

WEBCRAPING USING SELENIUM

1.

```
pip install selenium pandas beautifulsoup4
```

```
from selenium import webdriver
```

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

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

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

```
from selenium.webdriver.support import expected_conditions as EC
```

```
from bs4 import BeautifulSoup
```

```
import pandas as pd
```

```
import time
```

```
# Initialize the WebDriver
```

```
driver = webdriver.Chrome()
```

```
# Step 1: Get the web page
```

```
driver.get("https://www.naukri.com/")
```

```
# Step 2: Enter "Data Scientist" in the search field
```

```
search_field = driver.find_element(By.ID, "qsb-keyword-sugg")
```

```
search_field.send_keys("Data Scientist")
```

```
# Step 3: Click the search button
```

```
search_button = driver.find_element(By.CLASS_NAME, "search-btn")
search_button.click()
```

```
# Wait for the results to load
```

```
WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CLASS_NAME,
"list")))
```

```
# Step 4: Apply the location filter
```

```
location_filter = driver.find_element(By.XPATH, "//span[text()='Delhi / NCR']")
location_filter.click()
```

```
# Apply the salary filter
```

```
salary_filter = driver.find_element(By.XPATH, "//span[text()='3-6 Lakhs']")
salary_filter.click()
```

```
# Wait for the filters to apply
```

```
time.sleep(5)
```

```
# Step 5: Scrape the data for the first 10 job results
```

```
soup = BeautifulSoup(driver.page_source, 'html.parser')
```

```
job_cards = soup.find_all('article', class_='jobTuple bgWhite br4 mb-8')[:10]
```

```
jobs = []
```

```

for job in job_cards:

    job_title = job.find('a', class_='title fw500 ellipsis').text.strip()

    job_location = job.find('li', class_='fleft grey-text br2 placeHolderLi
location').text.strip()

    company_name = job.find('a', class_='subTitle ellipsis fleft').text.strip()

    experience = job.find('li', class_='fleft grey-text br2 placeHolderLi
experience').text.strip()

    jobs.append({

        'Job Title': job_title,

        'Location': job_location,

        'Company Name': company_name,

        'Experience Required': experience

    })

```

Step 6: Create a DataFrame of the scraped data

```

df = pd.DataFrame(jobs)

print(df)

```

Close the WebDriver

```

driver.quit()

```

2.

```

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

```

```
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from bs4 import BeautifulSoup
import pandas as pd
import time

# Initialize the WebDriver
driver = webdriver.Chrome()

# Step 1: Get the web page
driver.get("https://www.shine.com/")

# Step 2: Enter "Data Analyst" in the search field and "Bangalore" in the location field
job_title_field = driver.find_element(By.ID, "id_q")
job_title_field.send_keys("Data Scientist")

location_field = driver.find_element(By.ID, "id_loc")
location_field.send_keys("Bangalore")

# Step 3: Click the search button
search_button = driver.find_element(By.CLASS_NAME, "search-btn")
search_button.click()
```

```
# Wait for the results to load
```

```
WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CLASS_NAME, "jobTuple")))
```

```
# Step 4: Scrape the data for the first 10 job results
```

```
soup = BeautifulSoup(driver.page_source, 'html.parser')
```

```
job_cards = soup.find_all('li', class_='jobTuple')[0:10]
```

```
jobs = []
```

```
for job in job_cards:
```

```
    job_title = job.find('a', class_='job-title').text.strip()
```

```
    job_location = job.find('span', class_='loc').text.strip()
```

```
    company_name = job.find('a', class_='company-name').text.strip()
```

```
    experience = job.find('span', class_='exp').text.strip()
```

```
    jobs.append({
```

```
        'Job Title': job_title,
```

```
        'Location': job_location,
```

```
        'Company Name': company_name,
```

```
        'Experience Required': experience
```

```
    })
```

```
# Step 5: Create a DataFrame of the scraped data
```

```
df = pd.DataFrame(jobs)
```

```
print(df)
```

```
# Close the WebDriver
```

```
driver.quit()
```

3.

```
from selenium import webdriver
```

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

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

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

```
from selenium.webdriver.support import expected_conditions as EC
```

```
from bs4 import BeautifulSoup
```

```
import pandas as pd
```

```
import time
```

```
# Initialize the WebDriver
```

```
driver = webdriver.Chrome()
```

```
# Step 1: Get the web page
```

```
driver.get("https://www.flipkart.com/apple-iphone-11-black-64-gb/product-reviews/itm4e5041ba101fd?
```

```
pid=MOBFWQ6BXGJCEYNY&lid=LSTMObFWQ6BXGJCEYNYZXSHRJ&marketplace=FLIPKART")
```

```
# Wait for the reviews to load
```

```
WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CLASS_NAME, "_
```

```
1YokD2"))))
```

```
# Step 2: Scrape the data for the first 100 reviews
```

```
reviews = []
```

```
while len(reviews) < 100:
```

```
    soup = BeautifulSoup(driver.page_source, 'html.parser')
```

```
    review_cards = soup.find_all('div', class_='_1AtVbE')[:100 - len(reviews)]
```

```
    for card in review_cards:
```

```
        try:
```

```
            rating = card.find('div', class_='_3LWZIK').text.strip()
```

```
            review_summary = card.find('p', class_='_2-N8zT').text.strip()
```

```
            full_review = card.find('div', class_='t-ZTKy').text.strip()
```

```
            reviews.append({
```

```
                'Rating': rating,
```

```
                'Review Summary': review_summary,
```

```
                'Full Review': full_review
```

```
            })
```

```
        except AttributeError:
```

```
            continue
```

```
# Click the "Next" button to load more reviews
```

```
try:
```

```
        next_button = driver.find_element(By.CLASS_NAME, '_1LKTO3')
        next_button.click()
        time.sleep(2)
    except:
        break
```

Step 3: Create a DataFrame of the scraped data

```
df = pd.DataFrame(reviews)
```

```
print(df)
```

Close the WebDriver

```
driver.quit()
```

4.

```
from selenium import webdriver
```

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

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

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

```
from selenium.webdriver.support import expected_conditions as EC
```

```
from bs4 import BeautifulSoup
```

```
import pandas as pd
```

```
import time
```

Initialize the WebDriver


```
driver = webdriver.Chrome()
```

```
# Step 1: Get the web page
```

```
driver.get("https://www.flipkart.com/")
```

```
# Close the login popup if it appears
```

```
try:
```

```
    close_button = WebDriverWait(driver, 10).until(
        EC.element_to_be_clickable((By.XPATH, "//button[contains(text(), '×')]"))
    )
    close_button.click()
```

```
except:
```

```
    pass
```

```
# Step 2: Enter "sneakers" in the search field
```

```
search_field = driver.find_element(By.NAME, "q")
```

```
search_field.send_keys("sneakers")
```

```
search_field.send_keys(Keys.RETURN)
```

```
# Wait for the results to load
```

```
WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CLASS_NAME, "_1YokD2")))
```

```
# Step 3: Scrape the data for the first 100 sneakers
```

```
sneakers = []
```

```
while len(sneakers) < 100:
```

```
    soup = BeautifulSoup(driver.page_source, 'html.parser')
```

```
    sneaker_cards = soup.find_all('div', class_='_1AtVbE')[:100 - len(sneakers)]
```

```
    for card in sneaker_cards:
```

```
        try:
```

```
            brand = card.find('div', class_='_2WkVRV').text.strip()
```

```
            product_description = card.find('a', class_='lRpwTa').text.strip()
```

```
            price = card.find('div', class_='_30jeq3').text.strip()
```

```
            sneakers.append({
```

```
                'Brand': brand,
```

```
                'Product Description': product_description,
```

```
                'Price': price
```

```
            })
```

```
        except AttributeError:
```

```
            continue
```

```
# Click the "Next" button to load more sneakers
```

```
try:
```

```
    next_button = driver.find_element(By.CLASS_NAME, '_1LKTO3')
```

```
    next_button.click()
```

```

        time.sleep(2)

    except:

        break

# Step 4: Create a DataFrame of the scraped data
df = pd.DataFrame(sneakers)

print(df)


# Close the WebDriver
driver.quit()

5.

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.chrome.service import Service

from webdriver_manager.chrome import ChromeDriverManager

import time


# Set up the webdriver

driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))


# Open Amazon India

driver.get("https://www.amazon.in/")

```

```

# Search for "Laptop"

search_box = driver.find_element(By.ID, "twotabsearchtextbox")

search_box.send_keys("Laptop")

search_box.send_keys(Keys.RETURN)


# Wait for the page to load

time.sleep(3)


# Set CPU Type filter to "Intel Core i7"

cpu_filter = driver.find_element(By.XPATH, "//span[text()='Intel Core i7']")

cpu_filter.click()


# Wait for the page to load

time.sleep(3)


# Scrape the first 10 laptops

laptops = driver.find_elements(By.XPATH, "//div[@data-component-type='s-search-result']")[:10]


# Extract the required data

laptop_data = []

for laptop in laptops:

    title = laptop.find_element(By.XPATH, ".//span[@class='a-size-medium a-color-base

```

```
a-text-normal']").text
```

```
try:
```

```
    rating = laptop.find_element(By.XPATH, "//*[@class='a-icon-alt']").text
```

```
except:
```

```
    rating = "No rating"
```

```
try:
```

```
    price = laptop.find_element(By.XPATH, "//*[@class='a-price-whole']").text
```

```
except:
```

```
    price = "No price"
```

```
laptop_data.append({"Title": title, "Rating": rating, "Price": price})
```

```
# Print the scraped data
```

```
for data in laptop_data:
```

```
    print(data)
```

```
# Close the webdriver
```

```
driver.quit()
```

6.

```
from selenium import webdriver
```

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

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

```
from selenium.webdriver.chrome.service import Service
```

```
from webdriver_manager.chrome import ChromeDriverManager
```

```
import time

import csv


# Set up the webdriver

service = Service(ChromeDriverManager().install())

driver = webdriver.Chrome(service=service)


# Open the webpage

driver.get("https://www.azquotes.com/")


# Click on the "Top Quotes" link

top_quotes_link = driver.find_element(By.LINK_TEXT, "Top Quotes")

top_quotes_link.click()


# Wait for the page to load

time.sleep(5)


# Scrape the quotes

quotes = []

for i in range(1, 11): # Adjust the range to get more pages if needed

    quote_elements = driver.find_elements(By.CLASS_NAME, "title")

    author_elements = driver.find_elements(By.CLASS_NAME, "author")

    type_elements = driver.find_elements(By.CLASS_NAME, "tags")
```

```

for quote, author, type_ in zip(quote_elements, author_elements, type_elements):

    quotes.append({

        "quote": quote.text,

        "author": author.text,

        "type": type_.text

    })


# Go to the next page

next_button = driver.find_element(By.LINK_TEXT, "Next")

next_button.click()

time.sleep(5)


# Close the driver

driver.quit()


# Save the quotes to a CSV file

with open("top_1000_quotes.csv", "w", newline="", encoding="utf-8") as file:

    writer = csv.DictWriter(file, fieldnames=["quote", "author", "type"])

    writer.writeheader()

    writer.writerows(quotes)


print("Scraping completed. Quotes saved to top_1000_quotes.csv")

```

7.

```
from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.chrome.service import Service

from webdriver_manager.chrome import ChromeDriverManager

import pandas as pd

import time


# Set up the webdriver

service = Service(ChromeDriverManager().install())

driver = webdriver.Chrome(service=service)


# Open the webpage

driver.get("https://www.jagranjosh.com/general-knowledge/list-of-all-prime-ministers-
of-india-1473165149-1")


# Wait for the page to load

time.sleep(5)


# Find the table containing the data

table = driver.find_element(By.TAG_NAME, 'table')


# Initialize lists to store the data
```



```

names = []

born_dead = []

terms = []

remarks = []


# Iterate over the rows of the table

rows = table.find_elements(By.TAG_NAME, 'tr')[1:] # Skip the header row
for row in rows:

    cells = row.find_elements(By.TAG_NAME, 'td')

    if len(cells) == 4:

        names.append(cells[0].text.strip())

        born_dead.append(cells[1].text.strip())

        terms.append(cells[2].text.strip())

        remarks.append(cells[3].text.strip())


# Close the driver

driver.quit()


# Create a DataFrame

df = pd.DataFrame({

    'Name': names,

    'Born-Dead': born_dead,

    'Term of Office': terms,

```

```
'Remarks': remarks  
})
```

```
# Display the DataFrame
```

```
print(df)
```

8.

```
from selenium import webdriver
```

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

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

```
import pandas as pd
```

```
import time
```

```
# Initialize the WebDriver
```

```
driver = webdriver.Chrome()
```

```
# Step 1: Get the webpage
```

```
driver.get("https://www.motor1.com/")
```

```
# Step 2: Type in the search bar '50 most expensive cars'
```

```
search_bar = driver.find_element(By.NAME, "q")
```

```
search_bar.send_keys("50 most expensive cars")
```

```
search_bar.send_keys(Keys.RETURN)
```

```
# Wait for the search results to load
```

```
time.sleep(3)
```

```
# Step 3: Click on '50 most expensive cars in the world'
```

```
link = driver.find_element(By.PARTIAL_LINK_TEXT, "50 Most Expensive Cars in the  
World")
```

```
link.click()
```

```
# Wait for the page to load
```

```
time.sleep(3)
```

```
# Step 4: Scrape the data
```

```
car_names = driver.find_elements(By.CSS_SELECTOR, "h3")
```

```
car_prices = driver.find_elements(By.CSS_SELECTOR, "p")
```

```
# Extract the text and store in lists
```

```
names = [name.text for name in car_names]
```

```
prices = [price.text for price in car_prices if "$" in price.text]
```

```
# Create a DataFrame
```

```
df = pd.DataFrame({"Car Name": names, "Price": prices})
```

```
# Display the DataFrame
```

```
print(df)
```

```
# Close the WebDriver
```

```
driver.quit()
```