

Forum:[Hack Forums](#)

Thread: [How to get a simple string/number from website?](#)

FaberCastell - November 6, 2018, 8:44 am

Basically i just want to get integer value from the site.

<https://www.bitmex.com/app/trade/XBTUSD>

I want to get Open interest value, in html it is
814,106,194 USD

I tried with this code

```
# import libraries
import urllib.request
from bs4 import BeautifulSoup

# specify the url
quote_page = 'https://www.bitmex.com/app/trade/XBTUSD'

# query the website and return the html to the variable 'page'
page = urllib.request.urlopen(quote_page)

# parse the html using beautiful soup and store in variable `soup`
soup = BeautifulSoup(page, 'html.parser')

# Take out the <div> of name and get its value
name_box = soup.find('span', attrs={'class': 'value'})

name = name_box.text # strip() is used to remove starting and trailing
print (name)
```

But name_box doesn't return anything..well it returns "None"..
Anyidea how should i find the value?

gulliemam - November 6, 2018, 10:01 am

BitMEX offers a REST API for those things. Take a look at the Instrument section on bitmex.com/api/explorer . Just send an HTTP request and parse the response with a JSON library. You can also use the `requests` library, which is a bit simpler as it has a JSON decoder built-in (`response.json()`).

By the way, you don't need to comment every single part of your code, because it is already self-explanatory :)

FaberCastell - November 6, 2018, 11:59 am

(Nov 6, 2018 11:01 AM)gulliemam Wrote: [\[CITE\]](#) BitMEX offers a REST API for those things. Take a look at the Instrument section on bitmex.com/api/explorer . Just send an HTTP request and parse

the response with a JSON library. You can also use the `requests` library, which is a bit simpler as it has a JSON decoder built-in (`response.json()`).

By the way, you don't need to comment every single part of your code, because it is already self-explanatory :)
oh gosh, thanks lol.

i copied the code from some website D:

Infamia - November 6, 2018, 12:39 pm

get the url, split on `

to implement inherent conversion, simply split on `USD`, get the 0th item (to implement correct value filtration), replace commas with nothing and do a float conversion.

FaberCastell - November 6, 2018, 4:10 pm

ok will try. there's an API, but it shows combined value of perpetuals+march and december futures.. i want value for the perpetuals only, so i guess ill have to figure it out still

Qismon - November 7, 2018, 6:43 am

Regex is great for parsing text of any kind, for the string you pulled this should work if you are just trying to pull integers.

Code import re

```
htmlStr = r'<span class="value">814,106,194 USD</span>'
```

```
regexInt = re.findall(r'\d+', htmlStr)
```

```
for i in regexInt:  
    print(i)
```

If you need to extract the full value with the letters USD at the end, this regex will work more for your needs.

Code import re

```
htmlStr = r'<span class="value">814,106,194 USD</span>'
```

```
regexInt = re.findall(r'(([1-9]\d{0,2}(\,\d{3})*)|((([1-9]\d*)?\d))(\.\d\d)?([ ]))USD', htmlStr)
```

```
for i in regexInt:  
    print(i)
```

Infamia - November 7, 2018, 8:04 am

(Nov 7, 2018 07:43 AM)Qismon Wrote: [CITE](#) Regex is great for parsing text of any kind, for the string you pulled this should work if you are just trying to pull integers.

Code import re

```
htmlStr = r'<span class="value">814,106,194 USD</span>'
```

```
regexInt = re.findall(r'\d+', htmlStr)
```

```
for i in regexInt:
    print(i)
```

If you need to extract the full value with the letters USD at the end, this regex will work more for your needs.

Code import re

```
htmlStr = r'<span class="value">814,106,194 USD</span>'
```

```
regexInt = re.findall(r'((([1-9]\d{0,2}(\,\d{3})*)|((([1-9]\d*)?\d))(\.\d\d)?([ ]))USD', htmlStr)
```

```
for i in regexInt:
    print(i)
```

don't use regex for trivial purposes, thanks.

Qismon - November 7, 2018, 8:47 am

(Nov 7, 2018 09:04 AM)Infamia Wrote: [\[CITE\]](#) don't use regex for trivial purposes, thanks. Why not? it works fine.

Infamia - November 7, 2018, 9:08 am

(Nov 7, 2018 09:47 AM)Qismon Wrote: [\[CITE\]](#) Why not? it works fine.

much slower to compile and use the regex engine to use patterns opposed to using native string submethods, it accumulates whenever you loop it; which is usually the case for mass-scraping or scraping in general.

tu|pac - November 7, 2018, 11:23 am

(Nov 7, 2018 10:08 AM)Infamia Wrote: [\[CITE\]](#) much slower to compile and use the regex engine to use patterns opposed to using native string submethods, it accumulates whenever you loop it; which is usually the case for mass-scraping or scraping in general.

Well.... idk, if he is trying to get a simple price off a trading platform and doesn't want to use the REST api, he might not care about compile time. He would just need a number and regex is a very effective way to do that quick-n-dirty.

Infamia - November 7, 2018, 2:29 pm

(Nov 7, 2018 12:23 PM)tu|pac Wrote: [\[CITE\]](#) Well.... idk, if he is trying to get a simple price off a trading platform and doesn't want to use the REST api, he might not care about compile time. He would just need a number and regex is a very effective way to do that quick-n-dirty.

`regex is a very effective way to do that quick-n-dirty.`

it is neither effective nor quick

don't know what compile time you're talking about but okay.

Qismon - November 7, 2018, 2:38 pm

(Nov 7, 2018 12:23 PM)tu|pac Wrote: [\[CITE\]](#) Well.... idk, if he is trying to get a simple price off a trading platform and doesn't want to use the REST api, he might not care about compile time. He would just need a number and regex is a very effective way to do that quick-n-dirty.

Agreed! ?

A good Regex is extremely powerful and effective way, and as for compile time a few milliseconds doesn't matter.

@Infamia you should be more flexible, just because its not "your" way of doing it, doesn't mean its wrong or ineffective. Smoke a fucking joint and relax!