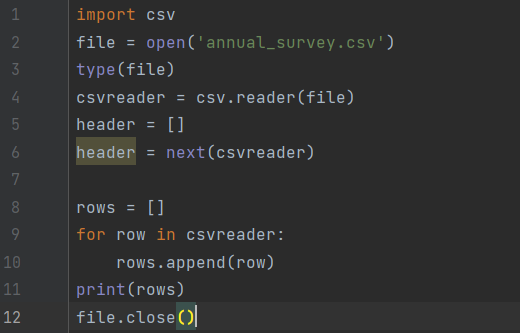
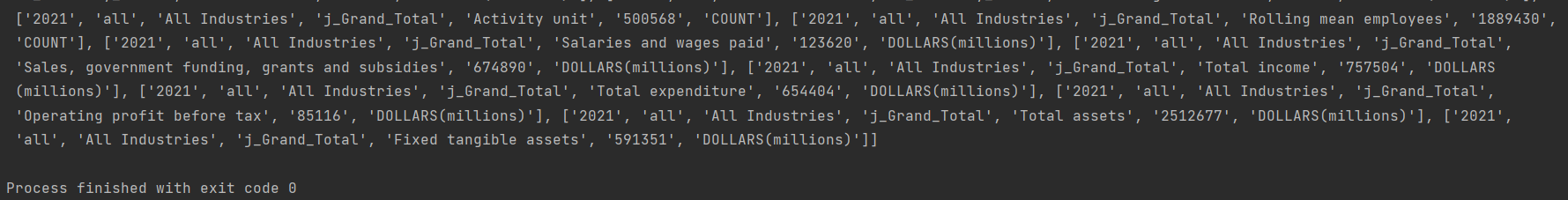
**Read data from csv file into python list:**

We can read data from a csv file by converting the data into a list of lists.

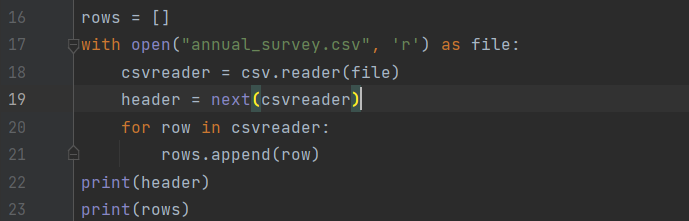
Here, this Python code utilizes the `csv` module to process data from a CSV file named 'annual\_survey.csv'. Initially, the file is opened in read mode, and its file object is assigned to the variable `file`. The code then checks the type of the file object using `type(file)`, although this is not crucial to the main functionality. A CSV reader object, `csvreader`, is created to parse the CSV data from the file. The first row, assumed to contain column headers, is stored in the `header` list. Subsequently, an empty list called `rows` is initialized to accumulate the data rows from the CSV file. A loop iterates through the remaining rows, appending each to the `rows` list. Finally, the list of rows is printed to the console, and the file is closed to release system resources. In essence, this code facilitates the extraction and printing of both column headers and data rows from the specified CSV file, employing Python's `csv` module for efficient CSV handling.



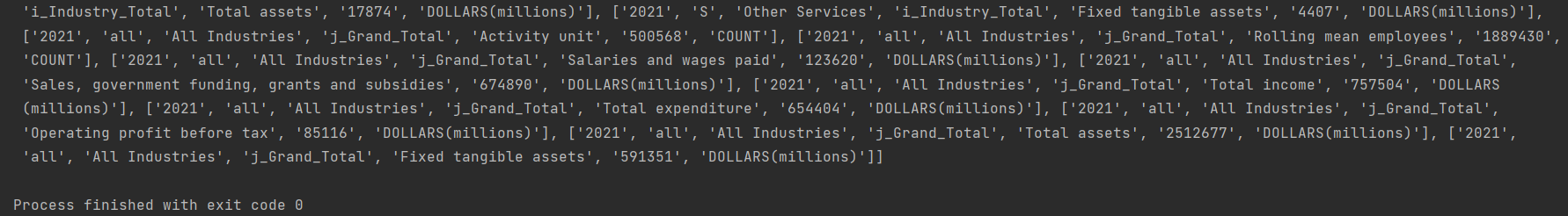
**Output:**



**Another way for reading the data is:**

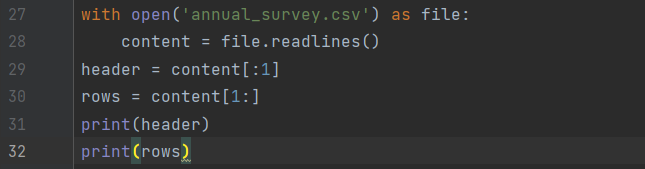


**Output:**



**Readlines():**

To retrieve both the header and rows from a CSV file without using the `csv` library, you can use the `open()` and `with()` statements along with the `.readlines()` method. This method reads all the lines from the file and returns them as a list, where each item in the list represents a row in your CSV file. This allows you to manually handle the separation and organization of data into headers and rows.



**Output:**

