

Python for Data Science

Web Scraping



Development in Africa with Radio Astronomy

Anna Scaife
University of Manchester



Science & Technology
Facilities Council

Training Data

All data science is only as good as the data we put in.

i.e. You can have the world's most perfect classifier, but if you have poor training data then it's worthless.

The data is worth more than the code.

Data is the New Oil



David Parkins

www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil

Libraries

Import `requests` library for handling weblinks:

```
import requests
```

Import `beautifulsoup` library for searching html:

```
from bs4 import BeautifulSoup as bs
```

Import `pytube` library for extracting data from youtube:

```
from pytube import YouTube
```

requests

```
# specify a web link:  
url = "https://www.somewebsite.com"  
  
# use the requests library to get the page:  
r = requests.get(url)
```

```
# extract the text from the web link:  
page = r.text
```

The text that is returned is the `html` of the webpage.

HTML

```
1 <!doctype html><html invert style="font-size: 10px;font-family: Roboto, Arial, sans-serif; background-color: #fafafa;"><head><meta http-equiv="origin-trial" data-feature="Media Capabilities" data-expires="2018-04-12" content="AjLq5un7MpG_eM34tWCJ3Dh8YzY1072ckfwdkbUNKGtUNazkrw55eq2tI60vGO1lsCNj33W9WmuV113EAsdHAWAAABpeyJvcmlnaW4i0iJodHRwcsovL31vdXR1YmUyZ9t0jQ0MyI slm1LYXrlcmJioiJNzWRpYUNhcGFiawxpdG1lcyc1sImV4cGlyeS16MTUyMzQ5MTIwMCwiAxNTdWJkb2lhaW4iOnRydWV9"><meta http-equiv="origin-trial" data-feature="Long Task Observer" data-expires="2017-04-17" content="Agxf9faupH8YmYNh1nbswBxxzTaTpz1j3At6FURcvdBzs018VxKDkfinT4bbXfPZX81XKFjotQzrhFvVnpzFwYAAABZeyJvcmlnaW4i0iJodHRwcsovL3d3dy55b3V0dWJ1LmNvbTo ONDiMiLCJmzWF0dxJ1ljoitG9suz1Rhct2PyN1ncnZlcisimV4cGlyeS16MTQ5Mj03NxWmH0"><script>var ytcfg = {d: function() {return (window.yt && yt.config_) || ytcfg.data_ || (ytcfg.data_ = ())},get: function(k, o) {return (k in ytcfg.d()) ? ytcfg.d_[k] : o},set: function() {var a = arguments;if (a.length > 1) (ytcfg.d_)[a[0]] = a[1];} else {for (var k in a[0]) (ytcfg.d_)[k] = a[0][k];}},window.ytcfg.set('EMERGENCY_BASE_URL', '/error_2047-client.name=1\u0026client.version=2.20180905\u0026level=ERROR\u0026t=jarserro');</script><link rel="shortcut icon" href="https://s.ytimg.com/yts/img/favicon-vfl8qSV2F.ico" type="image/x-icon"><link rel="icon" href="https://s.ytimg.com/yts/img/favicon_48-vflvjb_Qk.png" sizes="48x48"><link rel="icon" href="https://s.ytimg.com/yts/img/favicon_96-vflW99Ec0w.png" sizes="96x96"><link rel="icon" href="https://s.ytimg.com/yts/img/favicon_144-vfl1lAfab.png" sizes="144x144"><title>YouTube</title><script>var ytcsi = {gt: function(n) {n = [n || ''] + 'data';return ytcsi[n] || (ytcsi[n] = (tick: {}, info: {}))},now: window.performance && window.performance.timing && window.performance.now ? function() {return window.performance.timing.navigationStart + window.performance.timing && window.performance.now} : function() {return (new Date()).getTime()},getTime: function(l, t, n) {ticks = ytcsi.gt(n).ticks;var v = t || ytcsi.now();if (ticks[l]) {ticks['-' + l] = (ticks['-' + l] + 1) || [ticks[l]];ticks['-' + l].push(v);}ticks[l] = v;},info: function(k, v, n) {ytcsi.gt(n).info[k] = v;},setStart: function(s, t, n) {ytcsi.info['yt_sts', s, n];ytcsi.tick('_start', t, n);},(function(w, d) {ytcsi.setStart('dhs', w.performance ? w.performance.timing.responseStart : null);var isPrerender = (d.visibilityState || d.webkitVisibilityState) == 'prerender';var vName = (d.visibilityState || d.webkitVisibilityState) ? 'webkitvisibilitychange' : 'visibilitychange';if (isPrerender) {ytcsi.info['prerender', 1];var startTick = function() {ytcsi.setStart('dhs');d.removeEventListener(vName, startTick)};d.addEventListener(vName, startTick, false);};if (d.addEventListener) {d.addEventListener(vName, function() {ytcsi.tick('vc')}, false);}var s1t = function(el, t) {function(timeout(function() {var n = ytcsi.now();el.loadTime = n;if (el.s1t) {el.s1t();}}), t)};w._ytRIL = function(el) {if (!el.getAttribute('data-thumb')) {if (w.requestAnimationFrame) {w.requestAnimationFrame(function() {s1t(el, 0)});}} else {s1t(el, 16)};}};(window, document);</script>
2 <script src="https://s.ytimg.com/yts/jsbin/web-animations-next-lite.min-vflqEtsI7/web-animations-next-lite.min.js" type="text/javascript" name="web-animations-next-lite.min-web-animations-next-lite.min"></script>
3 <script>if (window.ytcsi) {window.ytcsi.tick('rsbe_dph', null, '')};</script>
4
5 <link rel="import" href="https://s.ytimg.com/yts/htmlbin/desktop_polymer-vflsQgFuS.html" name="desktop_polymer" > <script>if (window.ytcsi) {window.ytcsi.tick("rsae_dph", null, '')};</script>
6
```

beautifulsoup

beautifulsoup makes it easier for us to search through the html.

```
# use beautifulsoup to parse the html:  
soup=bs(page, 'html.parser')
```

```
# define the base url and the search string:  
base = "https://www.youtube.com/results?search_query="  
qstring = "cat+videos"  
  
# use the requests library to get the page:  
r = requests.get(base+qstring)  
  
# extract the text from the web link:  
page = r.text  
  
# use beautifulsoup to parse the html:  
soup=bs(page,'html.parser')
```

Find all the hyperlink elements with the html class attribute "yt-uix-tile-link":

```
vids = soup.findAll('a', attrs={'class':'yt-uix-tile-link'})
```

```
In [17]: print vids[0]
```

```
<a aria-describedby="description-id-172285" class="yt-uix-tile-link yt-ui-ellipsis yt-ui-ellipsis-2 yt-uix-sessionlink spf-link" data-sessionlink="itct=CFoQ3DAYACITCL_sxavhrd0CFU-o1QodmzYIFijoJFIFIY2F0IHZpZGVvcw" dir="ltr" href="/watch?v=pOmu0LtcI6Y" rel="spf-prefetch" title="It's TIME for SUPER LAUGH! - Best FUNNY CAT videos">It's TIME for SUPER LAUGH! - Best FUNNY CAT videos</a>
```

HTML reference:

<https://www.w3schools.com/html/default.asp>

YouTube

The funniest and most humorous cat videos ever! - Funny cat compilation

Tiger Productions 20M views • 1 year ago

Cats are awesome, and super funny too! Who doesn't like cats and kittens? They make us laugh and happy! Just look how they ...

It's TIME for SUPER LAUGH! - Best FUNNY CAT videos

Tiger Productions 1M views • 4 weeks ago

This was the first and funniest cat video ever! Get ready to wine and laugh in tears because this is en

Inspector Console Debugger Style Editor Performance Memory Network Storage

Rules Computed Layout Animations Fonts

```
< ytd-two-column-search-results-renderer class="style-scope ytd-search" center-results="guide-persistent-and-visible"> event
  <div id="primary" class="style-scope ytd-two-column-search-results-renderer">
    <ytd-section-list-renderer class="style-scope ytd-two-column-search-results-renderer"> event flex
      <div id="header-container" class="style-scope ytd-section-list-renderer"></div>
      <div id="contents" class="style-scope ytd-section-list-renderer"> event
        <ytd-item-section-renderer class="style-scope ytd-section-list-renderer"> event
          <div id="header" class="style-scope ytd-item-section-renderer"></div>
          <paper-spinner-lite class="style-scope ytd-item-section-renderer" aria-hidden="true"></paper-spinner-lite>
          <div id="contents" class="style-scope ytd-item-section-renderer">
            <ytd-video-renderer class="style-scope ytd-item-section-renderer"> event
              <div id="dismissible" class="style-scope ytd-video-renderer"> event flex
                <div id="dismissed" class="style-scope ytd-video-renderer"></div>
              </ytd-video-renderer>
            <ytd-video-renderer class="style-scope ytd-item-section-renderer"></ytd-video-renderer> event
            <ytd-video-renderer class="style-scope ytd-item-section-renderer"></ytd-video-renderer> event
          </div>
        </ytd-item-section-renderer>
      </div>
    </ytd-section-list-renderer>
  </div>
</ytd-two-column-search-results-renderer>
```

`beautifulsoup findAll` returns a list. We need to extract the `hyperref` for each element, i.e. the url:

```
videolist=[]
for v in vids:
    tmp = 'https://www.youtube.com' + v['href']
    videolist.append(tmp)
```

```
count=0
for item in videolist:

    # increment counter:
    count+=1

    # initiate the class:
    yt = YouTube(item)

    # have a look at the different formats available:
    #formats = yt.get_videos()

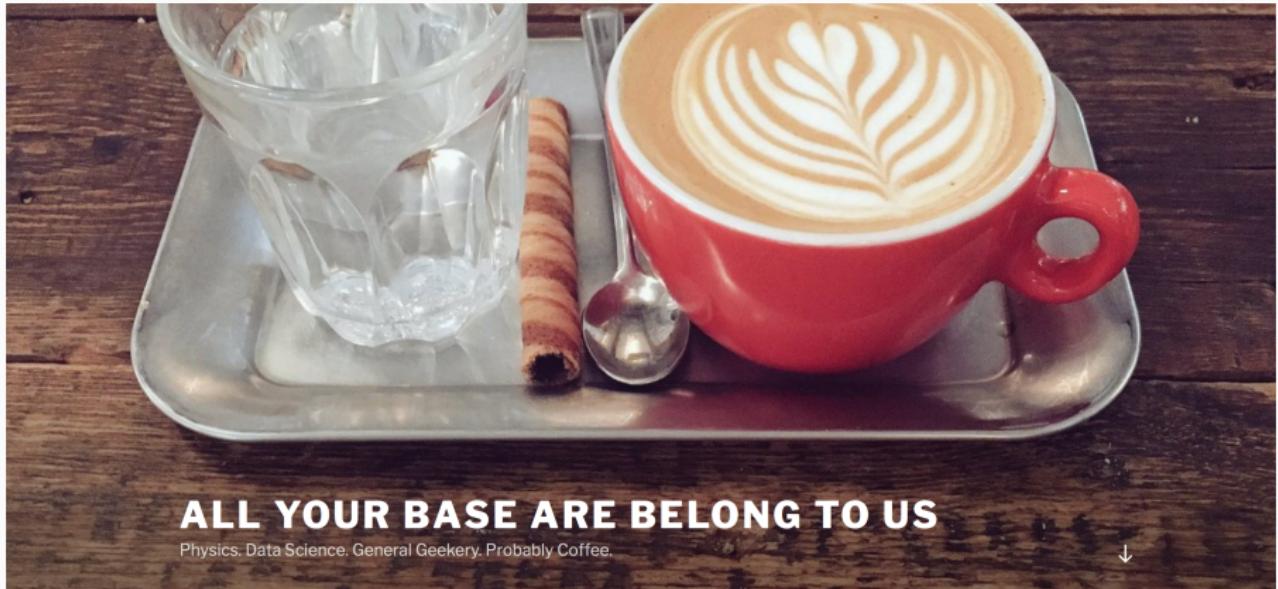
    # grab the video:
    video = yt.get('mp4', '360p')

    # set the output file name:
    yt.set_filename('Video_'+str(count))

    # download the video:
    video.download('./')
```

A cautionary note: the YouTube ToS do state that:

5.1 H. you agree not to use or launch any automated system (including, without limitation, any robot, spider or offline reader) that accesses the Service in a manner that sends more request messages to the YouTube servers in a given period of time than a human can reasonably produce in the same period by using a publicly available, standard (i.e. not modified) web browser;



ALL YOUR BASE ARE BELONG TO US

Physics. Data Science. General Geekery. Probably Coffee.

