Hands-on Lab : Web Scraping

Estimated time needed: **30 to 45** minutes

Objectives

In this lab you will perform the following:

- Extract information from a given web site
- Write the scraped data into a csv file.

Extract information from the given web site

You will extract the data from the below web site:

In [1]:
```
#this url contains the data you need to scrape
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA032
```

The data you need to scrape is the **name of the programming language** and **average annual salary**.
It is a good idea to open the url in your web broswer and study the contents of the web page before you start to scrape.

Import the required libraries

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

Download the webpage at the url

In [3]:
```
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA032
data = requests.get(url).text
```

Create a soup object

In [4]:
```
soup = BeautifulSoup(data, "html.parser")
```

Scrape the `Language name` and `annual average salary`.

In [5]:
```
# Initialize the list to store scraped data
scraped_data = []

# Find the table in the webpage
table = soup.find('table')  # Locate the first <table> tag

if table is None:
    print("No table found on the webpage.")
else:
    # Loop through all rows in the table
    for row in table.find_all('tr'):  # Each row in the table
        cols = row.find_all('td')  # Each column in the row
        if len(cols) == 2:  # Ensure the row has exactly 2 columns
            language = cols[0].get_text(strip=True)  # First column: Language name
            salary = cols[1].get_text(strip=True)  # Second column: Average salary
            scraped_data.append([language, salary])  # Add data to the list
print(soup.prettify())
```

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <title>
   Salary survey results of programming languages
  </title>
  <style>
   table, th, td {
  border: 1px solid black;
}
  </style>
 </head>
 <body>
  <hr/>
  <h2>
   Popular Programming Languages
  </h2>
  <hr/>
  <p>
   Finding out which is the best language is a tough task. A programming language is
created to solve a specific problem. A language which is good for task A may not be
able to properly handle task B. Comparing programming language is never easy. What w
e can do, however, is find which is popular in the industry.
  </p>
  <p>
   There are many ways to find the popularity of a programming languages. Counting t
he number of google searchs for each language is a simple way to find the popularit
y. GitHub and StackOverflow also can give some good pointers.
  </p>
  <p>
   Salary surveys are a way to find out the programmings languages that are most in
demand in the industry. Below table is the result of one such survey. When using any
survey keep in mind that the results vary year on year.
  </p>
  <hr/>
  <table>
   <tbody>
    <tr>
     <td>
      No.
     </td>
     <td>
      Language
     </td>
     <td>
      Created By
     </td>
     <td>
      Average Annual Salary
     </td>
     <td>
      Learning Difficulty
     </td>
    </tr>
    <tr>
     <td>
```

```
  1
 </td>
 <td>
  Python
 </td>
 <td>
  Guido van Rossum
 </td>
 <td>
  $114,383
 </td>
 <td>
  Easy
 </td>
</tr>
<tr>
 <td>
  2
 </td>
 <td>
  Java
 </td>
 <td>
  James Gosling
 </td>
 <td>
  $101,013
 </td>
 <td>
  Easy
 </td>
</tr>
<tr>
 <td>
  3
 </td>
 <td>
  R
 </td>
 <td>
  Robert Gentleman, Ross Ihaka
 </td>
 <td>
  $92,037
 </td>
 <td>
  Hard
 </td>
</tr>
<tr>
 <td>
  4
 </td>
 <td>
  Javascript
 </td>
```

```
 <td>
  Netscape
 </td>
 <td>
  $110,981
 </td>
 <td>
  Easy
 </td>
</tr>
<tr>
 <td>
  5
 </td>
 <td>
  Swift
 </td>
 <td>
  Apple
 </td>
 <td>
  $130,801
 </td>
 <td>
  Easy
 </td>
</tr>
<tr>
 <td>
  6
 </td>
 <td>
  C++
 </td>
 <td>
  Bjarne Stroustrup
 </td>
 <td>
  $113,865
 </td>
 <td>
  Hard
 </td>
</tr>
<tr>
 <td>
  7
 </td>
 <td>
  C#
 </td>
 <td>
  Microsoft
 </td>
 <td>
  $88,726
```

```
 </td>
 <td>
  Hard
 </td>
</tr>
<tr>
 <td>
  8
 </td>
 <td>
  PHP
 </td>
 <td>
  Rasmus Lerdorf
 </td>
 <td>
  $84,727
 </td>
 <td>
  Easy
 </td>
</tr>
<tr>
 <td>
  9
 </td>
 <td>
  SQL
 </td>
 <td>
  Donald D. Chamberlin, Raymond F. Boyce.
 </td>
 <td>
  $84,793
 </td>
 <td>
  Easy
 </td>
</tr>
<tr>
 <td>
  10
 </td>
 <td>
  Go
 </td>
 <td>
  Robert Griesemer, Ken Thompson, Rob Pike.
 </td>
 <td>
  $94,082
 </td>
 <td>
  Difficult
 </td>
</tr>
```

```
  </tbody>
 </table>
 <hr/>
</body>
</html>
```

Save the scrapped data into a file named *popular-languages.csv*

In [6]:
```
output_file = "popular-languages.csv"

with open(output_file, mode='w', newline='', encoding='utf-8') as file:
    writer = csv.writer(file)
    writer.writerow(["Language", "Average Annual Salary"])  # Write the header row
    writer.writerows(scraped_data)  # Write the scraped data

print(f"Data successfully saved to {output_file}")
```

```
Data successfully saved to popular-languages.csv
```

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-10-17	0.1	Ramesh Sannareddy	Created initial version of the lab