

# Importing flat files from the web

INTERMEDIATE IMPORTING DATA IN PYTHON



**Hugo Bowne-Anderson**  
Data Scientist at DataCamp

# You're already great at importing!

- Flat files such as .txt and .csv
- Pickled files, Excel spreadsheets, and many others!
- Data from relational databases
- You can do all these locally
- What if your data is online?

# Can you import web data?



## Wine Quality Data Set

Download: [Data Folder](#), [Data Set Description](#)

**Abstract:** Two datasets are included, related to red and white vinho verde wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests (see [Cortez et al., 2009], [Web Link]).



Data Set Characteristics:	Multivariate	Number of Instances:	4898	Area:	Business
Attribute Characteristics:	Real	Number of Attributes:	12	Date Donated	2009-10-07
Associated Tasks:	Classification, Regression	Missing Values?	N/A	Number of Web Hits:	349131

- You can: go to URL and click to download files
- BUT: not reproducible, not scalable

# You'll learn how to...

- Import and locally save datasets from the web
- Load datasets into pandas DataFrames
- Make HTTP requests (GET requests)
- Scrape web data such as HTML
- Parse HTML into useful data (BeautifulSoup)
- Use the urllib and requests packages

# The urllib package

- Provides interface for fetching data across the web
- `urlopen()` - accepts URLs instead of file names

# How to automate file download in Python

```
from urllib.request import urlopen
url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv'
urlopen(url, 'winequality-white.csv')
```

```
('winequality-white.csv', <http.client.HTTPMessage at 0x103cf1128>)
```

# Let's practice!

INTERMEDIATE IMPORTING DATA IN PYTHON

# HTTP requests to import files from the web

INTERMEDIATE IMPORTING DATA IN PYTHON



**Hugo Bowne-Anderson**  
Data Scientist at DataCamp



# URL

- Uniform/Universal Resource Locator
- References to web resources
- Focus: web addresses
- Ingredients:
  - Protocol identifier - http:
  - Resource name - datacamp.com
- These specify web addresses uniquely

# HTTP

- HyperText Transfer Protocol
- Foundation of data communication for the web
- HTTPS - more secure form of HTTP
- Going to a website = sending HTTP request
  - GET request
- `urlretrieve()` performs a GET request
- HTML - HyperText Markup Language

# GET requests using urllib

```
from urllib.request import urlopen, Request
url = "https://www.wikipedia.org/"
request = Request(url)
response = urlopen(request)
html = response.read()
response.close()
```

# GET requests using requests



- Used by “her Majesty’s Government, Amazon, Google, Twilio, NPR, Obama for America, Twitter, Sony, and Federal U.S. Institutions that prefer to be unnamed”

# GET requests using requests

- One of the most downloaded Python packages

```
import requests
url = "https://www.wikipedia.org/"
r = requests.get(url)
text = r.text
```

# Let's practice!

INTERMEDIATE IMPORTING DATA IN PYTHON

# Scraping the web in Python

INTERMEDIATE IMPORTING DATA IN PYTHON



**Hugo Bowne-Anderson**  
Data Scientist at DataCamp

# HTML

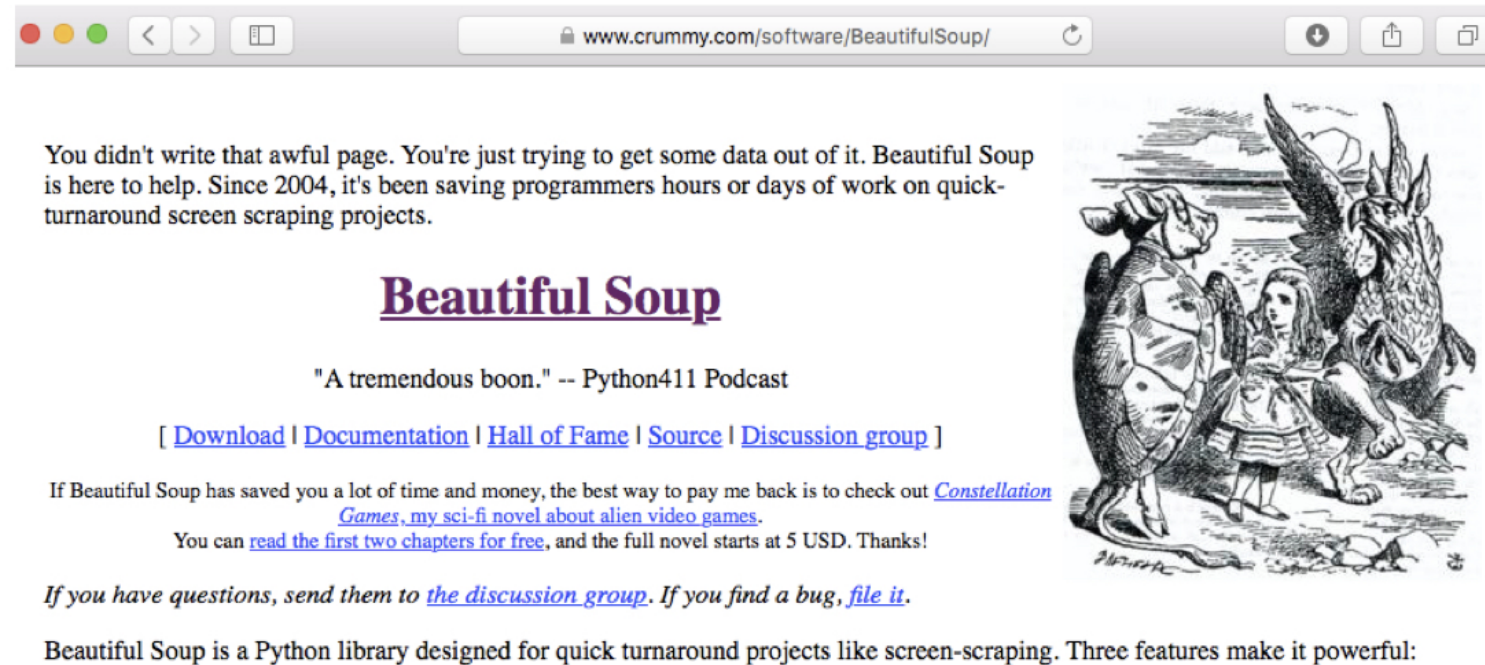
- Mix of unstructured and structured data
- Structured data:
  - Has pre-defined data model, or
  - Organized in a defined manner
- Unstructured data: neither of these properties

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!-- SEO -->
5     <meta charset="utf-8" />
6 <title>DataCamp: The Easy Way To Learn R & Data Science Online</title>
```



# BeautifulSoup

- Parse and extract structured data from HTML



- Make tag soup beautiful and extract information

# BeautifulSoup

```
from bs4 import BeautifulSoup
import requests
url = 'https://www.crummy.com/software/BeautifulSoup/'
r = requests.get(url)
html_doc = r.text
soup = BeautifulSoup(html_doc)
```

# Prettified Soup

```
print(soup.prettify())
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-html40/transitional.dtd">
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
    <title>
      Beautiful Soup: We called him Tortoise because he taught us.
    </title>
    <link href="mailto:leonardr@segfault.org" rev="made"/>
    <link href="/nb/themes/Default/nb.css" rel="stylesheet" type="text/css"/>
    <meta content="Beautiful Soup: a library designed for screen-scraping HTML and XML." name="Description"/>
    <meta content="Markov Approximation 1.4 (module: leonardr)" name="generator"/>
    <meta content="Leonard Richardson" name="author"/>
  </head>
  <body alink="red" bgcolor="white" link="blue" text="black" vlink="660066">
    
    <br/>
    <p>
```

# Exploring BeautifulSoup

- Many methods such as:

```
print(soup.title)
```

```
<title>Beautiful Soup: We called him Tortoise because he taught us.</title>
```

```
print(soup.get_text())
```

```
Beautiful Soup: We called him Tortoise because he taught us.  
You didn't write that awful page. You're just trying to  
get some data out of it. BeautifulSoup is here to  
help. Since 2004, it's been saving programmers hours or  
days of work on quick-turnaround screen scraping  
projects.
```

# Exploring BeautifulSoup

- `find_all()`

```
for link in soup.find_all('a'):  
    print(link.get('href'))
```

```
bs4/download/  
#Download  
bs4/doc/  
#HallOfFame  
https://code.launchpad.net/beautifulsoup  
https://groups.google.com/forum/?fromgroups#!forum/beautifulsoup  
http://www.candlemarkandgleam.com/shop/constellation-games/  
http://constellation.crummy.com/Constellation%20Games%20excerpt.html  
https://groups.google.com/forum/?fromgroups#!forum/beautifulsoup  
https://bugs.launchpad.net/beautifulsoup/  
http://lxml.de/  
http://code.google.com/p/html5lib/
```

# Let's practice!

INTERMEDIATE IMPORTING DATA IN PYTHON