**Document code Mike Minnebach**

**Name scraper = metacritic\_spider\_gamedetails**

Line 7/15:

Initiate the urls from which we want to start scraping. These urls cover all the platforms.

Line 22/23:

Get the href from all the links of the game pages. From this href we can compile the link of the individual game page, by appending it to [www.metacritic.com](http://www.metacritic.com)

Line 26/30:

Check for all pages of items. We check if the ‘next’ button is present on the web page. If that is true, the href of that button is extracted and called to follow.

Line 35/49:

Extract the information for each individual game website. For the sake of this assignment, both xpath and css selectors were used. A key was created from the title and the platform.

Line 53/67:

The respective data is returned.

**Name\_scraper = sql\_det**

Line 18/25:

Initiate the urls from which we want to start scraping. These urls cover all the platforms.

Line 32/33:

Get the href from all the links of the game pages. From this href we can compile the link of the individual game page, by appending it to [www.metacritic.com](http://www.metacritic.com)

Line 36/40:

Check for all pages of items. We check if the ‘next’ button is present on the web page. If that is true, the href of that button is extracted and called to follow.

Line 44/47:

Extract the information for each individual game website. For the sake of this assignment, both xpath and css selectors were used. A key was created from the title and the platform.

**Note:**

To make use of the SQLite output, the items.py, pipelines.py and settings.py were adjusted to facilitate a connection with SQLite.