

Code Description

web_scrappers

1. FetchDescription.py

- function fetch_desc_amazon
The function fetch_desc_amazon parses the link and gets the product description
- function fetch_desc_walmart
The function fetch_desc_walmart parses the link and gets the product description
- function fetch_desc_ebay
The function fetch_desc_ebay parses the link and gets the product description
- function fetch_desc_costco
The function fetch_desc_costco parses the link and gets the product description
- function fetch_desc_bestbuy
The function fetch_desc_bestbuy parses the link and gets the product description
- function fetch_desc_bjs
The function fetch_desc_bjs parses the link and gets the product description

2. WebScraper.py

- function scrapper(product_description):
Checks for each ecommerce platform the product_description(parameter of function) for details. Details include description, price and url. This function returns these details as a dictionary.

3. web_scrapper_amazon.py

- function get_url_amazon(search_term):
This function returns the Amazon url of the product (search_term) given as an argument.
- function scrap_amazon(driver, search_term)
Webpage of the product corresponding to search_term is retrieved using BeautifulSoup.
- function run(search_term)
Extracts the product description, price , URL and website name corresponding to search_term and returns it through a dictionary variable.

4. web_scrapper_bjs.py

- function get_url_bjs(search_term):
This function returns the bjs url of the product (search_term) given as an argument.
- function scrap_bjs(driver, search_term)
Webpage of the product corresponding to search_term is retrieved using BeautifulSoup.
- function run(search_term)
Extracts the product description, price , URL and website name corresponding to search_term and returns it through a dictionary variable.

5. web_scrapper_ebay.py

- function get_url_amazon(search_term):
This function returns the Amazon url of the product (search_term) given as an argument.
- function scrap_ebay(search_term)
Webpage of the product corresponding to search_term is retrieved using BeautifulSoup.
- function run(search_term)
Extracts the product description, price , URL and website name corresponding to search_term and returns it through a dictionary variable.

6.web_scrapper_walmart.py

- function get_url_walmart(search_term):
This function returns the Walmart url of the product (search_term) given as an argument.
- function scrap_walmart search_term)
Webpage of the product corresponding to search_term is retrieved using BeautifulSoup.
- function run(search_term)
Extracts the product description, price , URL and website name corresponding to search_term and returns it through a dictionary variable.

7. web_scrapper_costco.py

- function get_url_costco(search_term):
This function returns the Costco url of the product (search_term) given as an argument.
- function scrap_costco(search_term)
Webpage of the product corresponding to search_term is retrieved using BeautifulSoup.
- function run(search_term)

Extracts the product description, price , URL and website name corresponding to search_term and returns it through a dictionary variable.

8.web_scrapper_Bestbuy.py

- function get_url_bestbuy(search_term):
This function returns the Amazon url of the product (search_term) given as an argument.
- function scrap_bestbuy(search_term)
Webpage of the product corresponding to search_term is retrieved using BeautifulSoup.
- function run(search_term)
Extracts the product description, price , URL and website name corresponding to search_term and returns it through a dictionary variable.