Understanding Your Scraped Data

Ian Ozsvald
INTERIM CHIEF DATA SCIENTIST

@ianozsvald   ianozsvald.com

Overview

Chrome Developer Tools

Check div and span elements

Extract text from these elements

Add confidence with a unit test

Content vs. Styling

HTML

Page specific content

Structure

Text or voice

Not written for scraping

CSS

Shared styling

Selectors and Declarations

Divisions and Spans

Cascading rules

HTML

HEAD

BODY

DOM

Demo

Investigate our page’s structure with Chrome Developer Tools

Select elements

Make changes live

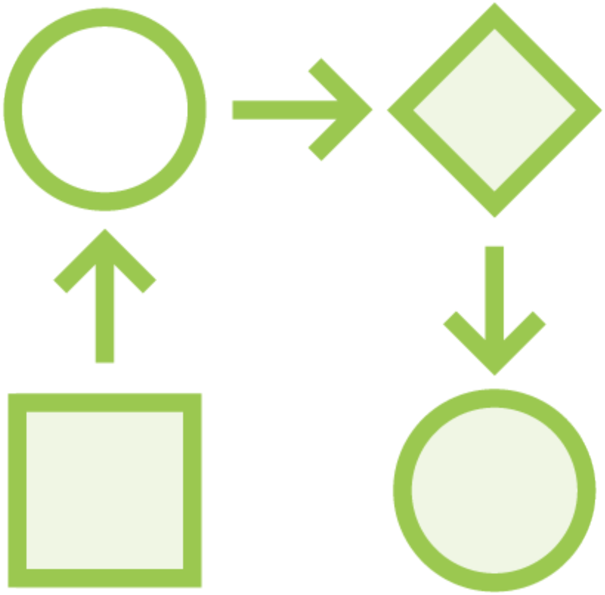
Debug errors in the page

What’s your strategy?

What do you need?

Find relevant tags and selectors

Note examples of the content you’ll extract



Demo

Extract information with BeautifulSoup4

Why not regular expressions?

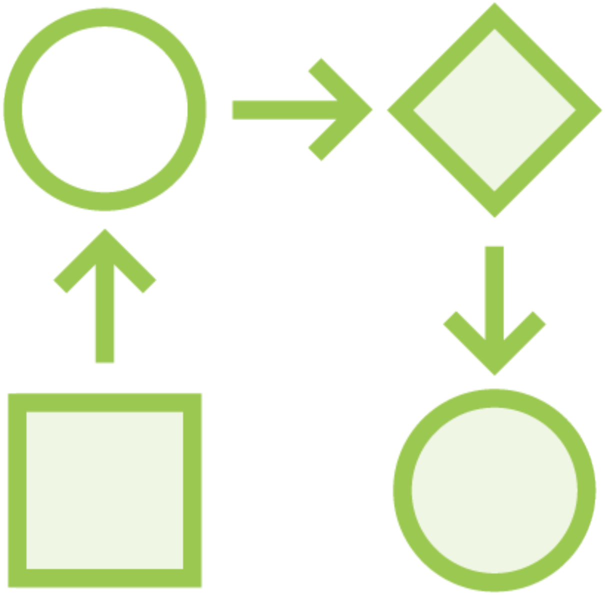
- They’re brittle!
- Very hard to read
- Very hard to support
- You end up recreating a parser!

Selecting your content

BeautifulSoup elements

Find functions

CSS Selectors



Demo

PyCharm is a modern IDE

Great debugging support

We’ll run and debug our code in it

Demo

Tests let us write robust code

We can explore edge cases

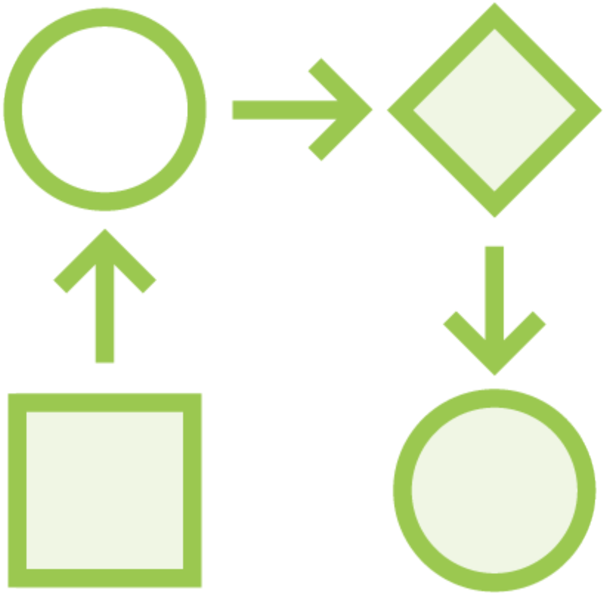
Adapt to changing situations with confidence

Processing online limits growth

Processing off-line is scalable

Adapt to changes

Extract incrementally



Summary

We’ve extracted key information

Our test gives us confidence

Ready to export information

Next we’ll visually explore our data