STEPS TO CREATE A PYTHON PACKAGE

Steps:

1. **Create a Directory:**
   1. Choose a descriptive name for your package that reflects its functionality.
   2. Create a new directory with that name. This will be the root directory of your package.

2. **Add `__init__.py` (Optional):**
   - While not strictly mandatory for all packages, it's a good practice to include an empty `__init__.py` file within the package directory. This file tells Python that the directory is a package and can be imported.

3. **Develop Modules:**
   - Create Python modules (`.py` files) within the package directory. These modules will contain your code (functions, classes, variables).

Example:

Let's create a simple package named `geometry` that contains a module `calculator.py` to perform geometric calculations:

1. Create a directory named `geometry`.
2. Inside `geometry`, create a file named `__init__.py` (you can leave it empty).
3. Create another file named `calculator.py` inside `geometry`.

   `calculator.py` (Example Code):

   Python
   ```python
   def area_square(side):
       """Calculates the area of a square."""
       return side * side


   def perimeter_square(side):
       """Calculates the perimeter of a square."""
       return 4 * side
   ```

Importing and Using the Package:

Now, you can import the module from your package in other Python scripts:

Python
```python
# Assuming your current script is outside the 'geometry' package

from geometry.calculator import area_square, perimeter_square
```

```
side_length = 5

area = area_square(side_length)
perimeter = perimeter_square(side_length)

print("Area of square:", area)
print("Perimeter of square:", perimeter)
```

This code will output:

```
Area of square: 25
Perimeter of square: 20
```

Additional Considerations:

- You can create sub-packages within your package by following the same directory structure.