

Preparing to Scrape

Before we can start scraping, we need to figure out what we want to do. We will be using my blog for this example. Our task will be to scrape the titles and links to the articles on the front page of my blog, which can be found here: <http://www.blog.pythonlibrary.org/>. You can use Python's **urllib2** module to download the HTML that we need to parse or you can use the **requests** library. For this example, I'll be using requests.

Most websites nowadays have pretty complex HTML. Fortunately most browsers provide tools to make figuring out where website elements are quite trivial. For example, if you open my blog in chrome, you can right click on any of the article titles and click the **Inspect** menu option (see below):



Once you've clicked that, you will see a sidebar appear that highlights the tag that contains the title. It looks like this:

Menu

339.83 x 30

Python 101: Redirecting stdout

June 16, 2016

Cross-Platform, Python

Python, wxPython

Mike

Redirecting stdout to something most developers will need to do at some point or other. It can be useful to redirect stdout to a file or to a file-like object. I have also redirected stdout to a text control in some of my desktop GUI projects. In this article we will look at the following:

- Redirecting stdout to a file (simple)
- The Shell redirection method
- Redirecting stdout using a custom context manager
- Python 3's contextlib.redirect_stdout()
- Redirect stdout to a wxPython text control

Continue reading →

0 Comments

Elements

Console

Sources

Network

Timeline

Profiles

1

:

```

<div id="primary" class="content-area">
  <div id="content" class="site-content" role="main">
    <article id="post-5522" class="post-5522 post type-post status-publish format-standard hentry category-cross-platform category-python tag-python tag-wxpython">
      <header class="entry-header">
        <h1 class="entry-title">
          <a href="http://www.blog.pythonlibrary.org/2016/06/16/python-101-redirecting-stdout/" rel="bookmark">Python 101: Redirecting stdout</a>
        </h1>
        <div class="entry-meta"></div>
        <!-- .entry-meta -->
      </header>
      <!-- .entry-header -->
      <div class="entry-content"></div>
      <!-- .entry-content -->
    </article>
  </div>
</div>

```

html

body

#page

#main

#primary

#content

#post-5522

header.entry-header

h1.entry-title

a

Styles

Event Listeners

DOM Breakpoints

Properties

Filter

:hov

.cls

+

```

element.style {
}

.entry-title a {
  color: #141412;
}

a:visited {
  color: #ac0404;
}

a {
  color: #ca3e08;
  text-decoration: none;
}

* {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}

```

style.css?ver=2013-07-18:1040

style.css?ver=2013-07-18:120

style.css?ver=2013-07-18:115

style.css?ver=2013-07-18:60

margin

border

padding

auto x auto

Filter

Show all

box-sizing

border

color

cursor

display

font-family

border-...

rgb(20.

auto

inline

Disto

The Mozilla Firefox browser has Developer tools that you can enable on a per page basis that includes an Inspector you can use in much the same way as we did in Chrome. Regardless which web browser you end up using, you will quickly see that the **h1** tag is the one we need to look for. Now that we know what we want to parse, we can learn how to do so!