# **Croma Scraper**

A powerful and efficient web scraper designed to extract product information from the Croma India e-commerce website. This tool is ideal for market research, price monitoring, data analysis, or building custom product catalogs.

## **Table of Contents**

* [Features](https://www.google.com/search?q=%23features)
* [Technologies Used](https://www.google.com/search?q=%23technologies-used)
* [Setup and Installation](https://www.google.com/search?q=%23setup-and-installation)
* [Usage](https://www.google.com/search?q=%23usage)
* [Output Example](https://www.google.com/search?q=%23output-example)
* [Contributing](https://www.google.com/search?q=%23contributing)
* [License](https://www.google.com/search?q=%23license)

## **Features**

* **Product Data Extraction:** Scrapes key product details such as:
  + Product Name
  + Price (Current and Original, if available)
  + Product URL
  + Category/Sub-category
  + Brand
  + Product Rating (if available)
  + Number of Reviews (if available)
  + Availability Status
* **Flexible Output:** Saves extracted data into a structured format (e.g., CSV, JSON) for easy integration with other tools or databases.
* **Pagination Handling:** Navigates through multiple pages of search results or categories to ensure comprehensive data collection.
* **Error Handling:** Includes basic error handling for network issues or changes in website structure.
* **Configurable:** Easily adaptable to scrape different product categories or search queries.

## **Technologies Used**

* **Python 3.x:** The primary programming language.
* **Requests:** For making HTTP requests to the Croma website.
* **BeautifulSoup4:** For parsing HTML content and extracting data.
* **(Optional) Pandas:** For efficient data manipulation and saving to CSV/Excel.
* **(Optional) Selenium:** If dynamic content loading (JavaScript rendering) requires a headless browser.

## **Setup and Installation**

Follow these steps to get the Croma Scraper up and running on your local machine.

1. **Clone the Repository:**  
   git clone [https://github.com/TanmaySingh007/Croma-Scraper.git](https://github.com/TanmaySingh007/Croma-Scraper.git)  
   cd Croma-Scraper
2. **Create a Virtual Environment (Recommended):**  
   python -m venv venv  
   # On Windows  
   .\venv\Scripts\activate  
   # On macOS/Linux  
   source venv/bin/activate
3. **Install Dependencies:**  
   pip install -r requirements.txt  
     
   *(If* requirements.txt is not present, you might need to *install requests and beautifulsoup4 manually: pip install requests beautifulsoup4)*

## **Usage**

To run the scraper, execute the main Python script. You might need to specify a category, search term, or product URL depending on the scraper's design.

**Example:**

# Assuming your main script is named `scraper.py`  
python scraper.py --category "mobiles" --output\_file "mobiles\_data.csv"

*(Adjust the command-line arguments --category, --output\_file, etc., based on how your scraper.py is designed to accept inputs.)*

The scraper will then start fetching data and save it to the specified output file in your project directory.

## **Output Example**

The output file (e.g., products\_data.csv) will typically contain columns similar to:

Product Name,Brand,Price,Original Price,Rating,Reviews,Availability,Product URL,Category  
"Apple iPhone 14 (128GB, Midnight)",Apple,79900,79900,4.5,120,In Stock,[https://www.croma.com/apple-iphone-14...,Smartphones](https://www.croma.com/apple-iphone-14...,Smartphones)  
"Samsung Galaxy S23 (256GB, Green)",Samsung,99999,109999,4.8,85,In Stock,[https://www.croma.com/samsung-galaxy...,Smartphones](https://www.croma.com/samsung-galaxy...,Smartphones)  
"HP Pavilion Laptop 15-eg2000",HP,65000,75000,4.2,50,In Stock,[https://www.croma.com/hp-pavilion-laptop...,Laptops](https://www.croma.com/hp-pavilion-laptop...,Laptops)  
# ... more product entries

## **Contributing**

Contributions are welcome! If you have suggestions for improvements, bug fixes, or new features, please feel free to:

1. Fork the repository.
2. Create a new branch (git checkout -b feature/your-feature-name).
3. Make your changes.
4. Commit your changes (git commit -m 'Add new feature').
5. Push to the branch (git push origin feature/your-feature-name).
6. Open a Pull Request.

## **License**

This project is licensed under the MIT License - see the [LICENSE](https://www.google.com/search?q=LICENSE) file for details.