Scraping with Python for Fun & Profit

@PyCon India 2010
http://in.pycon.org

Abhishek Mishra
hello@ideamonk.com

Now... I want you to put your _data_ on the web!

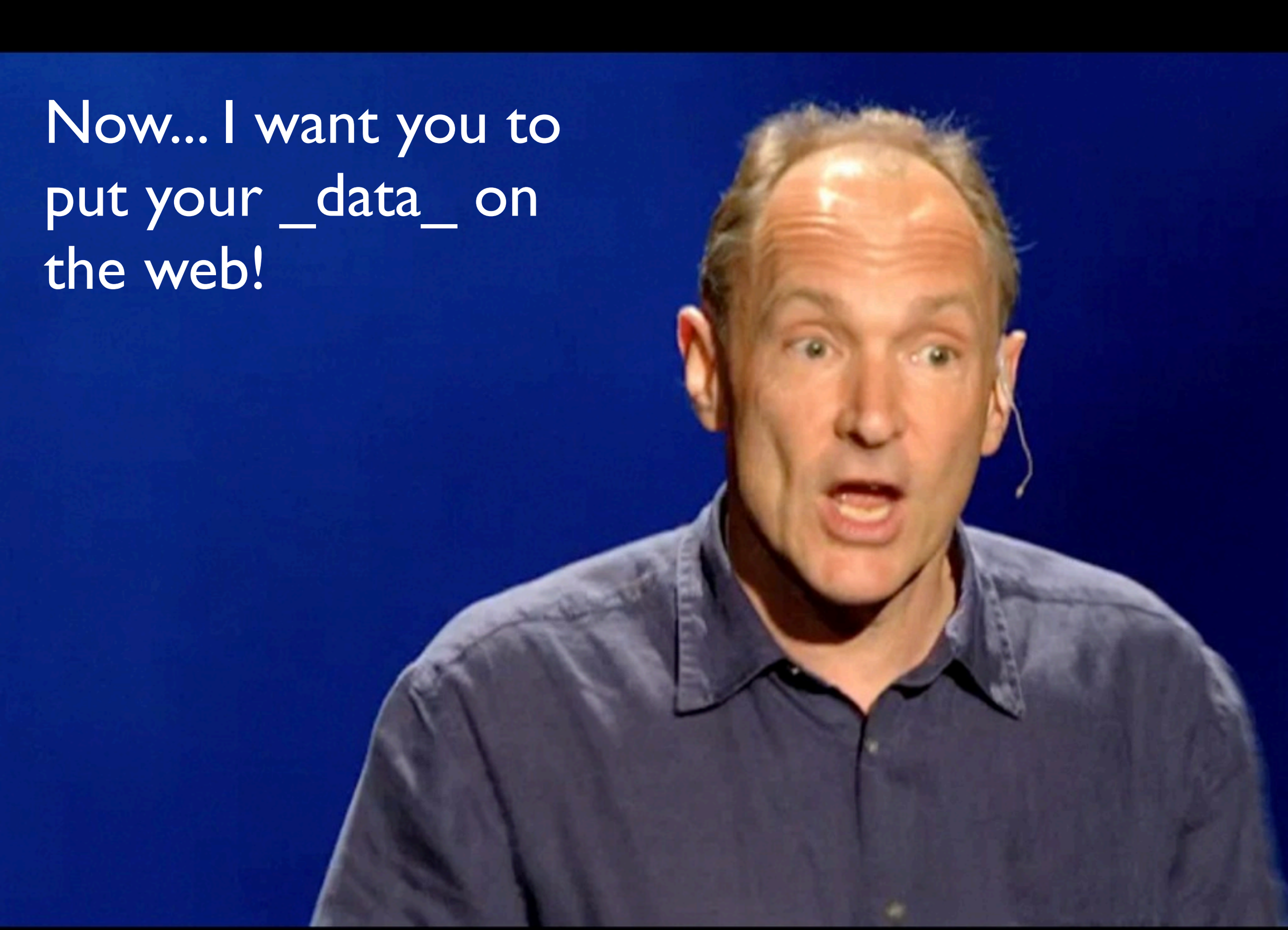
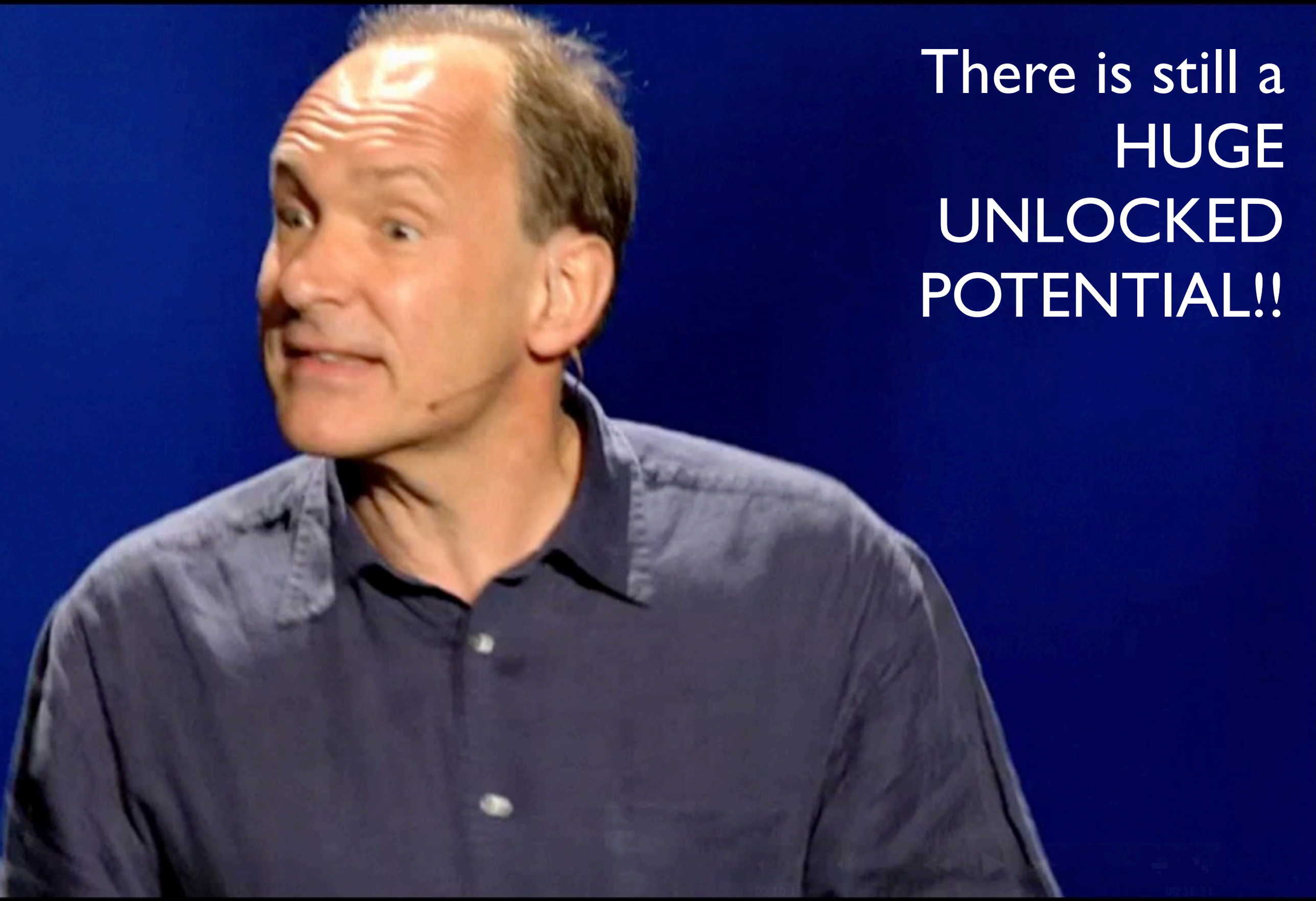


photo credit - TED - the next web http://www.ted.com/talks/lang/eng/tim_berners_lee_on_the_next_web.html



There is still a HUGE UNLOCKED POTENTIAL!!

photo credit - TED - the next web http://www.ted.com/talks/lang/eng/tim_berners_lee_on_the_next_web.html

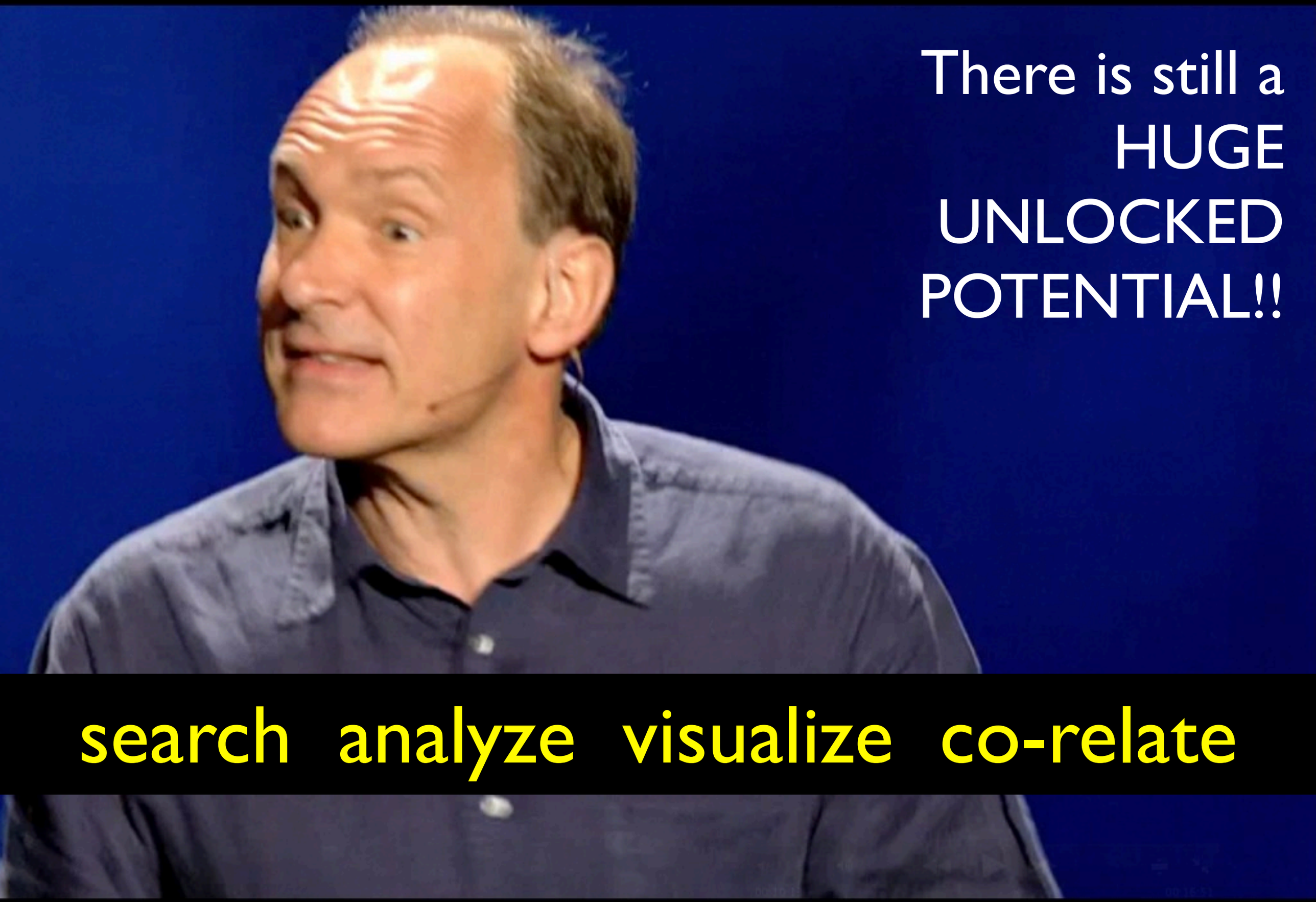


photo credit - TED - the next web http://www.ted.com/talks/lang/eng/tim_berners_lee_on_the_next_web.html

Haven't got DATA on the web as DATA!

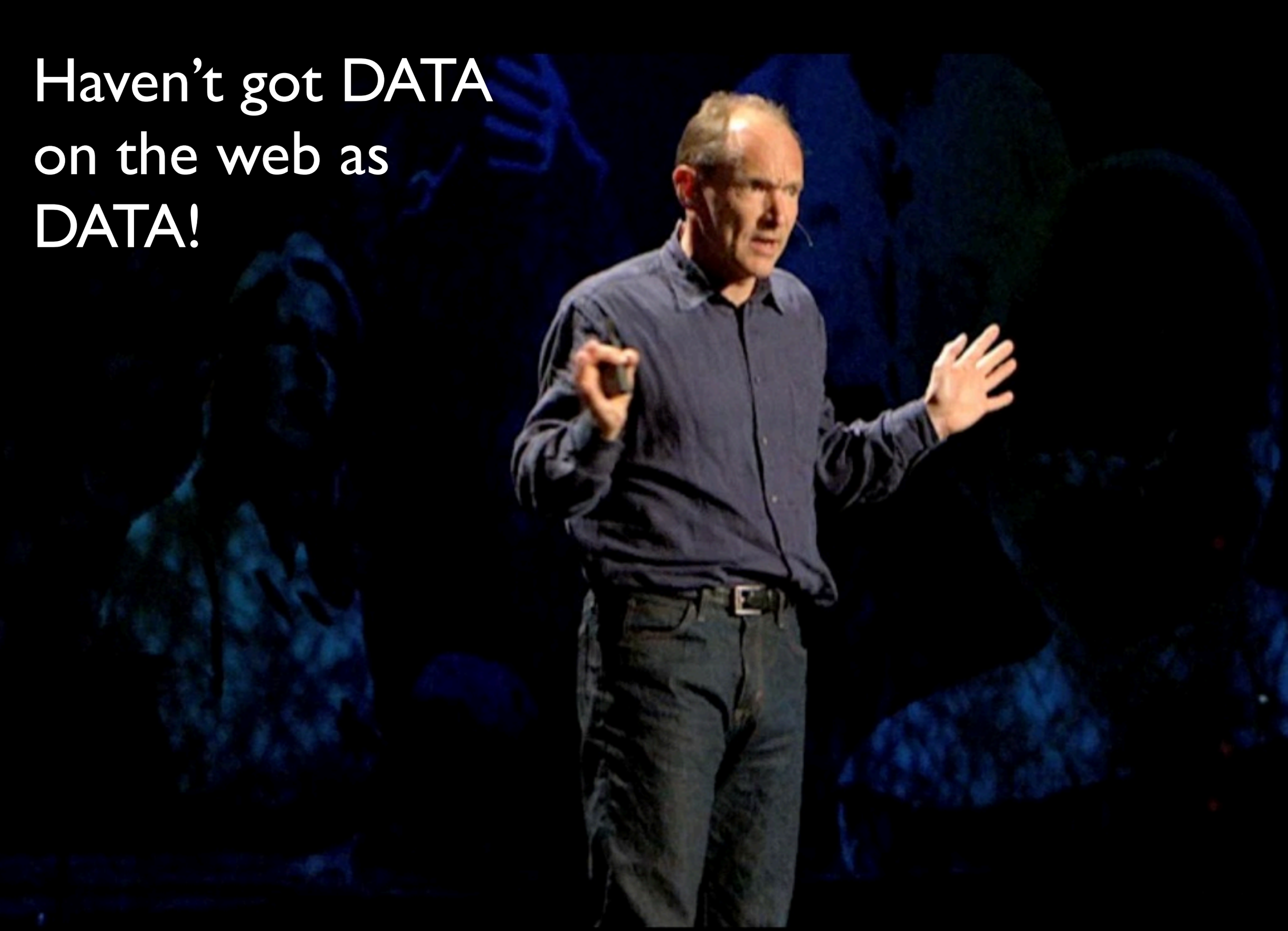


photo credit - TED - the next web http://www.ted.com/talks/lang/eng/tim_berners_lee_on_the_next_web.html

Haven't got DATA on the web as DATA!

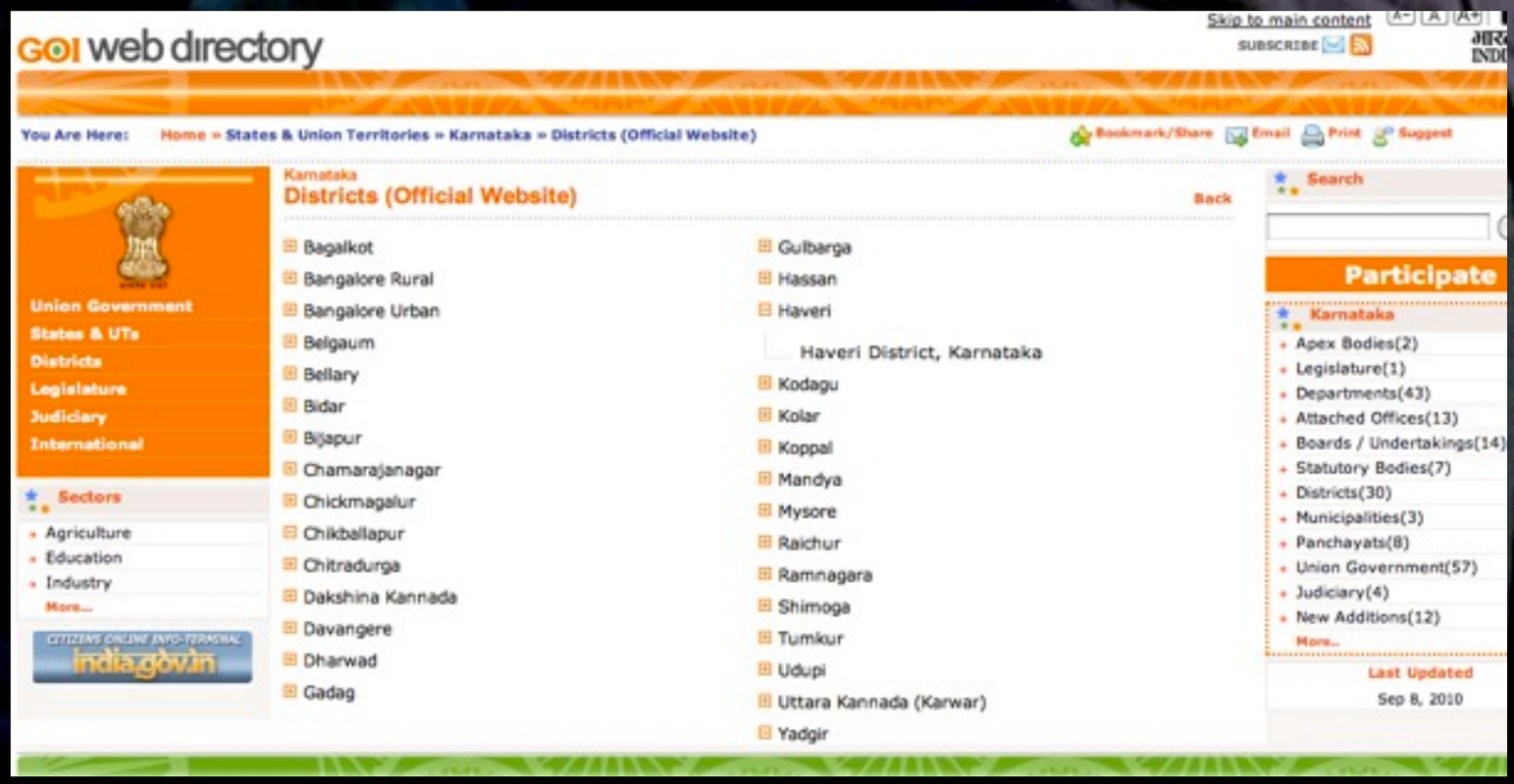


photo credit - TED - the next web http://www.ted.com/talks/lang/eng/tim_berners_lee_on_the_next_web.html

Haven't got DATA on the web as DATA!


```
648
649
    <ul  style="display:none" id="pKR009" class="bg_link">
650
    <li>
651
652
653
    <a
    href="javascript:openChild('http://goidirectory.nic.in
    /sitecounter.php?id=1439','win2');"
    class="heading_light_url"
    title="http://chickmagalur.nic.in">Chickmagalur
    District, Karnataka</a>
654
    </li>
655
    </ul>
```

photo credit - TED - the next web http://www.ted.com/talks/lang/eng/tim_berners_lee_on_the_next_web.html

FRUSTRATION !!!

1. will take a lot of time
2. the data isn't consumable
3. you want it right now!
4. missing source

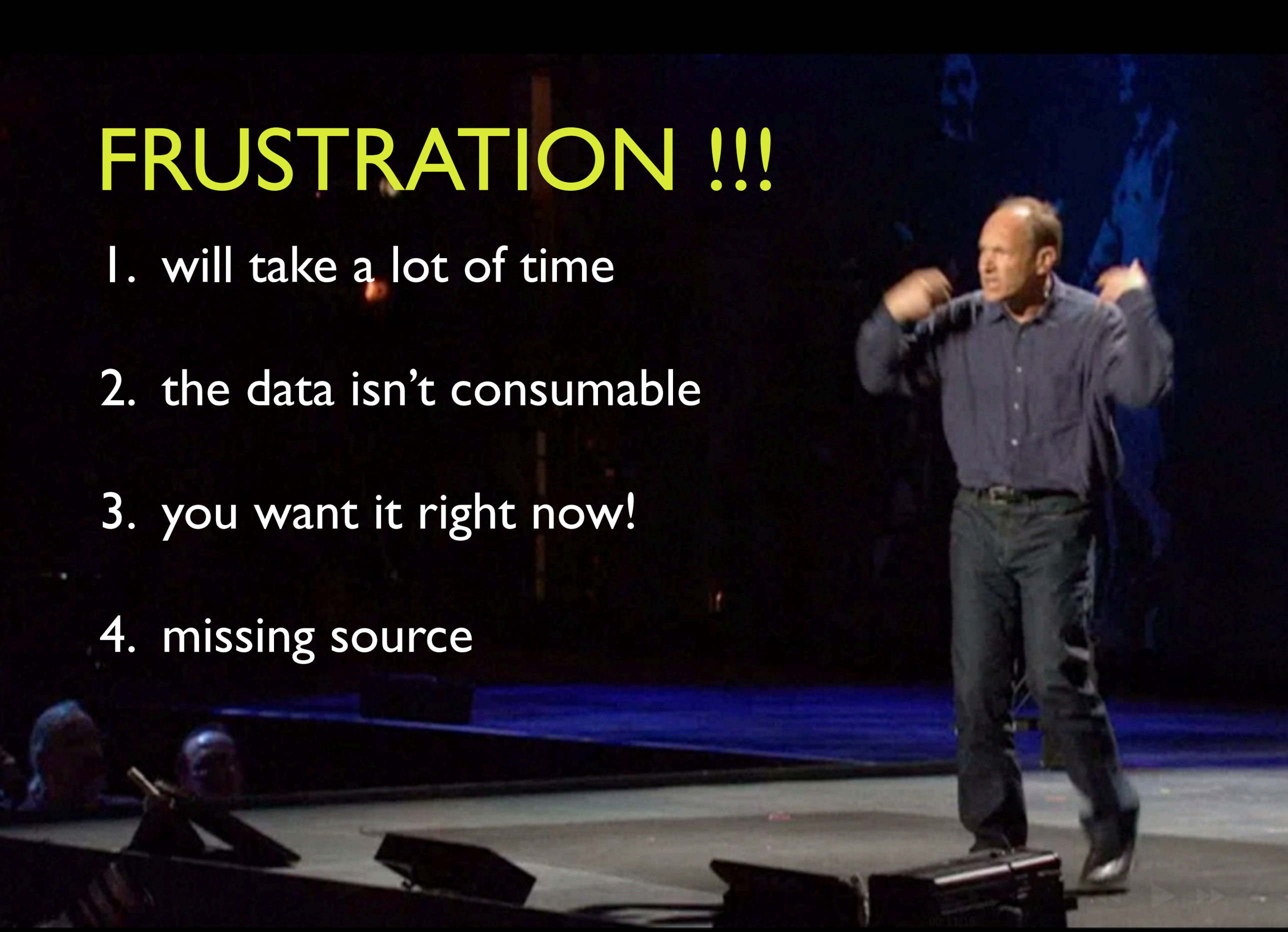


photo credit - TED - the next web http://www.ted.com/talks/lang/eng/tim_berners_lee_on_the_next_web.html

Solution ?

photo credit - Angie Pangie

Solution ?

just **scrape** it out!

photo credit - Angie Pangie

Solution ?

just **scrape** it out!

1. analyze request
2. find patterns
3. extract data

photo credit - Angie Pangie

Solution ?

just **scrape** it out!

1. analyze request
2. find patterns
3. extract data

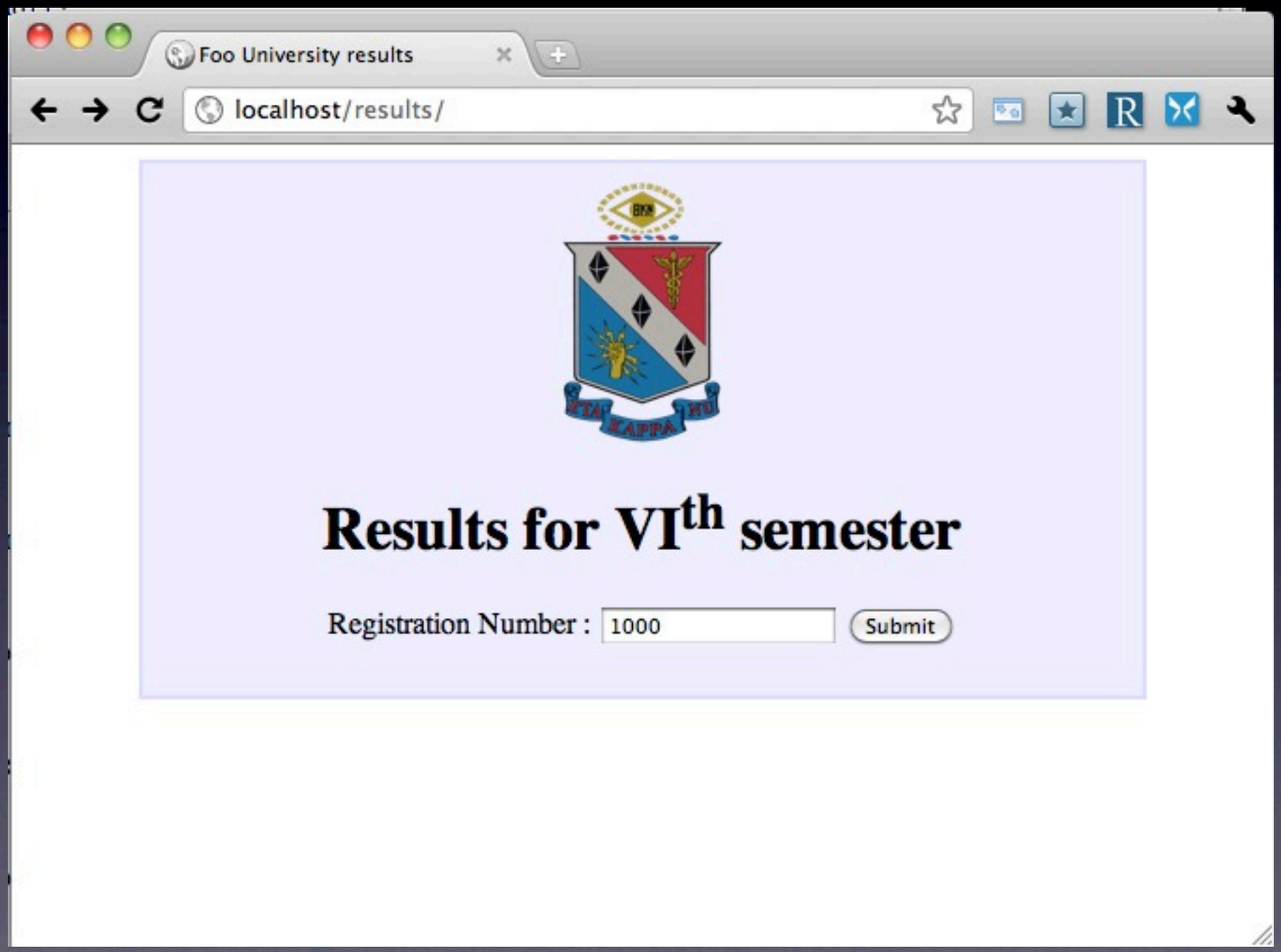
Use cases - **good** **bad** **ugly**

photo credit - Angie Pangie

Tools of Trade

- urllib2
  passing your data, getting back results
- BeautifulSoup
  parsing it out of the entangled web
- Mechanize
  programmatic web browsing
- Scrapy
  a web scraping framework

Our objective



Our objective

```
<body>
<div id="results">
<img src="shield.png" style="height:140px;"/>
<h1>Lohit Narayan
</h1>
<div>
Registration Number : 1234<br />
<table>
<tr>
<td>Object Oriented Programming</td>
<td> B+</td>
</tr>
<tr>
<td>Probability and Statistics</td>
<td> B+</td>
</tr>
<tr>
<td><b>CGPA</b></td>
<td><b>9.3</b></td>
</tr>
</table>
```

```
Name                Reg         CGPA
====                ===         ====
Lohit Narayan,      1234,       9.3
...                 ...         ...
```

urllib2

- Where to send?
- What to send?
- Answered by -
  firebug,
  chrome developer tools,
  a local debugging proxy

urllib2

```
import urllib
import urllib2

url = “http://localhost/results/results.php”

data = { ‘regno’ : ‘1000’ }
data_encoded = urllib.urlencode(data)

request = urllib2.urlopen( url, data_encoded )
html = request.read()
```

urllib2

```
import urllib
import urllib2

url = “http://localhost/results/results.php”

data = { ‘regno’ : ‘1000’ }
data_encoded = urllib.urlencode(data)

request = urllib2.urlopen( url, data_encoded )
html = request.read()
```

```
'<html>\n<head>\n\t<title>Foo University results</title>\n\t<style type="text/css">\n\t\t#results {\
n\t\t\tmargin:0px 60px;\n\t\t\tbackground: #efefff;\n\t\t\tpadding:10px;\n\t\t\tborder:2px solid #dd
ddff;\n\t\t\ttext-align:center;\n\t\t}\n\t</style>\n</head>\n<body>\n\t<div id="results">\n\t<img sr
c="shield.png" style="height:140px;"/>\n\t<h1>Rahim Kumar</h1>\n\t<div>\n\t\tRegistration Number   :
1000\t\t<br/>\n\t\t<table>\n\t\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t\t<td>Humanities Elective II</td>\n\t\t\t\t\
t<td> B-</td>\n\t\t\t\t</tr>\n\t\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t\t<td>Humanities Elective III</td>\n\t\t\t
\t\t<td> A</td>\n\t\t\t\t</tr>\n\t\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t\t<td>Discrete Mathematics</td>\n\t\t\t\
t\t<td> C-</td>\n\t\t\t\t</tr>\n\t\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t\t<td>Environmental Science and Engineer
ing</td>\n\t\t\t\t\t<td> B</td>\n\t\t\t\t</tr>\n\t\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Data Structures</t
d>\n\t\t\t\t\t<td> D</td>\n\t\t\t\t</tr>\n\t\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Science Elective I</td>\
n\t\t\t\t\t<td> D-</td>\n\t\t\t\t</tr>\n\t\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Object Oriented Programmin
g</td>\n\t\t\t\t\t<td> B</td>\n\t\t\t\t</tr>\n\t\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Probability and Stat
istics</td>\n\t\t\t\t\t<td> A-</td>\n\t\t\t\t</tr>\n\t\t\t\t\t\t\t\t<tr>\n\t\t\t\t<td><b>CGPA</b></td>\n
   \t\t\t\t<td><b>9.1</b></td>\n\t\t\t</tr>\n\t\t</table>\n\t</div>\n\t</div>\n</body>\n</html\n'
```

urllib2

```
import urllib   # for urlencode
import urllib2  # main urllib2 module

url = “http://localhost/results/results.php”

data = { ‘regno’ : ‘1000’ }
# ^ data as an object

data_encoded = urllib.urlencode(data)
# a string, key=val pairs separated by ‘&’

request = urllib2.urlopen( url, data_encoded )
# ^ returns a network object

html = request.read()
# ^ reads out its contents to a string
```

urllib2

```
import urllib
import urllib2

url = “http://localhost/results/results.php”
data = {}
scraped_results = []

for i in xrange(1000, 3000):
    data[‘regno’] = str(i)
    data_encoded = urllib.urlencode(data)
    request = urllib2.urlopen(url, data_encoded)
    html = request.read()
    scraped_results.append(html)
```

'<html>\n<head>\n\t<title>Foo University results</title>\n\t<style type="text/css">\n\t\t#results {\n\t\t\tmargin:0px 60px;\n\t\t\tbackground: #efefff;\n\t\t\tpadding:10px;\n\t\t\tborder:2px solid #ddddff;\n\t\t\ttext-align:center;\n\t\t}\n\t</style>\n</head>\n<body>\n\t \n\t\n\t<h1>Rahim Kumar</h1>\n\t \n\t\tRegistration Number : 1000\t\t
\n\t\t<table>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Humanities Elective II</td>\n\t\t\t\t\t<td> B-</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Humanities Elective III</td>\n\t\t\t\t\t<td> A</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Discrete Mathematics</td>\n\t\t\t\t\t<td> C-</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Environmental Science and Engineering</td>\n\t\t\t\t\t<td> B</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Data Structures</td>\n\t\t\t\t\t<td> D</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Science Elective I</td>\n\t\t\t\t\t<td> D-</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Object Oriented Programming</td>\n\t\t\t\t\t<td> B</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Probability and Statistics</td>\n\t\t\t\t\t<td> A-</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>CGPA</td>\n\t\t\t\t\t<td>9.1</td>\n\t\t\t</tr>\n\t\t</table>\n\t \n\t \n</body>\n</html\n'

Thats alright,
But this is still meaningless.

'<html>\n<head>\n\t<title>Foo University results</title>\n\t<style type="text/css">\n\t\t#results {\n\t\t\tmargin:0px 60px;\n\t\t\tbackground: #efefff;\n\t\t\tpadding:10px;\n\t\t\tborder:2px solid #ddddff;\n\t\t\ttext-align:center;\n\t\t}\n\t</style>\n</head>\n<body>\n\t \n\t\n\t<h1>Rahim Kumar</h1>\n\t \n\t\tRegistration Number : 1000\t\t
\n\t\t<table>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Humanities Elective II</td>\n\t\t\t\t\t<td> B-</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Humanities Elective III</td>\n\t\t\t\t\t<td> A</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Discrete Mathematics</td>\n\t\t\t\t\t<td> C-</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Environmental Science and Engineering</td>\n\t\t\t\t\t<td> B</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Data Structures</td>\n\t\t\t\t\t<td> D</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Science Elective I</td>\n\t\t\t\t\t<td> D-</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Object Oriented Programming</td>\n\t\t\t\t\t<td> B</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>Probability and Statistics</td>\n\t\t\t\t\t<td> A-</td>\n\t\t\t</tr>\n\t\t\t\t\t\t\t<tr>\n\t\t\t\t\t<td>CGPA</td>\n\t\t\t\t\t<td>9.1</td>\n\t\t\t</tr>\n\t\t</table>\n\t \n\t \n</body>\n</html\n'

Thats alright,
But this is still meaningless.

Enter BeautifulSoup

BeautifulSoup

http://www.flickr.com/photos/29233640@N07/3292261940/

BeautifulSoup

- Python HTML/XML Parser
- Wont choke you on bad markup
- auto-detects encoding
- easy tree traversal
- find all links whose url contains ‘foo.com’

BeautifulSoup

```
>>> import re
>>> from BeautifulSoup import BeautifulSoup

>>> html = """<a href="http://foo.com/foobar">link1</a> <a
href="http://foo.com/beep">link2</a> <a href="http://
foo.co.in/beep">link3</a>"""

>>> soup = BeautifulSoup(html)
>>> links = soup.findAll('a', {'href': re.compile
(".*foo.com.*")})
>>> links
[<a href="http://foo.com/foobar">link1</a>, <a href="http://
foo.com/beep">link2</a>]

>>> links[0][‘href’]
u'http://foo.com/foobar'

>>> links[0].contents
[u'link1']
```

BeautifulSoup

```
>>> html = """<html>\n <head>\n  <title>\n   Foo University results\n  </title>\n  <style type="text/css">\n   #results {\n
   \t\t\tmargin:0px 60px;\n\t\t\tbackground: #efefff;\n\t\t\tpadding:10px;\n\t\t\tborder:2px solid #ddddff;\n\t\t\ttext-
align:center;\n\t\t}\n  </style>\n </head>\n <body>\n  <div id="results">\n   <img src="shield.png" style="height:140px;" /
  >\n   <h1>\n    Bipin Narayan\n   </h1>\n   <div>\n    Registration Number : 1000\n    <br />\n    <table>\n     <tr>\n
 <td>\n       Humanities Elective II\n      </td>\n      <td>\n       C-\n      </td>\n     </tr>\n     <tr>\n      <td>\n
       Humanities Elective III\n      </td>\n      <td>\n       B-\n      </td>\n     </tr>\n     <tr>\n      <td>\n
         Discrete Mathematics\n      </td>\n      <td>\n       C\n      </td>\n     </tr>\n     <tr>\n      <td>\n
 Environmental Science and Engineering\n      </td>\n      <td>\n       C\n      </td>\n     </tr>\n     <tr>\n      <td>\n
Data Structures\n      </td>\n      <td>\n       A\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Science Elective
I\n      </td>\n      <td>\n       B-\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Object Oriented Programming\n
 </td>\n      <td>\n       B+\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Probability and Statistics\n      </
td>\n      <td>\n       A\n      </td>\n     </tr>\n     <tr>\n      <td>\n       <b>\n        CGPA\n       </b>\n      </
 td>\n      <td>\n       <b>\n        8\n       </b>\n      </td>\n     </tr>\n    </table>\n   </div>\n  </div>\n </body>
                                                \n</html>"""
```

BeautifulSoup

```
>>> html = """<html>\n <head>\n  <title>\n   Foo University results\n  </title>\n  <style type="text/css">\n   #results {\n
   \t\t\tmargin:0px 60px;\n\t\t\tbackground: #efefff;\n\t\t\tpadding:10px;\n\t\t\tborder:2px solid #ddddff;\n\t\t\ttext-
align:center;\n\t\t}\n  </style>\n </head>\n <body>\n  <div id="results">\n   <img src="shield.png" style="height:140px;" /
  >\n   <h1>\n    Bipin Narayan\n   </h1>\n   <div>\n    Registration Number : 1000\n    <br />\n    <table>\n     <tr>\n
 <td>\n       Humanities Elective II\n      </td>\n      <td>\n       C-\n      </td>\n     </tr>\n     <tr>\n      <td>\n
       Humanities Elective III\n      </td>\n      <td>\n       B-\n      </td>\n     </tr>\n     <tr>\n      <td>\n
         Discrete Mathematics\n      </td>\n      <td>\n       C\n      </td>\n     </tr>\n     <tr>\n      <td>\n
 Environmental Science and Engineering\n      </td>\n      <td>\n       C\n      </td>\n     </tr>\n     <tr>\n      <td>\n
Data Structures\n      </td>\n      <td>\n       A\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Science Elective
I\n      </td>\n      <td>\n       B-\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Object Oriented Programming\n
 </td>\n      <td>\n       B+\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Probability and Statistics\n      </
 td>\n      <td>\n       A\n      </td>\n     </tr>\n     <tr>\n      <td>\n       <b>\n        CGPA\n       </b>\n      </
 td>\n      <td>\n       <b>\n        8\n       </b>\n      </td>\n     </tr>\n    </table>\n   </div>\n  </div>\n </body>
                                                \n</html>"""
```

```
>>> soup.h1
<h1>Bipin Narang</h1>
>>> soup.h1.contents
[u'Bipin Narang']
>>> soup.h1.contents[0]
u'Bipin Narang'
```

BeautifulSoup

```
>>> html = """<html>\n <head>\n  <title>\n   Foo University results\n  </title>\n  <style type="text/css">\n   #results {\n
   \t\t\tmargin:0px 60px;\n\t\t\tbackground: #efefff;\n\t\t\tpadding:10px;\n\t\t\tborder:2px solid #ddddff;\n\t\t\ttext-
align:center;\n\t\t}\n  </style>\n </head>\n <body>\n  <div id="results">\n   <img src="shield.png" style="height:140px;" /
  >\n   <h1>\n    Bipin Narayan\n   </h1>\n   <div>\n    Registration Number : 1000\n    <br />\n    <table>\n     <tr>\n
 <td>\n       Humanities Elective II\n      </td>\n      <td>\n       C-\n      </td>\n     </tr>\n     <tr>\n      <td>\n
       Humanities Elective III\n      </td>\n      <td>\n       B-\n      </td>\n     </tr>\n     <tr>\n      <td>\n
         Discrete Mathematics\n      </td>\n      <td>\n       C\n      </td>\n     </tr>\n     <tr>\n      <td>\n
 Environmental Science and Engineering\n      </td>\n      <td>\n       C\n      </td>\n     </tr>\n     <tr>\n      <td>\n
Data Structures\n      </td>\n      <td>\n       A\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Science Elective
I\n      </td>\n      <td>\n       B-\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Object Oriented Programming\n
 </td>\n      <td>\n       B+\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Probability and Statistics\n      </
td>\n      <td>\n       A\n      </td>\n     </tr>\n     <tr>\n      <td>\n       <b>\n        CGPA\n       </b>\n      </
 td>\n      <td>\n       <b>\n        8\n       </b>\n      </td>\n     </tr>\n    </table>\n   </div>\n  </div>\n </body>
                                                \n</html>"""
```

```
>>> soup.div.div.contents[0]
u'\n\t\tRegistration Number : 1000\t\t'
>>> soup.div.div.contents[0].split(':')
[u'\n\t\tRegistration Number ', u' 1000\t\t']
>>> soup.div.div.contents[0].split(':')[1].strip()
u'1000'
```

BeautifulSoup

```
>>> html = """<html>\n <head>\n  <title>\n   Foo University results\n  </title>\n  <style type="text/css">\n   #results {\n
   \t\t\tmargin:0px 60px;\n\t\t\tbackground: #efefff;\n\t\t\tpadding:10px;\n\t\t\tborder:2px solid #ddddff;\n\t\t\ttext-
align:center;\n\t\t}\n  </style>\n </head>\n <body>\n  <div id="results">\n   <img src="shield.png" style="height:140px;" /
  >\n   <h1>\n    Bipin Narayan\n   </h1>\n   <div>\n    Registration Number : 1000\n    <br />\n    <table>\n     <tr>\n
 <td>\n       Humanities Elective II\n      </td>\n      <td>\n       C-\n      </td>\n     </tr>\n     <tr>\n      <td>\n
       Humanities Elective III\n      </td>\n      <td>\n       B-\n      </td>\n     </tr>\n     <tr>\n      <td>\n
         Discrete Mathematics\n      </td>\n      <td>\n       C\n      </td>\n     </tr>\n     <tr>\n      <td>\n
 Environmental Science and Engineering\n      </td>\n      <td>\n       C\n      </td>\n     </tr>\n     <tr>\n      <td>\n
Data Structures\n      </td>\n      <td>\n       A\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Science Elective
I\n      </td>\n      <td>\n       B-\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Object Oriented Programming\n
 </td>\n      <td>\n       B+\n      </td>\n     </tr>\n     <tr>\n      <td>\n       Probability and Statistics\n      </
 td>\n      <td>\n       A\n      </td>\n     </tr>\n     <tr>\n      <td>\n       <b>\n        CGPA\n       </b>\n      </
 td>\n      <td>\n       <b>\n        8\n       </b>\n      </td>\n     </tr>\n    </table>\n   </div>\n  </div>\n </body>
                                                \n</html>"""
```

```
>>> soup.find('b')
<b>CGPA</b>
>>> soup.find('b').findNext('b')
<b>8</b>
>>> soup.find('b').findNext('b').contents[0]
u'8'
```

urllib2 + BeautifulSoup

```
import urllib
import urllib2
from BeautifulSoup import BeautifulSoup

url = "http://localhost/results/results.php"
data={}
for i in xrange(1000,1100):
    data['regno'] = str(i)
    html = urllib2.urlopen( url, urllib.urlencode(data) ).read()
    soup = BeautifulSoup(html)
    name = soup.h1.contents[0]
    regno = soup.div.div.contents[0].split(':')[1].strip()
    # ^ though we alread know current regno ;) == str(i)
    cgpa = soup.find('b').findNext('b').contents[0]
    print "%s,%s,%s" % (regno, name, 'cgpa')
```

Mechanize

http://www.flickr.com/photos/flod/3092381868/

Mechanize

Stateful web browsing in Python

after Andy Lester's WWW:Mechanize

fill forms
follow links
handle cookies
browser history

http://www.flickr.com/photos/flod/3092381868/

Mechanize

```
>>> import mechanize
>>> br = mechanize.Browser()
>>> response = br.open(“http://google.com”)
>>> br.title()
‘Google’
>>> br.geturl()
'http://www.google.co.in/'
>>> print response.info()
Date: Fri, 10 Sep 2010 04:06:57 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
Server: gws
X-XSS-Protection: 1; mode=block
Connection: close
content-type: text/html; charset=ISO-8859-1
>>> br.open("http://yahoo.com")
>>> br.title()
'Yahoo! India'
>>> br.back()
>>> br.title()
'Google'
```

Mechanize

filling forms

```
>>> import mechanize
>>> br = mechanize.Browser()
>>> response = br.open("http://in.pycon.org/2010/account/
login")
>>> html = response.read()
>>> html.find('href="/2010/user/ideamonk">Abhishek |')
-1

>>> br.select_form(name="login")
>>> br.form["username"] = "ideamonk"
>>> br.form["password"] = "**********"
>>> response2 = br.submit()
>>> html = response2.read()
>>> html.find('href="/2010/user/ideamonk">Abhishek |')
10186
```

Mechanize

filling forms

```
>>> import mechanize
>>> br = mechanize.Browser()
>>> response = br.open("http://in.pycon.org/2010/account/
login")
>>> html = response.read()
>>> html.find('href="/2010/user/ideamonk">Abhishek |')
-1

>>> br.select_form(name="login")
>>> br.form["username"] = "ideamonk"
>>> br.form["password"] = "**********"
>>> response2 = br.submit()
>>> html = response2.read()
>>> html.find('href="/2010/user/ideamonk">Abhishek |')
10186
```

Mechanize

logging into gmail

Mechanize

logging into gmail

```
▼<form id="gaia_loginform" action="https:
  "return(gaia_onLoginSubmit());">
```

the form has no name!

cant select_form it

Mechanize

logging into gmail

```
>>> br.open("http://mail.google.com")
>>> login_form = br.forms().next()
>>> login_form["Email"] = "ideamonk@gmail.com"
>>> login_form["Passwd"] = "***************"
>>> response = br.open( login_form.click() )
>>> br.geturl()
'https://mail.google.com/mail/?shva=1'
>>> response2 = br.open("http://mail.google.com/mail/h/")
>>> br.title()
'Gmail - Inbox'
>>> html = response2.read()
```

Mechanize

reading my inbox

```
>>> soup = BeautifulSoup(html)
>>> trs = soup.findAll('tr', {'bgcolor':'#ffffff'})
>>> for td in trs:
...     print td.b
...
<b>madhukargunjan</b>
<b>NIHI</b>
<b>Chamindra</b>
<b>KR2 blr</b>
<b>Imran</b>
<b>Ratnadeep Debnath</b>
<b>yasir</b>
<b>Glenn</b>
<b>Joyent, Carly Guarcello</b>
<b>Zack</b>
<b>SahanaEden</b>
...
>>>
```

Mechanize

Can scrape my gmail chats !?

http://www.flickr.com/photos/ktylerconk/1465018115/

Scrapy

a Framework for web scraping

http://www.flickr.com/photos/lwr/3191194761

Scrapy

uses XPath
navigate through elements

http://www.flickr.com/photos/apc33/15012679/

Scrapy

XPath and Scrapy

```
<h1> My important chunk of text </h1>
//h1/text()

<div id=”description”> Foo bar beep &gt; 42 </div>
//div[@id='description']

<div id='report'> <p>Text1</p> <p>Text2</p> </div>
//div[@id='report']/p[2]/text()

// select from the document, wherever they are
/  select from root node
@  select attributes
```

scrapy.selector.XmlXPathSelector
scrapy.selector. HtmlXPathSelector

http://www.w3schools.com/xpath/default.asp

Scrapy

Interactive Scraping Shell!

```
$ scrapy-ctl.py shell http://my/target/website
```

http://www.flickr.com/photos/span112/2650364158/

Scrapy

Step 1

define a model to store Items

http://www.flickr.com/photos/bunchofpants/106465356

Scrapy

Step II

create your Spider to extract Items

http://www.flickr.com/photos/thewhitestdogalive/4187373600

Scrapy

Step III

write a `Pipeline` to store them

Scrapy

Lets write a Scrapy spider

SpojBackup

- Spoj? Programming contest website
- 4021039 code submissions
- 81386 users
- web scraping == backup tool
- uses Mechanize
- http://github.com/ideamonk/spojbackup

SpojBackup

- https://www.spoj.pl/status/ideamonk/signedlist/

```
**********************************************************************************
*      Signed document issued by Sphere Online Judge  (http://www.spoj.pl)       *
**********************************************************************************

----- BEGIN OF DATA -----
Submissions by [ideamonk] in contest [SPOJ].  DoI: 2010-09-10 12:18:26.
/-------------------------------------------------------------------------------\
|   ID    |         DATE         | PROBLEM |  RESULT   | TIME |  MEM   | LNG |
|---------|----------------------|---------|-----------|------|--------|-----|
| 3198919 | 2010-01-22 01:00:45  |   BOXES |        AC | 0.07 |   2544 | C++ |
| 3023085 | 2009-11-29 14:54:36  |   ACODE |        AC | 0.46 |   5832 | PYT |
| 3023081 | 2009-11-29 14:54:15  |   ACODE |        WA | 0.68 |   5844 | PYT |
| 3023078 | 2009-11-29 14:53:38  |   ACODE |        AC | 0.47 |   5836 | PYT |
| 3023066 | 2009-11-29 14:50:22  |   ACODE |        AC | 0.37 |   5560 | PYT |
| 3023061 | 2009-11-29 14:50:00  |   ACODE |        CE | 0.00 |      0 | PYT |
| 3023030 | 2009-11-29 14:46:01  |   ACODE |        AC | 0.54 |   5556 | PYT |
| 3022963 | 2009-11-29 14:36:16  |   ACODE |        RE | 0.14 |   4124 | PYT |
```

- https://www.spoj.pl/files/src/save/3198919
- Code walktrough

Conclusions

http://www.flickr.com/photos/bunchofpants/106465356

Conclusions

Python - large set of tools for scraping

Choose them wisely

Do not steal

http://www.flickr.com/photos/bunchofpants/106465356

Conclusions

Python - large set of tools for scraping

Choose them wisely

Do not steal

License and Site Access

IMDb grants you a limited license to access and make personal use of this site and not to download (other than page caching) or modify it, or any portion of it, except with express written consent of IMDb. This site or any portion of this site may not be reproduced, duplicated, copied, sold, resold, visited, or otherwise exploited for any commercial purpose without express written consent of IMDb. This license does not include any resale or commercial use of this site or its contents or any derivative use of this site or its contents.

Robots and Screen Scraping: You may not use data mining, robots, screen scraping, or similar data gathering and extraction tools on this site, except with our express written consent as noted below.

Framing: You may not frame or utilize framing techniques to enclose any trademark, logo, or other proprietary information (including images, text, page layout, or form) of IMDb without express written consent.

http://www.flickr.com/photos/bunchofpants/106465356

Conclusions

Python - large set of tools for scraping

Choose them wisely

Do not steal

Use the cloud - http://docs.picloud.com/advanced_examples.html

Share your scrapers - http://scraperwiki.com/

Thanks to @amatix , flickr, you, etc

@PyCon India 2010
http://in.pycon.org

Abhishek Mishra
hello@ideamonk.com

http://www.flickr.com/photos/bunchofpants/106465356