # XXX probably also want an abstract factory that knows when it
 makes

~\anaconda3\lib\urllib\request.py in _call_chain(self, chain, kind, met
h_name, *args)
    501         for handler in handlers:
    502             func = getattr(handler, meth_name)
--> 503             result = func(*args)
    504             if result is not None:
    505                 return result

~\anaconda3\lib\urllib\request.py in http_error_default(self, req, fp,
 code, msg, hdrs)
    647 class HTTPDefaultErrorHandler(BaseHandler):
    648     def http_error_default(self, req, fp, code, msg, hdrs):
--> 649         raise HTTPError(req.full_url, code, msg, hdrs, fp)
    650
    651 class HTTPRedirectHandler(BaseHandler):

HTTPError: HTTP Error 503: Service Unavailable
```

In [48]:
```
category_urls = [item.get("href") for item in soup_a.find_all('a')]
```

In [23]:
```
urls = list(dict.fromkeys(category_urls))
urls = list(filter(None, urls))
```

In [24]:
```
string = 'https://www.amazon.com/'
list_final = [string + s for s in urls]
```

In [25]:
```
list_final
```

Out[25]:
```
['https://www.amazon.com/https://www.amazon.com/gp/help/customer/displa
y.html/ref=footer_cou?ie=UTF8&nodeId=508088',
```

```
 'https://www.amazon.com/https://www.amazon.com/gp/help/customer/displa
y.html/ref=footer_privacy?ie=UTF8&nodeId=468496']
```

In [ ]: