CMPE 150
Introduction to Computing
Spring 2022

Slides adapted from Gökçe Uludoğan

For Loops

- Iterates over a collection of objects
- Syntax:

```
for item in sequence:
    statements
```

- At each iteration:
  -item takes the value of the next element in sequence
  -statements are executed for item

For Loops

Loop continues until the last item is reached.

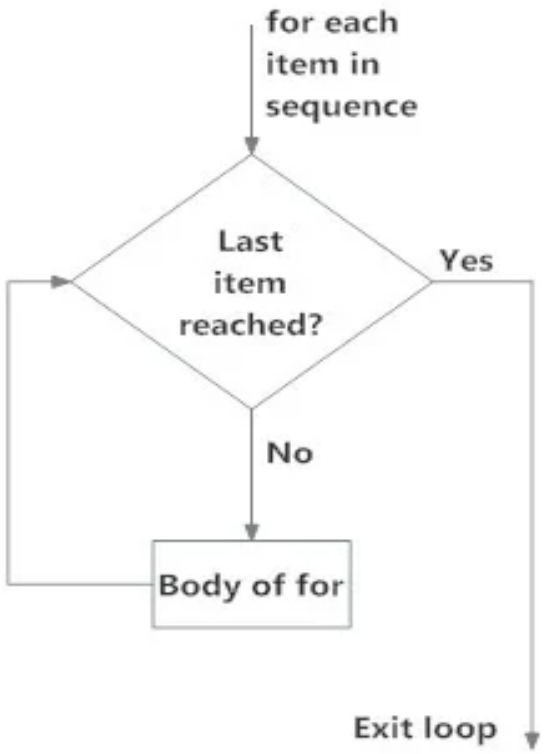


Fig: operation of for loop

Example: Iterating a sequence of numbers

```
for i in [1, 2, 3, 4]:
    print(i)
```

Output

```
1
2
3
4
```

For Loop Visualization

https://bouncmpe150.github.io/python-slides/lab3.html - /4

The range() function

- Generates numbers in given range
- range(start, stop, step_size)
  - **start:** lower limit: By default, 0.
  - **stop:** upper limit. Numbers are generated up to this number.
  - **step_size:** difference between each number. By default, 1.

- start and step_size are optional arguments.

Examples

```
for i in range(5):
    print(i, end=', ')
```

Output

```
0, 1, 2, 3, 4,
```

```
for i in range(5, 10):
    print(i, end=', ')
```

Output

```
5, 6, 7, 8, 9,
```

```
for i in range(2,8,2):
    print(i, end=', '
```

Output

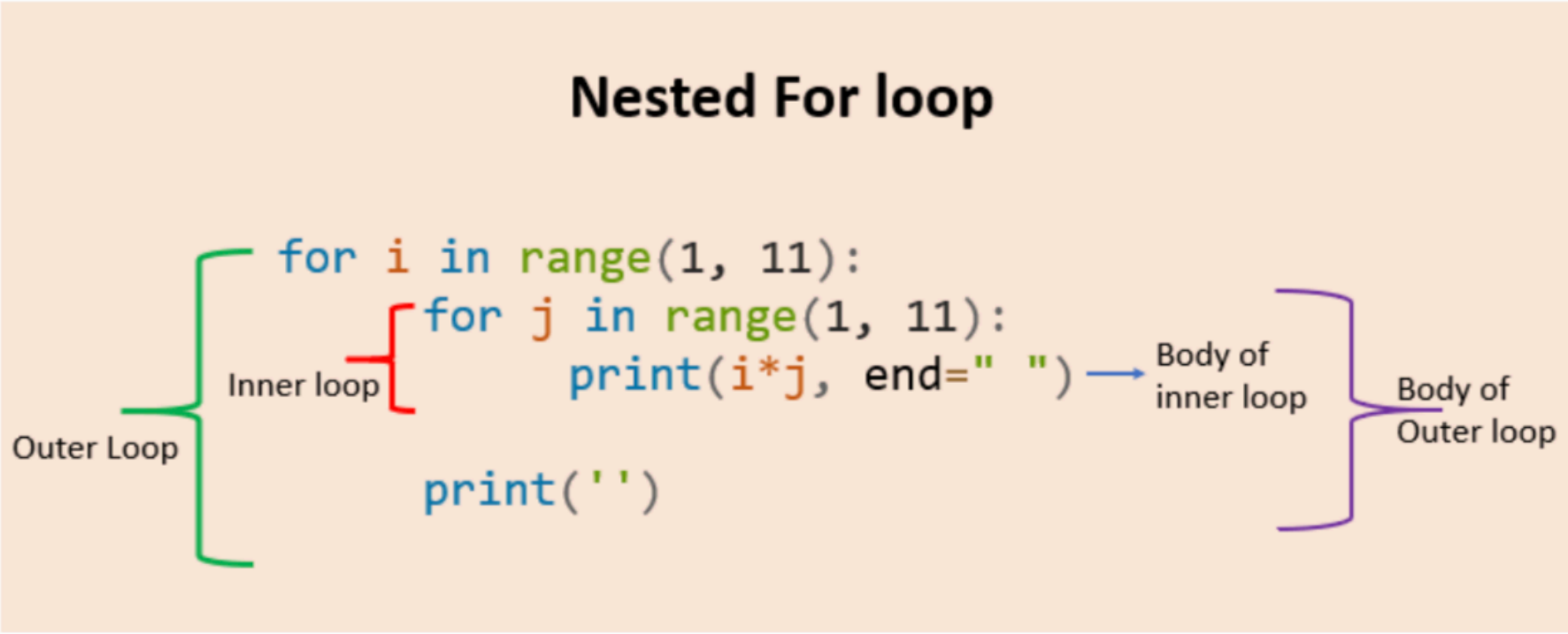
```
2, 4, 6,
```

Nested Loop

- A loop inside the body of the outer loop.
- In each iteration of the outer loop, inner loop execute all its iteration.
- Syntax

```
# outer for loop
for element in sequence:
    # inner for loop
    for element in sequence:
        body of inner for loop
    body of outer for loop
```

Example: Multiplication Table



Nested Loop Visualization

https://bouncmpe150.github.io/python-slides/lab3.html - /12

Example: Print Triangle

```
rows = 5
# outer loop
for i in range(1, rows + 1):
    # inner loop
    for j in range(1, i + 1):
        print("*", end=" ")
    print('')
```

the outer loop: the number of rows print.
The inner loop: the total number of columns in each row.

Loops with Turtle

Drawing a square in Turtle:

```
from turtle import *

drawing_area = Screen()
drawing_area.setup(width=750, height=500)

shape('square')
left(90)
forward(150)
left(90)
forward(150)
left(90)
forward(150)
left(90)
forward(150)

done()
```

Loops with Turtle

The same thing can be accomplished with a loop:

```
from turtle import *

drawing_area = Screen()
drawing_area.setup(width=750, height=500)

shape('square')
for i in range(4):
    left(90)
    forward(150)

done()
```

Python Random Module

- Python has a built-in module that you can use to make random numbers.
- The random module has a set of methods.
- Two of them are:
  -randint()
  -choice()

The randint() function

- Returns a random number between the given range.
- random.randint(start, stop)
  - **start:** an integer specifying at which position to start.
  - **stop:** an integer specifying at which position to stop.

Example: Return a number between 4 and 10 (both included)

```
import random

print(random.randint(4, 10))
```

The choice() function

- Returns a random element from the given sequence.
- The sequence can be a string, a range, a list, a tuple or any other kind of sequence.
- Random.choice(start, stop)
  - **start:** an integer specifying at which position to start.
  - **stop:** an integer specifying at which position to stop.

Example: Return a random element from a list

```
import random

mylist = ["apple", "banana", "cherry"]

print(random.choice(mylist))
```

Tuple

- Tuples are used to store multiple items in a single variable.
- A tuple is a collection which is ordered and **unchangeable**.
- Syntax: written with round brackets.

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```

Tuple Items

Ordered

- Items have a defined order and that order will not change.

Unchangeable

- We cannot changei add, or remove items after the tüple has been created.

Allow Duplicates

- Since tuples are indexed, they can have items with the same value.

Tuple Items – Data Types

Tuple items can be of any data type.

```
tuple1 = ("apple", "banana", "cherry")
tuple2 = (1, 5, 7, 9, 3)
tuple3 = (True, False, False)
```

A tuple can contain different data types.

```
tuple1 = ("abc", 34, True, 40, "male")
```

References

1. https://bouncmpe150.github.io/python-slides/lab3.html
2. https://pythontutor.com/
3. https://realpython.com/python-for-loop/
4. https://www.programiz.com/python-programming/for-loop
5. https://pynative.com/python-nested-loops/
6. https://www.w3schools.com/python/python_tuples.asp
7. https://www.w3schools.com/python/module_random.asp
8. https://vegibit.com/how-to-use-loops-with-python-turtle/