1)Scrape all product names and prices from the first two pages of "Books to Scrape" (http://books.toscrape.com/). Handle simple pagination and structure the output as a list of dictionaries with 'title' and 'price'.

In [2]:
```
import requests
from bs4 import BeautifulSoup

base_url = "http://books.toscrape.com/catalogue/page-{}.html"
books_list = []

for page_num in range(1, 3):
    url = base_url.format(page_num)
    response = requests.get(url)

    response.encoding = 'utf-8'

    if response.status_code != 200:
        print(f"Could not retrieve page {page_num}")
        continue

    soup = BeautifulSoup(response.text, "html.parser")
    products = soup.find_all('article', class_='product_pod')

    for product in products:
        title = product.h3.a['title']
        price = product.find('p', class_='price_color').text
        books_list.append({'title': title, 'price': price})

print(f"Total books scraped: {len(books_list)}")
print(books_list[:5])
```

```
Total books scraped: 40
[{'title': 'A Light in the Attic', 'price': '£51.77'}, {'title': 'Tipping the Vel
vet', 'price': '£53.74'}, {'title': 'Soumission', 'price': '£50.10'}, {'title':
'Sharp Objects', 'price': '£47.82'}, {'title': 'Sapiens: A Brief History of Human
kind', 'price': '£54.23'}]
```

2)Extract the current weather descriptions (like ‘clear’, ‘cloudy’) and temperatures for at least five cities from a public weather site (such as https://www.weather.com or https://wttr.in). Present your data in a tabular format (city, description, temperature).

In [5]:
```
import requests
import pandas as pd

cities = ["London", "New York", "Tokyo", "Sydney", "Paris"]
weather_data = []

for city in cities:
    url = f"https://wttr.in/{city}?format=j1"
    response = requests.get(url)

    if response.status_code != 200:
        print(f"Could not get weather for {city}")
        continue

    data = response.json()
    current_condition = data['current_condition'][0]
    description = current_condition['weatherDesc'][0]['value']
    temperature_c = current_condition['temp_C']

    weather_data.append({
        'City': city,
        'Description': description,
        'Temperature (°C)': temperature_c
    })

df_weather = pd.DataFrame(weather_data)
print(df_weather)

df_weather.to_csv('weather_data.csv', index=False)
print("Weather data saved to 'weather_data.csv'")
```

```
       City               Description Temperature (°C)
0    London                     Sunny               23
1  New York                     Clear               26
2     Tokyo             Partly cloudy               28
3    Sydney  Thunderstorm in vicinity               12
4     Paris                     Clear               18
Weather data saved to 'weather_data.csv'
```

3)From the "Real Python Fake Jobs" board (https://realpython.github.io/fake-jobs/), gather all job titles, companies, and locations listed on the first three pages. Save the results as a CSV file. Be sure to loop through the pagination and properly parse the HTML for structured data extraction.

In [6]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd

base_url = "https://realpython.github.io/fake-jobs/page/{}/"

all_jobs = []


for page in range(1, 4):
    url = base_url.format(page)
    print(f"Scraping page: {url}")

    response = requests.get(url)
    if response.status_code != 200:
        print(f"Couldn't access page {page}")
        continue

    soup = BeautifulSoup(response.text, "html.parser")

    job_cards = soup.find_all("div", class_="card-content")

    for card in job_cards:
        title = card.find("h2", class_="title").get_text(strip=True)
        company = card.find("h3", class_="company").get_text(strip=True)
        location = card.find("p", class_="location").get_text(strip=True)

        all_jobs.append({
            "Title": title,
            "Company": company,
            "Location": location
```

```python
        })

df_jobs = pd.DataFrame(all_jobs)

df_jobs.to_csv("fake_jobs.csv", index=False)

print(f"Saved {len(df_jobs)} job listings to 'fake_jobs.csv'")
```

```
Scraping page: https://realpython.github.io/fake-jobs/page/1/
Couldn't access page 1
Scraping page: https://realpython.github.io/fake-jobs/page/2/
Couldn't access page 2
Scraping page: https://realpython.github.io/fake-jobs/page/3/
Couldn't access page 3
Saved 0 job listings to 'fake_jobs.csv'
```

In [ ]: