

Summary of For Loops in Python

A **for** loop in Python is a powerful tool for iterating over sequences such as lists, tuples, dictionaries, sets, or strings. It allows you to execute a block of code for each element in the sequence, making it particularly useful for repetitive tasks and data processing.

Key Points:

- **Syntax:** The basic syntax of a **for** loop is:

```
for variable in sequence:  
    # Code to execute for each element
```

- **Examples:**
 - **Calculating Kinetic Energy:** Iterate over lists of masses and velocities to compute kinetic energy.
 - **Summing Forces:** Sum a list of forces to find the total force.
 - **Calculating Total Distance:** Sum distances traveled to find the total distance.
- **Indentation:** Proper indentation is crucial as it defines the scope of the loop.
- **Common Errors:** Be aware of common errors such as **IndentationError**, **SyntaxError**, **NameError**, **TypeError**, **IndexError**, and **LogicError**.
- **Using `range()`:** The `range()` function generates a sequence of numbers, useful for iterating a specific number of times.
- **List Slicing:** Slicing allows you to access and iterate over a subset of a list.

By mastering **for** loops, you can efficiently handle repetitive tasks and data processing, making them an essential construct in Python programming.