In [1]:

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

Python 3

Http, работа с web

MIPT 2020

основное про http - https://ru.wikipedia.org/wiki/HTTP (https://ru.wikipedia.org/wiki/HTTP)

HTML

In [2]:

```
%%file basic.html

<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8" />
      <title>HTML Document</title>
   </head>
   <body>
      <p>
         <b>
            Этот текст будет полужирным, <i>а этот — ещё и курсивным</i>.
         </b>
      </p>
   </body>
</html>
```

```
Overwriting basic.html
```

In [3]:

```
!firefox basic.html
```

Более продвинутые вещи нужно искать

PyPi - https://pypi.org/ (https://pypi.org/)

Urllib

In [4]:

```
import urllib
import http

with urllib.request.urlopen('http://yandex.ru') as f:
    type(f)
    f.read(100).decode('utf-8')
    f.getcode(), f.geturl(), f.headers
```

Out[4]:

```
http.client.HTTPResponse
```

Out[4]:

```
'<!DOCTYPE html><html class="i-ua_js_no i-ua_css_standart i-ua_browser_ i-ua_b
rowser_desktop document'
```

Out[4]:

```
(200, 'https://yandex.ru/', <http.client.HTTPMessage at 0x7fa69c4efb80>)
```

requests

Более высокоуровневая библиотека для запросов

In [5]:

```
import requests
```

In [6]:

```
with requests.get('http://yandex.ru') as f:
    f.text[:100], f.status_code, f.headers['Content-type']

# f.json()
```

Out[6]:

```
('<!DOCTYPE html><html class="i-ua_js_no i-ua_css_standart i-ua_browser_unknow
n i-ua_browser_desktop d',
 200,
 'text/html; charset=UTF-8')
```

aiohttp

In [7]:

```
import aiohttp

async with aiohttp.request('get', 'http://yandex.ru') as resp:
    resp_text = await resp.text()
    resp_text[:100], resp.status, resp.headers['Content-type']
```

Out[7]:

```
('<!DOCTYPE html><html class="i-ua_js_no i-ua_css_standart i-ua_browser_unknow
n i-ua_browser_desktop d',
 200,
 'text/html; charset=UTF-8')
```

Парсинг HTML

lxml

Warning

The xml.etree.ElementTree module is not secure against maliciously constructed data. If you need to parse untrusted or unauthenticated data see XML vulnerabilities

In [8]:

```
%%file my.xml

<cinema>
  <name>BestCinema</name>
  <films>
    <categories>
      <category>Action</category>
      <category>Thriller</category>
      <category>Soap opera</category>
    </categories>
  </films>
</cinema>
```

```
Overwriting my.xml
```

In [9]:

```
from lxml import etree

tree = etree.parse('my.xml')

root = tree.getroot()
root.tag

def print_all(node):
    print(f'{node.tag} {node.text}')
    for child in node:
        print_all(child)

print_all(root)
```

Out[9]:

```
'cinema'

cinema

name BestCinema
films

categories

category Action
category Thriller
category Soap opera
```

`root_iter` рекурсивно обходит xml

In [10]:

```
for child in root.iter('category'):
    print(f'{child.tag} {child.text}')
```

```
category Action
category Thriller
category Soap opera
```

In [11]:

```
data_string = """
<data>
    <country name="Liechtenstein">
        <rank>1</rank>
        <year>2008</year>
        <gdppc>141100</gdppc>
        <neighbor name="Austria" direction="E"/>
        <neighbor name="Switzerland" direction="W"/>
    </country>
    <country name="Singapore">
        <rank>4</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N"/>
    </country>
    <country name="Panama">
        <rank>68</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor name="Costa Rica" direction="W"/>
        <neighbor name="Colombia" direction="E"/>
    </country>
</data>
"""
```

In [12]:
```python
import xml.etree.ElementTree as ET

root = ET.fromstring(data_string)

countries = root.findall('country')
countries
```

Out[12]:
```
[<Element 'country' at 0x7fa694303bd0>,
 <Element 'country' at 0x7fa694290680>,
 <Element 'country' at 0x7fa694290810>]
```

In [13]:
```python
for country in countries:
    rank = country.find('rank').text
    name = country.get('name')
    print(rank, name)
```

```
1 Liechtenstein
4 Singapore
68 Panama
```

In [14]:
```python
for rank in root.iter('rank'):
    new_rank = int(rank.text) + 1
    rank.text = str(new_rank)
    rank.set('updated', 'yes')

ET.dump(root)
```

```
<data>
    <country name="Liechtenstein">
        <rank updated="yes">2</rank>
        <year>2008</year>
        <gdppc>141100</gdppc>
        <neighbor name="Austria" direction="E" />
        <neighbor name="Switzerland" direction="W" />
    </country>
    <country name="Singapore">
        <rank updated="yes">5</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N" />
    </country>
    <country name="Panama">
        <rank updated="yes">69</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor name="Costa Rica" direction="W" />
        <neighbor name="Colombia" direction="E" />
    </country>
</data>
```

In [15]:
```python
for neighbor in root.findall('./country/neighbor'):
    neighbor.get('name')
```

Out[15]: `'Austria'`

Out[15]: `'Switzerland'`

Out[15]: `'Malaysia'`

Out[15]: `'Costa Rica'`

Out[15]: `'Colombia'`

```
In [16]: for panama in root.findall("*[@name='Panama']"):
             panama.get('name')
             panama.find('year').text
```

```
Out[16]: 'Panama'

Out[16]: '2011'
```

```
In [17]: for year in root.findall("*[.='2011']"):
             year.text
```

BeautifulSoup

```
In [18]: from bs4 import BeautifulSoup
```

```
In [19]: async with aiohttp.request('get', 'http://yandex.ru') as resp:
             resp_text = await resp.text()
```

```
In [20]: soup = BeautifulSoup(resp_text, 'html')

         soup.title
```

```
Out[20]: <title>Яндекс</title>
```

```
In [21]: for child in soup.recursiveChildGenerator():
             if child.name == 'title':
                 child
```

```
Out[21]: <title>Яндекс</title>
```

```
In [22]: print(soup.prettify()[:1000])
```

```
<!DOCTYPE html>
<html class="i-ua_js_no i-ua_css_standart i-ua_browser_unknown i-ua_browser_de
sktop document_sticky-extra-logo_yes i-ua_platform_other" lang="ru">
 <head xmlns:og="http://ogp.me/ns#">
  <meta content="text/html;charset=utf-8" http-equiv="Content-Type"/>
  <meta content="IE=edge" http-equiv="X-UA-Compatible"/>
  <title>
   Яндекс
  </title>
  <link href="//yastatic.net/iconostasis/_/8lFaTHLDzmsEZz-5XaQg9iTWZGE.png" re
l="shortcut icon"/>
  <link href="//yastatic.net/iconostasis/_/5mdPq4V7ghRgzBvMkCaTzd2fjYg.png" re
l="apple-touch-icon" sizes="76x76"/>
  <link href="//yastatic.net/iconostasis/_/s-hGoCQMUosTziuARBks08IUxmc.png" re
l="apple-touch-icon" sizes="120x120"/>
  <link href="//yastatic.net/iconostasis/_/KnU823iWwj_vrPra7x9aQ-4yjRw.png" re
l="apple-touch-icon" sizes="152x152"/>
  <link href="//yastatic.net/iconostasis/_/wT9gfGZZ80sP0VsoR6dgDyXJf2Y.png" re
l="apple-touch-icon" sizes="180x180"/>
  <link href="https://yandex.ru/company/press_releases/news.rss" rel="alternat
```

```
In [23]: soup.find(rel="shortcut icon")
```

```
Out[23]: <link href="//yastatic.net/iconostasis/_/8lFaTHLDzmsEZz-5XaQg9iTWZGE.png" rel
         ="shortcut icon"/>
```

In [24]:
```
soup.find('title')
```

Out[24]:
```
<title>Яндекс</title>
```

In [25]:
```
import re

soup.find_all(string=re.compile("Я"))
```

Out[25]:
```
['Яндекс',
 'на Яндекс.Станции',
 'Яндекс.Учебник',
 'Популярные сервисы Яндекса',
 'Яндекс.Браузер',
 '©\xa0Яндекс',
 'setTimeout(function(){var bannerData={"pl_priority":3,"stat_delay_sec":2,"so
urce":"html5_hidpi_desktop_banner","image_alt":"Яндекс Станция","hidpi_imag
e":"https://awaps.yandex.net/0/c1/tVK-Oiz0m0j2bQxEwT0FkFckRV4pB11nzJKfRjDhmngf
-vJHRzvUbxq8s4+UD_t3K8IHaZahcDaDGGWQj2qFReNBf-hhZVQuUbpLA6IaXKMIr4DxyeX4-21pKm
e_t38mZqwZzSLjc3ZuhXamadj3MfQMQA5qGb3UOLlvM4m0BzwY8IeNQyt5-tAyE_tAktR75zj9JGzr
tXs0ZF4UA9AxkR35PSalnR6s3XDC3NC2xp1cJPILlBb1MWW_tOzoUaMoOAUJm4oBJ+eXZQiiPv5Wpg
mH3s5QZHA259IPmoSuw1zH-NTLcZqAr_tAYJAOWF4BdVwdSUElIaAjSc+HiC80F3WEWhUxfYGYAVZ9
ZLSrZhMqc+XC0JO_KYqB3MX0RSJIZtgAA_A_.gif","not_show_stat_url":"https://awaps.y
andex.net/99/c1/tx21lszVAoU5vGvVMTT5HdZERv2r9iSDrNklH+KEivNHYbVAZk0ZGO5vwtqEo_
tEzOCNa+pdVLr22jtt7kDWRJW1-IM2ewfxV-b5GhhOnplyNq7uJWcytRaBxK1_tk0UR01xzVVAr38C
5SCCkVbHkUKzO3FancmIVS0wshRI4qBruVcPX9HZk9kcT_tZ1AGkcmn4gChcbsdNW0-NSLhJW3708a
ksQkMVtB1I2-eN3+cZ5Ktu6uO9VCE_tOlnwcXRcI28mtFxKczvMlv+qEmNUY0LpgnwwgPL2iXG4CMM
LrZBTDTuLf71H_tWIeuEadc2m1eo2UBGaGBhmEQ+WDgyg6Srdijliltj1ZX2QnounMDPGc4Z+SF_S9
S0XIFsrNnkOsPZci5RpjkJ8_A_.gif?aa_reject=1","height":90,"bnCounts":["https://a
waps.yandex.net/0/c1/tx21lszVAoU5vGvVMTT5HdaEMHQi3x2zc22csYsgZfR1ZxT5tiRRyRsoX
7Dyy_tf5fAJyLjiSWFwrSl6MUK5A9iluB-MPLT+nL6dmX-SpRlWFwOoLJbQXuMeTSq_tRpLTNZ-UzM
g0eFrN0IFq0iV5dGlnOikfsnTszYlemcc4LNWZ5pX1Q7r4cVWj_thZJK98g77m0Eq+Uy9WCA4Jth4H
VzfEtQsW5m3ksAhkglys3we1qzJSHs8QSq_tsqkpNFI+0mZTYi+Fczu8kYtzE11B1G7ShiZSSLDKL4
oyUzbbSNUaIrGlFoyg_tRvmAX5Qpk-2BNi97G16Eh3LFwyHqoMHIOJ7TRAYYtZTvBBwy7JZ7g8wfC0
Yo_qj7XUGQYgRPshJFAhHch8D2M4pMBicHWJmysM66srm3leX51CV9gPR+GN_A_.gif?locale=ru&
morda_rid=1588128110.26039.82756.80703&pageview_id=1588128110.26039.82756.8070
3&slots=135685%2C0%2C99%3B228244%2C0%2C7%3B234859%2C0%2C19"],"width":728,"imag
e":"https://awaps.yandex.net/0/c1/tVK-Oiz0m0j2bQxEwT0FkFckRV4pB11nz7ftZxmhVLhP
7Yq6rzdarVmdzuZeH_t8PVDEU9uU7vP0veZRdanHf-HoXP7wHhJimBV0YzXNxRUr553DR1vv69IwdJ
1_tqrI3kt4DaSltt6EnUTvrTYohoLzsbhHC5cfiEhu40kTirI2qmBwLFb1aKS1x_tyVNEXnj58wuW5
k+H9jJmamGjtGlRWlSZAIEskLGWbYTGDMC2zA3HlAjRpj79_tO2BO3CSzsW0tbmCNHk+9n-H90gMBe
pt6+ySQa3QYwsyO1+ZPPFVxgLs9IOGG_ticsl676qvjl3TQh1Ihrml+IZENWBieuVjhBYQBNreluuF
CanQYzQZdiJSJ6u_KJh-0WMyOMFOrHwAA_A_.gif","win_notice":["https://awaps.yandex.
net/0/c1/tx21lszVAoU5vGvVMTT5HdaEMHQi3x2zc22csYsgZfR1ZxT5tiRRyRsoX7Dyy_tf5fAJy
LjiSWFwrSl6MUK5A9iluB-MPLT+nL6dmX-SpRlWFwOoLJbQXuMeTSq_tRpLTNZ-UzMg0eFrN0IFq0i
V5dGlnOhSTmKqf38Ur6BehNb7nUSni0TW1FyXe_tFc-+H3hqjhY2fUzKBY5M8ZMCZ8kff8U88iSMju
riGLKIBDEhNkrPtSH8Q5We_tHmNWhZZ+0URVhvrUx3ZeknuXMK2ALA64L+KoQso-0TkZLukYcNrZi6
j9fvmt_tWgXixlsUo8lJL7MyNpMEeuYI-qZlLWEZ67vh792EGMvL0N492zN55TJVeIil_tho8resp2
YakWX2wVTOjRBWeWSqm4L3fjr8c6wlWn2-mouHNlHvEjh6aZuqlC_NcwW5qok+OMzy3ZbRGwAA_A_.
gif"],"banner_id":3701606,"click_url":"https://awaps.yandex.net/1/c1/tx21lszVA
oU5Fo8Pyi8d5uyhSYZ-eo3P-rfZmDPO8PoZVdsGjhIYIEj5+G+Ud_tRSJZvoNIYz4GJJYyVSka2nQK
9ChjsOJ94UoIloVHaN21v2Jzo1VSPNtqVIZe_tNxfsZ6nZOayhoe5H9TAa7ykVB54MJoWrRDNF38dD
Q3OSEM6Iv16plz8mPwaa_tmJlfUnI17DwcsL-A+Qij+8rMyjVtFDtH60B1FKs5o0JDd4FWujEXDm7r
z0Ai_tUmZvNtilr8S+Q2f-6TLvorNLYnyX-d9W-bqQJiG8EiMYPKttjbW1lUWLaJh1_tFaLg95mh34
U+vaiHi+0YgLpybLWhWEYvO-QcdD3u1fDvX8BCJUtNk-6abamK_iYTZ29o5bmpvGADLDY2kR+y1l+V
wUSWYKs7jrELH-7BgewwAA_A_.htm","html5_iframe_src":"","contentCls":"b-banner__c
ontent","_refresh":{"limit":10,"tab_timeout":10,"watch_timeout":90},"darkThem
e":"","minFlashVersion":""}, bannerElem=document.querySelectorAll(\'.b-banner_
_wrap\')[0]; AwapsJsonAPI.Json.prototype.drawBanner(bannerElem, bannerData); A
wapsJsonAPI.Json.prototype.expand(bannerData);}, 100);']
```

Полноценный пример

In [26]:
```
import asyncio
```

In [27]:

```
sum_rating = 0

async with aiohttp.request('get', 'http://reddit.com') as resp:
    resp_text = await resp.text()

soup = BeautifulSoup(resp_text, 'html')
posts = soup.find_all(id=re.compile('t3_'))

for post in posts:
    upvotes = post.find(string=re.compile("[0-9].[0-9]k"))
    if upvotes:
        try:
            upvotes = float(upvotes[:-1])
        except:
            continue
        print(upvotes)
        sum_rating += upvotes

sum_rating /= 3
print(f'sum_rating is {sum_rating}')
```

```
37.0
37.0
37.0
21.6
21.6
21.6
53.8
53.8
53.8
45.7
45.7
45.7
14.9
14.9
14.9
72.9
72.9
72.9
25.3
25.3
25.3
sum_rating is 271.1999999999999
```

In [ ]: