

Loop Statements

- For Loop
- While Loop
- For Loop and List comprehension
- Files Sneak Peek: For loop for files

While Loop

- Syntax :
while <condition>:
 statements1
else: # optional
 statements2
- *Statements2* is executed when condition becomes false (but not in case of break)
- WAP to print first 10 natural numbers. Update the program to print their sum
- WAP to count vowels in a string input by user.
- WAP to print all multiples of **3** till **N** (input N from user).

trainer.cpp@gmail.com

For Loop

- Syntax :

```
for <variable> in <sequence>:  
    statements1  
else:  
    statements2
```
- Statements 2 are executed when the condition becomes false
- **WAP** to create a list of 10 random numbers
- Print the squares of the above generated random numbers

trainer.cpp@gmail.com

Break and Continue

- **break** statement is used to terminate the current loop
- On execution, **continue** statement skips the statements below it in the current loop and forces next iteration of the loop.
- Update the **rolling dice** program to ask user to roll again or exit(break).
- Update the **rolling dice** program to also check for invalid inputs(continue)

trainer.cpp@gmail.com

List Comprehension : For loop

- Syntax:
[expression(<variable>) **for** <variable> **in** <sequence type> [<condition>]]
condition is optional
- WAP to generate list of first 10 natural numbers (Generate a list of their squares also).
- WAP to count vowels using list comprehension
- WAP to find sum of the squares of first 10 even numbers
4 + 9 + 16 + 25

trainer.cpp@gmail.com

Command Line Arguments

- Command line arguments are a way to pass values or options to a script while executing it through the command line.
- They are available through the **sys** module
- To use, import the sys module and use the **argv** attribute to access the command line values.
- **Argv** is a list containing strings.

trainer.cpp@gmail.com