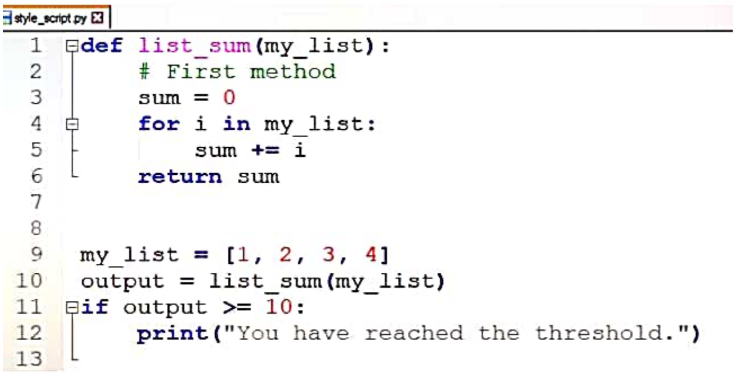
PRACTICAL 7

mplementing coding practices in Python using PEP8.

\s Guido van Rossum said, “Code is read much more often than it is written.” You may ;pend a few minutes, or a whole day, writing a piece of code to process user ıuthentication. Once you’ve written it, you’re never going to write it again. But you’ll lefinitely have to read it again. That piece of code might remain part of a project you’re vorking on. Every time you go back to that file, you’ll have to remember what that :ode does and why you wrote it, so readability matters. PEP stands for Python :nhancement Proposal, and there are several of them. A PEP is a document that lescribes new features proposed for Python and documents aspects of Python, like lesign and style, for the community.Writing clear, readable code shows professionalism. t’ll tell an employer that you understand how to structure your code well. If you have nore experience writing Python code, then you may need to collaborate with others. Vriting readable code here is crucial. Other people, who may have never met you or seen 'our coding style before, follow and recognize will make it easier for others to read your :ode.

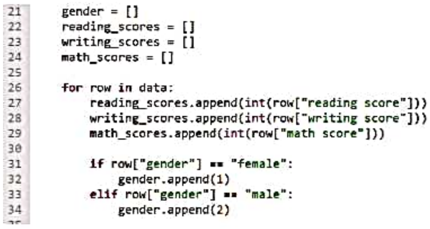
:oding using PEP 8:



```
def list_sum(my_list):
    # First method
    sum = 0
    for i in my_list:
        sum += i
    return sum


my_list = [1, 2, 3, 4]
output = list_sum(my_list)
if output >= 10:
    print("You have reached the threshold.")
```

\nother Example:



```
gender = []
reading_scores = []
writing_scores = []
math_scores = []

for row in data:
    reading_scores.append(int(row["reading score"]))
    writing_scores.append(int(row["writing score"]))
    math_scores.append(int(row["math score"]))

    if row["gender"] == "female":
        gender.append(1)
    elif row["gender"] == "male":
        gender.append(2)
```