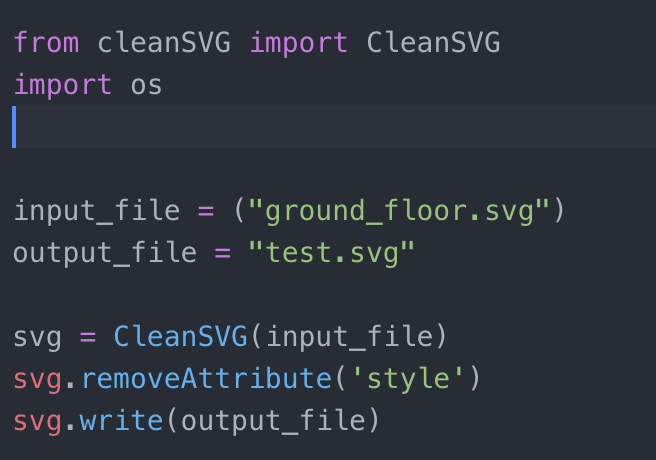
CO600 Toby James Log

Due to the precedence of CSS rules meaning that any in-line CSS has the highest priority we couldn’t alter or interact with any of the SVG objects. This meant we had to find some way of removing the style tag in all of the SVG’s XML. We briefly considered manually going into the files & deleting these tags – just so we could continue on the implementation of our main features but decided that hardcoding this would lead to problems later.

I decided to write a python script to parse through the XML inside the SVG file. I initially used pip (a package management tool for python) to install BeautifulSoup which is primarily a tool used for HTML parsing but can be applied to XML. This allowed me to navigate down the XML tree but wouldn’t allow any interaction with any style tags. After ruling BeautifulSoup out as it wasn’t a native SVG parser & any effort to try to transform it into one could be saved by looking elsewhere, I came across a project on github (<https://github.com/petercollingridge/SVG-Optimiser>) that added functionality to parsing SVGs. I took the main script, gutting a lot of the unnecessary features as we are only after removing style tags whilst maintaining the XML format.



I then called this script on the test svg file, taking the original XML, deleting the style tags & writing this to a new test SVG file.

As a time saving measure I wrote a Python script to scrape all of the computing staff