```
"""
Topic: Control Structures
@author: Varpe K. M.
Reference : NPTL:https://onlinecourses.nptel.ac.in/:Course-Python for Data Science
            https://www.geeksforgeeks.org/loops-in-python/
"""
"""
Control Structures
for and  while
 - Execute certain commands repeatedly and use a certain
   logic to stop the iterations
"""
"""
Syntax: while
    while expression:
    statement(s)
"""

# Python program to illustrate
# while loop
count = 0
while (count < 3):
    count = count + 1
    print("Hello VITians")
"""
Using else statement with while loops:
    -When the condition becomes false,
    the statement immediately after the loop is executed.
    The else clause is only executed when your while condition becomes false.
    If you break out of the loop, or if an exception is raised,
    it won't be executed.
If else like this:
if condition:
    # execute these statements
else:
    # execute these statements

Syntax While else:

while condition:
     # execute these statements
else:
     # execute these statements
"""

#Python program to illustrate
# combining else with while
count = 0
while (count < 3):
    count = count + 1
    print("Hello VITians")
else:
    print("In Else Bye VITians")


# Python program to illustrate
# Single statement while block
count = 0
while (count == 0): print("Hello VITians")
```

```
"""
Syntax: for
for iterator_var in sequence:
    statements(s)
"""

"""FOR LOOP EXAMPLES """
# Python program to illustrate for
for i in range(1,5,1):
    print(i)

for i in range(1,5):
    print(i)

for i in range(5):
    print(i)

for i in range(5,1,-1):
    print(i)

# Iterating over a list
print("List Iteration")
l = ["VIT", "for", "Innovation"]
for i in l:
    print(i)

# Iterating over a tuple (immutable) +
print("\nTuple Iteration")
t = ("VIT", "for", "geeks")
for i in t:
    print(i)

# Iterating over a String
print("\nString Iteration")
s = "Geeks"
for i in s :
    print(i)

# Iterating over dictionary 87

print("\nDictionary Iteration")
d = dict()
d['xyz'] = 123
d['abc'] = 345
for i in d :
    print("%s  %d" %(i, d[i]))
```