The following answers go with the Parser.py file inside the src folder. 4 and 5 are answered in the block comments of the Parsery.py file.

Explain the `with open(...) as collection:` construct in detail, then provide an example of code that uses it for the Syrian data CSV file.

From the Parser we see that the open(…) function used to import data from a file.

```
self.file = open(file_name, 'r', encoding=enc)
```

This line of code calls the open function from the built in Python functions. It returns a file object that is essentially a file pointer and allows for IO operations to be performed on the file. In this particular example, the open function takes in three arguments, the name of the file, the mode, and the file encoding. The mode describes the way that the file will be used, such as read or write operations but also allows for other operations such as appending to the file or creating a file that does not exist yet. The encoding describes the way that the file should be opened, such as in text mode or or binary mode. For the Syrian data file the encoding must be set to latin-1 when reading.

Explain the difference between a `list` and `dict` in python. Provide code samples of each, where you fill them with at least three items then iterate through and print items out.

In python lists are pretty literal in the sense that they are a list of values. Each value is mapped to an integer that represents their bucket, the first item is in index zero, the second is is index 1, etc.

When adding items to a list you must call the append method. This adds the value to the end of the list. You can iterate the items in order using a for loop.

```
list_example = []
list_example.append('one value')
list_example.append('another value')
list_example.append('the last value')

# iterate the list
for items in list_example:
    print(items)
```

In python dictionaries are a key-value map, where the bucket index is the key and that key maps to the value. The values aren't in any particular order though compared to the list.

```
dict_example = {}
dict_example['key'] = 'value'
dict_example['another'] = 'item'
dict_example['the final'] = 'value in the list'
```

When adding items to a dictionary, you do so by referencing the bucket key and then setting the value. You have several ways to interact through a dictionary. To iterate the values using a for loop the following can be done…

```
for value in dict_example:
    print(value)
```

To iterate the keys of the dictionary the following can be done…

```
for key in dict_example.keys():
    print(key)
```

Calling the keys() method on the dictionary returns a list of the keys for iteration.

Note the various styles of importing libraries and namespaces into Python. Please list an example of each style that you find; along with an explanation of how these vary in terms of the importation of libraries.

Python allows for importing libraries or portions of a library through the use of the import command. However, it allows allows for importing a specific object from a library also to avoid importing the unneeded objects.

Importing the entire library…

```
import csv
```

Importing a portion of the library…

```
from collections import deque
```

In the first example the entire csv library, but in the second one the deque object from the collections library is imported rather than the entire collections library.