**AUTOMATING THE BORING STUFF**

**Web scraping**

Practice problems

1.Briefly describe the differences between the webbrowser, requests, bs4, and selenium modules.

Ans-

Webbrowser is used to open web pages

Requests is used to makes HTTP requests to a server and retrieves web content.

BeautifulSoup is used to parses and extracts data from HTML or XML documents.

Selenium is used to automates browser interactions (e.g., clicking, typing, navigation).

2.What type of object is returned by requests.get()? How can you access the downloaded content as a string value?

Ans-

The requests.get() function returns a Response object..Use the se the .text attribute of the Response object to access the downloaded content.

3. What requests method checks that the download worked?

Ans-

By using the status\_code attribute of the Response object. HTTP status codes indicate whether the request was successful or not.

4. How can you get the HTTP status code of a requests response?

Ans-

Use the status\_code attribute

5. How do you save a requests response to a file

Ans-

To save the content of a requests response to a file, you can use the response.content (for binary data) or response.text (for text data) and write it to a file using Python's file handling methods.

6. What is the keyboard shortcut for opening a browser’s developer tools?

Ans-

Ctrl+shift+i

7. How can you view (in the developer tools) the HTML of a specific ele ment on a web page?

Ans-

1.open dev tools

2. Go to the elemnts tab and select an element.

8. What is the CSS selector string that would find the element with an id attribute of main

Ans-

Main

9. What is the CSS selector string that would find the elements with a CSS class of highlight?

.highligth would target all elements that would contain highloigfht

10. What is the CSS selector string that would find all the <div> elements inside another <div> element?

Div >div

11. What is the CSS selector string that would find the element with a value attribute set to favorite?

Ans-

button[value="favorite"]

12. Say you have a Beautiful Soup Tag object stored in the variable spam for the element

Hello, world!. How could you get a string 'Hello, world!' from the Tag object?

Ans-  
spam.string

13. How would you store all the attributes of a Beautiful Soup Tag object in a variable named linkElem?  
Ans-

linkElem = tag.attrs

14. Running import selenium doesn’t work. How do you properly import the selenium module?  
Ans-

from selenium import webdriver  
  
15. What’s the difference between the find\_element\_\* and find\_elements\_\* methods?

Ans-

find\_element\_\*: Returns **the first** matching element found on the page.

find\_elements\_\*: Returns a **list of all matching elements** on the page. If no matches are found, it returns an empty list.

16. What methods do Selenium’s WebElement objects have for simulating mouse clicks and keyboard keys?  
Ans-

Selenium’s WebElement objects have the following methods:

* For mouse clicks: click()
* For keyboard keys: send\_keys(Keys.<KEY>) (e.g., Keys.RETURN, Keys.ENTER)

17. You could call send\_keys(Keys.ENTER) on the Submit button’s WebElement object, but what is an easier way to submit a form with selenium?  
Ans-  
element.submit()  
18. How can you simulate clicking a browser’s Forward, Back, and Refresh buttons with selenium?

Ans-

Use Selenium's webdriver methods:

* **Forward button**: driver.forward()
* **Back button**: driver.back()
* **Refresh button**: driver.refresh()

**WORKING WITH EXCEL SPREADSHEETS**

1 What does the openpyxl.load\_workbook() function return?  
It returns a Workbook object representing the loaded Excel workbook.

2 What does the wb.sheetnames workbook attribute contain?  
It contains a list of strings, each representing the name of a sheet in the workbook.

3 How would you retrieve the Worksheet object for a sheet named 'Sheet1'?

sheet = wb['Sheet1']

4 How would you retrieve the Worksheet object for the workbook’s active sheet?

sheet = wb.active

5 How would you retrieve the value in the cell C5?

value = sheet['C5'].value

6 How would you set the value in the cell C5 to "Hello"?

sheet['C5'] = "Hello"

7 How would you retrieve the cell’s row and column as integers?

cell = sheet['C5']

row = cell.row

column = cell.column

8 What do the sheet.max\_column and sheet.max\_row attributes hold, and what is the data type of these attributes?

* sheet.max\_column: The number of the last column containing data (integer).
* sheet.max\_row: The number of the last row containing data (integer).

9 If you needed to get the integer index for column 'M', what function would you need to call?

from openpyxl.utils import column\_index\_from\_string

index = column\_index\_from\_string('M')

10 If you needed to get the string name for column 14, what function would you need to call?

from openpyxl.utils import get\_column\_letter

letter = get\_column\_letter(14)

11 How can you retrieve a tuple of all the Cell objects from A1 to F1?

cells = tuple(sheet['A1:F1'])

12 How would you save the workbook to the filename example.xlsx?

wb.save('example.xlsx')

13 How do you set a formula in a cell?

sheet['C5'] = "=SUM(A1:A10)"

14 If you want to retrieve the result of a cell’s formula instead of the cell’s formula itself, what must you do first?  
Enable formula evaluation by opening the workbook in an application like Excel, as OpenPyXL does not evaluate formulas. Alternatively, you could use a library like xlwings.

15 How would you set the height of row 5 to 100?

sheet.row\_dimensions[5].height = 100

16 How would you hide column C?

sheet.column\_dimensions['C'].hidden = True

17 What is a freeze pane?  
A freeze pane is a fixed row or column that remains visible while you scroll through the worksheet. It is used to keep headers or important labels in view.

18 What five functions and methods do you have to call to create a bar chart?  
To create a bar chart:

* Define chart data:

from openpyxl.chart import BarChart, Reference

data = Reference(sheet, min\_col=1, min\_row=1, max\_col=3, max\_row=10)

chart = BarChart()

* Add data to the chart:

chart.add\_data(data, titles\_from\_data=True)

* Set the title or axes labels (optional):

chart.title = "Sample Bar Chart"

chart.x\_axis.title = "X-Axis"

chart.y\_axis.title = "Y-Axis"

* Add the chart to the sheet:

sheet.add\_chart(chart, "E5").

**WORKING WITH GOOGLE SHEETS**

1 **What three files do you need for EZSheets to access Google Sheets?**

* **credentials.json**: The file containing your Google API credentials.
* **token.json**: The file generated during the authentication process, used for subsequent access without re-authentication.
* **service\_account.json**: An optional file for service account credentials if you're using a service account.

2 **What two types of objects does EZSheets have?**

* **Spreadsheet objects**: Represent an entire Google Sheet spreadsheet.
* **Sheet objects**: Represent individual sheets (tabs) within a spreadsheet.

3 **How can you create an Excel file from a Google Sheet spreadsheet?**

ss.downloadAsExcel('spreadsheet.xlsx')

4 **How can you create a Google Sheet spreadsheet from an Excel file?**

ss = ezsheets.upload('spreadsheet.xlsx')

5 **The ss variable contains a Spreadsheet object. What code will read data from the cell B2 in a sheet titled “Students”?**

data = ss['Students'].get('B2')

6 **How can you find the column letters for column 999?**

ezsheets.getColumnLetterOf(999)

7 **How can you find out how many rows and columns a sheet has?**

python

Copy code

rows = sheet.rowCount

columns = sheet.columnCount

8 **How do you delete a spreadsheet? Is this deletion permanent?**

ss.delete()

This deletion is **not permanent** immediately. The spreadsheet is moved to your Google Drive trash and can be restored from there unless permanently deleted.

9 **What functions will create a new Spreadsheet object and a new Sheet object, respectively?**

* **New Spreadsheet object**:

ss = ezsheets.createSpreadsheet('My Spreadsheet')

* **New Sheet object**:

ss.createSheet('New Sheet')

10 **What will happen if, by making frequent read and write requests with EZSheets, you exceed your Google account’s quota?**  
Google will temporarily block further requests from your account. You’ll need to wait for the quota to reset, which typically happens after 24 hours.

**WORKING WITH PDF AND WORD DOCUMENTS**

1. **What do you pass to the .PdfFileReader() function instead of a string value of the PDF filename?**  
   You pass a **file object** opened in binary read mode (rb).
2. **What modes do the File objects for PdfFileReader() and PdfFileWriter() need to be opened in?**
   * PdfFileReader(): **Read-binary mode** (rb).
   * PdfFileWriter(): **Write-binary mode** (wb).
3. **How do you acquire a Page object for page 5 from a PdfFileReader object?**

page = pdfReader.getPage(4) # Pages are zero-indexed

1. **What PdfFileReader variable stores the number of pages in the PDF document?**

num\_pages = pdfReader.numPages

1. **If a PdfFileReader object’s PDF is encrypted with the password swordfish, what must you do before you can obtain Page objects from it?**

pdfReader.decrypt('swordfish')

1. **What methods do you use to rotate a page?**
   * **Rotate clockwise**:

page.rotateClockwise(90)

* + **Rotate counterclockwise**:

page.rotateCounterClockwise(90)

1. **What method returns a Document object for a file named demo.docx?**

from docx import Document

doc = Document('demo.docx')

1. **What is the difference between a Paragraph object and a Run object?**
   * **Paragraph object**: Represents a block of text in a document.
   * **Run object**: Represents a sequence of characters within a Paragraph that share the same style.
2. **How do you obtain a list of Paragraph objects for a Document object that’s stored in a variable named doc?**

paragraphs = doc.paragraphs

1. **What type of object has bold, underline, italic, strike, and outline variables?**  
   A **Run object**.
2. **What is the difference between setting the bold variable to True, False, or None?**
   * True: Applies bold formatting explicitly.
   * False: Removes bold formatting explicitly.
   * None: Leaves the formatting unchanged or uses the inherited style.
3. **How do you create a Document object for a new Word document?**

from docx import Document

doc = Document()

1. **How do you add a paragraph with the text 'Hello, there!' to a Document object stored in a variable named doc?**

doc.add\_paragraph('Hello, there!')

1. **What integers represent the levels of headings available in Word documents?**  
   Heading levels range from **1** to **9**.

doc.add\_heading('Sample Heading', level=1)

**Working with CSV Files and JSON Data**

1. **What are some features Excel spreadsheets have that CSV spreadsheets don’t?**
   * Excel supports:
     + **Multiple sheets** in a single workbook.
     + **Cell formatting** (font styles, colors, etc.).
     + **Formulas and functions**.
     + **Charts** and embedded images.
     + **Macros** and advanced features like data validation.
   * CSV files are plain text and lack these features.
2. **What do you pass to csv.reader() and csv.writer() to create reader and writer objects?**
   * A **file object** opened in the appropriate mode (r for reader, w for writer).
3. **What modes do File objects for reader and writer objects need to be opened in?**
   * For csv.reader(): **Read mode** (r or rb for binary).
   * For csv.writer(): **Write mode** (w or wb for binary).
4. **What method takes a list argument and writes it to a CSV file?**

writer.writerow(['value1', 'value2', 'value3'])

1. **What do the delimiter and lineterminator keyword arguments do?**
   * delimiter: Specifies the character that separates values in the CSV file (default is a comma ,).

csv.writer(file, delimiter=';')

* + lineterminator: Specifies the string used to terminate lines (default is \r\n for Windows).

csv.writer(file, lineterminator='\n')

1. **What function takes a string of JSON data and returns a Python data structure?**

import json

python\_data = json.loads(json\_string)

1. **What function takes a Python data structure and returns a string of JSON data?**

json\_string = json.dumps(python\_data)

**KEEPING TIME, SCHEDULING TASKS, AND LAUNCHING PROGRAMS**

1. **What is the Unix epoch?**  
   The Unix epoch is **00:00:00 UTC on January 1, 1970**, representing the starting point for Unix time.
2. **What function returns the number of seconds since the Unix epoch?**

import time

seconds = time.time()

1. **How can you pause your program for exactly 5 seconds?**

time.sleep(5)

1. **What does the round() function return?**  
   The round() function returns a **number rounded to the nearest integer or to a specified number of decimal places**.  
   Example:

round(3.14159, 2) # Returns 3.14

round(5.6) # Returns 6

1. **What is the difference between a datetime object and a timedelta object?**
   * datetime object: Represents a specific **date and time**.  
     Example: datetime.datetime(2023, 11, 29, 15, 30)
   * timedelta object: Represents a **duration** or difference between two dates/times.  
     Example: datetime.timedelta(days=5, hours=2)
2. **Using the datetime module, what day of the week was January 7, 2019?**

import datetime

day\_of\_week = datetime.datetime(2019, 1, 7).strftime('%A') # Returns 'Monday'

1. **Say you have a function named spam(). How can you call this function and run the code inside it in a separate thread?**

import threading

thread = threading.Thread(target=spam)

thread.start()

1. **What should you do to avoid concurrency issues with multiple threads?**
   * Use **thread synchronization techniques**, such as:

**Locks**:

lock = threading.Lock()

with lock:

**Semaphores or Events** for managing resources or communication.

* + Use thread-safe data structures like queue.Queue.
  + Minimize shared data or use **thread-local storage**.

**SENDING EMAIL AND TEXT MESSAGES**

1. **What is the Unix epoch?**  
   The Unix epoch is **00:00:00 UTC on January 1, 1970**, representing the starting point for Unix time.
2. **What function returns the number of seconds since the Unix epoch?**

import time

seconds = time.time()

1. **How can you pause your program for exactly 5 seconds?**

time.sleep(5)

1. **What does the round() function return?**  
   The round() function returns a **number rounded to the nearest integer or to a specified number of decimal places**.  
   Example:

round(3.14159, 2)

round(5.6)

1. **What is the difference between a datetime object and a timedelta object?**
   * datetime object: Represents a specific **date and time**.  
     Example: datetime.datetime(2023, 11, 29, 15, 30)
   * timedelta object: Represents a **duration** or difference between two dates/times.  
     Example: datetime.timedelta(days=5, hours=2)
2. **Using the datetime module, what day of the week was January 7, 2019?**

import datetime

day\_of\_week = datetime.datetime(2019, 1, 7).strftime('%A') # Returns 'Monday'

1. **Say you have a function named spam(). How can you call this function and run the code inside it in a separate thread?**

import threading

thread = threading.Thread(target=spam)

thread.start()

1. **What should you do to avoid concurrency issues with multiple threads?**

Use **thread synchronization techniques**, such as:

**Locks**:

lock = threading.Lock()

with lock:

**Semaphores or Events** for managing resources or communication.

Use thread-safe data structures like queue.Queue.

Minimize shared data or use **thread-local storage**.