



Welcome to the Web Application Development with Python L-4 Training Program





MD. SHAKIL

Asst. Trainer

Web Application Development with Python

Daffodil International Professional Training Institute

Cell: +880 1793-299882

Email: shakil.eub.cse@gmail.com

Lecture

07

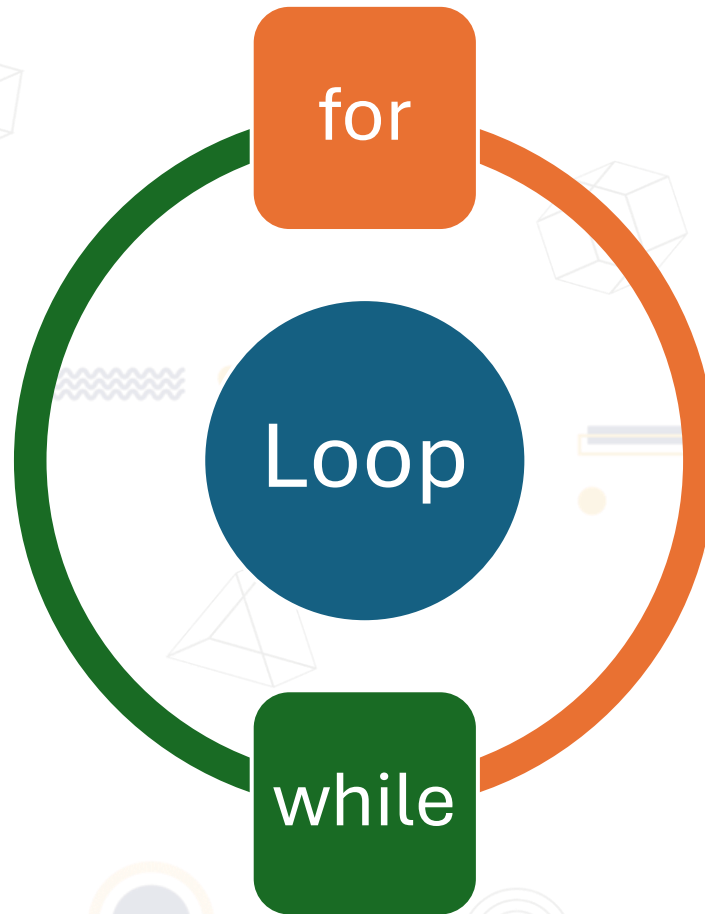


✓ Python Loop

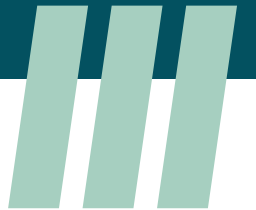
Why use **Loops**?

- ☐ To reduce redundancy
- ☐ To process collections (like lists)
- ☐ To repeat a task multiple times

Types of Loop in Python



Python for Loop



for Loop Syntax



```
for variable in sequence:  
    # code block
```

Loop Through a List

With the **for loop** we can execute a set of statements, once for each item in a list, tuple, set etc.

```
for-loop.py

languages = ['Swift', 'Python', 'Go']

# access elements of the list one by one
for lang in languages:
    print(lang)
```

```
output.txt

Swift
Python
Go
```


Loop Through a String

```
language = 'Python'

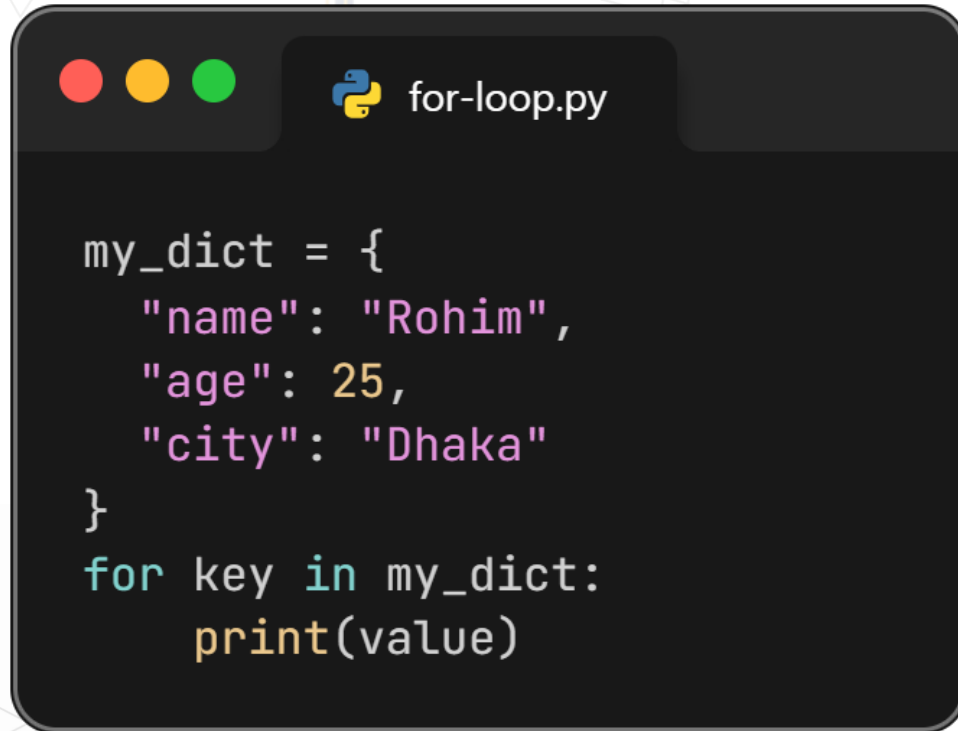
# iterate over each character in language
for x in language:
    print(x)
```

```
output.txt

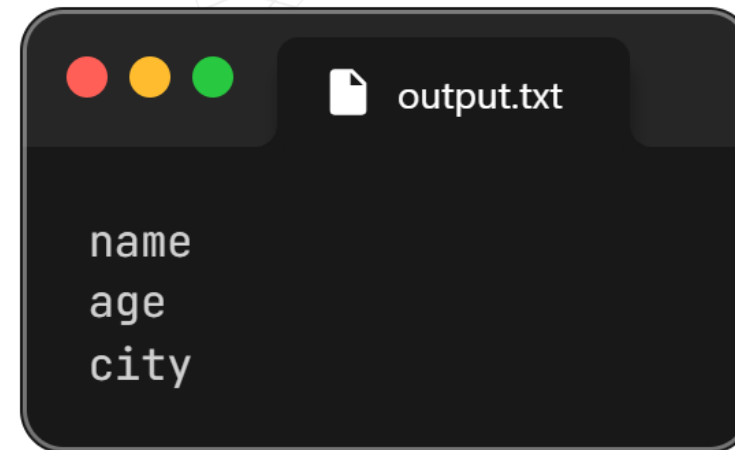
P
y
t
h
o
n
```

Loop Through a Dictionary

a) Loop Through Keys



```
my_dict = {  
    "name": "Rohim",  
    "age": 25,  
    "city": "Dhaka"  
}  
for key in my_dict:  
    print(value)
```



```
name  
age  
city
```

Loop Through a Dictionary

b) Loop Through Values

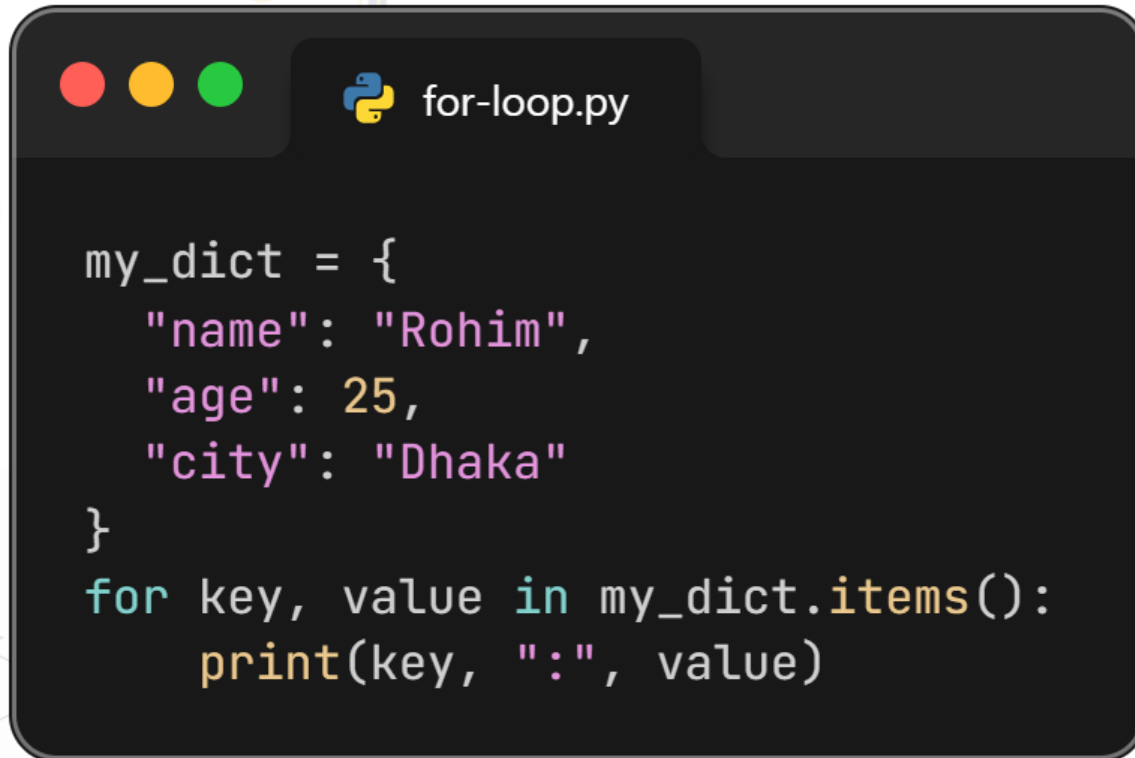
```
my_dict = {  
    "name": "Rohim",  
    "age": 25,  
    "city": "Dhaka"  
}  
for value in my_dict.values():  
    print(value)
```

output.txt


```
Rohim  
25  
Dhaka
```

Loop Through a Dictionary

c) Loop Through Key-Value Pair



```
my_dict = {  
    "name": "Rohim",  
    "age": 25,  
    "city": "Dhaka"  
}  
for key, value in my_dict.items():  
    print(key, ":", value)
```



```
name : Shakil  
age : 25  
city : Dhaka
```

for Loop with Python **range()**

In Python, the **range()** function returns a sequence of numbers.

```
for-loop.py

#Example: 1
for x in range(4):
    print(x)
#Example: 2
for x in range(2, 4):
    print(x)
```

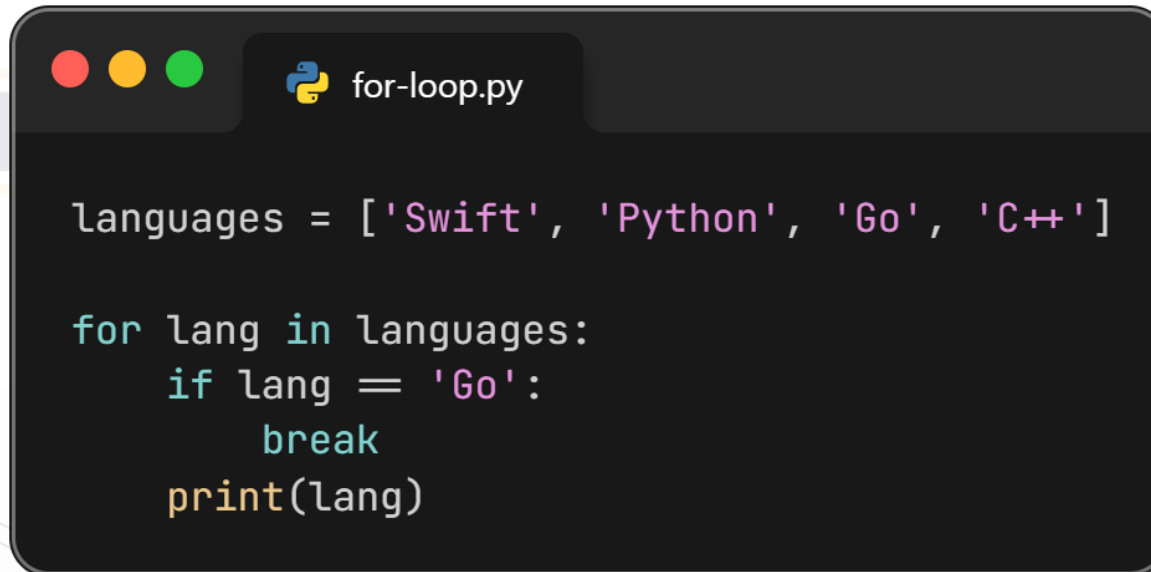
```
output.txt

#Output: 1
0
1
2
3

#Output: 2
2
3
```

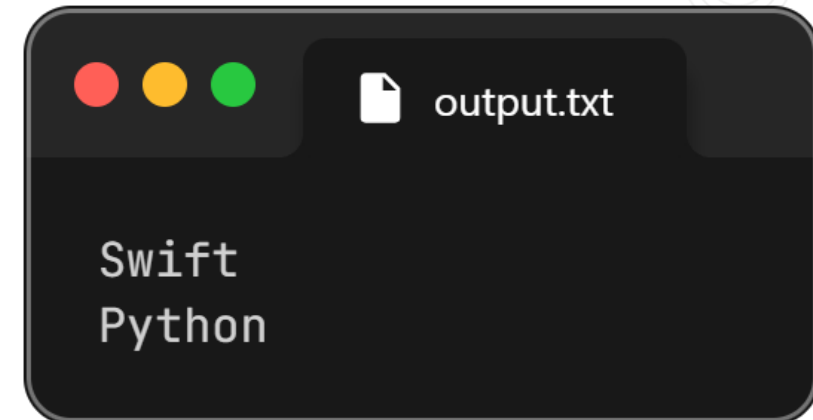
Break Statement

The **break** statement terminates the **for loop** immediately before it loops through all the items.



```
languages = ['Swift', 'Python', 'Go', 'C++']

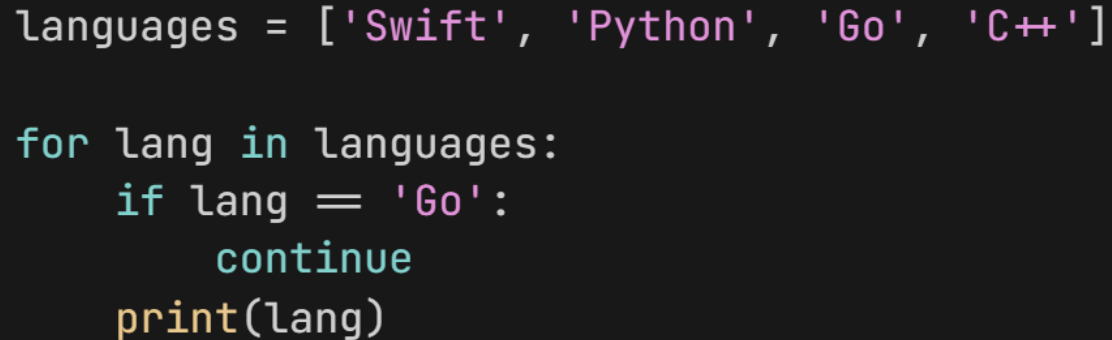
for lang in languages:
    if lang == 'Go':
        break
    print(lang)
```



```
Swift
Python
```

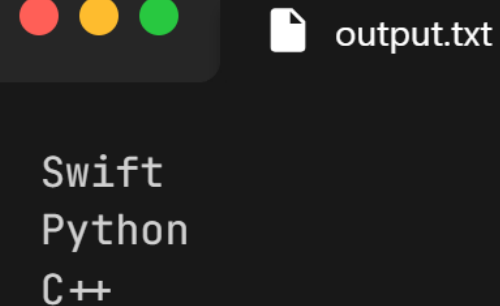
Continue Statement

The **continue** statement skips the current iteration of the loop and continues with the next iteration.



```
languages = ['Swift', 'Python', 'Go', 'C++']

for lang in languages:
    if lang == 'Go':
        continue
    print(lang)
```



```
Swift
Python
C++
```



Thanks All

