Web Scraping



Web Scraping

Two ways to mine data from the web

- The hard way, by web scraping
- The easy way, using web service APIs

We'll see examples of both.

Web Scraping

Web scraping, a.k.a. screen scraping, means getting data from a web page. Suppose we want to get the current wind data for a city from Open Weather Map.



3/9

What is a Web Page?

A web page is a chunk of text containing HTML code. The browser "renders" the HTML graphically. So web scraping means analyzing text using Python's text processing features.

City	x 🕐 view-source:openweathern x		Chris
> C D view-source:openwe	thermap.org/city/2994160 😪 🍳 🥊	M	≡
1 <idoctvpe html=""></idoctvpe>			
shtml lange"en">shead>smeta	ttn-emily="Content_Type" contents"text(html: charact=UTF_8">		
<pre>s smeta charset="utf-8"></pre>	the offer severe size constraint, constraint of		
4 <1 <meta http-equiv="X-</p></td><td>IA-Compatible' content='IE=edge"/> >			
<pre>smeta names"viewport" cc</pre>	tent="width=device-width, initial=scale=1.0">		
s smeta names"keywords" co	tent="weather, world, Openstreetman, forecast, Openweather, current" >		
z smota names"description"	contents "Current weather conditions in cities for world wide" >		
s Smeta name="author" cont	nt="OpenNeatherNap.org">		
<pre>smeta name="domain" cont</pre>	nts OpenNeatherNan.org >		
smeta http=equive*pragma	content="no-cache" >		
smota http-emily="Bypire	" contents"-1" >		
stitlesCitys/titles			
to t			
crota paper's domain yes	fw" content="34fe329eab9562dea90f4a5962ff90a9"/>		
Class Rootstran, sore CSS			
<pre><l-link href="ces/boot</l-</pre>			
	Analysis for selections .		
Cl Custom styles for t	terniste>		
<pre>clink brof="/orrhine/70s</pre>	113 Complete		
stank neer- /company/ive	ACTIVITIES AND A		
clink brafe" (combine (64)			
VIIIK HIGI- /CONVINE/ OIL	accontraction of the second se		
CI Recourses>			
click brof- bttp://fonto	mentionnin conferentianternald-400, 200, pol-terletalenbert, tumor tert (conto		
<pre>clink href='http://fonte</pre>	And the second s		
stink hiel- heepit/ione	Addieapis.com/debrianity-Robotor400,/00.300.4001taile.3001taile.7001taile 101- Stylesneet		
cype- cexc/css >			
contint orest (combine (d)	11693439773-000-0201466-251340- 1400726584">		
Security and Tournalian	109443977380960201401622713408-1400710394 >>/8011pc>		
coorint types"text/invacorin	- M		
sacripe cype- text) javaserip			
var _yaygad [];	110 21601610 1111.		
yeq.push([setDomainName	ansatorester ;;;		
gaq.push([_secoularinalie	, openweachernap.org []]		
(function()) (
(runceron() {	Planesk/'semisk'), as kuns - 'kosk/isussemisk', as sound - kuns.		
var ga - document.create	stement; extrint); yettype - text/javaeotipt ; yetäyne = true;		
ya.oro - (https:// ac	<pre>imment.ideation.protector / netper//sst / netper//WWW) + '.google=analytics.com/ga.js ; imment.ideation.protector / netper//sst / netper//www) + '.google=analytics.com/ga.js ;</pre>		
var s = uocument.getElen	ncsbylagname(script)[0]; s.parenchous.insercherof@(gå, S)}		
2107			
v/script/			
<pre>soript type="text/javasorip"</pre>			
J runction snowSuccessNess(mes	1)		
1 1			
5 var html = ' <div class="</p></td><td><pre>ilert alert-success: >×'+mess+'</div> ';			

eorc

・ロン ・日ン・ ・日ン・ ・日ン

Finding The Data On the Page

First you need to find the data within the HTML code for a page so you can construct a regex. Your browser's developer features can help you find the data:

City	×	Chris			
← → C 📋 openweathe	rmap.org/city/2994160	🛣 🧶 🛡 😫 🗏			
		Immedia Control Succes Network >> 011 ; X			
Weather in your city Metz, FR * 16.7 °C					
Wind	1.5 m/s NorthEast (40)				
Cloudiness	sky is clear				
Pressure	1022 hpa	Clear			
Humidity	88 %	▶ , Large modal			
Sunrise	5:51	Wrong data modal Hoday id="wrong-data-modal" class="modal fade"			
Sunset	21:33	tabindex="-1" role="dialog">_ Wrong data modal			
Geo coords	[6.17, 49.12]	<pre>v-table class="table table-striped table-bordered table-condensed"> v v </pre>			
amazon.fr 435 - 335		v v -vito - vito -			

Georgia Tech

Getting the Web Page's HTML Code

To get the HTML code of the web page into a Python string variable that you can play with, use Python's urllib.request module:

```
import urllib.request
# 2994160 is the city code for Metz, FR
request = urllib.request.Request("http://www.openweathermap.com/city/2994160")
response = urllib.request.urlopen(request)
page_bytes = response.read()
page_text = page_bytes.decode()
# page_text is Python str containing the HTML code
```

or with the requests module, which is what we'll use:

```
import requests
resp = requests.get("http://www.openweathermap.com/city/2994160")
resp.text # the text of the web page
```

Georgia Tech • ০ > • ক্লি > • ই > • ই - ৩৭. 6/9 Looks like the wind data is in the second element after the <div class="weather-widget"> tag, following a Wind element. We can play around with the HTML text in the Python REPL. We eventually end up with:

wind = re.findall(r'Wind(.+?)', page_text.replace("\n",""))[0]

Notice that we used a capture group to get the element data.

HTML is context free language, which roughly means that it supports arbitrary nesting of elements. For example, you could have arbitrarily nested div elements with "leaf" elements containing text data, e.g.:

```
<div>
<div>
some text</div>
</div>
</div>
```

By the rules of HTML, you could nest div tags as deeply as you want. Regular expressions match regular laguages, which don't support arbitrary nesting. So how can we use regexes to "parse" HTML?

Regex Matching in HTML Code

Parsing means scanning the linear sequence of symbols in a string to determine its structure (usually by putting the symbols in a tree). We don't need to parse HTML to find data on a web page. While the HTML language supports arbitrary nesting, a particular web page will be nested to a particular depth, resulting a simple linear sequence of symbols that we can match with a regular expression.