**Problem:** The script is eating up memory and not releasing it properly.

The python script provided in the GitHub repository has a memory management issue. To help understand the issue I’ll go over the basic idea of what the script does. The script has multiple functions that all try to achieve the goal of scrapping all items/cars in a used car selling website. It first gets the max number of pages to properly manipulate the URL, then it gets the links for each page and puts them in a python list to go over each listing and grab the necessary information. The problem I’m facing is that for each iteration/page it grabs with beautifulsoup it fails to release it from memory properly, so you end up a linear increase of memory over time, and eventually your device/server runs out of memory and the script stops working.

A graph with a line

Description automatically generated

Figure 1: A plot of the memory usage of a small sample.

I’ve done some investigating by memory profiling the functions in the script and I’ll go over my findings with you to hopefully help narrow down the solution.

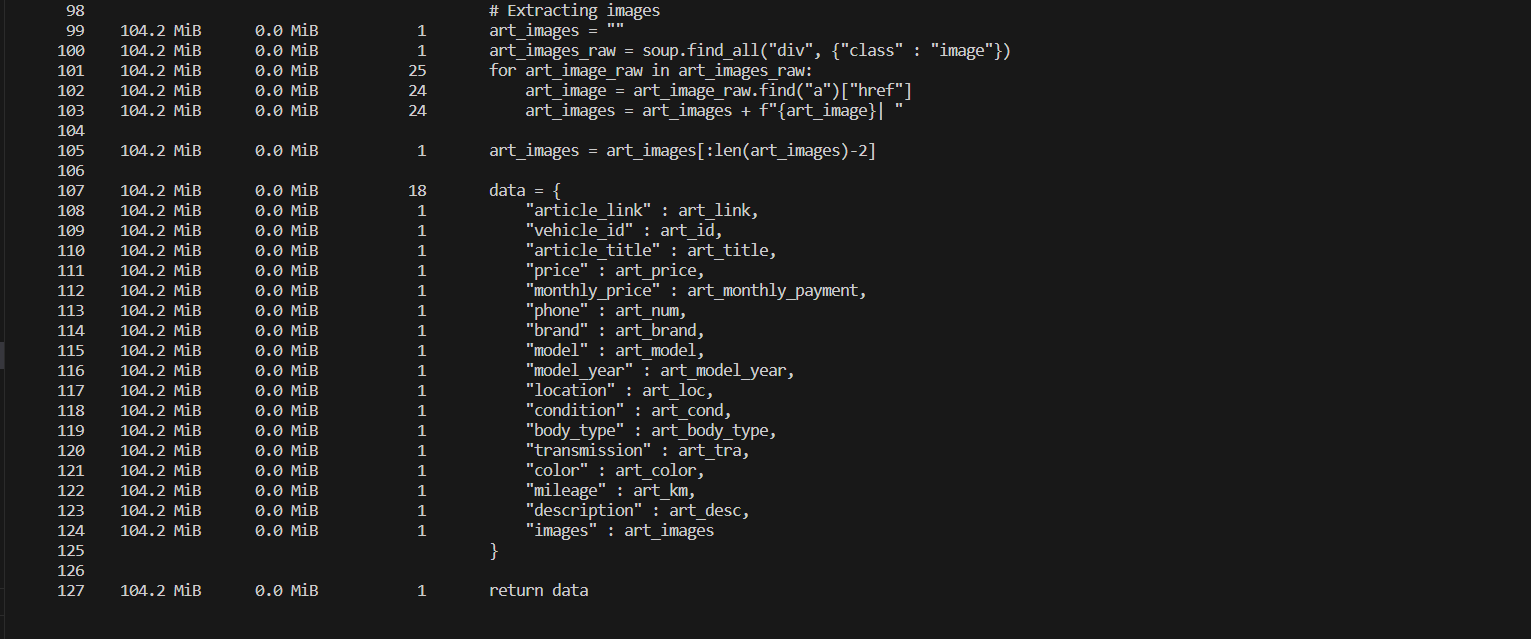
Below I’ll present screenshots of the first iteration and it’s memory usage for each line and function. A computer screen shot of text

Description automatically generatedA computer screen shot of text

Description automatically generated

A screen shot of a computer program

Description automatically generatedA computer screen with text on it

Description automatically generated

As you can see the main problematic lines are 41, 42, 43. Where I store the request response and soup into variables. Even when I do try to release them manually by setting them to the value “None” it only releases a portion of the used memory (i.e. if the used memory is 3.0 MiB the released memory is 1.4 MiB). I’ve also tried doing a manual garbage collection call by writing “gc.collect()” but to no avail.

My main goal from these scripts is to make a couple and leave them running on an EC2 instance but because of the memory leak the instance ends up crashing after it runs out of memory.

Lastly, below are screenshots of the same functions and lines memory usage after running it for 10 pages or around 200 iterations.

A screen shot of a computer program

Description automatically generatedA screen shot of a computer code

Description automatically generated