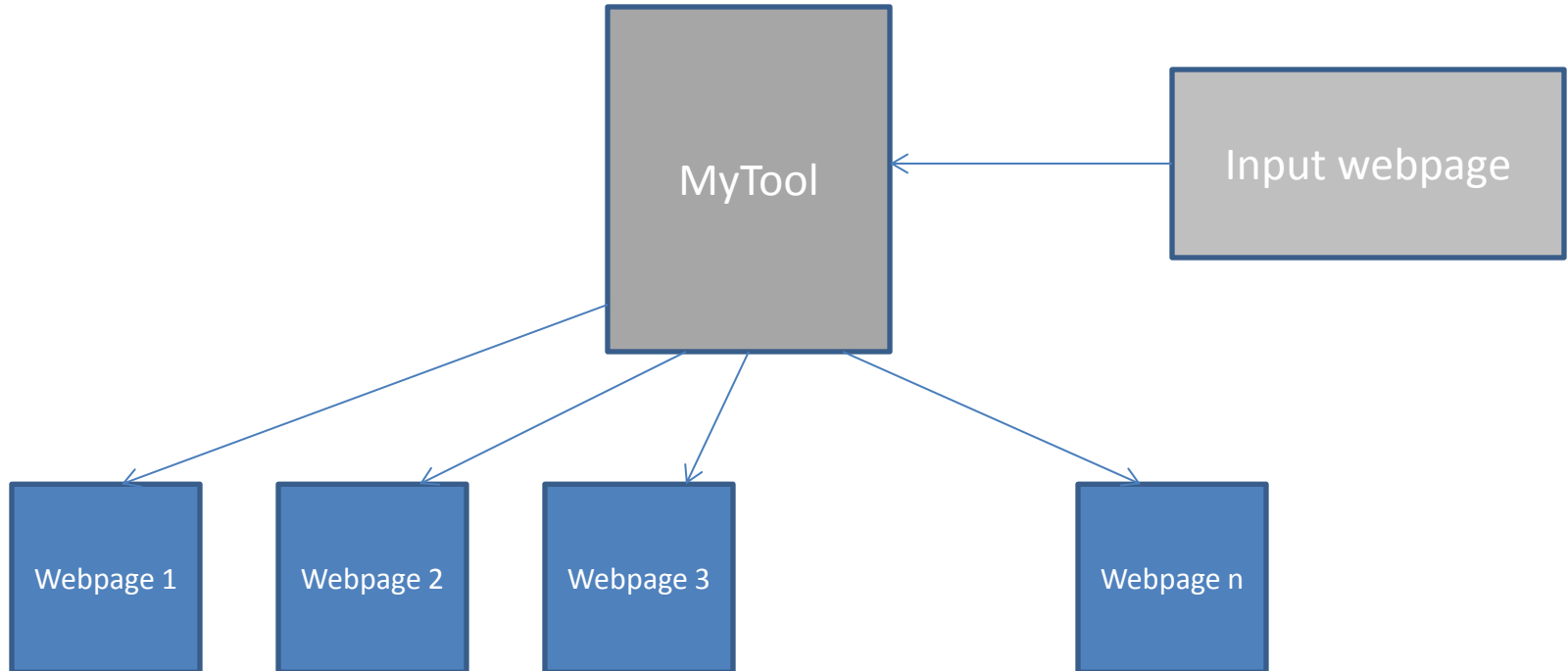


# Microbench tool 2 for testing Data Observatory

Submitted by Neha Gupta

# Requirements



# Requirements

- Take input html using any parsers like OpenWPM/ BeautifulSoup/ Python html parsers..
- Modify any random attribute of this input html and render it as a different webpage.
- Number of modifications to each webpage should be configurable.
- Modification to any webpage thus generated should be completely random

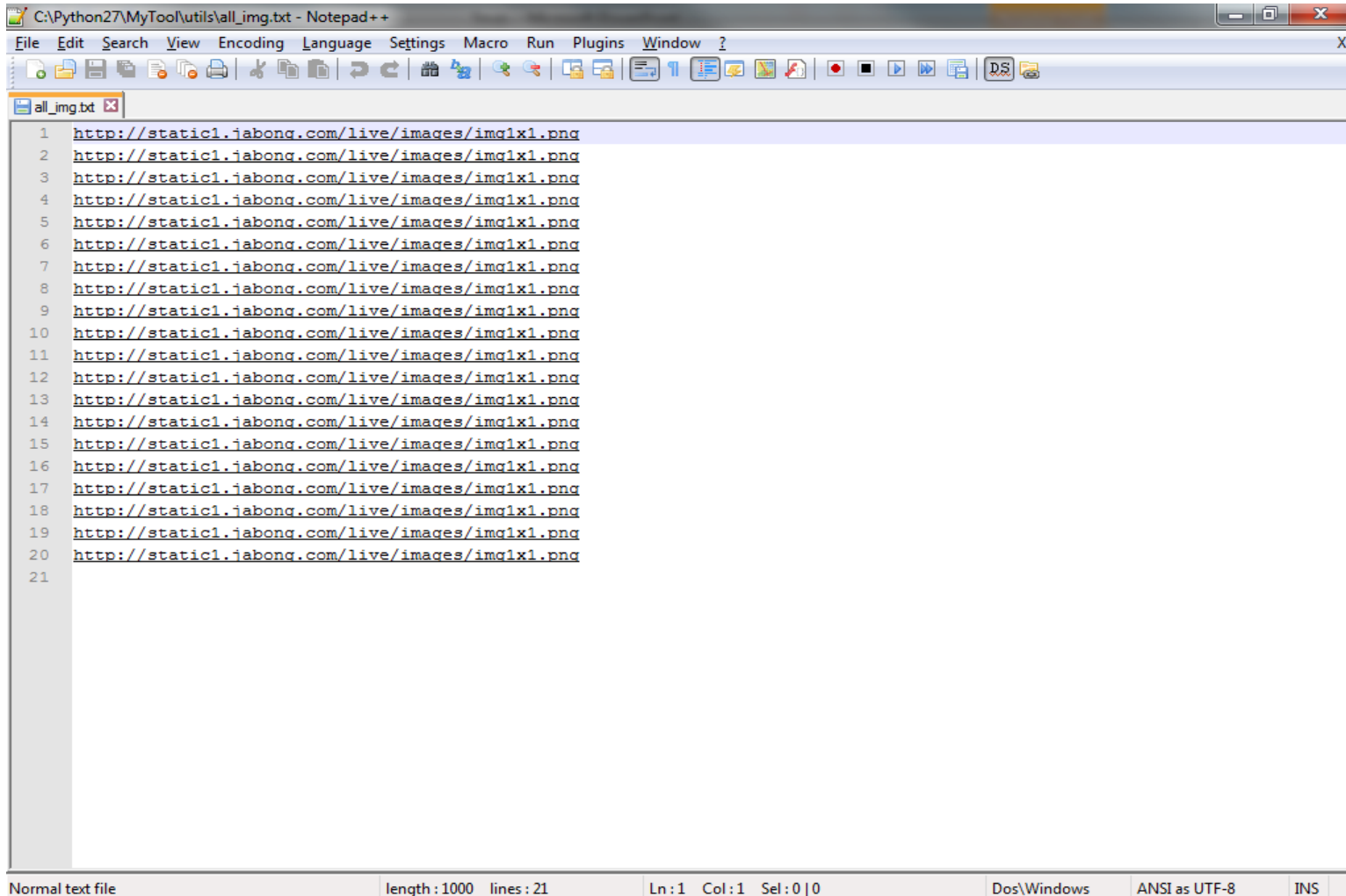
# Requirements

- The input URL should be user configurable.
- It should be possible to add and also remove these random elements
- Also, store the differences between the webpages.
- Write unit tests for the module.

# Step 1

- Build a web scraping tool that given a URL, scrapes all the links, images and text from the webpage and stores them.
- This was quite enjoyable using the python beautifulsoup library. It gives ample playground for common features.
- Other features, need to workaround or write helper functions

# Step 1 - results

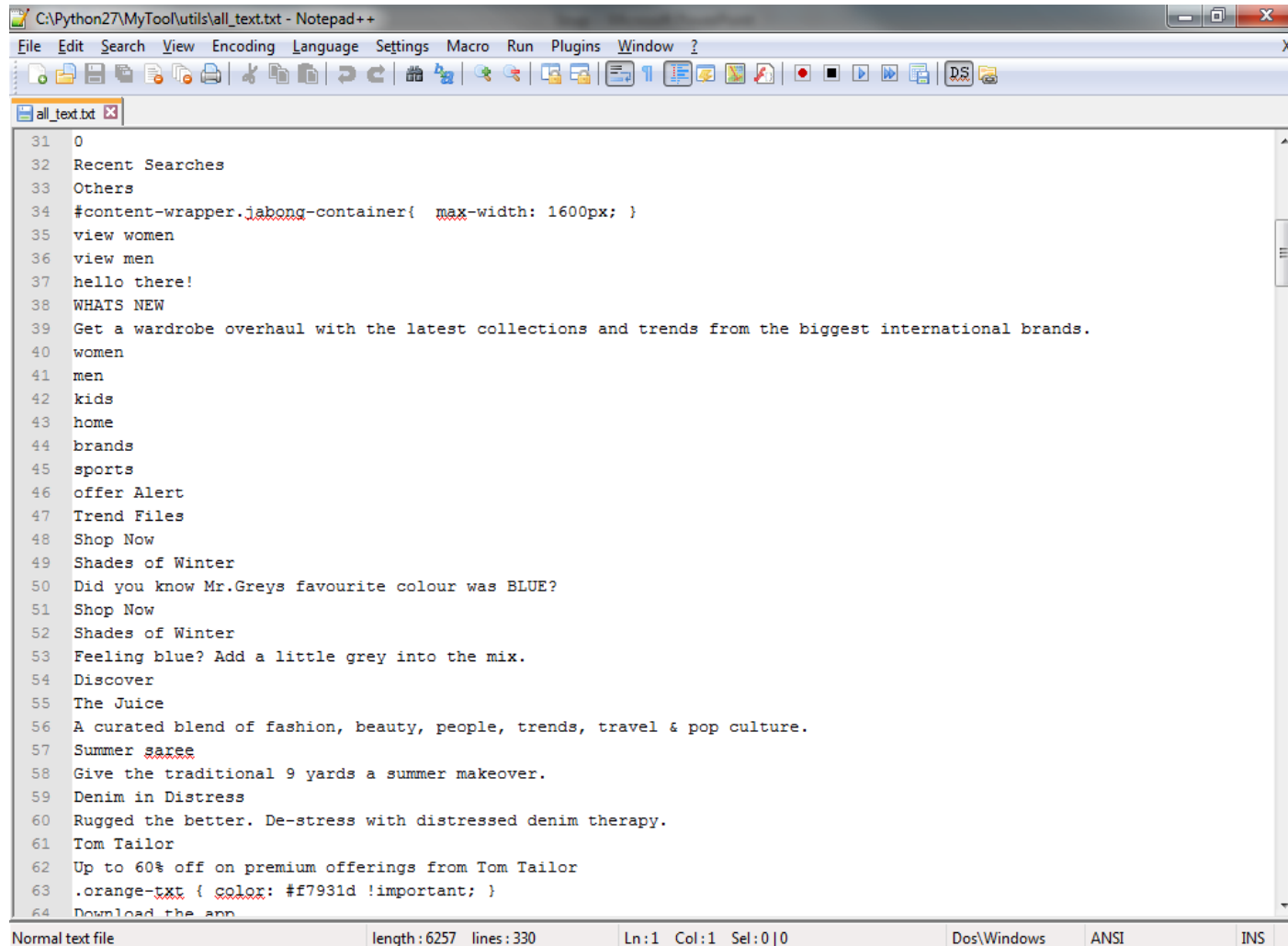


The screenshot shows a Notepad++ window titled "C:\Python27\MyTool\utils\all\_img.txt - Notepad++". The window contains a list of 20 identical URLs, each on a new line, numbered 1 through 20. The URLs are: <http://static1.jabong.com/live/images/img1x1.png>. The status bar at the bottom indicates "Normal text file", "length : 1000 lines : 21", "Ln : 1 Col : 1 Sel : 0 | 0", "Dos\Windows", "ANSI as UTF-8", and "INS".

```
1 http://static1.jabong.com/live/images/img1x1.png
2 http://static1.jabong.com/live/images/img1x1.png
3 http://static1.jabong.com/live/images/img1x1.png
4 http://static1.jabong.com/live/images/img1x1.png
5 http://static1.jabong.com/live/images/img1x1.png
6 http://static1.jabong.com/live/images/img1x1.png
7 http://static1.jabong.com/live/images/img1x1.png
8 http://static1.jabong.com/live/images/img1x1.png
9 http://static1.jabong.com/live/images/img1x1.png
10 http://static1.jabong.com/live/images/img1x1.png
11 http://static1.jabong.com/live/images/img1x1.png
12 http://static1.jabong.com/live/images/img1x1.png
13 http://static1.jabong.com/live/images/img1x1.png
14 http://static1.jabong.com/live/images/img1x1.png
15 http://static1.jabong.com/live/images/img1x1.png
16 http://static1.jabong.com/live/images/img1x1.png
17 http://static1.jabong.com/live/images/img1x1.png
18 http://static1.jabong.com/live/images/img1x1.png
19 http://static1.jabong.com/live/images/img1x1.png
20 http://static1.jabong.com/live/images/img1x1.png
21
```

Normal text file      length : 1000 lines : 21      Ln : 1 Col : 1 Sel : 0 | 0      Dos\Windows      ANSI as UTF-8      INS

# Step 1 results



The screenshot shows a Notepad++ window with the title bar 'C:\Python27\MyTool\utils\all\_text.txt - Notepad++'. The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, Plugins, and Window. The toolbar contains various icons for file operations and editing. The text area displays the following content:

```
31 0
32 Recent Searches
33 Others
34 #content-wrapper.jabong-container{ max-width: 1600px; }
35 view women
36 view men
37 hello there!
38 WHATS NEW
39 Get a wardrobe overhaul with the latest collections and trends from the biggest international brands.
40 women
41 men
42 kids
43 home
44 brands
45 sports
46 offer Alert
47 Trend Files
48 Shop Now
49 Shades of Winter
50 Did you know Mr.Greys favourite colour was BLUE?
51 Shop Now
52 Shades of Winter
53 Feeling blue? Add a little grey into the mix.
54 Discover
55 The Juice
56 A curated blend of fashion, beauty, people, trends, travel & pop culture.
57 Summer saree
58 Give the traditional 9 yards a summer makeover.
59 Denim in Distress
60 Rugged the better. De-stress with distressed denim therapy.
61 Tom Tailor
62 Up to 60% off on premium offerings from Tom Tailor
63 .orange-txt { color: #f7931d !important; }
64 Download the app
```

The status bar at the bottom indicates 'Normal text file', 'length: 6257 lines: 330', 'Ln: 1 Col: 1 Sel: 0 | 0', 'Dos\Windows', 'ANSI', and 'INS'.

# Step 2

- Generate random mutations of the webpage by
  - Adding new text elements in the html structure
  - Adding new images
  - Removing text
  - Removing image
- $O(2^n)$  possible permutations with known data
- Infinite possible combinations of new data



# Findings

- Studied OpenWPM tool provided by Princeton and found that they parse all the URLs in a given webpage but do not store any data from those webpages. The corresponding table xpath is empty.
- So, turned to BeautifulSoup.
- Add/ remove text and image operations were time taking but making them completely random took a lot more time in fact most of the time!

# Findings

- Interesting find: BeautifulSoup simply returns `<None Type>` object if the object is wrapped under a lot of tags e.g.

```
markup = '<a href="http://example.com/">I  
        linked to <i>example.com</i></a>'
```

So workaround: iterate through all elements and check against them for the required tag.

# Findings

- To print the items extracted from a webpage to console or file, we must use `encoding(latin1)` or such
- To print html soup objects, we must use `unicode utf-8`. Python supports this through libraries `unicode dammit`.
- Basic browser functions provided by using `webbrowser.py` to display html generated from `beautifulsoup` directly onto a browser.

# Descoped

- Had to descope the following:
  - Unit test for
    - lack of time
  - OpenWPM for taking input webpages for both:
    - lack of time and
    - little difference in the results we get. (OpenWPM uses advanced browser features like recording browser sessions as well as browser profiles, cookies, browser logging etc)

# Test results

- Demo
- Can generate as many different webpages we want to test the data observatory.
- All user configurable requirements are met.
- All modifications in the results are completely random.
- Dynamic pages are rendered in a much better way than static pages eg. [www.jabong.com](http://www.jabong.com)

# Questions?

Thank You