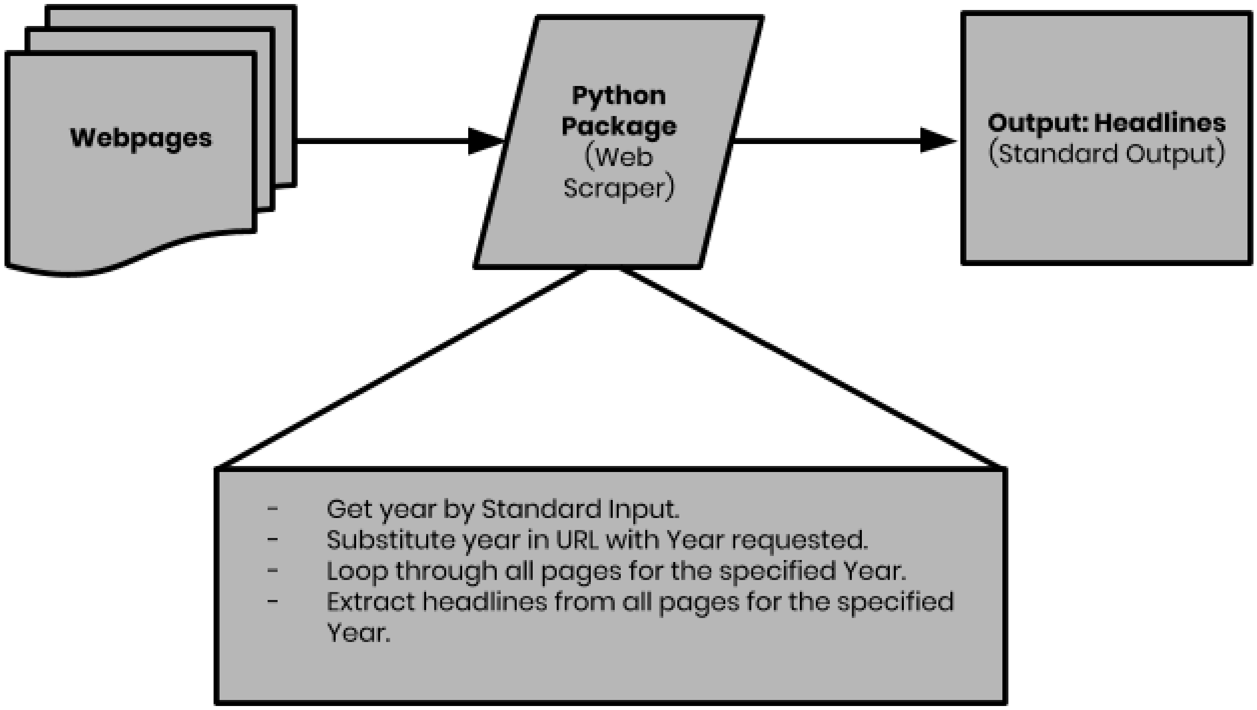
Paessler AG: Programming Exercise

1.) Describe how you are going to solve the problem in words.

For the purpose of extracting data i.e. headlines in this case, I would Web Scrape data from the Slashdot website.

According to Wikipedia, “Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. Web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol, or through a web browser.” (En.wikipedia.org, 2019)

I would use **BeautifulSoup** which is a well know Python package for web scrapping. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree. (PyPI, 2019)


Webpages
Python Package (Web Scraper)
Output: Headlines (Standard Output)

- Get year by Standard Input.
- Substitute year in URL with Year requested.
- Loop through all pages for the specified Year.
- Extract headlines from all pages for the specified Year.

1. **Get year by Standard Input**: The input from the user is taken i.e. the year. It must fall into the range 1998 - current year. If no year is mentioned, the current year is considered.
2. **Substitute year in URL with Year requested**: The year entered or left blank(current year) is substituted in the URL.
3. **Loop through all pages for the specified Year**: Since all the headlines from the specified year need to be included, looping over multiple pages are required. This means that specific tags of the website need to be identified in order to check the termination of the number of pages. One case could be that a specific tag might be available for inspection on all but the last page. Once this tag is identified, it would be easy to trace the last page and that's when the program stops fetching data.
4. **Extract headlines from all pages of the specified Year**: After the year in the URL has been substituted, the headlines are extracted from all the pages of the year and are broken down into lists. These lists are then appended and shown to the user.

2.) Develop the CLI-Application in Python and make it available to us (with instructions on how to run it)

You would need to have python3 installed and updated.

If you don't have it, please install it here: https://www.python.org/downloads/

Once installed, on your terminal, follow these steps:

1. Installing dependencies:
   ```
   pip3 install beautifulsoup4 requests
   ```
2. Run the script headlines_scraper.py
   ```
   python3 headlines_scraper.py
   ```

After these steps, you should see something like this on your terminal:

```
Enter Year Here >>
```

Enter the year you want the headlines to be displayed for after >> and then click Enter. If you don't enter anything, the current year's headlines will be displayed. All the headlines for your entered year will be displayed one after the other.

3.) What would you do differently if you could use the RSS feed?

Since RSS feed is a format for delivering regularly changing web content. This format helps to extract information from these websites in a well-structured manner. This also aids in simplifying the extraction process by querying the keys of the entries. In addition, scraping a website if not required. I would be sending the request, getting a structured response and filtering it as I wished. This would more specifically be parsing the URL and extracting the headlines by processing the keys i.e. title(for the headline) and published(for the year) for our use from the entries of RSS.

References

1. En.wikipedia.org. (2019). *Web scraping*. [online] Available at: https://en.wikipedia.org/wiki/Web_scraping [Accessed 4 May 2019].
2. PyPI. (2019). beautifulsoup4. [online] Available at: https://pypi.org/project/beautifulsoup4/ [Accessed 4 May 2019].