# Web Scraping

Steve Iannaccone • 05.02.2017

# Overview

## Objectives

- Gain a basic understanding of how the internet works
- Understand simple HTML and CSS and their differences
- Use Python to retrieve information from websites

# The Interwebs

## In Short:

*“...the Internet is not something that you just dump something on. It's not a big truck. It's a series of tubes.”*

*-Sen. Ted Stevens on Net Neutrality*

# The Internet vs World Wide Web

## They’re Not the Same Thing

When you type a domain name into your browser it first has to look-up the associated IP address for where the server lives and the communication protocol required to talk to it.

The WWW is the interconnection of these servers and resources.

The internet is the infrastructure that enables this interaction *(the tubes)*.

# Uniform Resource Locators

Anatomy of a URL:

# HTML and CSS

# HTML and CSS

## The Mantra:

Thou shalt keep thine content separate from thine style...

*But why?*

# HTML and CSS

## Redesign Not Rewrite:

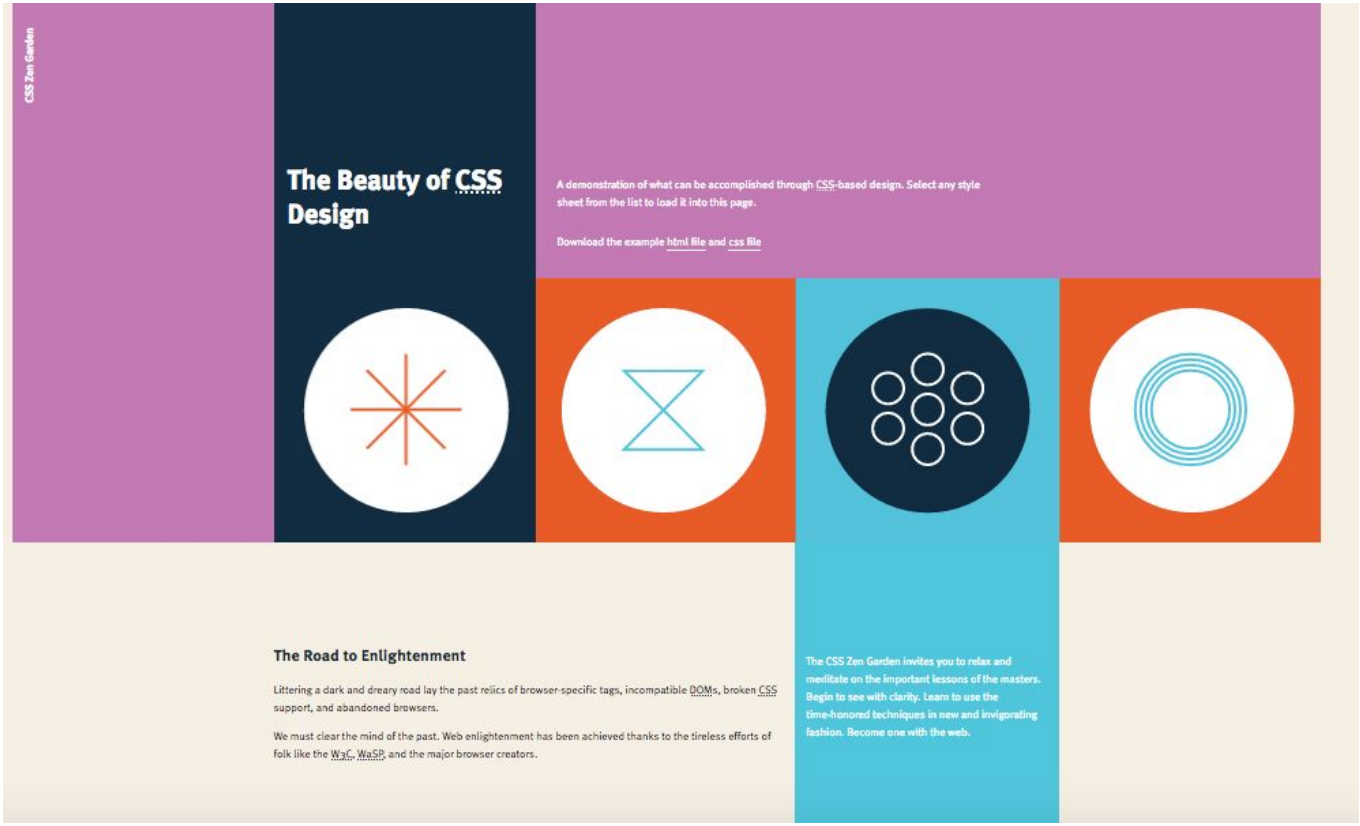
Take a look at some examples from Dave Shea's CSS Zen Garden. Radical site redesigns are possible with only modifying the style, not rewriting any of the content.

```
<body id="css-zen-garden">
<div class="page-wrapper">

    <section class="intro" id="zen-intro">
        <header role="banner">
            <h1>CSS Zen Garden</h1>
            <h2>The Beauty of <abbr title="Cascading Style Sheets">CSS</abbr> Design</h2>
        </header>

        <div class="summary" id="zen-summary" role="article">
            <p>A demonstration of what can be accomplished through <abbr title="Cascading Style Sheets">CSS</
            <p>Download the example <a href="/examples/index" title="This page's source HTML code, not to be
        </div>

        <div class="preamble" id="zen-preamble" role="article">
            <h3>The Road to Enlightenment</h3>
            <p>Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible <a
            <p>We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless
            <p>The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. 
        </div>
    </section>
```


CSS Zen Garden
The Beauty of CSS Design
A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.
Download the example html file and css file
The Road to Enlightenment
Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.
We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP, and the major browser creators.
The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

# CSS Zen Garden

*The Beauty of CSS Design*

*A demonstration of what can be accomplished through CSS based design. Select any style sheet from the list to load it into this page.*
*Download the example html file and css file*

## The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP, and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

## So What is This About?

There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The HTML remains the same, the only thing that has changed is the external CSS file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. Designers and coders alike have contributed to the beauty of the web; we can always push it further.

## Participation

Strong visual design has always been our focus. You are modifying this page, so strong CSS skills are necessary too, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and tips on working with CSS.


C.S.S.
ZEN GARDEN
A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.
DOWNLOAD THE EXAMPLE HTML FILE AND CSS FILE
THE ROAD TO ENLIGHTENMENT
Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.
We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP, and the major browser creators.
The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the
SELECT A DESIGN
Mid Century Modern
by Andrew Lohman
Garments
by Dan Mall
Steel
by Steffen Knoeller
Apothecary
ICTINEO-MONTURIOL

# Web Scraping

# First Things First

## This is a Legal Gray Area

At best.

While enforcement is difficult, don't take that risk. At the very least check the Terms of Service for the webpage you are trying to access. If what you are trying to do is not permitted, try other sources.

**Legal advice comes as-is and makes no claims that you will not get in any trouble whatsoever. If you do not understand the inherent risk involved with potentially stealing others work, then we cannot help you. Seriously, we are all adults, and we all live in an outrageously litigious society. So, deal with it accordingly.*

# But What If...

We aren’t IP lawyers or judges.

# Web Scraping

## Tags / Classes / IDs

All those HTML tags, style classes and ids make for great hooks to find just the content you want from a webpage.

```
<div class="summary" id="zen-summary" role="article">...

    <p>A demonstration of what can be accomplished...
```

# Web Scraping

## Tags / Classes / IDs

Everybody go to:

https://flukeout.github.io/

Practice using CSS selectors, how many levels can you get through in the next 5 minutes?

# Web Scraping

## Workflow

In your browser:

- Get the URL of the page with the info you need
- Use inspector to find tag / id you can use to extract info

In Python:

- Use **`requests`** to get html content as a string
- Use **`BeautifulSoup`** to parse the string and pull out info using your tag / id

# Web Scraping

```
import requests

from bs4 import BeautifulSoup


url = "http://www.foxsports.com/college-basketball/teams"

req = requests.get(url)

soup = BeautifulSoup(req.text, 'html.parser')

logos = soup.find_all('img', attrs = {'class': "wisfb_logoImage"})
```

# Web Scraping

## Pandas

Yes, pandas can read and parse HTML. So, if you only need data from tables from a single page this is probably the easiest approach.

```
tables = pd.read_html("http://www.website-in-question.com")
```

This returns a list of DataFrames where each DF is made from a table in the source page. But, depending on formatting quirks, this might be significantly worse than bs4.

# APIs

# APIs

## Application Programming Interface

Webmasters don't want you pinging their websites every 10ms to scrape out some information.

Often times they build APIs to help you collect just the info you need *(and so that you don't crush their servers)*.

It's Win-Win because you use less bandwidth on their server, and you get the data you need in a consistently formatted fashion.

# APIs

## API Keys

You will often need to register to gain access to an site's API.

Sometimes to to track usage, often to limit the number of calls, and occasionally as a business model where you pay per call or by amount of transferred data.

This is essentially your login credentials, NEVER publish your keys! Don't push up any script to GitHub that has your keys in plain text.

# APIs

## API Keys

There are several ways to deal with this. But here is one suggestion: Store your key as an Environment Variable.

In your bash profile (*~/.bash-profile*, *~/.bashrc*, or *~/.profile*) set the following:

```
export API_KEY="my_api_key"
```

# APIs

## API Keys

Now we can use the variable `API_KEY` in our python scripts using the `os` package without fear of publishing our private key when we push our repos.

```
import os

my_key = os.environ['MY_KEY']
```

# APIs

## They’re All Different

You’ll have to dig into the docs to figure out how to use the API you need. But, if you want an intro on the broad strokes of how to interact with an API in Python this blog does a good job:

https://www.dataquest.io/blog/python-api-tutorial/

# Recap

1. Why are HTML and CSS separated?
2. What hooks can we use to select data from a website?
3. How can we protect our API keys when pushing files up to public repos?