```python
data = pd.read_csv('/Users/burhanudin/Study_burhanudin_6/
```

```python
# Mari kita lihal lagi info data
data.info()
```

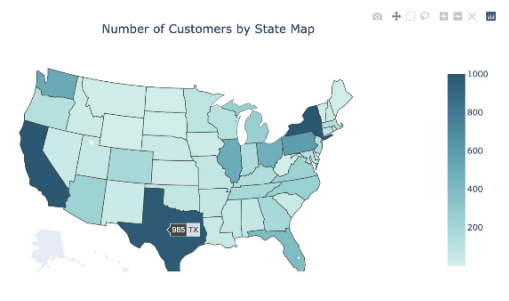
```
<class 'pandas.core.frame.DataFrame'>
Index: 9993 entries, 1 to 9994
Data columns (total 18 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Order ID      9993 non-null   object
 1   Order Date    9993 non-null   object
 2   Ship Date     9993 non-null   object
 3   Ship Mode     9993 non-null   object
 4   Customer ID   9993 non-null   object
 5   Segment       9993 non-null   object
 6   City          9993 non-null   object
 7   State         9993 non-null   object
 8   Postal Code   9993 non-null   int64
 9   Region        9993 non-null   object
 10  Product ID    9993 non-null   object
 11  Category      9993 non-null   object
 12  Sub-Category  9993 non-null   object
 13  Product Name  9993 non-null   object
 14  Sales         9993 non-null   float64
 15  Quantity      9993 non-null   int64
 16  Discount      9993 non-null   float64
 17  Profit        9993 non-null   float64
dtypes: float64(3), int64(2), object(13)
memory usage: 1.4+ MB
```

- Jumlah kolom tingal 18
- Juml;ah row jadi tinggal 9993 dari awalnya 9994

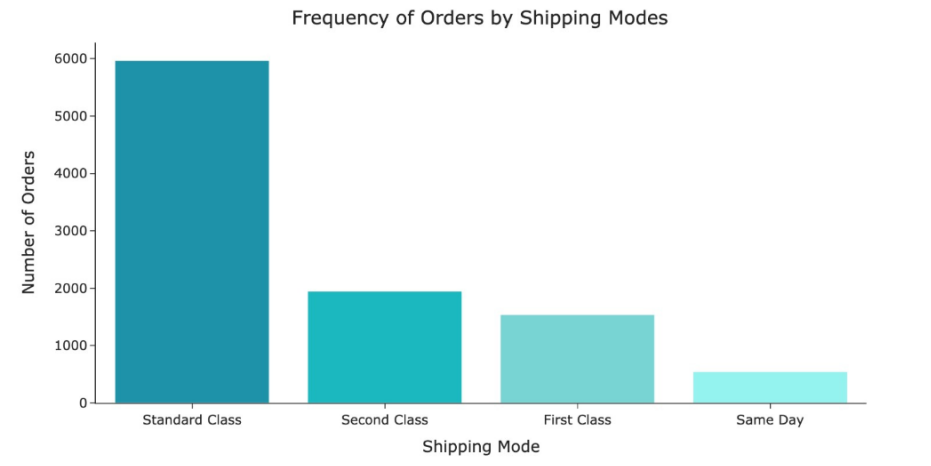
```python
# creating a custom palette
colors = ['#1192AA','#00B8C0', '#73D4D3', '#90F2F0']

sns.palplot(colors)
plt.gcf().set_size_inches(8,2.5)

for idx,values in enumerate(colors):
    plt.text(idx-0.375,0, colors[idx],{'font': "Courier New", 'size':16, 'weight':'bold','color':'black'}, alpha =0.7)
plt.gcf().set_facecolor('white')

plt.show()
```

```python
us_state_to_abbrev = {
    "Alabama": "AL",
    "Alaska": "AK",
    "Arizona": "AZ",
    "Arkansas": "AR",
    "California": "CA",
    "Colorado": "CO",
```

```python
import plotly.graph_objects as go

# Create new column with State Abbreviations
data['State_abb'] = data['State'].replace(us_state_to_abbrev)

# Plot
fig = go.Figure(data=go.Choropleth(
    locations= data['State_abb'].value_counts().index, # Spatial coordinates
    z = data['State_abb'].value_counts(), # Data to be color-coded
    locationmode = 'USA-states',
    colorscale = 'teal', zmin = 1, zmax = 1000
))

fig.update_layout(
    font = dict(
            size = 14
            ),
    title={
        'text': "Number of Customers by State Map",
        'y':0.95,
        'x':0.5
        },
    geo_scope='usa', # limite map scope to USA
)

fig.show()
```



```python
fig = px.bar(x = data['Ship Mode'].value_counts().index, y = data['Ship Mode'].value_counts(),
             color = data['Ship Mode'].value_counts().index,
             template = 'simple_white', color_discrete_sequence = colors)

#Set Title and x/y axis labels
fig.update_layout(
    xaxis_title="Shipping Mode",
    yaxis_title="Number of Orders",
    showlegend = False,
    font = dict(
            size = 14
            ),
    title={
        'text': "Frequency of Orders by Shipping Modes",
        'y':0.95,
        'x':0.5
        }
    )

fig.show()
```



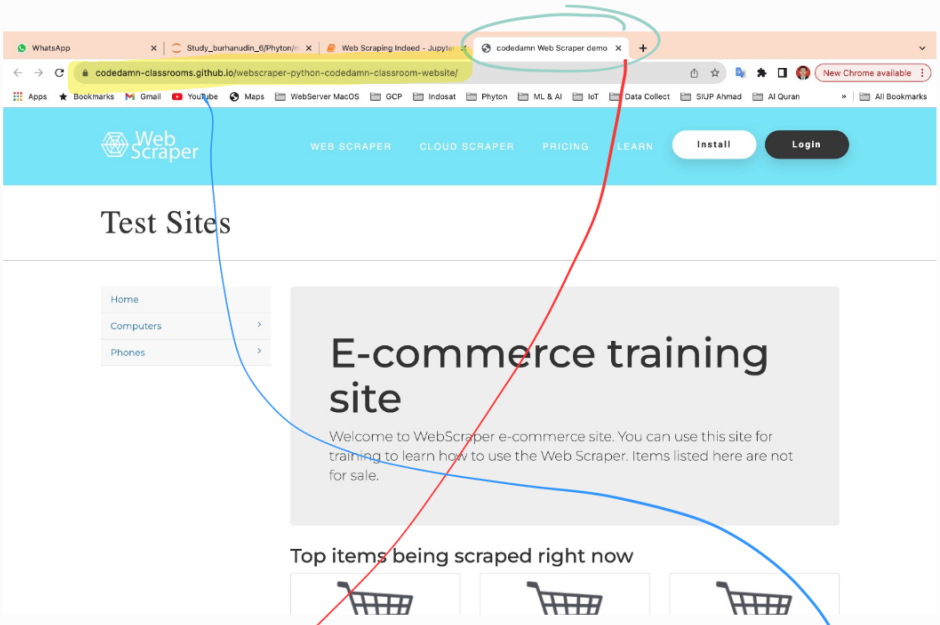
Loading web page with 'request'

```python
import requests

# Make a request to https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/
# Store the result in 'res' variable
res = requests.get(
    'https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/')
txt = res.text
status = res.status_code

print(txt, status)
# print the result
```

```
<!DOCTYPE html>
<html lang="en">
        <head>
                <!-- Anti-flicker snippet (recommended)  -->
                <style>
                        .async-hide {
                                opacity: 0 !important;
                        }
                </style>
                <title>codedamn Web Scraper demo</title>
                <meta charset="utf-8" />
                <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

                <meta
                        name="keywords"
                        content="web scraping,Web Scraper,Chrome extension,Crawling,Cross platform scraper, "
                />
                <meta name="description" content="The most popular web scraping website." />
                <link
```

satus '200'

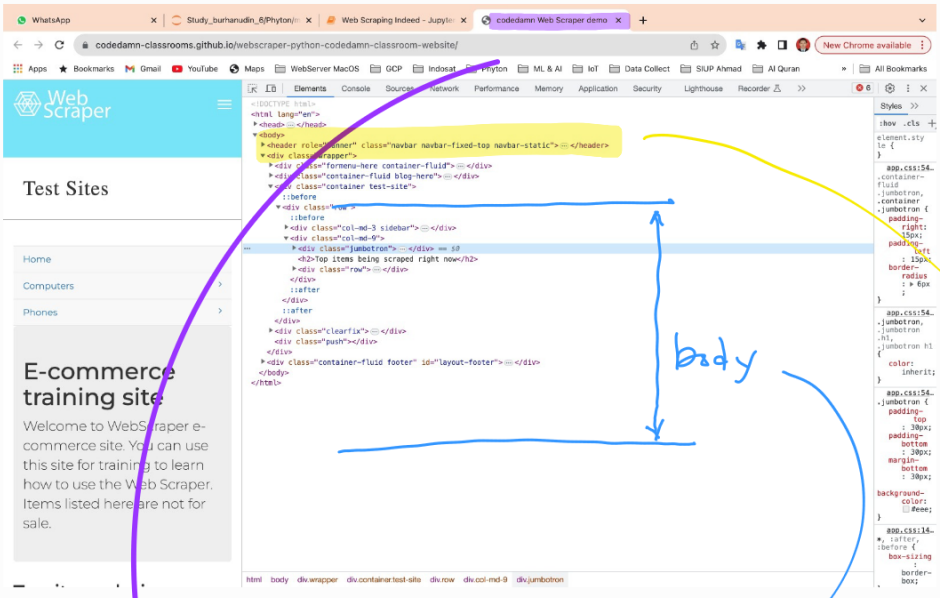
Exctract 'title'




```python
import requests
from bs4 import BeautifulSoup

# Make a request to https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/
page = requests.get(
    "https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/")
soup = BeautifulSoup(page.content, 'html.parser')

# Extract title of page
page_title = soup.title.text

# print the result
print(page_title)
```

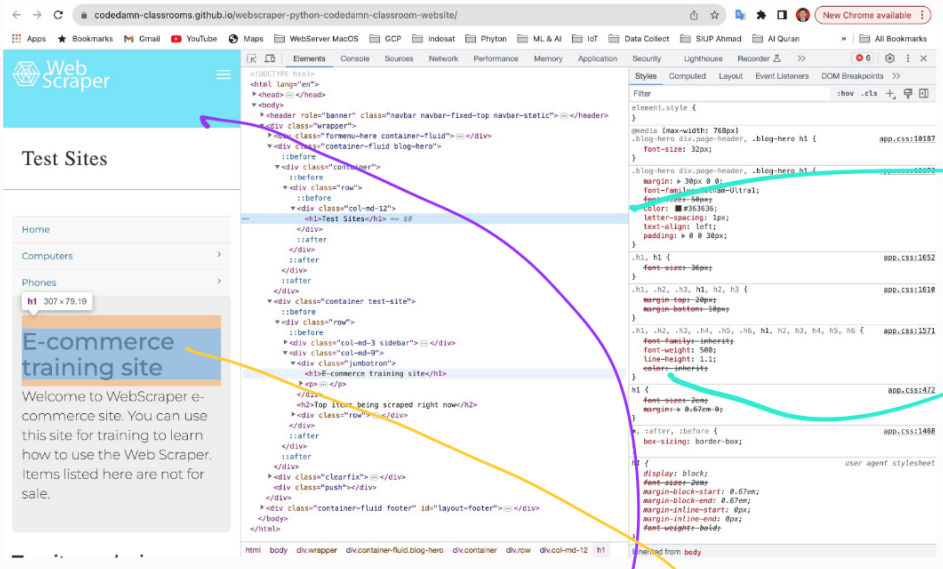
```
codedamn Web Scraper demo
```

Exctract bagian body and head




```python
import requests
from bs4 import BeautifulSoup

# Make a request
page = requests.get(
    "https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/")
soup = BeautifulSoup(page.content, 'html.parser')

# Extract title of page
page_title = soup.title

# Extract body of page
page_body = soup.body

# Extract head of page
page_head = soup.head

# print the result
print(page_title, page_head)
```

Select Element dengan BeautifulSoup



h1

```python
import requests
from bs4 import BeautifulSoup

# Make a request
page = requests.get(
    "https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/")
soup = BeautifulSoup(page.content, 'html.parser')

# Extract first <h1>(...)</h1> text
first_h1 = soup.select('h1')[0].text
```

```python
import requests
from bs4 import BeautifulSoup
# Make a request
page = requests.get(
    "https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/")
soup = BeautifulSoup(page.content, 'html.parser')

# Create all_h1_tags as empty list
all_h1_tags = []

# Set all_h1_tags to all h1 tags of the soup
for element in soup.select('h1'):
    all_h1_tags.append(element.text)

# Create seventh_p_text and set it to 7th p element text of the page
seventh_p_text = soup.select('p')[6].text

print(all_h1_tags, seventh_p_text)
```

select semua <h1>

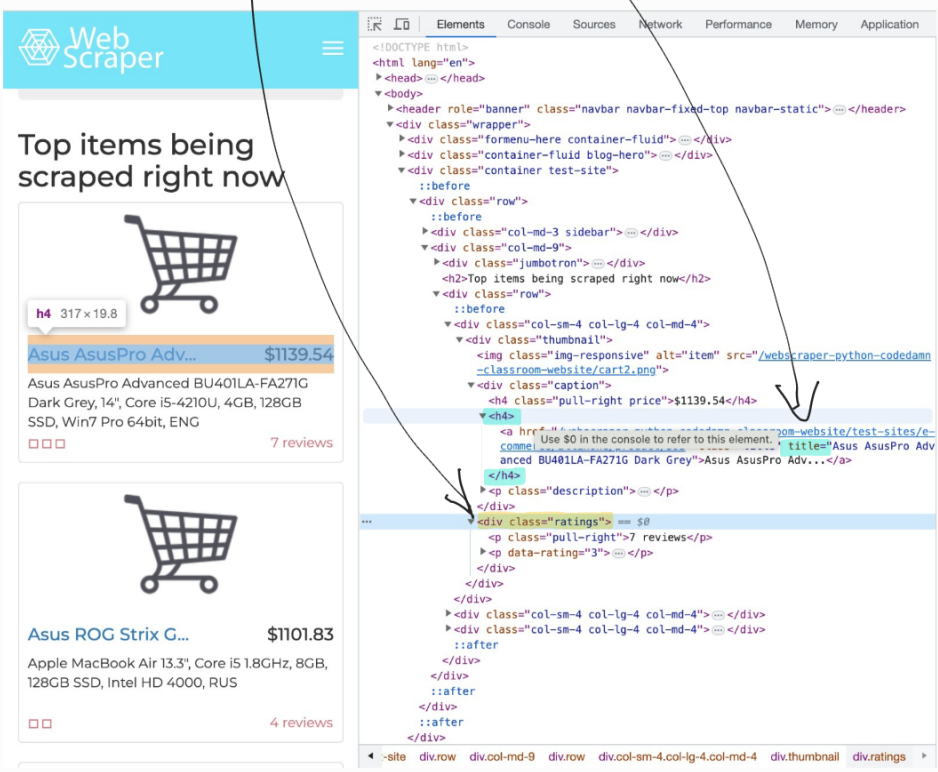
```
['Test Sites', 'E-commerce training site'] 7 reviews
```

simpan teks element

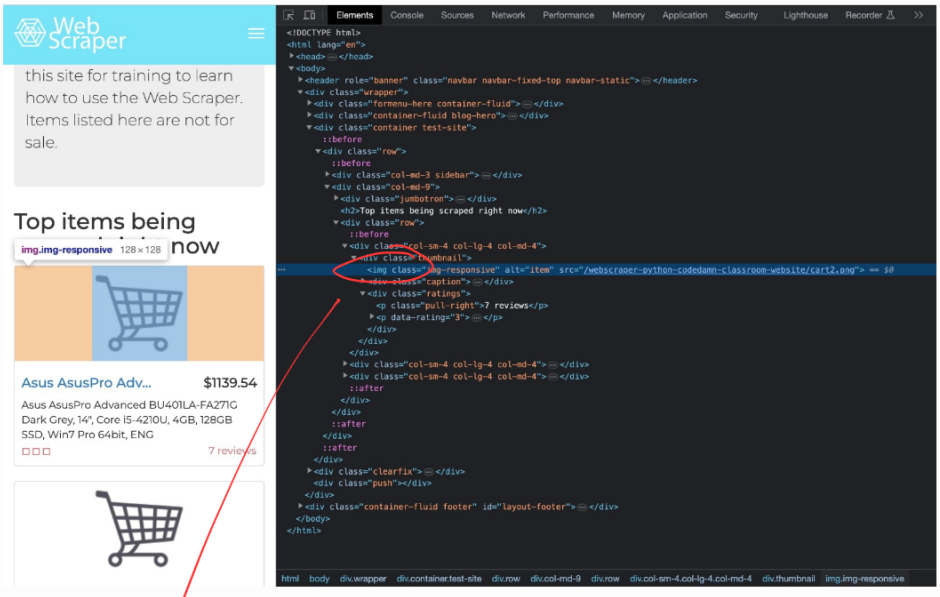
Top Items being scrapted right now

```python
import requests
from bs4 import BeautifulSoup
# Make a request
page = requests.get(
    "https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/")
soup = BeautifulSoup(page.content, 'html.parser')

# Create top_items as empty list
top_items = []

# Extract and store in top_items according to instructions on the left
products = soup.select('div.thumbnail')
for elem in products:
    title = elem.select('h4 > a.title')[0].text
    review_label = elem.select('div.ratings')[0].text
    info = {
        "title": title.strip(),
        "review": review_label.strip()
    }
    top_items.append(info)

print(top_items)
```

```
[{'title': 'Asus AsusPro Adv...', 'review': '7 reviews'}, {'title': 'Asus ROG Strix G...', 'review': '4 reviews'},
{'title': 'Acer Aspire 3 A3...', 'review': '2 reviews'}]
```

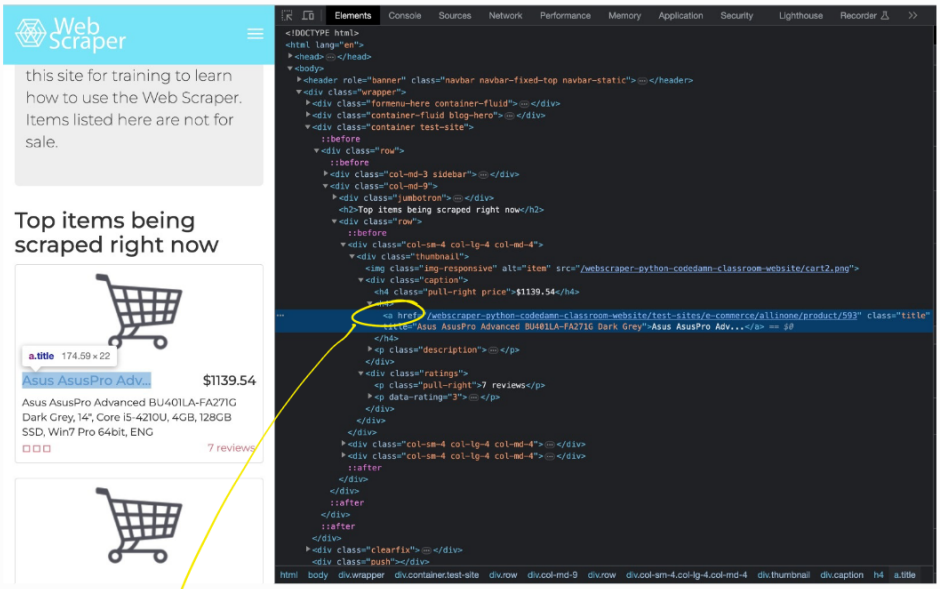


Exctracting Links




```python
import requests
from bs4 import BeautifulSoup
# Make a request
page = requests.get(
    "https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/")
soup = BeautifulSoup(page.content, 'html.parser')

# Create top_items as empty list
image_data = []

# Extract and store in top_items according to instructions on the left
images = soup.select('img')
for image in images:
    src = image.get('src')
    alt = image.get('alt')
    image_data.append({"src": src, "alt": alt})

print(image_data)
```

```
[{'src': '/webscraper-python-codedamn-classroom-website/logo_white.svg', 'alt': 'Web Scraper'}, {'src': '/webscraper-
python-codedamn-classroom-website/cart2.png', 'alt': 'item'}, {'src': '/webscraper-python-codedamn-classroom-website/
cart2.png', 'alt': 'item'}, {'src': '/webscraper-python-codedamn-classroom-website/cart2.png', 'alt': 'item'}, {'sr
c': '/webscraper-python-codedamn-classroom-website/fbicon.png', 'alt': 'Web Scraper on Facebook'}, {'src': '/webscrap
er-python-codedamn-classroom-website/twicon.png', 'alt': 'Web Scraper on Twitter'}]
```




```python
import requests
from bs4 import BeautifulSoup
# Make a request
page = requests.get(
    "https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/")
soup = BeautifulSoup(page.content, 'html.parser')

# Create top_items as empty list
all_links = []

# Extract and store in top_items according to instructions on the left
links = soup.select('a')
for ahref in links:
    text = ahref.text
    text = text.strip() if text is not None else ''

    href = ahref.get('href')
    href = href.strip() if href is not None else ''
    all_links.append({"href": href, "text": text})

print(all_links)
```

Extract atribut { href, text } → Ada, spaci

```
[{'href': '', 'text': 'Toggle navigation'}, {'href': '/webscraper-python-codedamn-classroom-website/', 'text': ''},
{'href': '#page-top', 'text': ''}, {'href': '/webscraper-python-codedamn-classroom-website/', 'text': 'Web Scraper'},
{'href': '/webscraper-python-codedamn-classroom-website/cloud-scraper', 'text': 'Cloud Scraper'}, {'href': '/webscrap
er-python-codedamn-classroom-website/pricing', 'text': 'Pricing'}, {'href': '#section3', 'text': 'Learn'}, {'href':
'/webscraper-python-codedamn-classroom-website/documentation', 'text': 'Documentation'}, {'href': '/webscraper-python
-codedamn-classroom-website/tutorials', 'text': 'Video Tutorials'}, {'href': '/webscraper-python-codedamn-classroom-w
ebsite/how-to-videos', 'text': 'How to'}, {'href': '/webscraper-python-codedamn-classroom-website/test-sites', 'tex
t': 'Test Sites'}, {'href': 'https://forum.webscraper.io/', 'text': 'Forum'}, {'href': 'https://chrome.google.com/web
store/detail/web-scraper/jnhgnonknehpejjnehehllkliplmbmhn?hl=en', 'text': 'Install'}, {'href': 'https://cloud.webscra
per.io/', 'text': 'Login'}, {'href': '/webscraper-python-codedamn-classroom-website/test-sites/e-commerce/allinone',
'text': 'Home'}, {'href': '/webscraper-python-codedamn-classroom-website/test-sites/e-commerce/allinone/computers',
'text': 'Computers'}, {'href': '/webscraper-python-codedamn-classroom-website/test-sites/e-commerce/allinone/phones',
'text': 'Phones'}, {'href': '/webscraper-python-codedamn-classroom-website/test-sites/e-commerce/allinone/product/59
3', 'text': 'Asus AsusPro Adv...'}, {'href': '/webscraper-python-codedamn-classroom-website/test-sites/e-commerce/all
inone/product/583', 'text': 'Asus ROG Strix G...'}, {'href': '/webscraper-python-codedamn-classroom-website/test-site
s/e-commerce/allinone/product/576', 'text': 'Acer Aspire 3 A3...'}, {'href': '/webscraper-python-codedamn-classroom-w
ebsite/', 'text': 'Web Scraper browser extension'}, {'href': '/webscraper-python-codedamn-classroom-website/pricing'
```

Generating CSV from data

```python
import requests
from bs4 import BeautifulSoup
import csv
# Make a request
page = requests.get(
    "https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/")
soup = BeautifulSoup(page.content, 'html.parser')

all_products = []

products = soup.select('div.thumbnail')
for product in products:
    # TODO: Work
    print("Work on product here")


keys = all_products[0].keys()

with open('products.csv', 'w', newline='') as output_file:
    dict_writer = csv.DictWriter(output_file, keys)
    dict_writer.writeheader()
    dict_writer.writerows(all_products)
```

```
Work on product here
Work on product here
Work on product here

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[12], line 17
     12 for product in products:
     13     # TODO: Work
     14     print("Work on product here")
---> 17 keys = all_products[0].keys()
     19 with open('products.csv', 'w', newline='') as output_file:
     20     dict_writer = csv.DictWriter(output_file, keys)

IndexError: list index out of range
```

Error di koreksi dibawah

```python
import requests
from bs4 import BeautifulSoup
import csv
# Make a request
page = requests.get(
    "https://codedamn-classrooms.github.io/webscraper-python-codedamn-classroom-website/")
soup = BeautifulSoup(page.content, 'html.parser')

# Create top_items as empty list
all_products = []

# Extract and store in top_items according to instructions on the left
products = soup.select('div.thumbnail')
for product in products:
    name = product.select('h4 > a')[0].text.strip()
    description = product.select('p.description')[0].text.strip()
    price = product.select('h4.price')[0].text.strip()
    reviews = product.select('div.ratings')[0].text.strip()
    image = product.select('img')[0].get('src')

    all_products.append({
        "name": name,
        "description": description,
        "price": price,
        "reviews": reviews,
        "image": image
    })


keys = all_products[0].keys()

with open('products.csv', 'w', newline='') as output_file:
    dict_writer = csv.DictWriter(output_file, keys)
    dict_writer.writeheader()
    dict_writer.writerows(all_products)
```

Coding Detile disini
extract data
1. Product name
2. Price
3. Description
4. Reviews
5. Product Image