ANS1. CSV files are simple, lacking many of the features of an Excel spreadsheet. For example, CSV files:

* Don’t have types for their values—everything is a string
* Don’t have settings for font size or color
* Don’t have multiple worksheets
* Can’t specify cell widths and heights
* Can’t have merged cells
* Can’t have images or charts embedded in them

ANS2. To read a CSV file with the csv module, first open it using the open() function , just as you would any other text file. But instead of calling the read() or readlines() method on the File object that open() returns, pass it to the csv.reader() function . This will return a reader object for you to use.

First, call open() and pass it 'w' to open a file in write mode . This will create the object you can then pass to csv.writer()  to create a writer object.

ANS3.  File objects need to be opened in read-binary ('rb') for reader objects and write-binary ('wb') for writer objects.

ANS4. The writerow() method takes a list argument and writes it to a CSV file

ANS5. The *delimiter* is the character that appears between cells on a row. By default, the delimiter for a CSV file is a comma. The *line terminator* is the character that comes at the end of a row. By default, the line terminator is a newline.

We can change characters to different values by using the delimiter and lineterminator keyword,

arguments With csv.writer().

Passing delimiter='\t' and lineterminator='\n\n'  changes the character between cells to a tab and the character between rows to two newlines.

ANS6. call loads() and pass it a string of JSON data. It will return that data as a Python dictionary.

ANS7. The json.dumps() function will translate a Python value into a string of JSON-formatted data