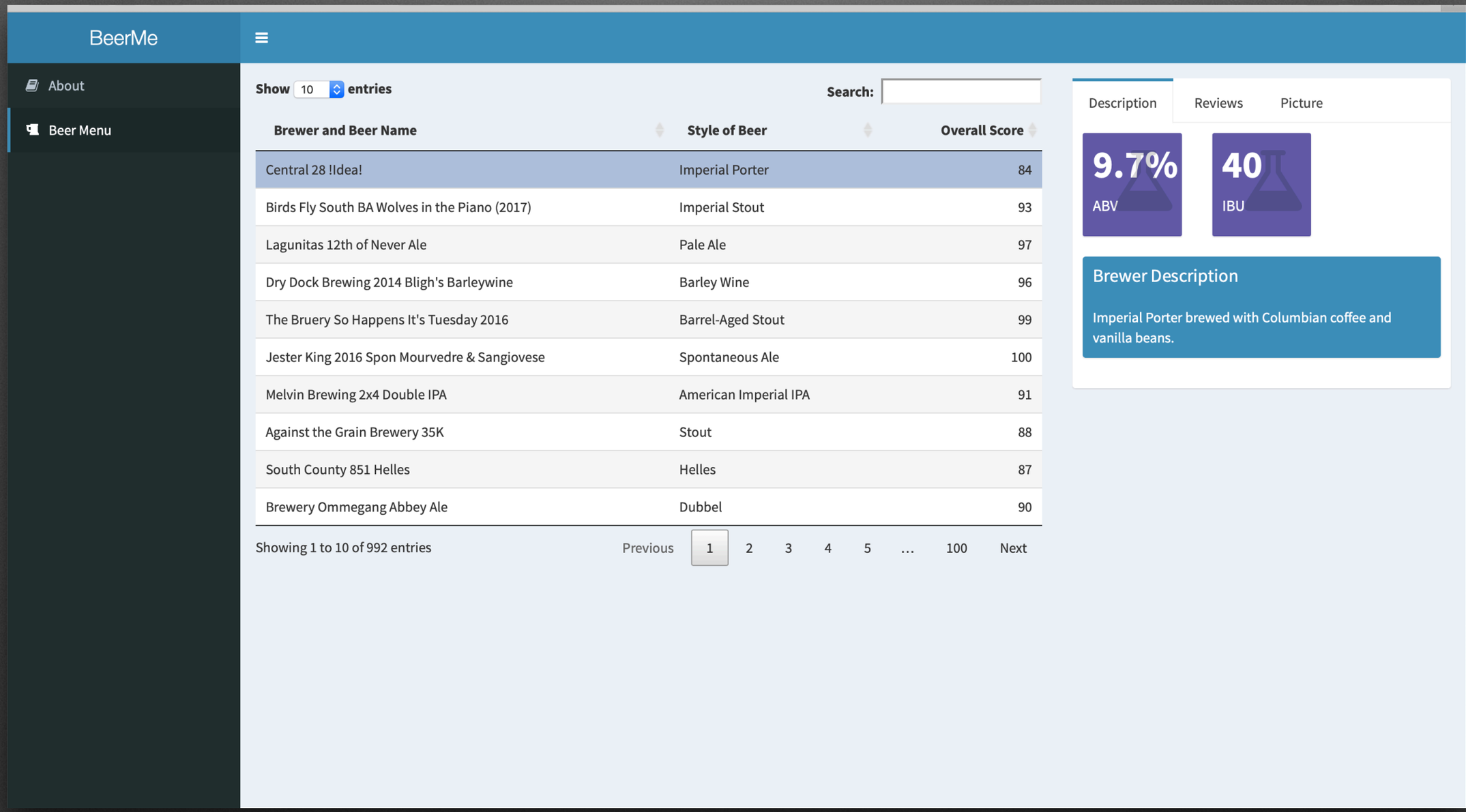
```
xcept Exception as e
    print(e)
    csv_file.close()
    driver.close()
    break
```

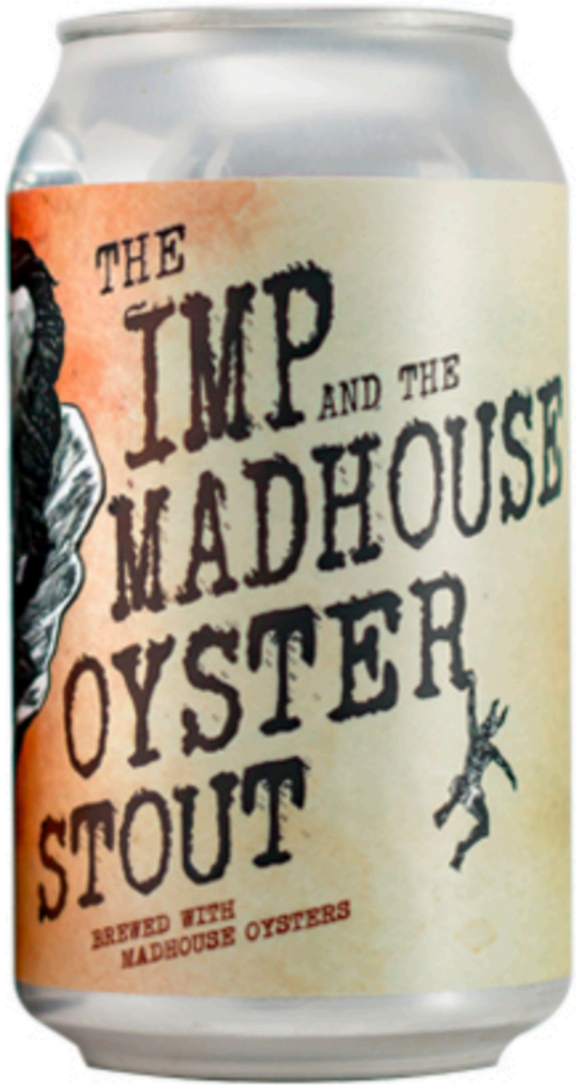
BeerMe: A Comedy of Errors

Devon Blumenthal



The App

https://dmblumen.shinyapps.io/BeerMe2/



The Imp and the Madhouse Oyster Stout

RavenBeer

Experimental Beer | Year-Round | Maryland | United States

92

Judges Ratings

How We Score | Read Full Review

1 Review

Aroma: 22 / 24

Flavor: 36 / 40

Appearance: 6 / 6

Mouthfeel: 10 / 10

Overall Impression: 18 / 20

The First Website

https://beerconnoisseur.com/search-beer

```
    start_urls = ['https://beerconnoisseur.com/search-beer']

    def parse(self,response):
        result_urls = ['https://beerconnoisseur.com/search-beer?keys=&field_beer_style_tid=All&field_state_value=All&field_country_value=United%20States&so

        for url in result_urls:
            yield Request(url=url, callback=self.parse_result_page)

    def parse_result_page(self,response):
        condensed_urls = response.xpath('//div[@class ="views-field views-field-title"]//a/@href').extract()
        full_urls = []
        for url in condensed_urls:
            full_urls.append('https://beerconnoisseur.com' + url)

        for url in full_urls:
            yield Request(url=url, callback=self.parse_detail_page)

    def parse_detail_page(self,response):
        name = response.xpath('//h1/text()').extract()
        image = response.xpath('//div[@class="field field-name-field-beer-image field-type-image field-label-hidden"]//img/@src').extract()
        brewer = response.xpath('//div[@class="field field-name-field-brewery field-type-entityreference field-label-hidden"]//a/text()').extract()
        beer_type = response.xpath('//div[@class="field field-name-field-beer-style field-type-taxonomy-term-reference field-label-hidden"]//a/text()').extr
        release = response.xpath('//div[@class="field field-name-field-availability field-type-taxonomy-term-reference field-label-hidden"]//a/text()').extr

        ###State needs the second element pulled
        state = response.xpath('//div[@class="field field-name-field-state field-type-list-text field-label-hidden"]//div/text()').extract()[1]

        description = response.xpath('//div[@class="field field-name-body field-type-text-with-summary field-label-above"]//p/text()').extract()

        ### abv needs the % sign removed
        abv_p = response.xpath('//div[@class="field field-name-field-abv field-type-text field-label-inline clearfix"]//div[@class ="field-item even"]/text(
        abv = list(map(lambda s: s.replace('%',""),abv_p))

        ibu = response.xpath('//div[@class="field field-name-field-ibus field-type-text field-label-inline clearfix"]//div[@class ="field-item even"]/text()
        serv_temp = response.xpath('//div[@class="field field-name-field-served-at field-type-text field-label-inline clearfix"]//div[@class ="field-item ev
        hops = response.xpath('//div[@class="field field-name-field-hops field-type-text field-label-inline clearfix"]//div[@class ="field-item even"]/text(
        malts = response.xpath('//div[@class="field field-name-field-malts field-type-text field-label-inline clearfix"]//div[@class ="field-item even"]/te

        ### Aroma is going to require Regex
        aroma_p = response.xpath('//div[@class="views-field views-field-field-judge-aroma"]//div[@class ="field-content"]/text()').extract_first()
        aroma = re.findall('(\d+) /',aroma_p)

```

The Scrapy

100 Lines of Code. Took 2 minutes to scrape 2000 websites.

Wiley Roots Galaxy DH Funk Yo Couch

Print Shelf Talker

How we review

What the brewers say

"Tart farmhouse saison fermented with wild captured Brettanomyces, dry hopped with more than 1 pound per barrel of Australian Galaxy hops. Winner of a gold medal at the 2017 GABF."

What our panel thought

Aroma: "Very fruity hops and ester aroma with passion fruit, peach, pear, and apple. Behind that's there's tangerine, lemon, and a touch of herbal. Spicy phenols, light Brett funk. Slight leather."

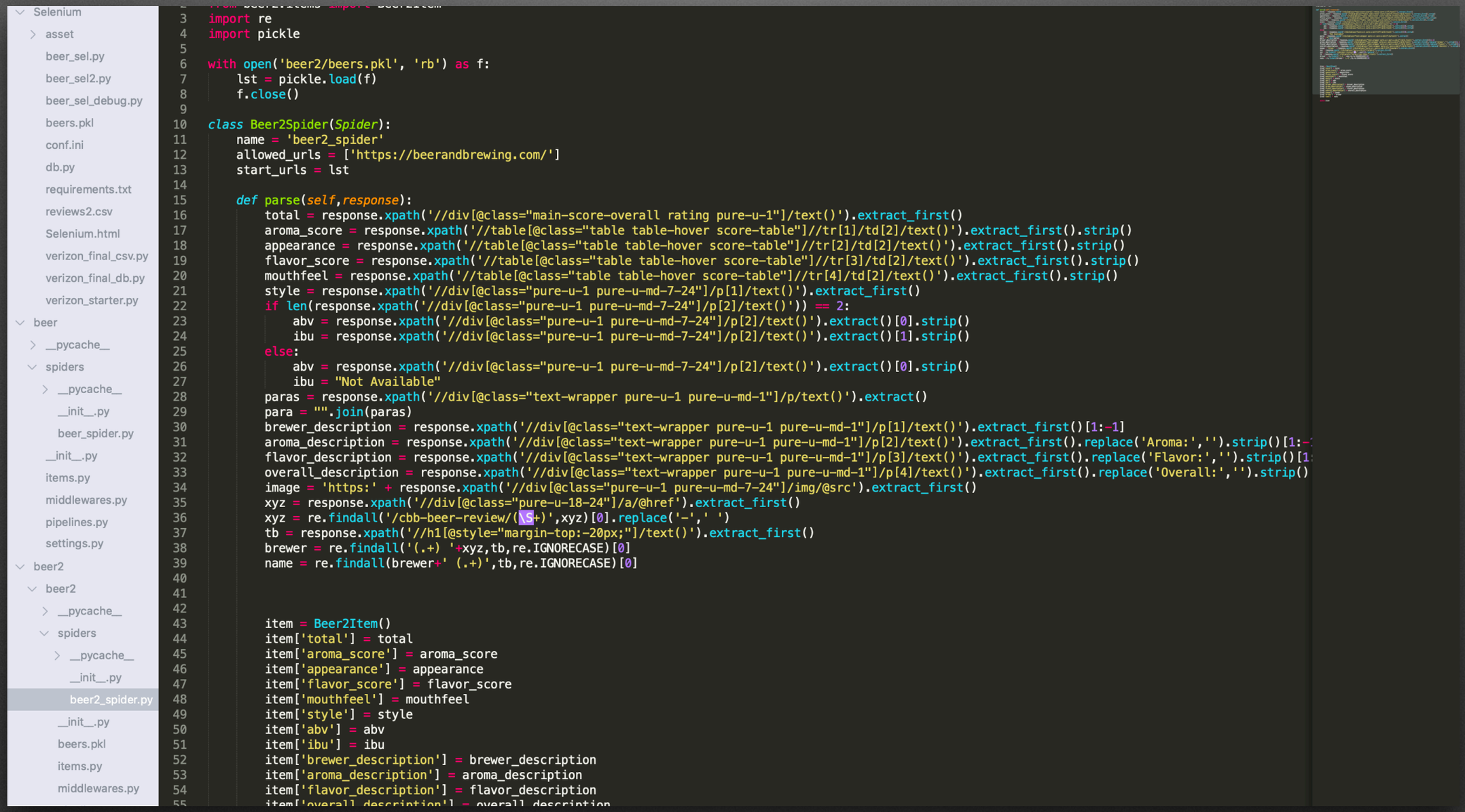
Flavor: "Very juicy hops flavor with a mix of tropical and citrus. Moderately low sweetness with moderately high sour that fades into a dry finish. The body supports the beer with a malt character that seems a bit toothier than others in the category. Light Brett funk and tart. Lingering aftertaste of sweet fruit balanced with light tartness. Acidity is masterful placed to accentuate the fruit flavors from the hops."

Overall: "Almost a fruit-cocktail blend with some funk, but it all finishes nice and dry and drinkable. A fantastic beer that delivers an ultimately satisfying hops character with a

Style: Mixed-Fermentation Farmhouse ale

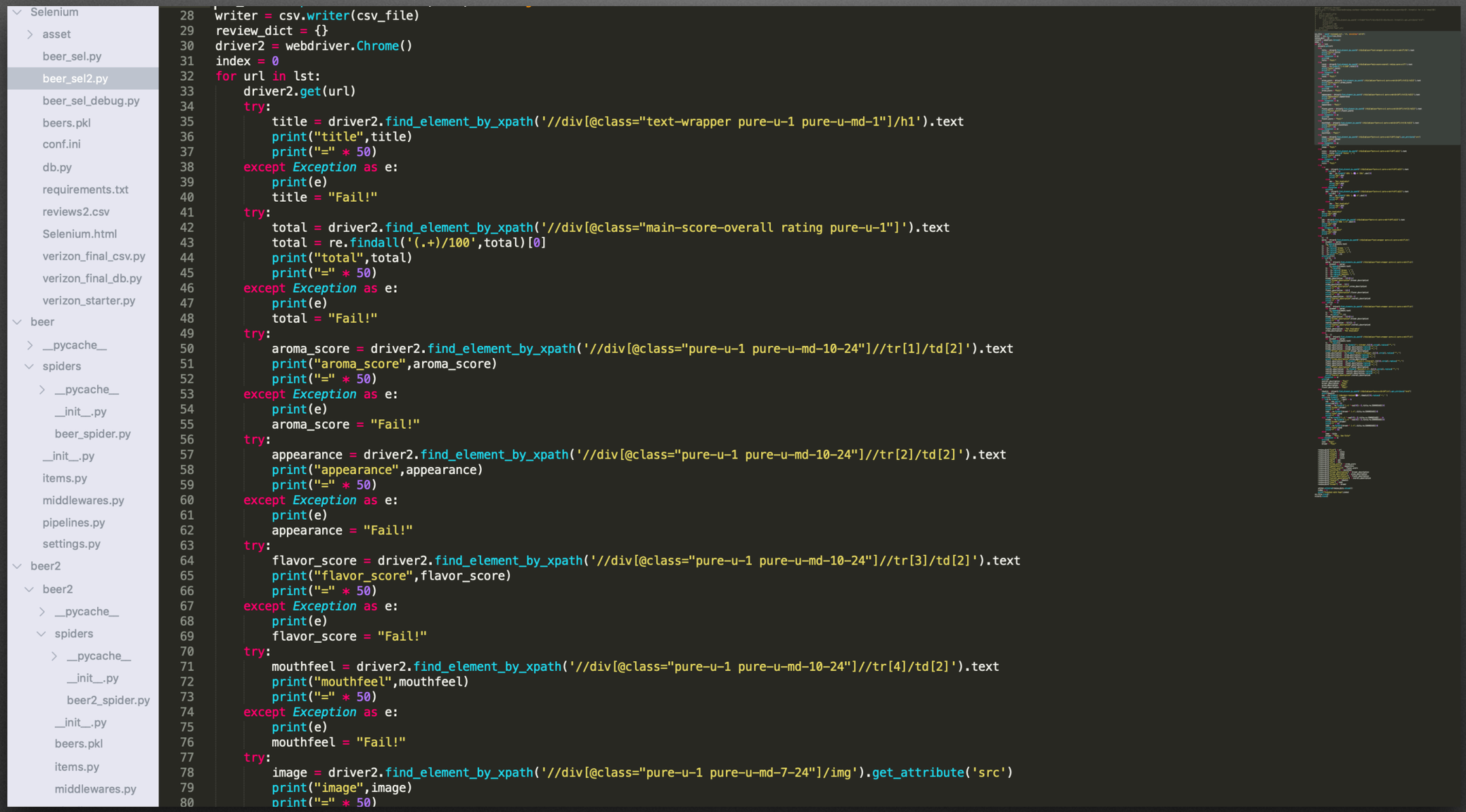
The Siren: The Second Website

https://beerandbrewing.com/beer-reviews/



The Second Scrapy

Only 60 lines of code! The problem:
Some (but not all) the pages use AJAX



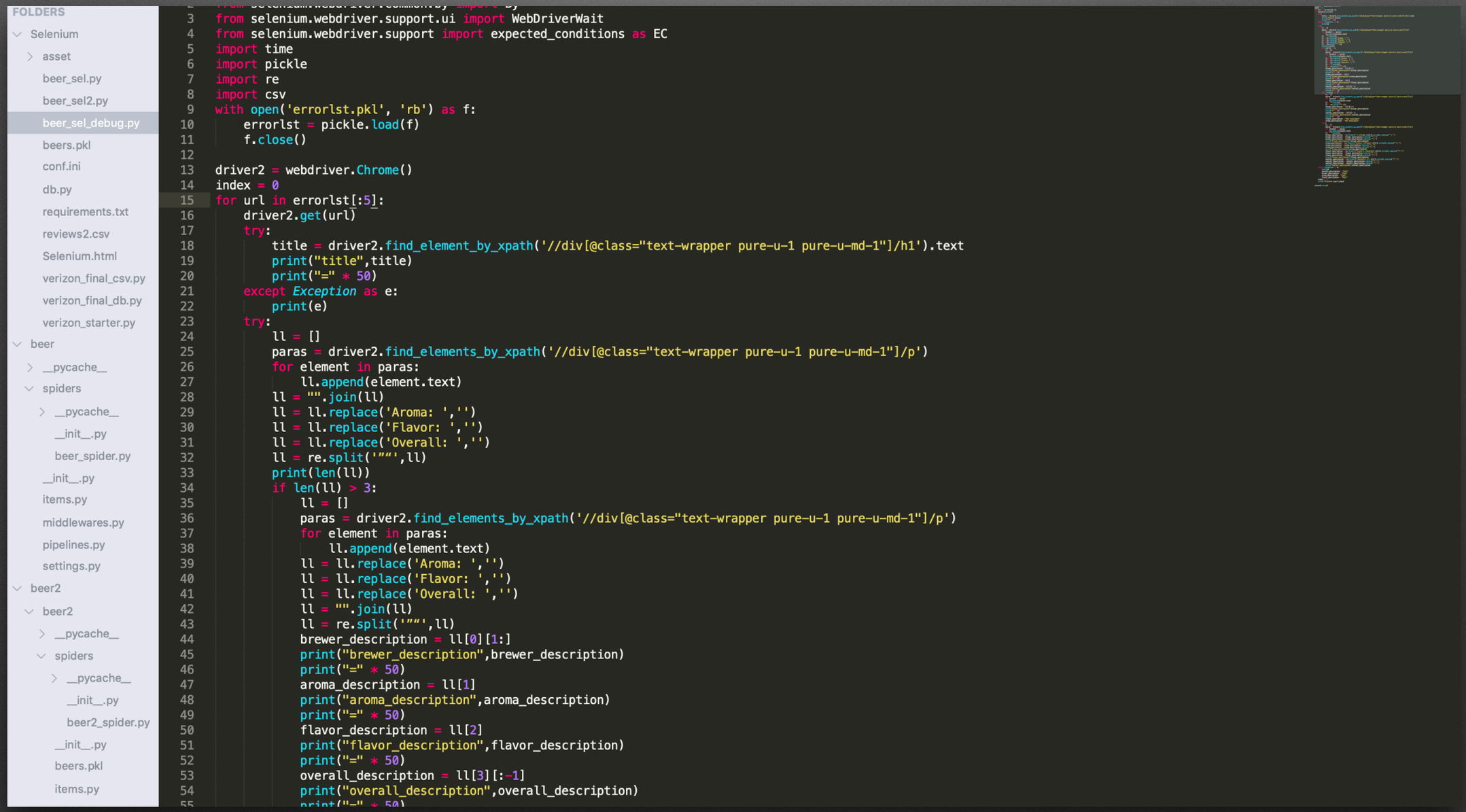
Selenium: The (Almost) Finished Product

250 lines of code for just over 1000 pages. The recurring problem:
Not all pages had the same pattern.

```
xcept Exception as e
    print(e)
    csv_file.close()
    driver.close()
    break
```

This was a huge problem:

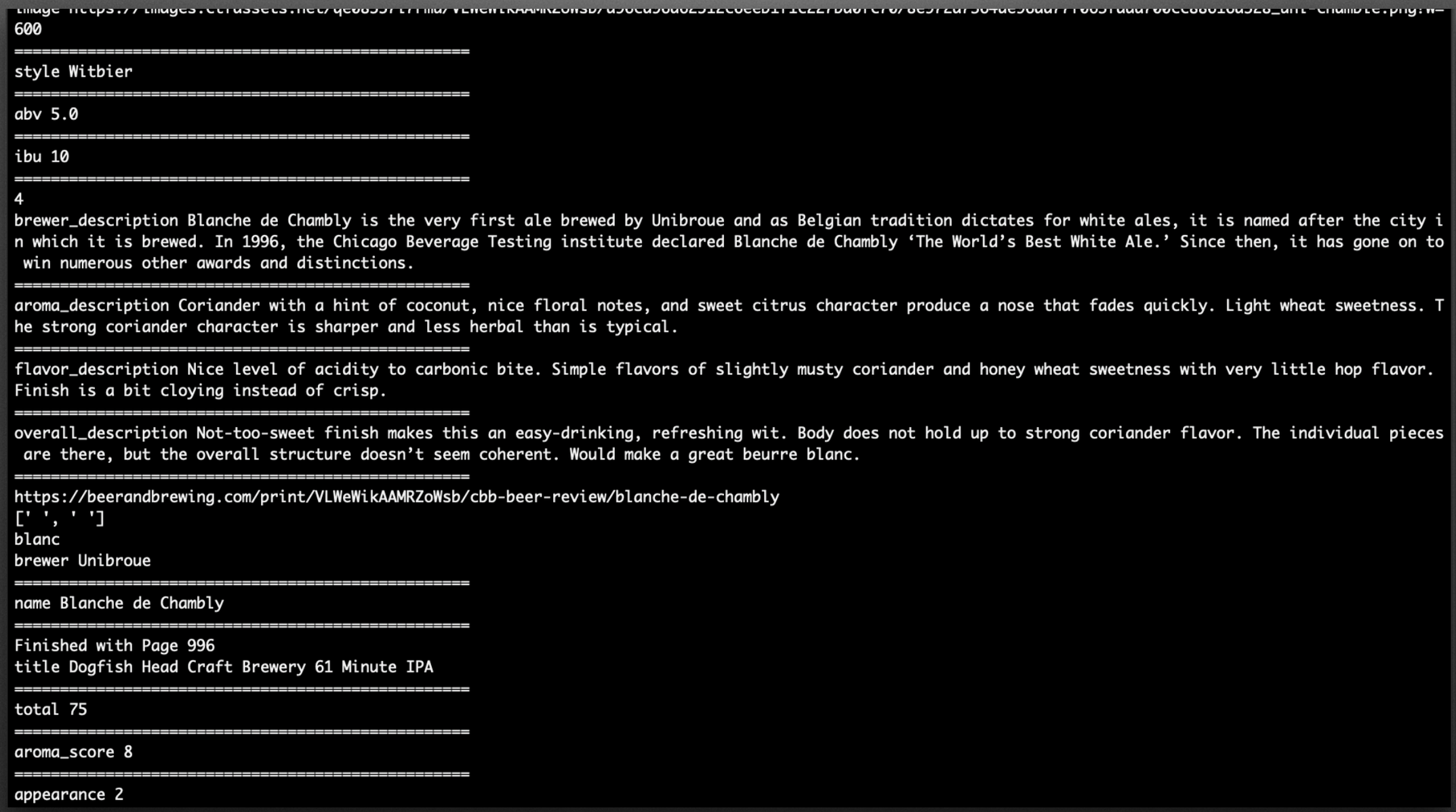
An exception would occur; Python would tell me what the error was and then just break out of the loop. It wouldn't really tell me the change in the pattern that caused the error to occur.



The Solution: A Debugging Selenium

What's happening in the background:

Upload the csv, filter the data based on "Fail!" for a given variable, create a list of the urls that had the issue, use pickle package to save the list and then load said list into the debugger. Then figure out the pattern. Copy and paste the working code for the variable and repeat.



Why this is handy:

Allows you to target the specific issues, determine where the changes in pattern are and create code to accommodate changes in patterns.

The Future

- Potentially take the two CSVs and combine them.
- Natural Language Processing With the aroma, flavor, and overall reviews
- Potentially scrape a third website, using the names of the beer to scrape links of where you can buy the beer if it interests you.