

# Ultra Short E-commerce Product Scraper (Python Script)

```
#!/usr/bin/env python3
"""Ultra Short E-commerce Product Scraper"""

import requests, csv, json, time, re
from bs4 import BeautifulSoup
from urllib.parse import urljoin

class Scraper:
    def __init__(self):
        self.s = requests.Session()
        self.s.headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.3'}

    def get(self, url):
        try:
            r = self.s.get(url, timeout=10)
            time.sleep(1)
            return BeautifulSoup(r.content, 'html.parser')
        except: return None

    def txt(self, el, sel):
        e = el.select_one(sel)
        return e.get_text(strip=True) if e else ""

    def clean_price(self, p):
        m = re.search(r'[\p{C}]{1,3}\p{C}?\s?[\d,]+\.\d*', p or "")
        return m.group().strip() if m else ""

    def extract(self, soup, base, site=""):
        products = []

        if "amazon" in site:
            for c in soup.select('[data-component-type="s-search-result"]'):
                p = {
                    'name': self.txt(c, 'h2 a span'),
                    'price': self.clean_price(self.txt(c, '.a-price-whole')),
                    'rating': re.search(r'(\d+\.\d*)', self.txt(c, '.a-icon-alt')) or "",
                    'url': urljoin(base, c.select_one('h2 a')['href']) if c.select_one('h2 a') else ""
                }
                p['rating'] = p['rating'].group(1) if p['rating'] else ""
                if p['name']: products.append(p)

        elif "flipkart" in site:
            for c in soup.find_all('div', class_=re.compile(r'_1AtVbE|_13oc-S')):
                n = c.find('a', class_=re.compile(r'IRpwTa|_1fQZEK'))
                if n:
                    p = {
                        'name': n.get_text(strip=True),
                        'price': self.clean_price(self.txt(c, '[class*="_30jeq3"]')),
                        'rating': re.search(r'(\d+\.\d*)', self.txt(c, '[class*="_3LWZlK"]')) or ""
                        'url': urljoin(base, n.get('href', ''))
                    }
                    p['rating'] = p['rating'].group(1) if p['rating'] else ""
                    products.append(p)

        else: # Generic
            for c in soup.select('.product, .item, [class*="product"], [class*="item"]')[0:20]:
                p = {
                    'name': self.txt(c, 'h1, h2, h3, .title, .name, [class*="title"]'),
                    'price': self.clean_price(self.txt(c, '.price, [class*="price"]')),
                    'rating': re.search(r'(\d+\.\d*)', self.txt(c, '.rating, [class*="rating"]')) or ""
                    'url': urljoin(base, c.select_one('a')['href']) if c.select_one('a') else ""
                }
                p['rating'] = p['rating'].group(1) if p['rating'] else ""
                if len(p['name']) > 3: products.append(p)

        return products

    def scrape(self, url, pages=3, site=""):
        products, base = [], f"{url.split('/')[0]}://{url.split('/')[2]}"
        for i in range(1, pages + 1):
```

```

        page_url = url if i == 1 else f"{url}{'&' if '?' in url else '?'}page={i}"
        soup = self.get(page_url)
        if not soup: continue

        prods = self.extract(soup, base, site)
        if prods:
            products.extend(prods)
            print(f"Page {i}: {len(prods)} products")
        else: break

    return products

def save_csv(self, products, file="products.csv"):
    if products:
        with open(file, 'w', newline='', encoding='utf-8') as f:
            csv.DictWriter(f, products[0].keys()).writeheader()
            csv.DictWriter(f, products[0].keys()).writerows(products)
        print(f"Saved {len(products)} products to {file}")

def save_json(self, products, file="products.json"):
    if products:
        json.dump(products, open(file, 'w', encoding='utf-8'), indent=2)
        print(f"Saved {len(products)} products to {file}")

# Usage
def scrape_products(url, pages=3, site="", format="csv"):
    s = Scraper()
    products = s.scrape(url, pages, site)
    if format == "json": s.save_json(products)
    else: s.save_csv(products)
    return products

# Examples
def amazon(): return scrape_products("https://amazon.com/s?k=laptop", 2, "amazon")
def flipkart(): return scrape_products("https://flipkart.com/search?q=phone", 2, "flipkart")
def generic(): return scrape_products("https://example.com/products", 3)

if __name__ == "__main__":
    import sys
    if len(sys.argv) > 1:
        url, pages, site = sys.argv[1], int(sys.argv[2] if len(sys.argv) > 2 else 3), sys.argv[3] if
        scrape_products(url, pages, site)
    else:
        print("Usage: python scraper.py <URL> [pages] [site]")
        print("Example: python scraper.py 'https://amazon.com/s?k=laptop' 3 amazon")

```