Exercises Hand-In 2

Group 30 (Oliver Nilsson)

In [1]:

```
# Import required libraries
import pandas as pd
import requests as req
import bs4
import re

# Print the versions of the libraries to check if they are installed correctly
print(f"Pandas version: {pd.__version__}")
print(f"Requests version: {req.__version__}")
print (f"BeautifulSoup version: {bs4.__version__}")
print(f"Regular Expression version: {re.__version__}")
```

```
Pandas version: 1.5.3
Requests version: 2.31.0
BeautifulSoup version: 4.12.3
Regular Expression version: 2.2.1
```

1. Wikipedia scraping

In [2]:

```
# Define the base address and the start address
base_address = 'https://en.wikipedia.org/'
start_address = base_address + 'wiki/Programming_languages_used_in_most_popular_websites'

# Test response, should return 200 if request was successful
# Use try-except to catch any errors and prevent the script from crashing
try:
    response = req.get(start_address)
    response.raise_for_status()
    print(f"Response code: {response.status_code}")
except req.exceptions.HTTPError as err:
    print(err)
```

```
Response code: 200
```

In [3]:

```
# Parse the response with BeautifulSoup
soup = bs4.BeautifulSoup(response.text, 'lxml')

# Find the table with the programming languages for most popular websites
table = soup.find('table', {'class': 'wikitable sortable'})
# Find all the rows in the table
full_table = table.find_all('tr')
# Print the headers to check if the table was found
print(f"Headers:\n {full_table[0].text}")
```

```
Headers:

Websites

Popularity(unique visitors per month)[1]

Front-end(Client-side)

Back-end(Server-side)

Database
```

Notes

In [4]:

```
# Create a list of headers and a list of data
headers = [re.sub(r'\[.*?\]', '', header) for header in full_table[0].text.split('\n') i
f header]
data = []

# Loop through the rows and extract the data
for row in full_table[1:]:
    hidden_texts = [element.text for element in row.find_all('span', {'style': 'display:
none'})]
    row_text = row.text # Get the text of the row
    pattern = '|'.join(map(re.escape, hidden_texts)) # Create a pattern to remove hidden
text
    row_text = re.sub(pattern, '', row_text).strip() # Remove hidden text and strip the
row
    row_data = [re.sub(r'\[.*?\]', '', value) for value in row_text.split('\n') if value
] # Split the row into values

    # Check if the first value is a number and convert it to an integer
    if row_data[1][0].isdigit():
        value = ''.join(filter(str.isdigit, row_data[1].split(' ')[0])) # Remove any non
-digit characters
        # Check if the value is a digit and convert it to an integer
        if value.isdigit():
            row_data[1] = int(value)

    data.append(row_data) # Append the row to the data list

# Create a DataFrame from the data and headers
df = pd.DataFrame(data, columns=headers)
# Print the DataFrame to check if the data was extracted correctly
df.head()
```

Out[4]:

	Websites	Popularity(unique visitors per month)	Front-end(Client-side)	Back-end(Server-side)	Database	Notes
0	Google	2800000000	JavaScript, TypeScript	C, C++, Go, Java, Python, Node	Bigtable, MariaDB	The most used search engine in the world.
1	Facebook	1120000000	JavaScript, Typescript, Flow	Hack/HHVM, Python, C++, Java, Erlang, D, Haskell	MariaDB, MySQL, HBase, Cassandra	The most visited social networking site.
2	YouTube	1100000000	JavaScript, TypeScript	Python, C, C++, Java, Go	Vitess, BigTable, MariaDB	The most popular video sharing site.
3	Yahoo	750000000	JavaScript	PHP	PostgreSQL, HBase, Cassandra, MongoDB,	None
4	Etsy	516000000	JavaScript	PHP	MySQL, Redis	E-commerce website.

In [5]:

```
# Save the data to a Excel file
try:
    df.to_excel('programming_languages.xlsx', index=False)
    print(f"Data saved to Excel file:\n{soup.title.text} -> programming_languages.xlsx")
except Exception as e:
    print(f"Error saving data to Excel file: {e}")
```

```
Data saved to Excel file:
Programming languages used in most popular websites - Wikipedia -> programming_languages.
xlsx
```