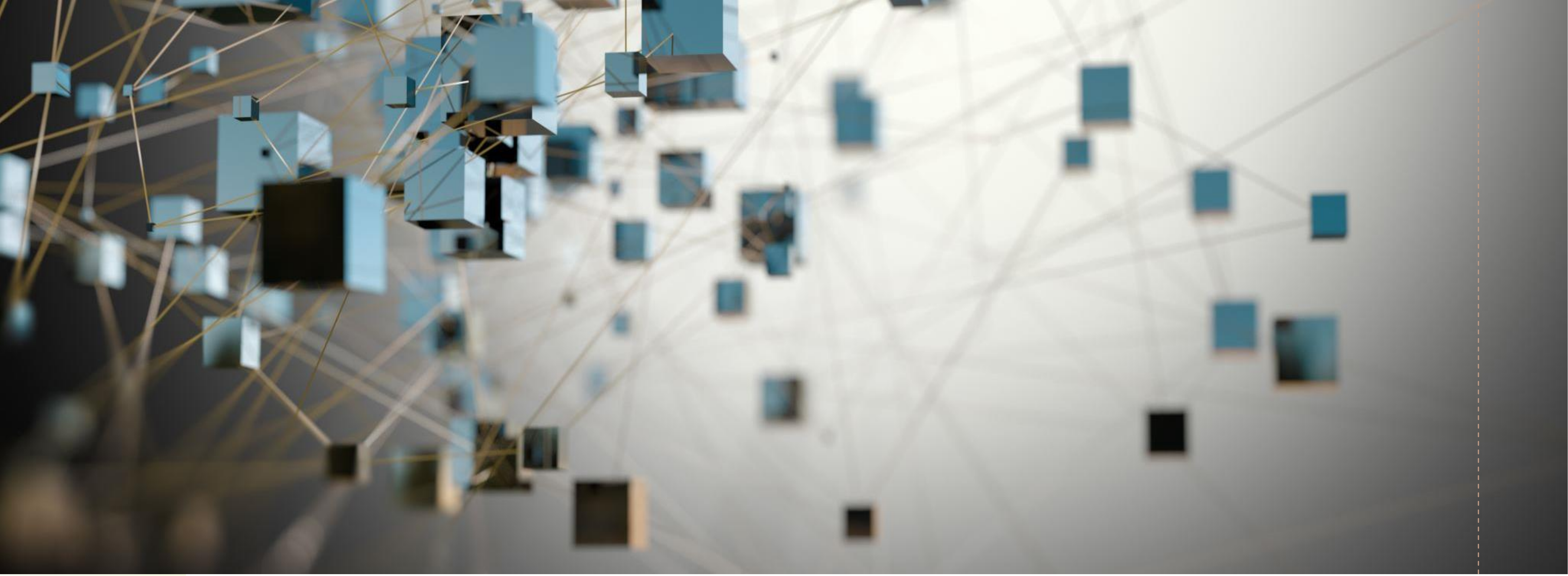


FINAL PROJECT

DATA ENGINEERING - Web Scraping

(DS103)

WONG WOON YONG

Introduction

‘Web Scraping’

A technique that gather structured data or information from web pages.

Step 1: Accessing target Website using HTTP library “requests”.

Step 2: Parse content of web using Web Parsing library Beautiful Soup.

Step 3: Save result to DataFrame format.

Objective

➔ From NEWS website, shows process of Parsing and Collecting information. Target to retrieve Title, Header and HTML elements by the tag name (any 2 tag).

Step 1: Accessing target Website

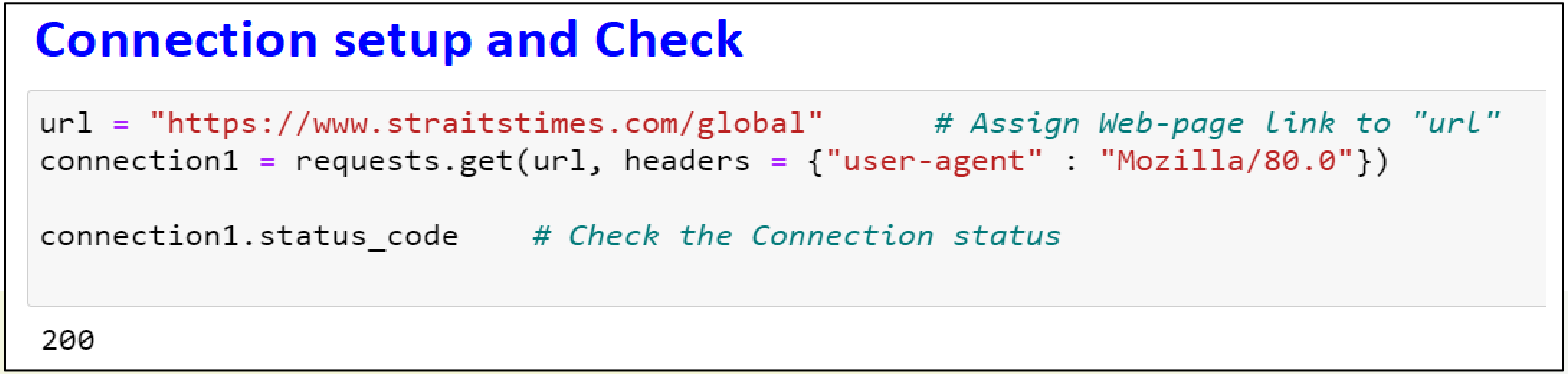
Established a Connection using “requests” and check for **Connection status**:

1) Code 200 series: Connection successful

2) Code 400 series: "Forbidden, cannot access due to blocked or protected"

3) Code 500 series: "Server Error"

Important: Only when having connection status with Code 200 then can proceed to next step.



Connection setup and Check

```
url = "https://www.straitstimes.com/global"      # Assign Web-page link to "url"
connection1 = requests.get(url, headers = {"user-agent" : "Mozilla/80.0"})

connection1.status_code    # Check the Connection status
```

```
200
```

Step 2: Parse content of website

- Requests fetch a page
- BeautifulSoup to parse content and extracting information

Parser Example:

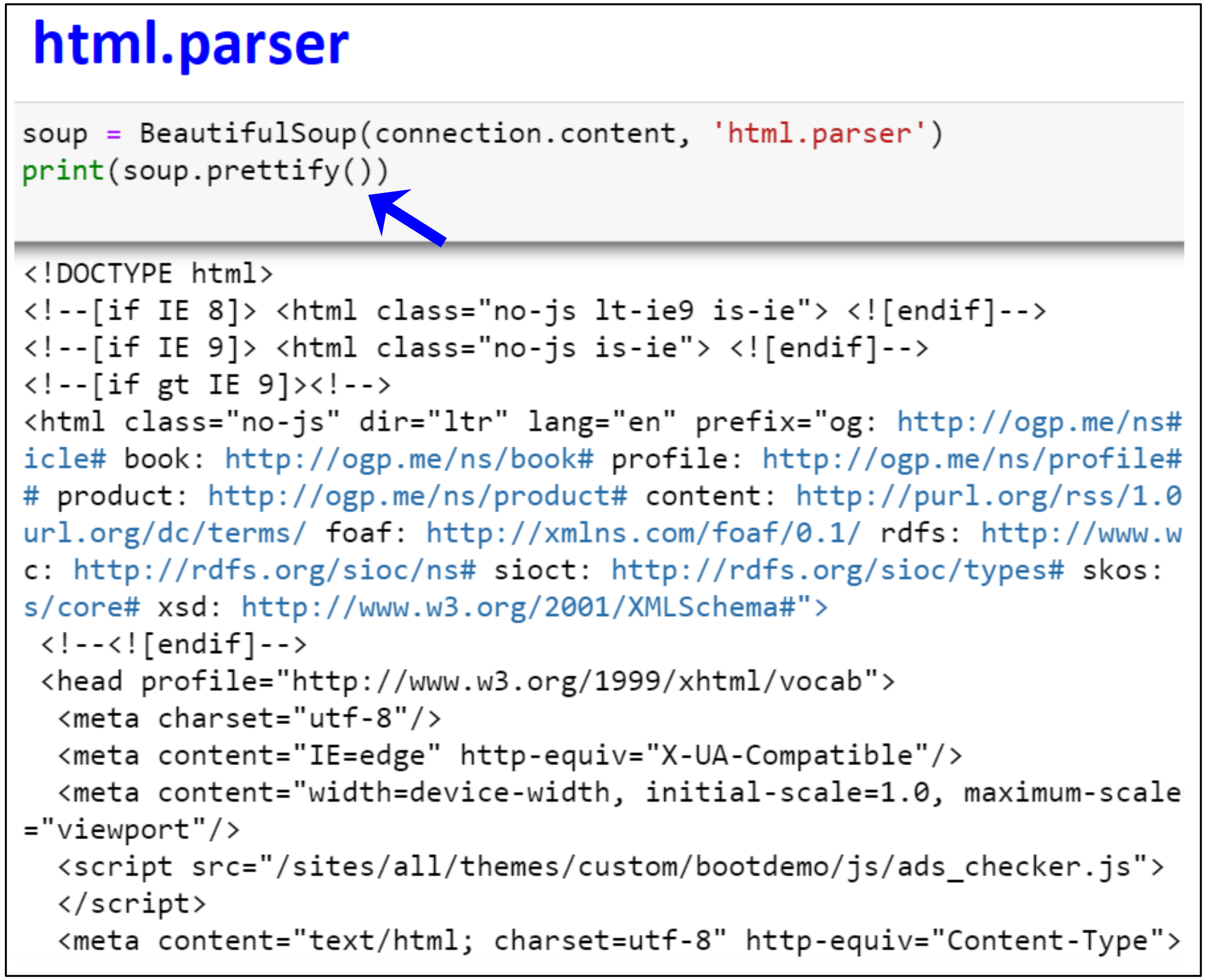
(a) "html.parser“ (b) " html5lib "

Observation

1) Contents extracted is 95% similar.

2) 'html.parser' speed is decent and "html5lib" is slower.

3) "html5lib" better handling tangling tag issues.



html.parser

```
soup = BeautifulSoup(connection.content, 'html.parser')
print(soup.prettify())
```

```
<!DOCTYPE html>
<!--[if IE 8]> <html class="no-js lt-ie9 is-ie"> <![endif]-->
<!--[if IE 9]> <html class="no-js is-ie"> <![endif]-->
<!--[if gt IE 9]><!-->
<html class="no-js" dir="ltr" lang="en" prefix="og: http://ogp.me/ns#
icle# book: http://ogp.me/ns/book# profile: http://ogp.me/ns/profile#
# product: http://ogp.me/ns/product# content: http://purl.org/rss/1.0
url.org/dc/terms/ foaf: http://xmlns.com/foaf/0.1/ rdfs: http://www.w
c: http://rdfs.org/sioc/ns# sioct: http://rdfs.org/sioc/types# skos:
s/core# xsd: http://www.w3.org/2001/XMLSchema#">
 <!--<![endif]-->
 <head profile="http://www.w3.org/1999/xhtml/vocab">
  <meta charset="utf-8"/>
  <meta content="IE=edge" http-equiv="X-UA-Compatible"/>
  <meta content="width=device-width, initial-scale=1.0, maximum-scale
="viewport"/>
  <script src="/sites/all/themes/custom/bootdemo/js/ads_checker.js">
  </script>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
```

Step 2: Parse content of website

a) Extracting Title

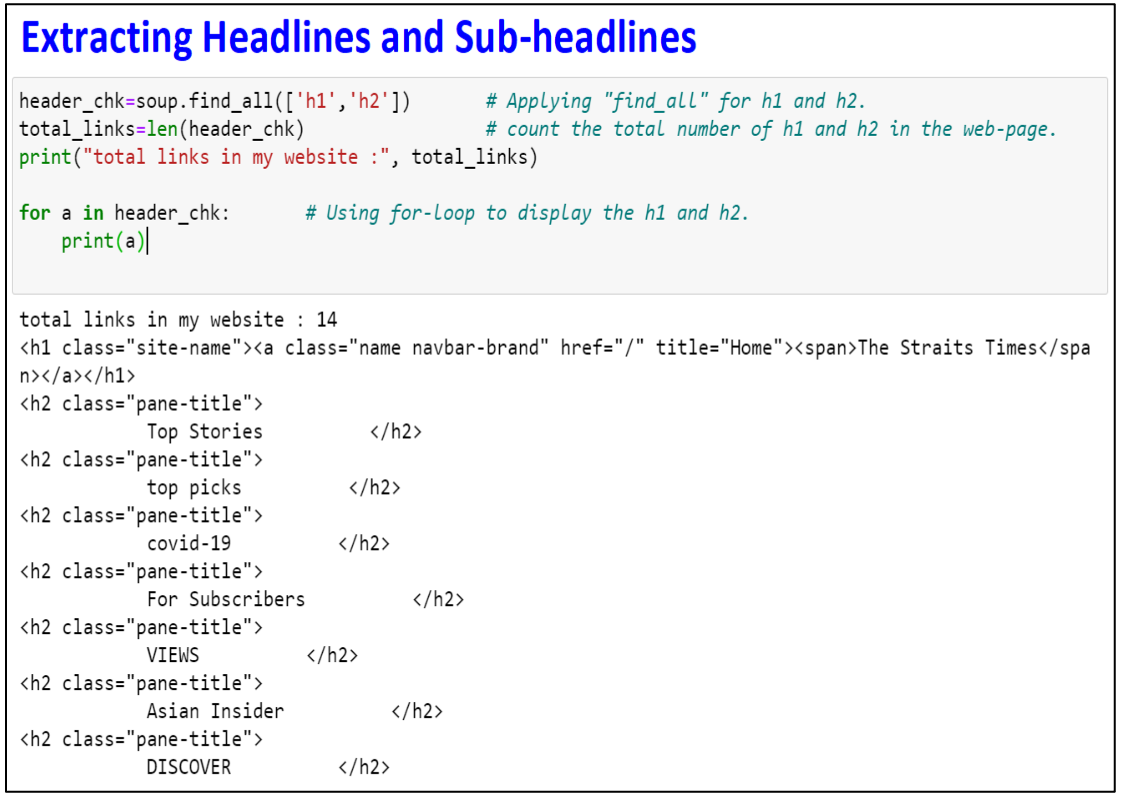
Title element: Assigning title to HTML document.

b) Extracting Headlines and Sub-headlines

- Header contains targeted keywords and close to related page title and content.
- Sub-header should contain similar keywords as header tag.

c) Extracting other HTML tags - "a" & "span"

- Anchor (a) element: Create hyperlink for web-page or a location within web-page itself.
- Span element: Select inline content for purely styling purposes.

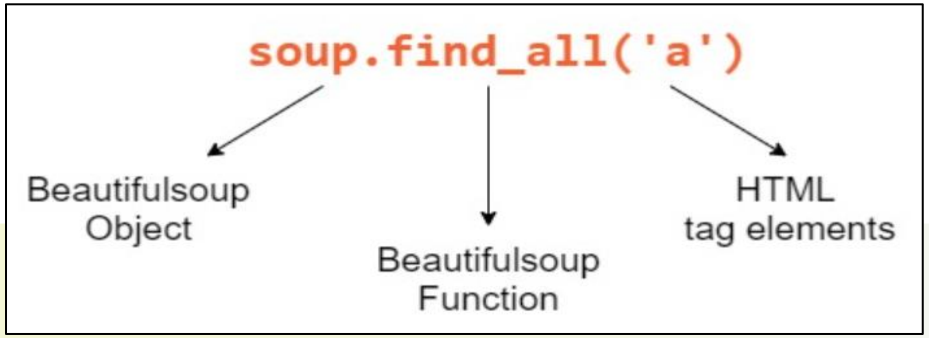


Extracting Headlines and Sub-headlines

```
header_chk=soup.find_all(['h1','h2'])       # Applying "find_all" for h1 and h2.
total_links=len(header_chk)                 # count the total number of h1 and h2 in the web-page.
print("total links in my website :", total_links)

for a in header_chk:       # Using for-loop to display the h1 and h2.
    print(a)
```

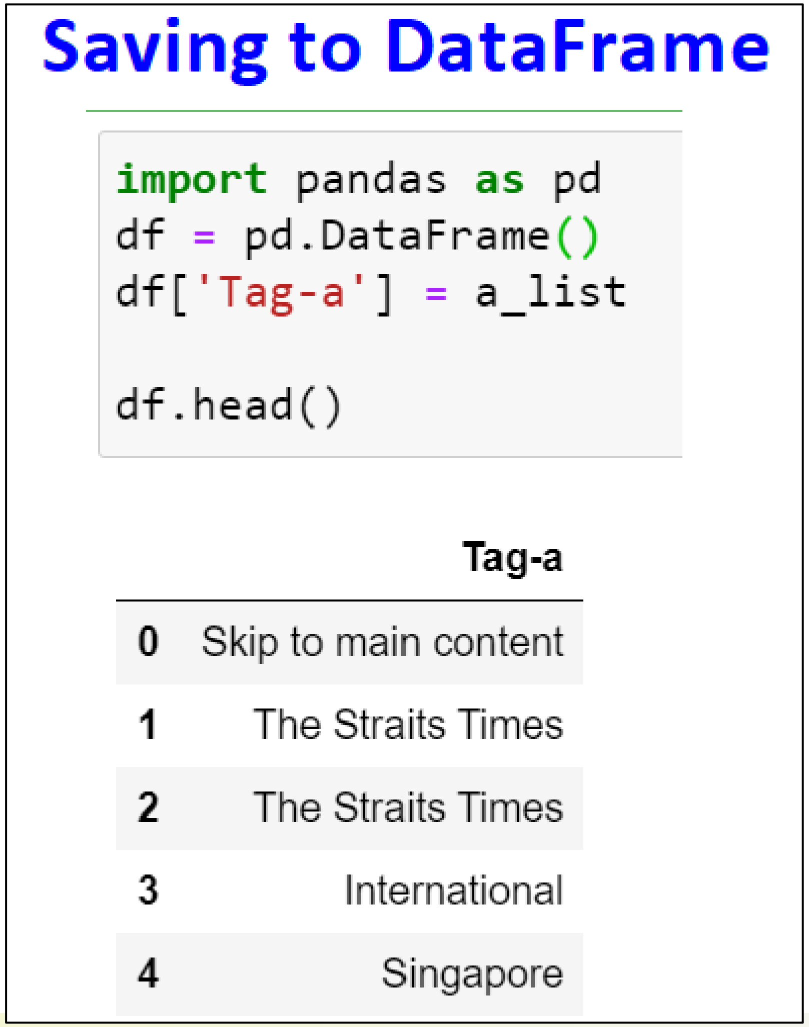
```
total links in my website : 14
<h1 class="site-name"><a class="name navbar-brand" href="/" title="Home"><span>The Straits Times</spa
n></a></h1>
<h2 class="pane-title">
            Top Stories          </h2>
<h2 class="pane-title">
            top picks          </h2>
<h2 class="pane-title">
            covid-19          </h2>
<h2 class="pane-title">
            For Subscribers          </h2>
<h2 class="pane-title">
            VIEWS          </h2>
<h2 class="pane-title">
            Asian Insider          </h2>
<h2 class="pane-title">
            DISCOVER          </h2>
```



Step 3: Save to DataFrame format

Saving extracted information into DataFrame by using Pandas.

A Dataframe format can be save into another format like CSV, Excel and etc.



Saving to DataFrame

```
import pandas as pd
df = pd.DataFrame()
df['Tag-a'] = a_list

df.head()
```

	Tag-a
0	Skip to main content
1	The Straits Times
2	The Straits Times
3	International
4	Singapore

Q & A

- Thank You -