

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

---



## **NGHIÊN CỨU TỐT NGHIỆP 1**

Đề tài: **Thu thập, lưu trữ và phân tích dữ liệu từ web.**

Chủ đề: **Thu thập lưu trữ và phân tích dữ liệu trên các sàn giao dịch  
điện tử về thiết bị di động.**

Mã học phần: IT5021

Mã lớp học: 139420

Giảng viên hướng dẫn:

TS.Nguyễn Hữu Đức

Họ và tên: Nguyễn Thu Trang

MSSV: 20205035

Lớp: Việt Nhật 05 - K65

**HÀ NỘI, 08/2023**

## LỜI CẢM ƠN

Kính gửi thầy Nguyễn Hữu Đức!

Trước tiên, em xin gửi lời cảm ơn chân thành và sâu sắc tới thầy về sự hướng dẫn, hỗ trợ của thầy tận tình của thầy cho em trong việc định hướng, tìm hiểu về chủ đề nghiên cứu tốt nghiệp 1. Trong suốt thời gian hướng dẫn, thầy đã đưa ra những bước cần thực hiện trong quá trình nghiên cứu, kịp thời giải đáp những thắc mắc, cũng như điều chỉnh những sai sót trong quá trình tìm hiểu và nghiên cứu đề tài.

Xin gửi lời cảm ơn đến các bạn đã giúp đỡ mình trong quá trình trao đổi, tìm kiếm thông tin, nội dung, cũng như cách thực hiện đề tài nghiên cứu.

Với thời gian cũng như kinh nghiệm còn hạn chế, bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong muốn nhận được sự điều chỉnh từ thầy cũng như góp ý của các bạn để hoàn thành tốt hơn trong các học phần nghiên cứu đồ án kế tiếp.

Em xin chân thành cảm ơn!

## TÓM TẮT NỘI DUNG

Nội dung nghiên cứu tốt nghiệp 1 với đề tài phân tích, lưu trữ, thu thập dữ liệu về web các thiết bị di động, tập trung vào phân tích, tìm kiếm nội dung thông tin của các thiết bị di động như điện thoại trên các sàn giao dịch điện tử. Trong báo cáo này, em sẽ tập trung thu thập thông tin trên trang web <https://www.thegioididong.com/dtdd>

Để thu thập dữ liệu web, em có tìm hiểu một số thư viện Selenium, Scrapy, ... Với ưu điểm dễ dàng tương tác, truy vấn và xử lý dữ liệu trên trang web sử dụng Javascript, kiểm thử linh hoạt, ngôn ngữ được chọn sử dụng trong nghiên cứu lần này là Python với thư viện Selenium và Pandas.

Bước đầu, tiến hành truy cập vào trang web, sử dụng các câu lệnh selenium để lấy được thông tin cơ bản từ mã nguồn hiển thị trên homepage của <https://www.thegioididong.com/dtdd> như tên (Title), hình ảnh (Image), link (URL), kích thước (Size), giá gốc (Old price), giá khuyến mãi (Price), chiều khấu giảm giá (Discount), giá đặc biệt (Special price), tổng số lượt bình luận (Total comment). Thông tin sau khi thu thập được lưu vào dataframe với cấu trúc dạng bảng (productView).

Sau đó, tiếp tục truy cập vào các URL đã thu thập trước đó, đi đến chi tiết từng trang web chứa thông tin sản phẩm. Lấy một số thông tin chi tiết bổ sung cho sản phẩm mà homepage chưa hiển thị: màu sắc (Color), cấu hình (Configuration), đánh giá trên thang điểm 5 (Rate). Bổ sung vào bảng dữ liệu đã có tạo thành một dataframe mới (Product).

Tạo dataframe mới để (productDetailComments) thu thập chi tiết các comment từng sản phẩm (tên sản phẩm (Title), người comment (Name), chứng nhận mua hàng (Confirm Buy), nội dung comment (Content), số lượt thích (Likes)) để thu thập thông tin liên quan đến comment về sản phẩm.

Sau khi thu được 2 bảng dữ liệu dataframe trên (Product, productDetailComments), ta tiến hành merge chúng lại theo tên sản phẩm để thu được dữ liệu cuối cùng (Smartphone/ Laptop...) là Smartphone, chứa tất cả dữ liệu tìm kiếm được.

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<b>2</b>
<b>TÓM TẮT NỘI DUNG .....</b>	<b>3</b>
<b>I. TÌM HIỂU CHUNG .....</b>	<b>5</b>
1. Sơ bộ về thế giới di động: .....	5
2. Nguồn thông tin từ: .....	5
3. Khuân dạng dữ liệu: .....	5
a) Homepage: .....	5
b) Thông tin chi tiết sản phẩm: .....	7
<b>II. PHƯƠNG THỨC THU THẬP .....</b>	<b>8</b>
1. Selenium: .....	8
a) Các thành phần Selenium: .....	9
b) Các câu lệnh điều khiển trong selenium: .....	9
c) Các lớp thường xuất hiện trong Selenium: .....	9
d) Phiên làm việc: .....	10
e) Phương thức lấy phần tử bằng selenium: .....	10
2. Pandas .....	13
<b>III. THỬ NGHIỆM VÀ BÁO CÁO KẾT QUẢ .....</b>	<b>14</b>
Tìm link ảnh: .....	17
Tìm size: .....	18
Kết quả: .....	27
<b>IV. THAM KHẢO: .....</b>	<b>28</b>

## I. TÌM HIỂU CHUNG

### 1. Sơ bộ về thế giới di động:

Là nơi hợp tác với nhiều hãng điện thoại nhằm cung cấp sản phẩm uy tín, độc quyền. Các sản phẩm đa dạng về mẫu mã, thương hiệu, cũng như giá cả. Ngoài ra, thegioididong còn cung cấp thêm các sản phẩm laptop, máy tính bảng, phụ kiện, smartwatch, đồng hồ, máy tính, máy in, máy cũ...

Đối tượng thông tin thu thập: điện thoại di động.

### 2. Nguồn thông tin từ:

<https://www.thegioididong.com/>

<https://viettelstore.vn/>

<https://fptshop.com.vn/>

<https://tiki.vn/>

<https://shopee.vn/>

<https://cellphones.com.vn/>

<https://hoanghamobile.com/>

Trang chứa thông tin sẽ thu thập:





















- Điện thoại di động:

<https://www.thegioididong.com/dtdd>

### 3. Khuôn dạng dữ liệu:

- a) Homepage:

Màn hình homepage hiển thị 20 sản phẩm trong lần đầu tiên truy cập.

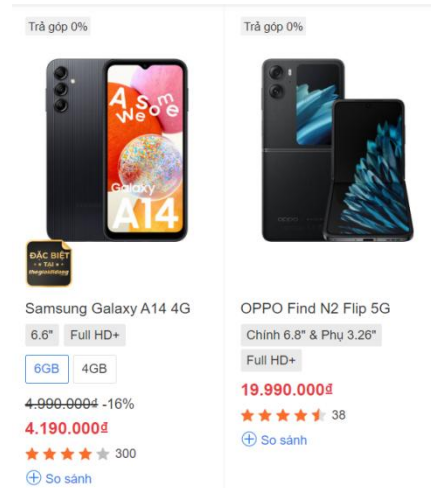
 <p>Samsung Galaxy A14 5G</p> <p>6.8" Full HD+</p> <p>8GB 4GB</p> <p><del>4.100.000đ</del> -10%</p> <p>4.100.000đ</p> <p>★★★★☆ 300</p> <p>Go sánh</p>	 <p>OPPO Find N2 Flip 5G</p> <p>Chính 6.8" &amp; Phụ 5.5"</p> <p>Full HD+</p> <p><del>12.200.000đ</del></p> <p>★★★★☆ 30</p> <p>Go sánh</p>	 <p>iPhone 14 Pro Max</p> <p>6.7" Super Retina XDR</p> <p>128GB 256GB 512GB</p> <p>1TB</p> <p><del>26.400.000đ</del> -11%</p> <p>26.400.000đ</p> <p>★★★★☆ 340</p> <p>Go sánh</p>	 <p>iPhone 14 Pro</p> <p>6.1" Super Retina XDR</p> <p>128GB 256GB 512GB</p> <p><del>24.400.000đ</del> -12%</p> <p>24.400.000đ</p> <p>Giá đặc biệt: 24.290.000đ</p> <p>★★★★☆ 100</p> <p>Go sánh</p>	 <p>Xiaomi Redmi 12</p> <p>6.7" Full HD+</p> <p>8GB 6GB</p> <p><del>3.700.000đ</del> -11%</p> <p>3.700.000đ</p> <p>★★★★☆ 400</p> <p>Go sánh</p>
 <p>Samsung Galaxy A24</p> <p>6.5" Full HD+</p> <p>8GB 6GB</p> <p><del>6.000.000đ</del> -4%</p> <p>6.000.000đ</p> <p>★★★★☆ 20</p> <p>Go sánh</p>	 <p>vivo Y30</p> <p>6.5" Full HD+ 6.8"</p> <p>128GB 256GB</p> <p><del>5.900.000đ</del> -1%</p> <p>5.900.000đ</p> <p>Go sánh</p>	 <p>Xiaomi Redmi 12C</p> <p>6.7" HD+</p> <p>8GB 128GB</p> <p>Online giá rẻ quá</p> <p><del>2.400.000đ</del> -30%</p> <p>2.400.000đ</p> <p>★★★★☆ 170</p> <p>Go sánh</p>	 <p>Samsung Galaxy S21 FE 5G</p> <p>6.4" Full HD+</p> <p>8GB - 128GB 6GB - 128GB 8GB - 256GB</p> <p><del>2.000.000đ</del> -33%</p> <p>2.000.000đ</p> <p>★★★★☆ 69</p> <p>Go sánh</p>	 <p>iPhone 11</p> <p>6.1" Liquid Retina</p> <p>64GB 128GB</p> <p><del>10.800.000đ</del> -4%</p> <p>10.800.000đ</p> <p>Giá đặc biệt: 10.690.000đ</p> <p>★★★★☆ 700</p> <p>Go sánh</p>
 <p>Xiaomi Redmi Note 12 5G</p> <p>6.67" Full HD+</p> <p>8GB - 256GB 6GB - 128GB 8GB - 128GB</p> <p><del>5.000.000đ</del> -7%</p> <p>5.000.000đ</p> <p>★★★★☆ 10</p> <p>Go sánh</p>	 <p>vivo Y02T</p> <p>6.51" HD+</p> <p>3.200.000đ</p> <p>★★★★☆ 30</p> <p>Go sánh</p>	 <p>OPPO Reno8 T 5G</p> <p>6.7" Full HD+</p> <p>8GB - 256GB 8GB - 128GB</p> <p><del>10.000.000đ</del></p> <p>10.000.000đ</p> <p>★★★★☆ 60</p> <p>Go sánh</p>	 <p>Realme C55</p> <p>6.72" Full HD+</p> <p>8GB - 128GB 6GB - 256GB</p> <p><del>4.700.000đ</del> -1%</p> <p>4.700.000đ</p> <p>Giá đặc biệt: 4.290.000đ</p> <p>★★★★☆ 100</p> <p>Go sánh</p>	 <p>Samsung Galaxy S23 Ultra 5G</p> <p>6.8" Quad HD+ (2640x1080)</p> <p>8GB - 256GB 12GB - 512GB</p> <p><del>25.000.000đ</del> -10%</p> <p>25.000.000đ</p> <p>Giá rẻ quá: 24.290.000đ</p> <p>★★★★☆ 10</p> <p>Go sánh</p>
 <p>Samsung Galaxy A34 5G</p> <p>6.8" Full HD+</p> <p>128GB 256GB</p> <p><del>7.400.000đ</del> -11%</p> <p>7.400.000đ</p> <p>Giá rẻ quá: 6.990.000đ</p> <p>★★★★☆ 240</p> <p>Go sánh</p>	 <p>iPhone 14</p> <p>6.1" Super Retina XDR</p> <p>128GB 256GB</p> <p><del>10.100.000đ</del> -12%</p> <p>10.100.000đ</p> <p>Giá đặc biệt: 9.080.000đ</p> <p>★★★★☆ 180</p> <p>Go sánh</p>	 <p>Vivo Y16</p> <p>6.51" HD+</p> <p>8GB 128GB</p> <p>Online giá rẻ quá</p> <p><del>3.000.000đ</del> -15%</p> <p>3.000.000đ</p> <p>★★★★☆ 50</p> <p>Go sánh</p>	 <p>Realme C60x</p> <p>6.51" HD+</p> <p>3GB 4GB 3GB</p> <p><del>2.390.000đ</del> -1%</p> <p>2.390.000đ</p> <p>Giá rẻ quá: 1.790.000đ</p> <p>★★★★☆ 77</p> <p>Go sánh</p>	 <p>Samsung Galaxy S23+ 5G</p> <p>6.6" Full HD+</p> <p>8GB - 256GB 8GB - 512GB</p> <p><del>23.990.000đ</del> -11%</p> <p>23.990.000đ</p> <p>★★★★☆ 10</p> <p>Go sánh</p>

Muốn xem tất cả điện thoại đang có, click vào “Xem thêm”, sau mỗi lần click, màn hình hiển thị thêm 20 sản phẩm.

Xem thêm 83 Điện thoại ▾

Tại homepage, thông tin sản phẩm gồm có:

- Title (tên sản phẩm)
- Image (hình ảnh)
- URL (link sản phẩm)
- Size (kích thước)
- Storage (Dung lượng)
- Price (giá bán)
- Old Price (giá gốc - có hoặc không)
- Discount (giảm giá - có hoặc không)
- Special Price (giá đặc biệt - có hoặc không)
- Total comment (tổng số comment - có hoặc không)



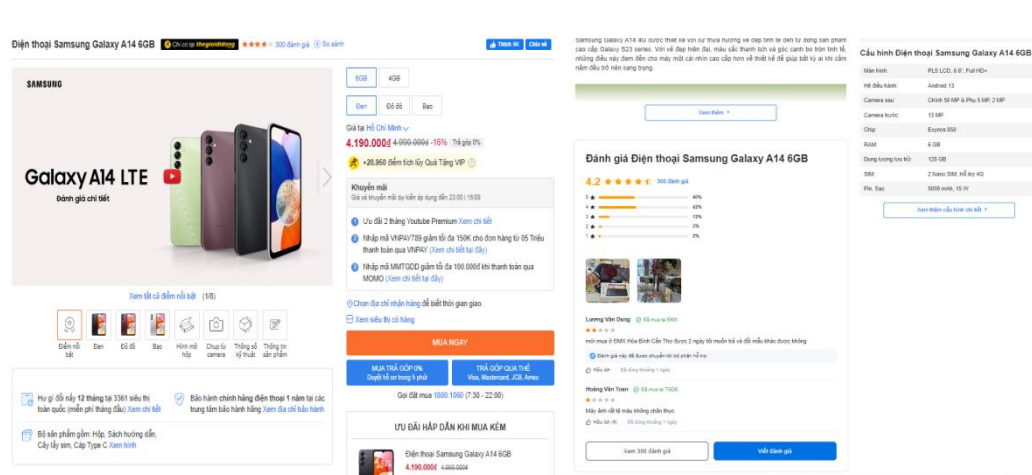
```
productView = pd.DataFrame(list(zip(productTitles, productImages,
productLinks, productSizes, productStorage, productPrice, productPercent,
productOldPrice, productSpecialPrice, totalComments)),
columns=["Title", "Image", "URL", "Size", "Storage", "Price", "Discount", "Old Price",
"Special Price", "Total Comments"])
```

→Tại sao không dùng ID:

Trong mã nguồn của web thegioididong, ID được đánh số từ 1-20 cho 20 sản phẩm đầu tiên, sau khi mở rộng dữ liệu bằng cách click vào “Xem thêm”, các sản phẩm tiếp theo được đánh ID lại từ 1 - 20. Do ID bị lặp lại nên không sử dụng ID để phân biệt các sản phẩm.

Trong chi tiết các sản phẩm, bổ sung thêm một số hình ảnh và thông tin cấu hình, đánh giá, bình luận.

b) Thông tin chi tiết sản phẩm:



Khuôn dạng dữ liệu thông tin bổ sung:

- Color (màu sắc)
- Rate (đánh giá)
- Configuration (cấu hình)

```
detailProduct = pd.DataFrame(list(zip(productColor, productRate,
productConfiguration)), columns= ["Color", "Rate", "Configuration"])
```

Khuôn dạng dữ liệu comment chi tiết:

- Title
- Name
- Confirm Buy
- Content
- Likes

```
productDetailComments = pd.DataFrame(list(zip(cmtProduct, cmtNames,
confirmBuy, cmtContents, cmtLikes)), columns=["Title", "Name", "Confirm Buy",
"Content", "Likes"])
```

## II. PHƯƠNG THỨC THU THẬP

Sử dụng thư viện Selenium và Pandas kết hợp với API.

### 1. Selenium:

Selenium là công cụ kiểm thử tự động dành cho nhiều trình duyệt. Selenium hỗ trợ nhiều ngôn ngữ khác nhau như Python, Javascript, Java, C#, Ruby...



a) Các thành phần Selenium:

- Selenium WebDriver: cung cấp giao diện lập trình để xây dựng các chương trình kiểm thử tự động và tự động hóa

Các ràng buộc Selenium Python cung cấp API đơn giản để viết bài kiểm tra chức năng/ chấp nhận, truy cập tất cả các chức năng 1 cách trực quan thông qua Selenium Web Driver.

- Selenium IDE: Là một plugin dành cho trình duyệt cho phép người dùng ghi lại và phát lại các thao tác tương tác với trang web, thường được áp dụng cho các kiểm thử đơn giản.
- Selenium Grid: là công cụ cho phép thực hiện kiểm thử đa hình trình duyệt và đa nền tảng song song trên nhiều máy, nhiều trình duyệt cùng một lúc.

Trong nghiên cứu lần này, công cụ sử dụng là Selenium WebDriver với trình duyệt Chrome thông qua trình điều khiển Chrome Driver. Trình điều khiển sẽ ủy quyền cho trình duyệt và xử lý thông điệp giao tiếp đến từ Selenium.

b) Các câu lệnh điều khiển trong selenium:

- Đầu tiên, cần phải cài đặt môi trường làm việc với Python (trong bài sử dụng IDE Pycharm).
- Cài thư viện Selenium: `pip install selenium`
- Tải Chrome Driver từ <https://sites.google.com/chromium.org/driver/>

c) Các lớp thường xuất hiện trong Selenium:

- WebDriver: lớp chính tạo và quản lý phiên làm việc với trình duyệt web.

```
from selenium import webdriver
```

- By: chứa các phương thức tìm kiếm, xác định phần tử trang web (element/elements) dựa trên ID, NAME, TAG\_NAME, CLASS\_NAME, CSS\_SELECTOR, XPATH, LINK\_TEXT, PARTIAL\_LINK\_TEXT.

```
from selenium.webdriver.common.by import By
```

- Keys: chứa các hằng số đại diện cho các phím trên bàn phím.

```
from selenium.webdriver.common.keys import Keys
```

- Select: được sử dụng để tương tác với các phần tử select, giúp tùy chọn từ danh sách và thực hiện các thao tác liên quan đến ô danh sách: click(), move\_to\_element(), release()...

```
from selenium.webdriver.support.ui import Select
```

- WebDriverWait: cho phép chờ đợi các lệnh điều kiện nhất định được thực hiện trước khi chuyển sang tác vụ khác.
- ActionChains: cho phép thực hiện hành động tương tác phức tạp.

- Các lớp xử lý ngoại lệ (exceptions):

NoSuchElementException: ngoại lệ khi không tìm được phần tử tương ứng

NoSuchAttributeException: ngoại lệ khi attribute không tìm thấy trong element

ElementClickInterceptedException: không lệnh click không thực hiện được

ElementNotInteractableException: không tương tác được

ElementClickSelectableException: không chọn được...

```
from selenium.common.exceptions import NoSuchElementException, NoSuchElementException,  
ElementClickInterceptedException
```

#### d) Phiên làm việc:

Khởi tạo một phiên làm việc với trình duyệt web (Chrome) thông qua webdriver và đường dẫn đến tệp thực thi của trình duyệt executable\_path (Chrome Driver). Đường dẫn được sử dụng trong chương trình này là D:\chromedriver.

```
driver = webdriver.Chrome(executable_path='D:\chromedriver')
```

Ủy quyền điều khiển truy cập trang web (thegioidadong.com) cho driver. Mở trang theo đường link url đã cung cấp.

```
driver.get(url)
```

```
driver.get('https://www.thegioidadong.com')
```

#### e) Phương thức lấy phần tử bằng selenium:

- Find\_element():

Trả về giá trị đầu tiên trùng khớp với dữ liệu tìm kiếm. Nếu không tìm thấy kết quả nào phù hợp, chương trình sẽ đưa ra cảnh báo lỗi.

- Find\_elements():

Trả về tất cả các phần tử trùng khớp với dữ liệu tìm kiếm. Nếu không tìm thấy kết quả nào phù hợp, chương trình sẽ đưa ra cảnh báo lỗi. Phương thức này thường được kết hợp với By để tìm kiếm phần tử trong DOM.

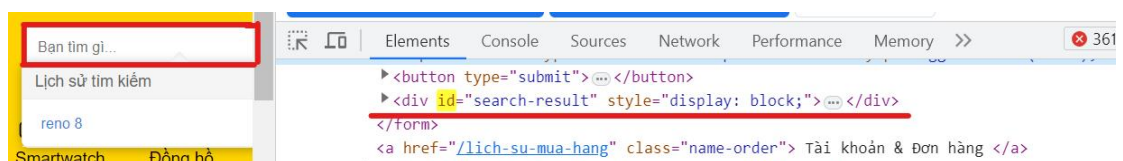
Ngoài ra, chúng ta có thể tìm phần tử con từ phần tử gốc. (Find element/elements from element). Không nhất thiết phần tử gốc phải là driver.

### Cách định vị phần tử truyền thống:

Locator	Mô tả
ID	Định vị các phần tử có thuộc tính ID khớp với giá trị tìm kiếm
NAME	Định vị các phần tử có thuộc tính NAME khớp với giá trị tìm kiếm
CLASS_NAME	Định vị các phần tử có tên lớp chứa giá trị tìm kiếm (không được đặt tên lớp ghép)
TAG_NAME	Định vị các phần tử có tên thẻ khớp với giá trị tìm kiếm: [a], [href], [h1], [h2], [h3], [strong]...
CSS_SELECTOR	Định vị các phần tử khớp với bộ chọn CSS
XPAT	Định vị các phần tử khớp với biểu thức XPath
LINK_TEXT	Định vị các phần tử neo có văn bản hiển thị khớp với giá trị tìm kiếm
PARTIAL_LINK_TEXT	Định vị các phần tử neo có văn bản hiển thị chứa giá trị tìm kiếm. Nếu nhiều phần tử phù hợp, chỉ phần tử đầu tiên sẽ được chọn.

Ví dụ:

- By.ID: Tìm theo ID phần tử



Đây là phần tử chứa attribute id đầu tiên trong DOM nên ta có thể sử dụng truy vấn:

```
Search_result = driver.find_element(By.ID, "search_result")
```

sẽ trả về phần tử ô dữ liệu “Bạn tìm gì...”

- By.CLASS\_NAME: tìm theo tên lớp

Vd: lấy phần tử danh sách các sản phẩm (điện thoại) listProducts

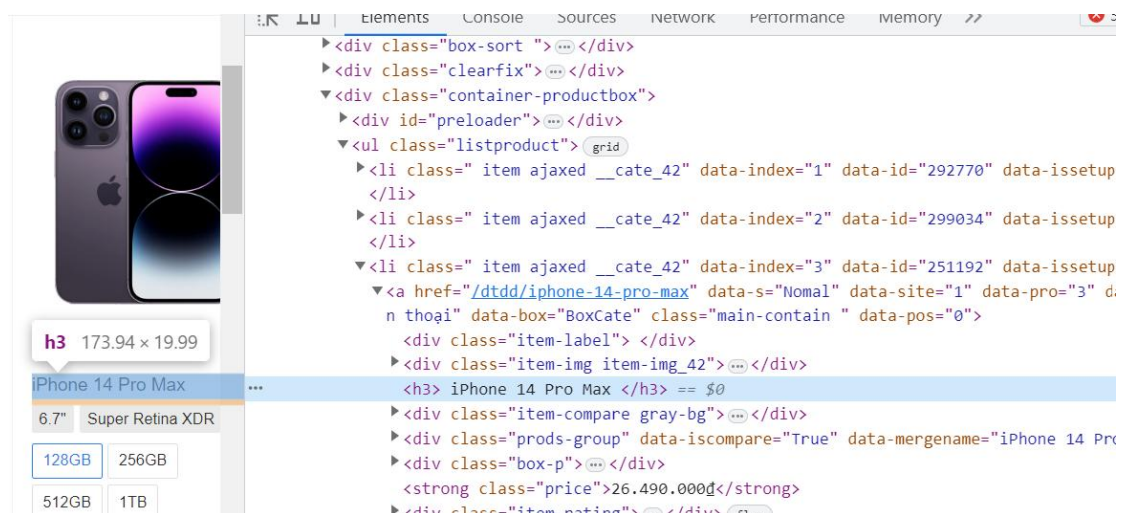


Câu lệnh:

```
listProducts = driver.find_element(By.CLASS_NAME, 'listproduct')
```

listProducts bao gồm toàn bộ danh sách nên nó chỉ là element duy nhất.

- By.TAG\_NAME



Giả sử muốn lấy danh sách tên các điện thoại, mỗi điện thoại sẽ nằm trong 1 thẻ li, class item-img\_42, tên nằm trong thẻ h3, ta có thể sử dụng câu lệnh:

```
titles = listProducts.find_elements(By.TAG_NAME, 'h3')
productTitles = [title.text for title in titles]
```

- By.CSS\_SELECTOR

Cũng là truy vấn tên sản phẩm, nếu muốn lấy tên “iphone 14 Pro Max”, có thể sử dụng css\_selector của phần tử

#categoryPage > div.container-productbox > ul > li:nth-child(3) > a.main-contain > h3

```
title = listProducts.find_element(By.CSS_SELECTOR, "#categoryPage > div.container-productbox > ul > li:nth-child(3) > a.main-contain > h3").text
```

- By.XPATH:

Tương tự, “iphone 14 Pro Max” có XPATH là:

/html/body/div[6]/section/div[3]/ul/li[3]/a[1]/h3

```
title=listProducts.find_element(By.XPATH, "/html/body/div[6]/section/div[3]/ul/li[3]/a[1]/h3").text
```

Định vị phần tử theo vị trí tương đối:

Định vị vị trí tương đối của các phần tử sẽ định vị phần tử cần tìm dựa trên vị trí tương đối so với một phần tử có vị trí dễ xác định.

Vị trí tương đối	Mô tả
Above	Phần tử cần tìm nằm trên phần tử để định vị.
Below	Phần tử cần tìm nằm dưới phần tử để định vị.
Left of	Phần tử cần tìm nằm bên trái phần tử để định vị.
Right of	Phần tử cần tìm nằm bên phải phần tử để định vị.
Near	Xác định phần tử nằm gần phần tử để định vị trong phạm vi 50px.

Sau khi thực hiện các thao tác tìm kiếm, muốn kết thúc phiên làm việc, có thể sử dụng:

driver.quit(): kết thúc tất cả phiên làm việc với selenium, đóng trình duyệt.

driver.close(): kết thúc phiên làm việc hiện tại, đóng cửa sổ trình duyệt hiện tại.

## 2. Pandas

Pandas là thư viện mã nguồn mở của Python, hỗ trợ phân tích và dữ liệu dữ liệu cơ bản. Pandas linh hoạt trong thao tác dữ liệu và lập chỉ mục, cho phép đọc, ghi dữ liệu chủ yếu dạng bảng ở nhiều định dạng khác nhau: sql

database, csv, excel... Dễ dàng tái cấu trúc bố cục dữ liệu, tự động đưa dữ liệu về dạng cấu trúc, dễ dàng thêm, sửa xóa các cột dữ liệu, vì vậy tối ưu về hiệu năng sử dụng.

Import pandas:

```
import pandas as pd
```

a) Cấu trúc dữ liệu thường thấy trong pandas:

i) Series:

Cấu trúc dữ liệu một chiều chứa dữ liệu thuộc bất kì loại nào, giống như một cột trong bảng. Có thể truy cập Pandas series giống như truy cập mảng.

ii) DataFrame:

Cấu trúc dữ liệu hai chiều, giống như mảng hai chiều, bao gồm các hàng và cột.

Các thao tác có thể sử dụng với DataFrame:

Tạo dataframe, thêm, xóa cột dữ liệu mới, gộp dữ liệu, thống kê, tính toán, đọc và xuất dữ liệu, sắp xếp dữ liệu, khám phá dữ liệu, xử lý dữ liệu thiếu,...

Ngoài ra Pandas cho phép nhập dữ liệu từ nhiều nguồn khác nhau như sql database, json, csv, excel...

Trong báo cáo lần này, em sẽ sử dụng dataframe để tổng hợp dữ liệu thu thập được.

### III. THỬ NGHIỆM VÀ BÁO CÁO KẾT QUẢ

Mã nguồn: <https://github.com/TrangPC/NCTN1>

Import các thư viện cần thiết:

```
from selenium import webdriver
import time
import numpy as np
import pandas as pd
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.keys import Keys
from selenium.common.exceptions import NoSuchElementException,
NoSuchAttributeException, ElementClickInterceptedException
```

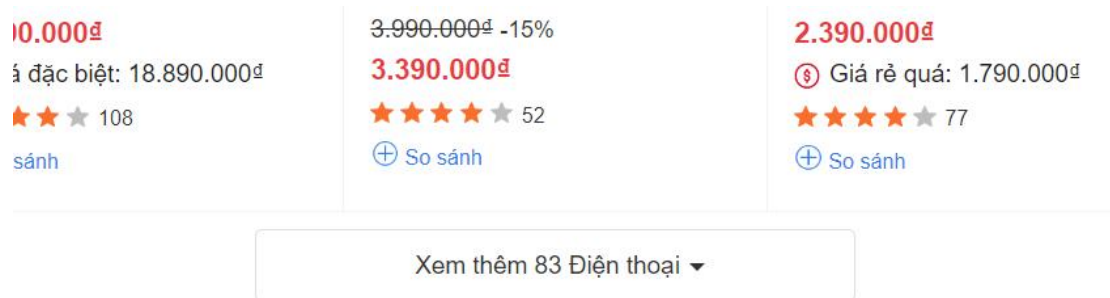
Chọn API:

```
driver = webdriver.Chrome(executable_path='D:\chromedriver')
```

Chọn trang web cần lấy dữ liệu:

```
driver.get('https://www.thegioididong.com/dtdd')
```

Load toàn bộ dữ liệu bằng cách điều khiển chương trình ấn vào “Xem thêm”.



Mỗi lần load được 20 sản phẩm, lần load cuối, số lượng máy còn lại sẽ nhỏ hơn hoặc bằng 20.

Ta có vòng lặp:

```
loop = True
while loop:
    try:
        viewMore = driver.find_element(By.CLASS_NAME, 'view-more')
        view = viewMore.find_element(By.TAG_NAME, 'a')
        remain = viewMore.find_element(By.CLASS_NAME, 'remain').text
        if int(remain) >= 20:
            view.click()
        else:
            view.click()
            loop = False

        print(remain)

    except ElementClickInterceptedException:
        print('Don\'t load data')
        break
    except NoSuchElementException:
        print('Don\'t load data')
        break
```

Điều kiện lặp sẽ đúng nếu số lượng máy còn lại lớn hơn 20 (loop = True).

Ngược lại, nếu số máy còn lại nhỏ hơn hoặc bằng 20, nó chỉ cần load thêm đúng 1 lần cuối cùng rồi thoát khỏi vòng lặp (loop = False).

Để tránh xảy ra lỗi không tìm thấy Class “view-more”, sử dụng NoSuchElementException để thông báo ra màn hình thoát, và không click được vào nút “Xem thêm”, sử dụng ElementClickInterceptedException để thông báo ra màn hình và thoát.

**Kết quả:**

```
83
63
43
23
3
|
Process finished with exit code 0
```

M239 4G

QQVGA

34 -22%

10đ

★ ★ 125

ảnh

Masstel IZI 10 4G

1.77" QVGA

370.000đ

★★★★★ 176

So sánh

Bạn có hài lòng với trải nghiệm tìm kiếm thông tin, sản phẩm trên website không?

Hài lòng

Không hài lòng

Bắt đầu thu thập dữ liệu các sản phẩm từ homepage.

**Dữ liệu mẫu:**

Trả góp 0%

Samsung Galaxy A14 4G

6.6" Full HD+

6GB 4GB

4.990.000đ -16%

4.190.000đ

★★★★★ 301

So sánh

Trả góp 0%

OPPO Find N2 Flip 5G

Chính 6.8" & Phụ 3.26" Full HD+

19.990.000đ

★★★★★ 38

So sánh

iPhone 14 Pro Max

6.7" Super Retina XDR

128GB 256GB 512GB

1TB

29.990.000đ -11%

26.490.000đ

★★★★★ 345

So sánh

iPhone 14 Pro

6.1" Super Retina XDR

128GB 256GB 512GB

27.990.000đ -12%

24.490.000đ

Giá đặc biệt: 24.290.000đ

★★★★★ 165

So sánh

Trả góp 0%

Xiaomi Redmi 12

6.73" Full HD+

4GB 8GB

4.290.000đ -6%

3.990.000đ

★★★★★ 433

So sánh

● **Tìm tên:**

Mã nguồn:

<h3> Samsung Galaxy A14 4G </h3>

Kết quả:



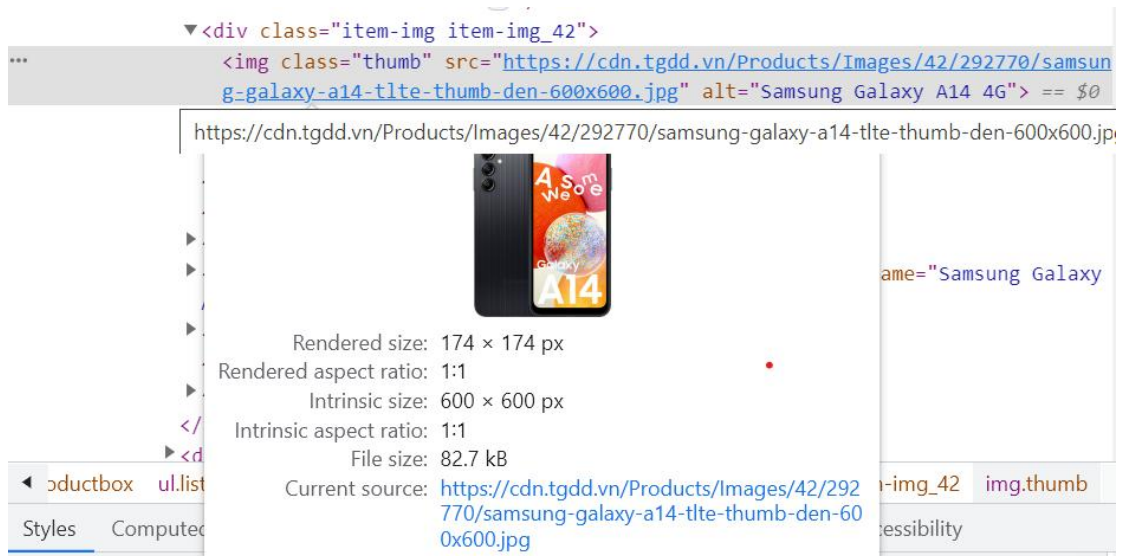
```
36 listProducts = driver.find_element(By.CLASS_NAME, 'listproduct')
37
38 titles = listProducts.find_elements(By.TAG_NAME, 'h3')
39 productTitles = [title.text for title in titles]
40
41 # imgsClass = listProducts.find_elements(By.CLASS_NAME, 'item-img_42')
```

Run copyCrawlTGDD

```
D:\PYCHARM\venv\Scripts\python.exe D:\PYCHARM\copyCrawlTGDD.py
D:\PYCHARM\copyCrawlTGDD.py:10: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
  driver = webdriver.Chrome(executable_path='D:\chromedriver')
['Samsung Galaxy A14 4G', 'OPPO Find N2 Flip 5G', 'iPhone 14 Pro Max', 'iPhone 14 Pro', 'Xiaomi Redmi 12', 'Samsung Galaxy A24', 'vivo Y36', 'Xiaomi Redmi 12 5G']
```

## ● Tìm link ảnh:

Mã nguồn:



Mỗi một sản phẩm có 1 link ảnh nằm trong class item-img\_42, thẻ img, attribute src.

Kết quả:

```
41 imgsClass = listProducts.find_elements(By.CLASS_NAME, 'item-img_42')
42 productImages = []
43 for i in range(0, len(imgsClass)):
44     try:
45         img = imgsClass[i].find_element(By.CSS_SELECTOR, 'img')
46         productImages.append(img.get_attribute('src'))
47     except NoSuchElementException:
48         productImages.append("NULL")
49
```

Run copyCrawlTGDD

```
D:\PYCHARM\venv\Scripts\python.exe D:\PYCHARM\copyCrawlTGDD.py
D:\PYCHARM\copyCrawlTGDD.py:10: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
  driver = webdriver.Chrome(executable_path='D:\chromedriver')
['https://cdn.tgdd.vn/Products/Images/42/292770/samsung-galaxy-a14-tlte-thumb-den-600x600.jpg', 'https://cdn.tgdd.vn/Products/Images/42/299034/oppo-find-n2-flip-5g-thumb-den-600x600.jpg']
```

## ● Tìm link sản phẩm cụ thể:

Mã nguồn:

```

▼<li class=" item ajaxed __cate_42" data-index="1" data-id="292770" data-issetup=
"0" data-maingroup="13" data-subgroup="1491" data-type="1" data-vehicle="1" data-
productcode="0131491003651" data-price="4990000.0" data-ordertypeid="2" data-pos=
"1">
...
▶<a href="/dtdd/samsung-galaxy-a14" data-s="Nomal" data-site="1" data-pro="3"
data-cache="True" data-sv="webtgdd-26-122" data-name="Điện thoại Samsung Galaxy
A14 6GB" data-id="292770" data-price="4190000.0" data-brand="Samsung" data-
cate="Điện thoại" data-box="BoxCate" class="main-contain " data-pos="0"> ...
</a> == $0

```

Link sản phẩm có trong href thẻ a. Thẻ a nằm trong thẻ li, được xác định bằng cách tìm tên sản phẩm vì mỗi link href có 1 tên để xác định.

Kết quả:

```

53 ts.find_elements(By.CSS_SELECTOR, 'li [data-name]')
54 nk.get_attribute('href') for link in links]
55
56
Run copyCrawlTGDD x
D:\PYCHARM\venv\Scripts\python.exe D:\PYCHARM\copyCrawlTGDD.py
D:\PYCHARM\copyCrawlTGDD.py:10: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
driver = webdriver.Chrome(executable_path='D:\chromedriver')
103
['https://www.thegioididong.com/dtdd/samsung-galaxy-a14', 'https://www.thegioididong.com/dtdd/oppo-find-n2-flip', 'https://www.thegioididong.com/']

```

## ● Tìm size:

Mã nguồn:

```

... ▼<div class="item-compare gray-bg"> == $0
    <span>6.6</span>
    <span>Full HD</span>
</div>

49 try:
50     productSizes = []
51     for i in range(0, len(productTitles)):
52         size = listProducts.find_element(By.CLASS_NAME, '__cate_42')
53         sizes = size.find_element(By.CLASS_NAME, 'gray-bg')
54         productSizes.append(size.text)
55 except NoSuchElementException:
56     productSizes.append("NULL")
57
58
Run copyCrawlTGDD x
D:\PYCHARM\venv\Scripts\python.exe D:\PYCHARM\copyCrawlTGDD.py
D:\PYCHARM\copyCrawlTGDD.py:10: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
driver = webdriver.Chrome(executable_path='D:\chromedriver')
103
['Trà góp 0%\nSamsung Galaxy A14 4G\nó.6" Full HD+\nó6B\n46B\n4.990.000đ -16%\n4.190.000đ\n301\nSo sánh', 'Trà góp 0%\nSamsung Galaxy A14 4G\nó.6" Full HD+\nó6B\n46B\n4.990.000đ -16%\n4.190.000đ\n301\nSo sánh']

```

Do một số sản phẩm không có size như hình dưới đây, nên size của chúng sẽ được tính là NULL.

## ● Tìm giá bán:

Mã nguồn:

```

... ▼<strong class="price">4.490.000đ</strong> == $0


```

Chỉ có duy nhất giá bán là được in đậm (strong) nên em sẽ tìm theo TAG\_NAME là “strong”.

```

63 productOldPrice,productPercent, productPrice, productSpecialPrice, productStorage = [], [], [], [], []
64
65 for i in range(0, len(productTitles)):
66     try:
67         price = links[i].find_element(By.TAG_NAME, 'strong')
68         productPrice.append(price.text)
69     except NoSuchElementException:
70         productPrice.append("NULL")
71
72

```

Run  copyCrawlTGDD x

```

D:\PYCHARM\venv\Scripts\python.exe D:\PYCHARM\copyCrawlTGDD.py
D:\PYCHARM\copyCrawlTGDD.py:10: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
driver = webdriver.Chrome(executable_path='D:\chromedriver')
103
['4.190.000đ', '19.990.000đ', '26.490.000đ', '24.490.000đ', '3.990.000đ', '6.090.000đ', '5.990.000đ', '2.590.000đ', '9.990.000đ', '10.890.000đ']

```

### Kết quả:

```
95 for i in range(0, len(productTitles)):
96     try:
97         specialPrice = links[i].find_element(By.CLASS_NAME, 'fightprice')
98         productSpecialPrice.append(specialPrice.text)
99     except NoSuchElementException:
100         productSpecialPrice.append('NULL')
101 #
102 #
103
```

Run copyCrawlTGDD

```
D:\PYCHARM\venv\Scripts\python.exe D:\PYCHARM\copyCrawlTGDD.py
D:\PYCHARM\copyCrawlTGDD.py:10: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
driver = webdriver.Chrome(executable_path='D:\chromedriver')
103
['NULL', 'NULL', 'NULL', 'Giá đặc biệt: 24.290.000đ', 'NULL', 'NULL', 'NULL', 'NULL', 'NULL', 'Giá đặc biệt: 10.490.000đ', 'NULL', 'NULL', 'NULL']
```

## ● Tìm dung lượng:

Mã nguồn:

```
<div class="prods-group" data-iscompare="True" data-mergename="Samsung Galaxy A14 4G" data-lstarranged="292770,303579">
  <ul> flex == $0
    <li data-url="/dtdd/samsung-galaxy-a14" data-id="292770" data-index="0" class="merge__item item act">6GB</li> flex
    <li data-url="/" data-id="303579" data-index="1" class="merge__item item">4GB</li> flex
  </ul>
```

Một số sản phẩm không ghi dung lượng.

Kết quả:

```
88 for i in range(0, len(productTitles)):
89     try:
90         storage = links[i].find_element(By.CLASS_NAME, 'prods-group')
91         productStorage.append(storage.text)
92     except NoSuchElementException:
93         productStorage.append('NULL')
94 #
95 # for i in range(0, len(productTitles)):

```

Run copyCrawlTGDD

```
D:\PYCHARM\venv\Scripts\python.exe D:\PYCHARM\copyCrawlTGDD.py
D:\PYCHARM\copyCrawlTGDD.py:10: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
driver = webdriver.Chrome(executable_path='D:\chromedriver')
103
['6GB\n4GB', 'NULL', '12GB\n256GB\n512GB\n1TB', '12GB\n256GB\n512GB', '4GB\n8GB', '6GB\n8GB', '12GB\n256GB', '64GB\n128GB', '6GB - 128GB\n8GB']
```

## ● Tìm số lượng bình luận sản phẩm:

```
<div class="item-rating"> flex
  <p> ... </p>
  <p class="item-rating-total">301</p> == $0
</div>
```

Sản phẩm chưa ai bình luận sẽ hiển thị NULL.

Kết quả:

```
totalComments = []
for i in range(0, len(productTitles)):
    try:
        comments = links[i].find_element(By.CLASS_NAME, 'item-rating-total')
        totalComments.append(comments.text)
    except NoSuchElementException:
        totalComments.append('NULL')
```

copyCrawlTGDD x

0:\PYCHARM\copyCrawlTGDD.py:10: DeprecationWarning: executable\_path has been deprecated, please pass in a Service object  
driver = webdriver.Chrome(executable\_path='D:\chromedriver')

103  
['301', '38', '345', '165', '433', '28', 'NULL', '174', '89', '782', '14', '36', '80', '154', '61', '242', '108', '52', '77', '43', '92', '43',

Như vậy từ trang homepage đầu tiên, ta có thể thu thập được một số thông tin sản phẩm như trên.

Sở dĩ một số thuộc tính khuyết thiếu sẽ được thêm thành NULL để khi gộp cột dữ liệu, những ô khuyết thiếu đó sẽ không ảnh hưởng đến thứ tự sắp xếp của sản phẩm.

Ta gộp các trường dữ liệu lại với nhau thành một dataframe:

```
productView = pd.DataFrame(list(zip(productTitles, productImages,
productLinks, productSizes, productStorage, productPrice, productPercent,
productOldPrice, productSpecialPrice, totalComments)),
columns=["Title", "Image", "URL", "Size", "Storage", "Price", "Discount", "Old Price",
"Special Price", "Total Comments"])
productView.info()
```

Thông tin dữ liệu thu thập được:

#	Column	Non-Null Count	Dtype
0	Title	103 non-null	object
1	Image	103 non-null	object
2	URL	103 non-null	object
3	Size	103 non-null	object
4	Storage	103 non-null	object
5	Price	103 non-null	object
6	Discount	103 non-null	object
7	Old Price	103 non-null	object
8	Special Price	103 non-null	object
9	Total Comments	103 non-null	object

Sau khi thu thập xong ở homepage, tiến hành tìm kiếm thông tin ở từng trang sản phẩm.



Do đã có danh sách links các sản phẩm ở trên, em tiến hành duyệt lần lượt từng trang để thu thập thêm thông tin.

```
productColor, productRate, productConfiguration = [], [], []
for i in range(0, len(productTitles)):
    driver.get(productLinks[i])
```

Tuy nhiên, do lượng thông tin lớn, trong báo cáo này, em sẽ xét 1 links đầu tiên làm mẫu.

Link trang web mẫu: productLinks[0]

<https://www.thegioididong.com/dtdd/samsung-galaxy-a14>

Em sẽ lấy được thêm thông tin: màu sắc, điểm đánh giá, và mô tả sản phẩm.

```
try:
    color = driver.find_element(By.CLASS_NAME, 'box03.color.group.desk').text
    productColor.append(color)
except NoSuchElementException:
    productColor.append("NULL")
point = driver.find_element(By.CLASS_NAME, 'point')
rate = point.find_element(By.TAG_NAME, 'p').text
productRate.append(rate)

parameters = driver.find_element(By.CLASS_NAME, 'parameter')
productConfiguration.append(parameters.text)

print(productColor)
print(productRate)
print(parameters.text)
```

Kết quả: [Màu sắc][Điểm đánh giá] Cấu hình.

```
['Đen Đỏ đô Bạc']
['4.1']
Màn hình:
PLS LCD 6.6" Full HD+
Hệ điều hành:
Android 13
Camera sau:
Chính 50 MP & Phụ 5 MP, 2 MP
Camera trước:
13 MP
Chip:
Exynos 850
RAM:
6 GB
Dung lượng lưu trữ:
128 GB
SIM:
2 Nano SIM Hỗ trợ 4G
Pin, Sạc:
5000 mAh 15 W
```

Các thông tin này được lưu vào dataframe mới, sau khi hoàn thiện sẽ được nối với dataframe tại homepage.

```
detailProduct = pd.DataFrame(list(zip(productColor, productRate,
productConfiguration)), columns= ["Color", "Rate", "Configuration"])
detailProduct.info()

Product = pd.concat([productView, detailProduct], axis=1)
Product.info()
```

Các cột thông tin từ bảng dataframe detailProduct sẽ được thêm vào bên trái dataframe productView để tạo thành dataframe mới là Product.


Axis = 1, tức là ghép theo hàng, cột mới được thêm vào bên trái các cột cũ.




Kết quả:

```
dtypes: object(3)
memory usage: 152.0+ bytes
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103 entries, 0 to 102
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Title                 103 non-null   object
1   Image                 103 non-null   object
2   URL                   103 non-null   object
3   Size                  103 non-null   object
4   Storage               103 non-null   object
5   Price                 103 non-null   object
6   Discount              103 non-null   object
7   Old Price             103 non-null   object
8   Special Price         103 non-null   object
9   Total Comments       103 non-null   object
10  Color                 1 non-null     object
11  Rate                  1 non-null     object
12  Configuration         1 non-null     object
```

Thông tin sản phẩm 1:

```
168 print(Product.head(1).T)
169 #
```

Run  copyCrawlTGDD x

```
driver = webdriver.Chrome(executable_path= D:\chromedriver )
```

↑		0
↓	Title	Samsung Galaxy A14 4G
↔	Image	<a href="https://cdn.tgdd.vn/Products/Images/42/292770/...">https://cdn.tgdd.vn/Products/Images/42/292770/...</a>
⇅	URL	<a href="https://www.thegioididong.com/dtdd/samsung-gal...">https://www.thegioididong.com/dtdd/samsung-gal...</a>
🖨	Size	Trả góp 0%\nSamsung Galaxy A14 4G\n6.6" Full...
🗑	Storage	6GB\n4GB
	Price	4.190.000đ
	Discount	-16%
	Old Price	4.990.000đ
	Special Price	NULL
	Total Comments	301
	Color	Đen Đỏ đô Bạc
	Rate	4.1
	Configuration	Màn hình:\nPLS LCD 6.6" Full HD+\nHệ điều hành...

Bước tiếp theo là lấy các comment sản phẩm.

Trang web chỉ hiển thị 2 comment gần nhất nên muốn xem toàn bộ phải click vào “Xem 301 đánh giá”.



Mã nguồn:

```
▼<div class="box-flex"> flex
  <a href="/dtdd/samsung-galaxy-a14/danh-gia" class="c-btn-rate bt
n-view-all">Xem 301 đánh giá</a> == $0
```



```
try:
    more = driver.find_element(By.CLASS_NAME, 'box-flex')
    cmtLink = more.find_element(By.TAG_NAME, 'a')
    cmtLink.click()
```

Sau khi click sẽ hiển thị toàn bộ comments.

Nếu click lỗi, tức là sản phẩm chưa có comment nào.

```
except ElementClickInterceptedException:
    cmtNames, confirmBuy, cmtContents, cmtLikes = [], [], [], []
```

Nếu click thành công, tiến hành thu thập các thông tin liên quan đến comment: Người comment, đã chứng nhận mua hàng hay chưa, nội dung comment và số lượt thích comment.

Mã nguồn:

```
<li id="r-55145171" class="par">
  <div class="cmt-top">
    <p class="cmt-top-name">Hoàng văn toan</p>
    <div class="confirm-buy">
      <i class="iconcmt-confirm"></i>
      " Đã mua tại TGDĐ "
    </div>
  </div>
  <div class="cmt-intro">...</div> flex
  <div class="cmt-content">
    <p class="cmt-txt">Máy ảnh rất tệ màu không chân thực</p>
  </div>
  <div class="cmt-command"> == $0
    <a href="javascript:likeRating(55145171);" class="cmt1 dot-circle-ava" data-like="5">...</a>
    <span class="cmt1 dot-line">...</span>
  </div>
</li>
```

```

try:
    more = driver.find_element(By.CLASS_NAME, 'box-flex')
    cmtLink = more.find_element(By.TAG_NAME, 'a')
    cmtLink.click()

    listComments = driver.find_element(By.CLASS_NAME, "comment-list")

    names = listComments.find_elements(By.CLASS_NAME, 'cmt-top-name')
    cmtNames = [cmtname.text for cmtname in names]

    buy = listComments.find_elements(By.CLASS_NAME, 'confirm-buy')
    confirmBuy = [cfBuy.text for cfBuy in buy]

    contents = listComments.find_elements(By.CLASS_NAME, 'cmt-content')
    cmtContents = [content.text for content in contents]

    likes = listComments.find_elements(By.CLASS_NAME, 'cmt-command')
    cmtLikes = [like.find_element(By.TAG_NAME, 'a').text for like in likes]

    cmtProduct.append(productTitles[i])
#
except ElementClickInterceptedException:
    cmtNames, confirmBuy, cmtContents, cmtLikes = [], [], [], []

```

Kết quả:

```

['Thuy', 'Lương Văn Dung', 'Hoàng Văn Toàn', 'Nguyễn Văn Sao', 'Nguyễn Đức Đức', 'Tú Anh', 'TRẦN VĂN DŨNG', 'Nông Thị Kim', 'Thanh Thạch', 'Quang',
'Dã mua tại TGDĐ', 'Đã mua tại ĐMX', 'Đã mua tại TGDĐ', 'Đã mua tại TGDĐ', 'Đã mua tại TGDĐ', 'Đã mua tại TGDĐ', 'Đã mua tại ĐMX', 'Đã mua tại ',
'May xai mình thay tam on..nhưng có 1 vấn đề là lúc mình xai hết pin bi tat nguồn..dem sac..den lúc mư nguồn lên thì rất rất rất lâu luôn ay...<
'Hữu ích', 'Hữu ích (3)', 'Hữu ích (5)', 'Hữu ích', 'Hữu ích (1)', 'Hữu ích', 'Hữu ích', 'Hữu ích (1)', 'Hữu ích (1)', 'Hữu ích (4)', 'Hữu ích '

```

Sau khi có các thông tin như trên, em tiến hành tạo 1 dataframe mới cho chúng tên là productDetailComments.

```

productDetailComments = pd.DataFrame(list(zip(cmtProduct, cmtNames,
confirmBuy, cmtContents, cmtLikes)),columns=["Title", "Name", "Confirm Buy",
"Content", "Likes"])
productDetailComments.info()

```

Kết quả:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1 entries, 0 to 0
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Title           1 non-null     object
1   Name            1 non-null     object
2   Confirm Buy     1 non-null     object
3   Content         1 non-null     object
4   Likes          1 non-null     object
dtypes: object(5)
memory usage: 168.0+ bytes

```

Sở dĩ, lấy thêm title của sản phẩm để có thể tạo khóa chung, ghép 2 dataframe Product và productDetailComments lại với nhau.

Cuối cùng t merge 2 bảng Product và productDetailComments lại với nhau, theo khóa chung là title, giữ nguyên bảng productDetailComments ra bảng cuối cùng Smartphone.

Kết quả:

```
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Title                  103 non-null   object
1   Image                  100 non-null   object
2   URL                    103 non-null   object
3   Size                   103 non-null   object
4   Storage                103 non-null   object
5   Price                  103 non-null   object
6   Discount               103 non-null   object
7   Old Price              103 non-null   object
8   Special Price          103 non-null   object
9   Total Comments        103 non-null   object
10  Color                   1 non-null     object
11  Rate                   1 non-null     object
12  Configuration           1 non-null     object
13  Name                    1 non-null     object
14  Confirm Buy             1 non-null     object
15  Content                 1 non-null     object
16  Likes                   1 non-null     object
dtypes: object(17)
```

Kết quả:

```
print(Smartphone.head(1).T)
```

	0
Title	Samsung Galaxy A14 4G
Image	<a href="https://cdn.tgdd.vn/Products/Images/42/292770/...">https://cdn.tgdd.vn/Products/Images/42/292770/...</a>
URL	<a href="https://www.thegioididong.com/dtdd/samsung-gal...">https://www.thegioididong.com/dtdd/samsung-gal...</a>
Size	Trả góp 0%\nSamsung Galaxy A14 4G\n6.6" Full...
Storage	6GB\n4GB
Price	4.190.000đ
Discount	-16%
Old Price	4.990.000đ
Special Price	NULL
Total Comments	301
Color	Đen Đỏ đô Bạc
Rate	4.1
Configuration	Màn hình:\nPLS LCD 6.6" Full HD+\nHệ điều hành...
Name	Thuy
Confirm Buy	Đã mua tại TGDĐ
Content	Máy xài mình thay tạm on..nhưng có 1 vấn đề là...
Likes	Hữu ích

#### IV. THAM KHẢO:

<https://www.selenium.dev/>

<https://selenium-python.readthedocs.io/>

<https://pandas.pydata.org/>

<https://www.w3schools.com/>