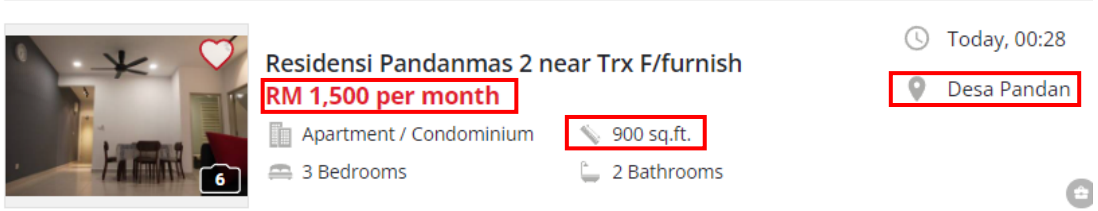
Question 1

- Beautiful Soup primarily works with static content—HTML and XML documents as they are loaded in the browser.
- For dynamic websites such as **Mudah.com** that use JavaScript to load content, Beautiful Soup alone is not sufficient because it cannot execute JavaScript.
- Therefore, we use Selenium library in Python to load dynamic content from Mudah.com.

Question 1 cont.

- From Table A we can see that the information that we want to scrape is Property name, Area, Size and Pricing.
- We can see that the listing page on Mudah.com contains Property type, no of Bedrooms, no of Bathrooms, Pricing, Area, and the listing datetime.
- One thing to note is that the property name is not shown on the listing page. We need to click on the link to get the details about the Property.
- So, in this case we scrape the Pricing, Size, Area on the listing page, and the Property name on the details page.



Residensi Pandanmas 2 near Trx F/furnish

RM 1,500 per month

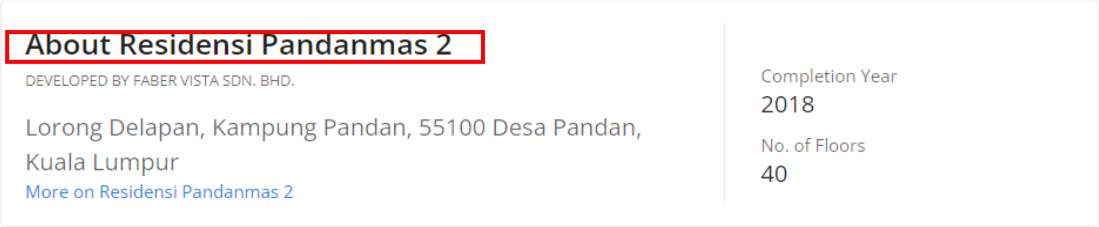
Apartment / Condominium   900 sq.ft.

3 Bedrooms   2 Bathrooms

Today, 00:28

Desa Pandan

Pricing, Area, Size



About Residensi Pandanmas 2

DEVELOPED BY FABER VISTA SDN. BHD.

Lorong Delapan, Kampung Pandan, 55100 Desa Pandan, Kuala Lumpur

More on Residensi Pandanmas 2

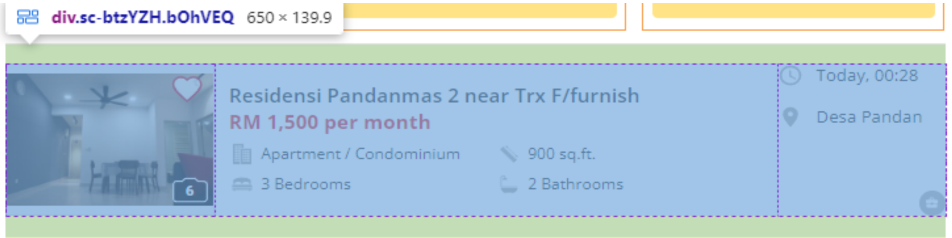
Completion Year
2018

No. of Floors
40

Property Name

Question 1 cont.

– The first step of scrapping is to identify the tag that contains the information that we want to scrap.



div.sc-btzYZH.bOhVEQ 650 × 139.9

Residensi Pandanmas 2 near Trx F/furnish

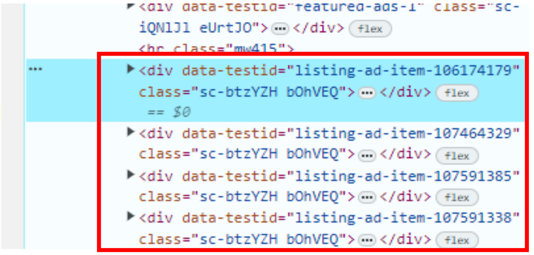
RM 1,500 per month

Apartment / Condominium 900 sq.ft.

3 Bedrooms 2 Bathrooms

Today, 00:28

Desa Pandan

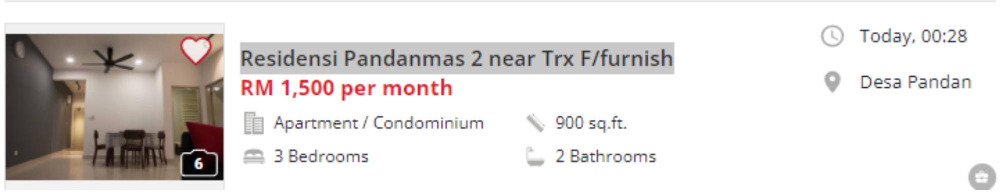


```
▸<div data-testid="featured-ads-1" class="sc-
  iQN1J1 eUrtJO"> ⋯ </div> flex
  <hr class="mw415">
▸<div data-testid="listing-ad-item-106174179"
  class="sc-btzYZH bOhVEQ"> ⋯ </div> flex
   == $0
▸<div data-testid="listing-ad-item-107464329"
  class="sc-btzYZH bOhVEQ"> ⋯ </div> flex
▸<div data-testid="listing-ad-item-107591385"
  class="sc-btzYZH bOhVEQ"> ⋯ </div> flex
▸<div data-testid="listing-ad-item-107591338"
  class="sc-btzYZH bOhVEQ"> ⋯ </div> flex
```

– We cannot directly use the class name as this is dynamic web. The class name changes when we refresh the web.

– From the html code we can see that the all the listing tags have an attribute called 'data-testid', and they contain substrings 'listing-ad-item'. We can make use of this characteristics to scrape the information for each property listed on the page.

– As mentioned in the previous page, we need to scrape the Property name on the details page.



Residensi Pandanmas 2 near Trx F/furnish

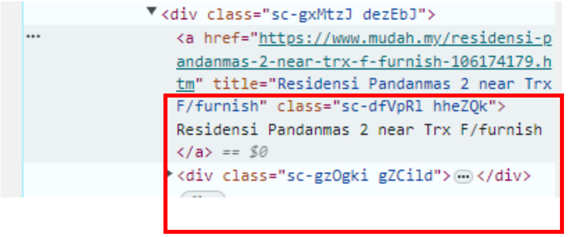
RM 1,500 per month

Apartment / Condominium 900 sq.ft.

3 Bedrooms 2 Bathrooms

Today, 00:28

Desa Pandan

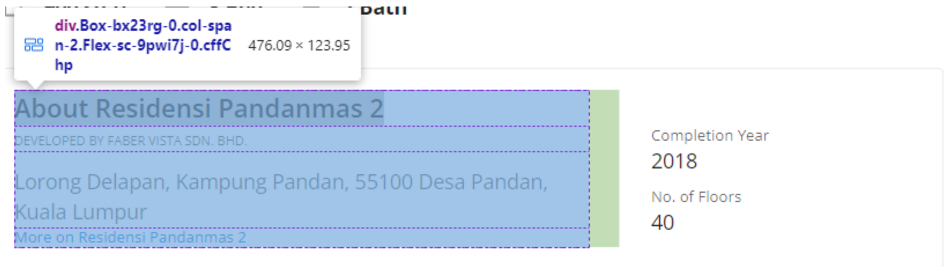


```
▾<div class="sc-gxMtzJ dezEbJ">
   <a href="https://www.mudah.my/residensi-p
   andanmas-2-near-trx-f-furnish-106174179.h
   tm" title="Residensi Pandanmas 2 near Trx
   F/furnish" class="sc-dfVpRl hheZQk">
   Residensi Pandanmas 2 near Trx F/furnish
   </a> == $0
  ▸<div class="sc-gzOgki gZCild"> ⋯ </div>
```

– Within the listing tag, we find the tag that contains the link of the details page <a href=“”>, and then scrape the Property name information from it.

Question 1 cont.

- On the property details page, we identify the tag that contains property information and define a function to extract the property name.



Bath
div.Box-bx23rg-0.col-span-2.Flex-sc-9pwi7j-0.cffChp 476.09 × 123.95

About Residensi Pandanmas 2

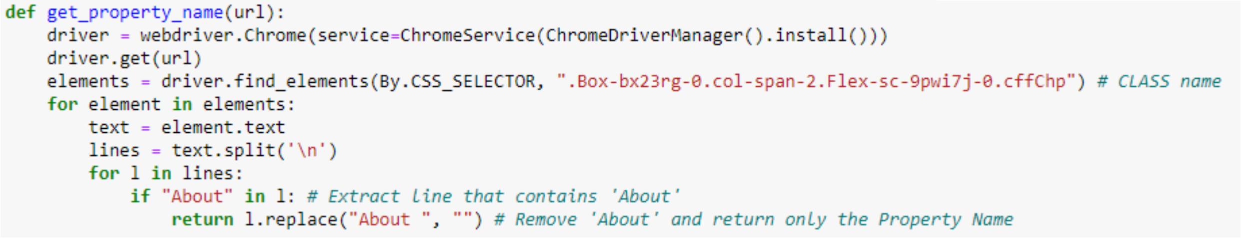
DEVELOPED BY FABER VISTA SDN. BHD.

Lorong Delapan, Kampung Pandan, 55100 Desa Pandan, Kuala Lumpur

More on Residensi Pandanmas 2

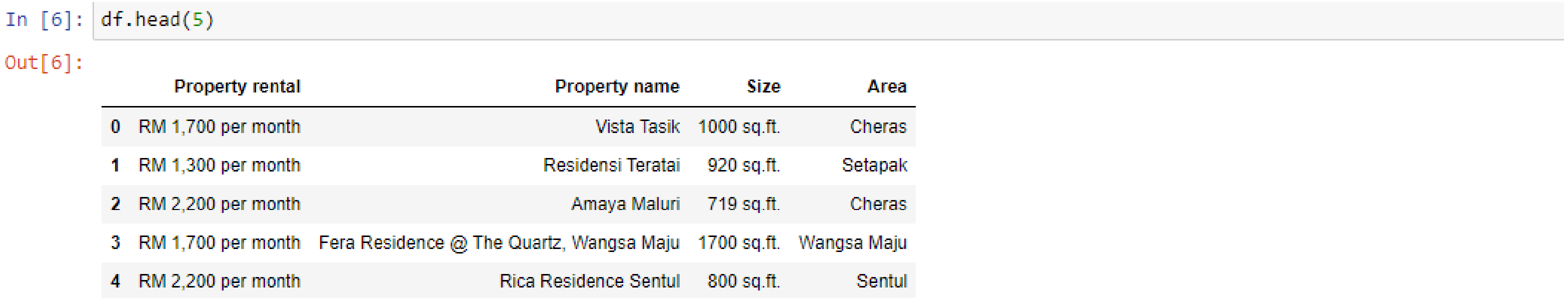
Completion Year
2018

No. of Floors
40



```python
def get_property_name(url):
    driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()))
    driver.get(url)
    elements = driver.find_elements(By.CSS_SELECTOR, ".Box-bx23rg-0.col-span-2.Flex-sc-9pwi7j-0.cffChp") # CLASS name
    for element in elements:
        text = element.text
        lines = text.split('\n')
        for l in lines:
            if "About" in l: # Extract line that contains 'About'
                return l.replace("About ", "") # Remove 'About' and return only the Property Name
```

- After getting all the information we want, we store it into a list then append to pandas dataframe for further processing.

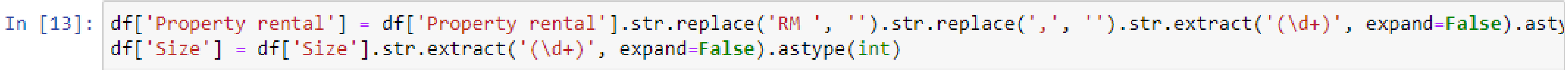


```
In [6]: df.head(5)
Out[6]:
```

	Property rental	Property name	Size	Area
0	RM 1,700 per month	Vista Tasik	1000 sq.ft.	Cheras
1	RM 1,300 per month	Residensi Teratai	920 sq.ft.	Setapak
2	RM 2,200 per month	Amaya Maluri	719 sq.ft.	Cheras
3	RM 1,700 per month	Fera Residence @ The Quartz, Wangsa Maju	1700 sq.ft.	Wangsa Maju
4	RM 2,200 per month	Rica Residence Sentul	800 sq.ft.	Sentul

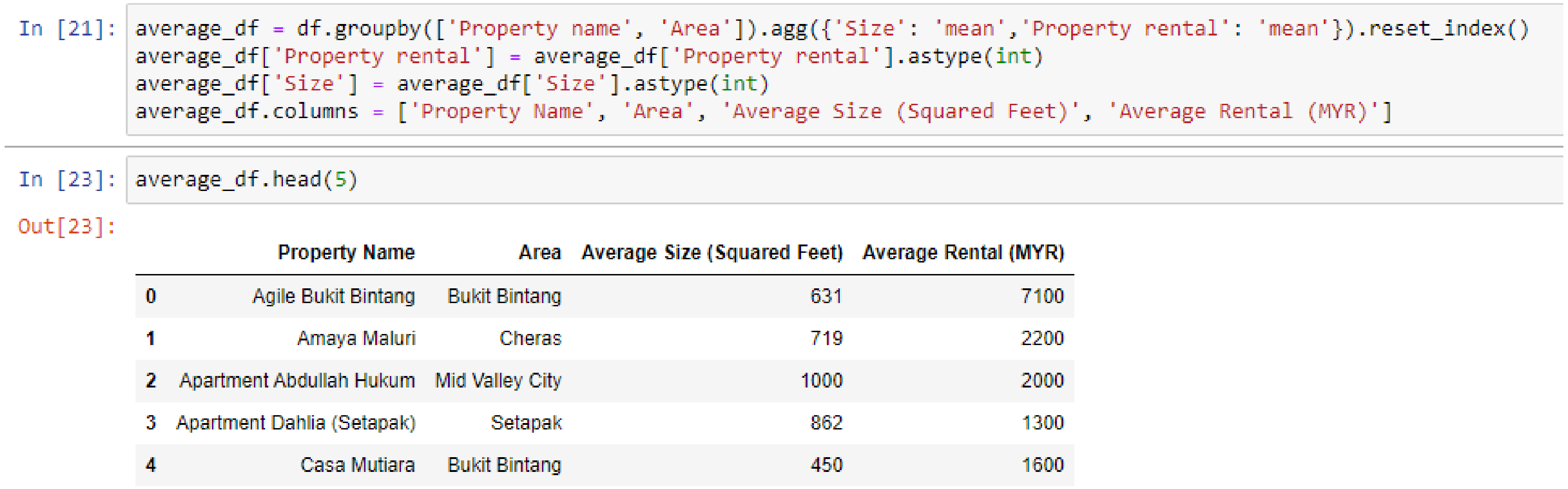
Question 1 cont.

– We do a little pre-processing stuff to clean the data by removing the non-numeric characters and commas, making it suitable for analysis.



```
In [13]: df['Property rental'] = df['Property rental'].str.replace('RM ', '').str.replace(',', '').str.extract('(\d+)', expand=False).asty
         df['Size'] = df['Size'].str.extract('(\d+)', expand=False).astype(int)
```

– After that, we use 'groupby' function in Pandas library to calculate the average price and size for each property, then convert the dataframe to excel.

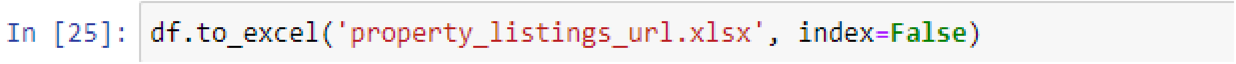


```
In [21]: average_df = df.groupby(['Property name', 'Area']).agg({'Size': 'mean','Property rental': 'mean'}).reset_index()
         average_df['Property rental'] = average_df['Property rental'].astype(int)
         average_df['Size'] = average_df['Size'].astype(int)
         average_df.columns = ['Property Name', 'Area', 'Average Size (Squared Feet)', 'Average Rental (MYR)']
```

```
In [23]: average_df.head(5)
```

Out[23]:

	Property Name	Area	Average Size (Squared Feet)	Average Rental (MYR)
0	Agile Bukit Bintang	Bukit Bintang	631	7100
1	Amaya Maluri	Cheras	719	2200
2	Apartment Abdullah Hukum	Mid Valley City	1000	2000
3	Apartment Dahlia (Setapak)	Setapak	862	1300
4	Casa Mutiara	Bukit Bintang	450	1600



```
In [25]: df.to_excel('property_listings_url.xlsx', index=False)
```