

Forum:[Hack Forums](#)

Thread: [Looking for references to read](#)

0NUR0UX - January 8, 2017, 2:31 pm

Hey,

I want to make a project on Big Data analysis using Python.

Anyone has good links?

Thanks.

Dialog - January 8, 2017, 3:21 pm

<http://shop.oreilly.com/product/0636920023784.do>

LovelyBot - January 8, 2017, 5:11 pm

I highly recommend you this site/book

<http://guidetodatamining.com/>

I found it very useful, learning to datamine and to do analysis on big sets of data

Goodies - January 8, 2017, 11:00 pm

Highly recommended:

<https://pymotw.com/3/>

<https://pymotw.com/2/>

0NUR0UX - January 9, 2017, 7:08 am

Thank you all.

I appreciate it a lot!

0NUR0UX - January 15, 2017, 8:48 pm

So far, I can scrap data from the IMDb website.

Code def scrapData(url):

import urllib.request

from bs4 import BeautifulSoup

req = urllib.request.Request(url, headers={'User-Agent': 'Mozilla/5.0'})

open = urllib.request.urlopen(req)

readHtml = open.read()

soup = BeautifulSoup(readHtml, "html.parser")

tags = soup.find_all("tbody")

for movies in tags:

tdHeader = movies.findAll("td", attrs = {'class':"titleColumn"})

for link in tdHeader:

titleLink = link.find_all("a")

for title in titleLink:

```

theTitle = str(title)
start = theTitle.index(">")
end = theTitle[start:].index("<")
print(theTitle[start:(start+end)])

```

```
scrapData("http://www."+imdb.com/chart/top?ref_=nv_mv_250_6")
```

Goodies - January 16, 2017, 4:29 pm

(Jan 15, 2017 09:48 PM)ONUR0UX Wrote: [CITE](#) So far, I can scrap data from the IMDb website.

```

Code def scrapData(url):
    import urllib.request
    from bs4 import BeautifulSoup
    req = urllib.request.Request(url, headers={'User-Agent': 'Mozilla/5.0'})
    open = urllib.request.urlopen(req)
    readHtml = open.read()

    soup = BeautifulSoup(readHtml, "html.parser")

    tags = soup.find_all("tbody")
    for movies in tags:
        tdHeader = movies.findAll("td", attrs = {'class':"titleColumn"})
        for link in tdHeader:
            titleLink = link.find_all("a")
            for title in titleLink:

                theTitle = str(title)
                start = theTitle.index(">")
                end = theTitle[start:].index("<")
                print(theTitle[start:(start+end)])

```

```
scrapData("http://www."+imdb.com/chart/top?ref_=nv_mv_250_6")
```

If that's the main purpose of the program, there's no need to import the module within a function. You're not doing optional paths with different libraries, so you should just move the imports to the top.

Also, you should follow the [SRP](#). Having a function that does too many things (gets data and parses it) is not ideal. I recommend you separate them out into two functions and perhaps have one that combines them.

This will let you, for example, easily change out parsers.

PHP Code

```

<?php
def parse(url, parser):
    try:
        req = urllib.request.Request(url, headers={'User-Agent': 'Mozilla/5.0'})
        open = urllib.request.urlopen(req)
        response = open.read()
    except Exception as e:
        print(e)
    else:

```

```

    return parser(response)

def imdb_parser(html):
    # do the parsing here

def other_parser(html):
    # parse some other site

data = parse("http://imdb.com/whatever", imdb_parser)
print(data)

```

ONUR0UX - January 16, 2017, 6:35 pm

(Jan 16, 2017 05:29 PM)Goodies Wrote: [\[CITE\]](#) If that's the main purpose of the program, there's no need to import the module within a function. You're not doing optional paths with different libraries, so you should just move the imports to the top.

Also, you should follow the [SRP](#). Having a function that does too many things (gets data and parses it) is not ideal. I recommend you separate them out into two functions and perhaps have one that combines them.

This will let you, for example, easily change out parsers.

PHP Code

```

<?php
def parse(url, parser):
    try:
        req = urllib.request.Request(url, headers={'User-Agent': 'Mozilla/5.0'})
        open = urllib.request.urlopen(req)
        response = open.read()
    except Exception as e:
        print(e)
    else:
        return parser(response)

def imdb_parser(html):
    # do the parsing here

def other_parser(html):
    # parse some other site

data = parse("http://imdb.com/whatever", imdb_parser)
print(data)

```

Thanks man, you are awesome I really appreciate your help.