Idea

Time and again I have found myself with a pesky internet connection and/or an old device on which Facebook lags. Facebook pages are still one of the most popular ways to convey information, ask questions, share updates and post memes. I prepare for quiz competitions and there are a lot of pages which post daily grills. To go through all of them within a week (when a grand quiz is nearing) could be made easier using some automation and web scraping.

Current Scope

The description, pictures (if any) and a link to the original post are captured in a plain easy-to-read CSS without any distractive elements. It will only consume bandwidth once (while scraping) and then can be read without internet as well. Facebook's own API need not be used.

Technologies used

Selenium: for the webdriver tool to automate the browser

BeautifulSoup: for scraping information from a webpage

HTML and CSS: for creating the plain easy-to-read webpage

Libraries used: time, webbrowser, os

Implementation details

The overall task can be divided into three subtasks—

1. Automate the scrolling of the page till the first (oldest) post is reached. This can be managed by using a object of the *webdriver* class which selenium supplies us and using functions like *driver.execute_script*, *window.scrollTo*, *time.sleep* to wait for the page to scroll.
2. Read all the *divs* (posts) from the page. This can be achieved using a BeautifulSoup object and using in-built functions like *soup.findAll*, *post.find*, *post_description.text*, etc. Images can also be found using appropriate functions.
3. Writing into our HTML file using file handling features of Python like *open*, *f.write*, etc. HTML tags like *<p>*, *<b>*, *<li>*, *<br>*, etc are used appropriately to make a page.

My experience

I learnt to read the documentation of various packages and then tried test scenarios and experimented a lot. I faced the challenge of working under a deadline with many bugs to squash.