

Lecture 6

Repetition: Python for Loop

for Loop

- In Python for loop is used to iterate over the items of any sequence including the Python list, string, tuple, etc.
- For for loop is also used to access elements from container (for example list, string, tuple) using built-in function range().

for Loop

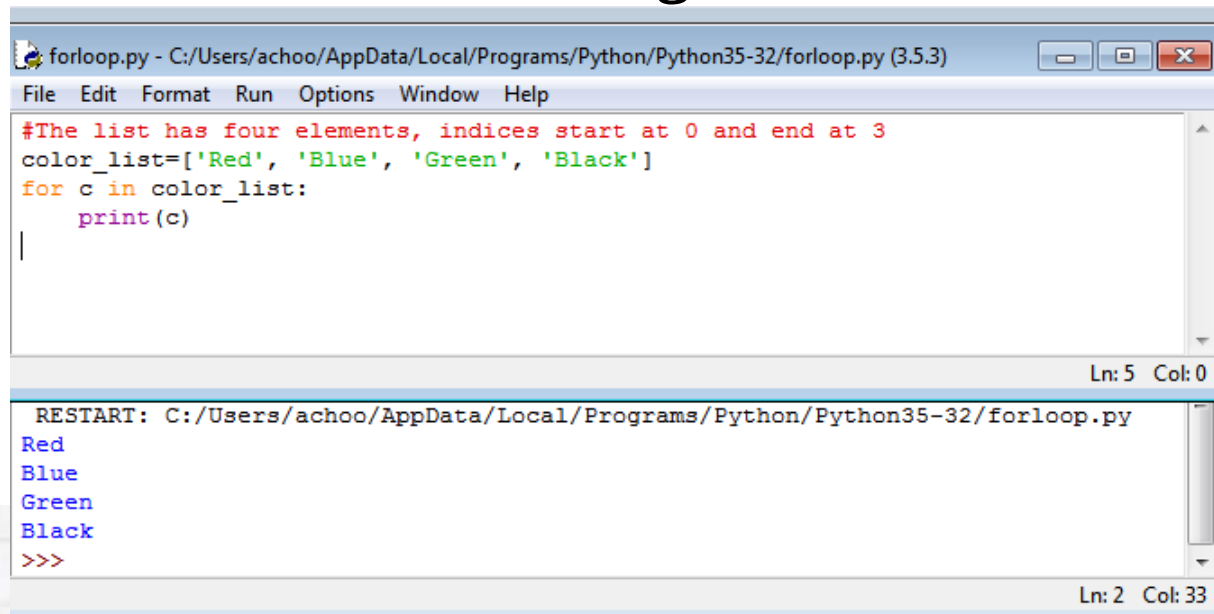
- **Syntax :** for variable_name in sequence:
 statement_1
 statement_2

- **Parameter:**

Name	Description
variable_name	It indicates target variable which will set a new value for each iteration of the loop.
sequence	A sequence of values that will be assigned to the target variable variable_name. Values are provided using a list or a string or from the built-in function range().
statement_1, statement_2,	Block of program statements.

for Loop

- When the for loop executed the first item (i.e. Red) is assigned to the variable c.
- The print statement will execute and the process will continue until reaching the end of the list.



The screenshot shows a Python IDE window titled 'forloop.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py (3.5.3)'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The editor contains the following code:

```
#The list has four elements, indices start at 0 and end at 3
color_list=['Red', 'Blue', 'Green', 'Black']
for c in color_list:
    print(c)
```

The status bar at the bottom right of the editor indicates 'Ln: 5 Col: 0'. Below the editor, the output console shows the results of the script execution:

```
RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py
Red
Blue
Green
Black
>>>
```

The output console status bar at the bottom right indicates 'Ln: 2 Col: 33'.

Python for Loop and range() function

- The range() function returns a list of consecutive integers.
- The function has one, two or three parameters where last two parameters are optional.
- It is widely used in for loops.
- Syntax:

`range(a)`

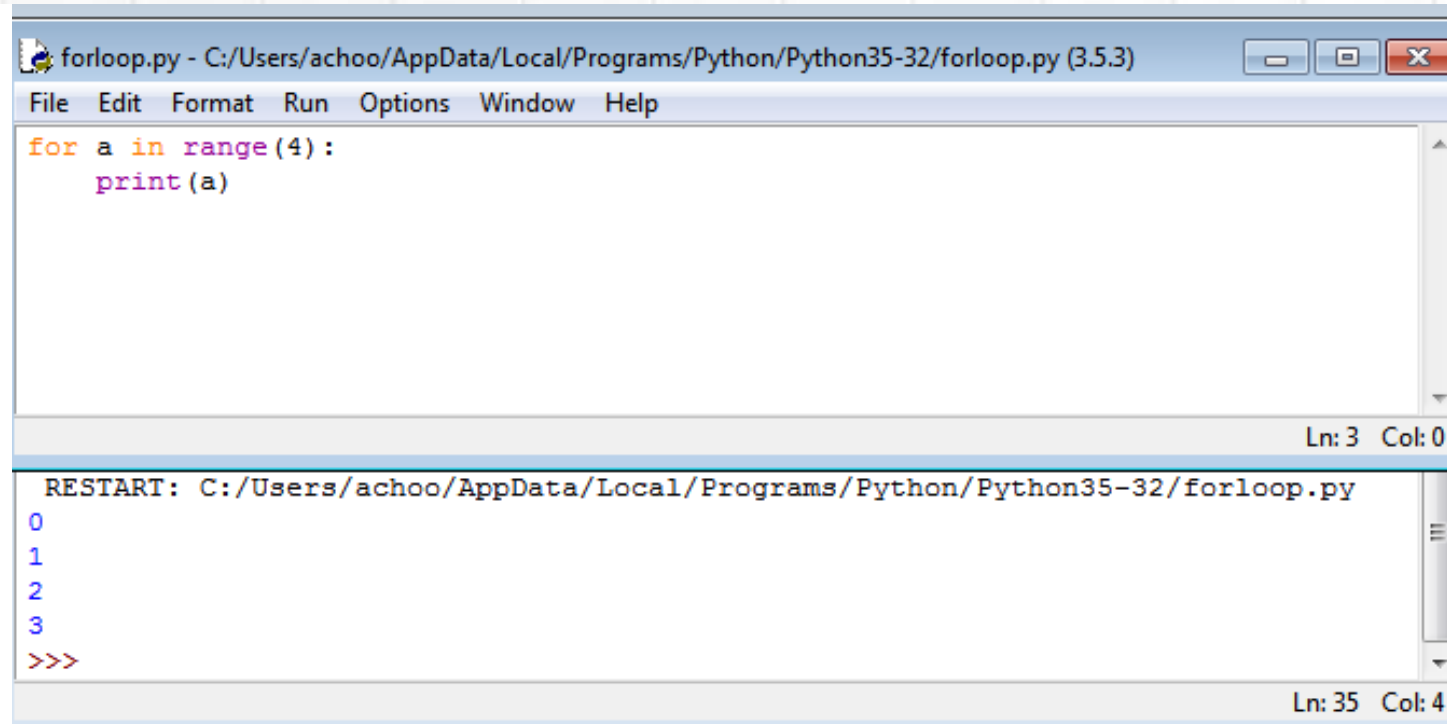
`range(a,b)`

`range(a,b,c)`

Python for Loop and range() function

- range(a): Generates a sequence of numbers from 0 to a, excluding a, incrementing by 1.
- Syntax:
`for<variable> in range(<number>):`

Python for Loop and range() function



The screenshot shows a Python IDE window titled "forloop.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py (3.5.3)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The editor contains the following code:

```
for a in range(4):  
    print(a)
```

The status bar at the bottom right of the editor indicates "Ln: 3 Col: 0". Below the editor is a console window showing the output of the program:

```
RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py  
0  
1  
2  
3  
>>>
```

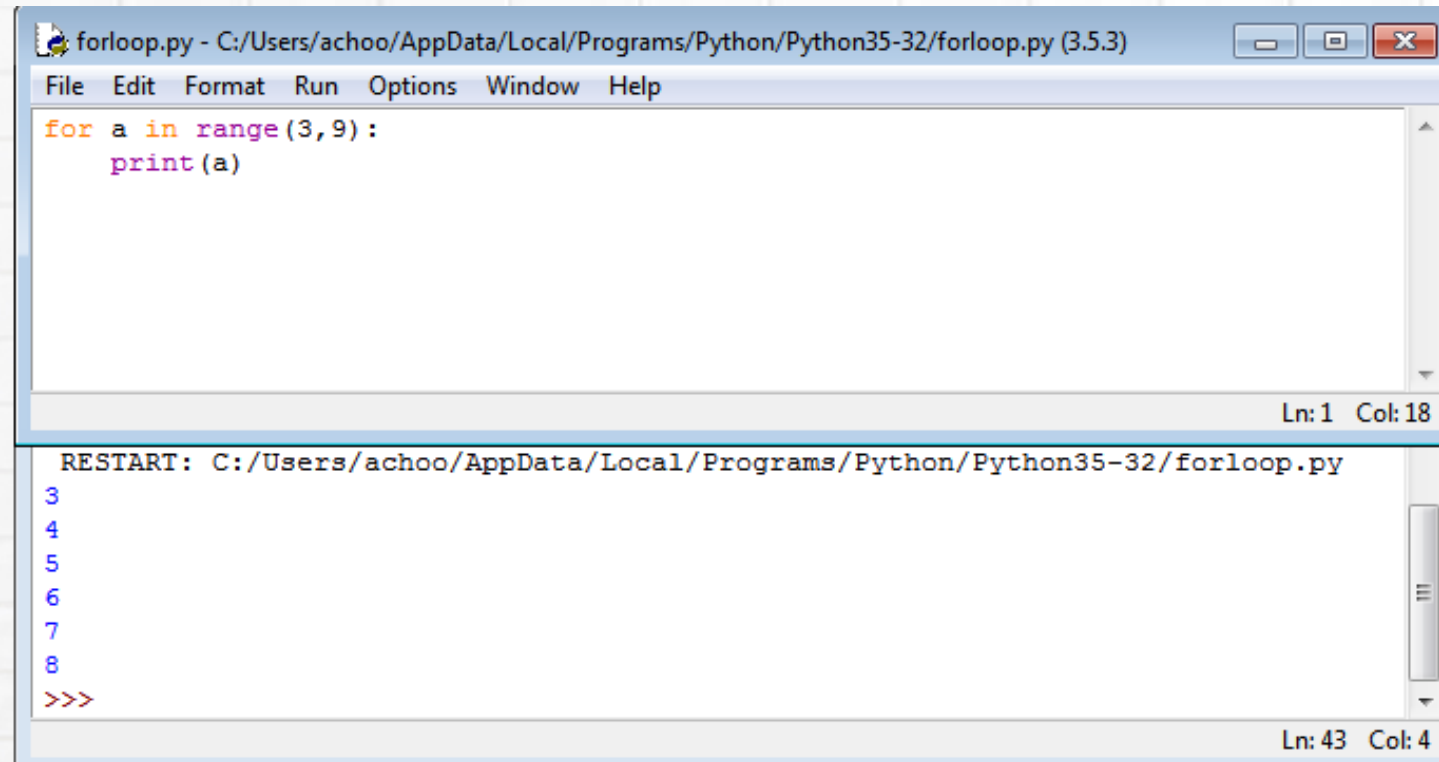
The console window status bar at the bottom right indicates "Ln: 35 Col: 4".

Python for Loop and range() function

- range(a,b): Generates a sequence of number from a to b excluding b, incrementing by 1.
- Syntax:

```
for<variable> in range(<"start_number", "end_number">):
```


Python for Loop and range() function



The screenshot shows a Python IDE window titled "forloop.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py (3.5.3)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The editor area contains the following code:

```
for a in range(3, 9):  
    print(a)
```

The status bar at the bottom right of the editor indicates "Ln: 1 Col: 18". Below the editor is a console window showing the output of the program:

```
RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py  
3  
4  
5  
6  
7  
8  
>>>
```

The console status bar at the bottom right indicates "Ln: 43 Col: 4".

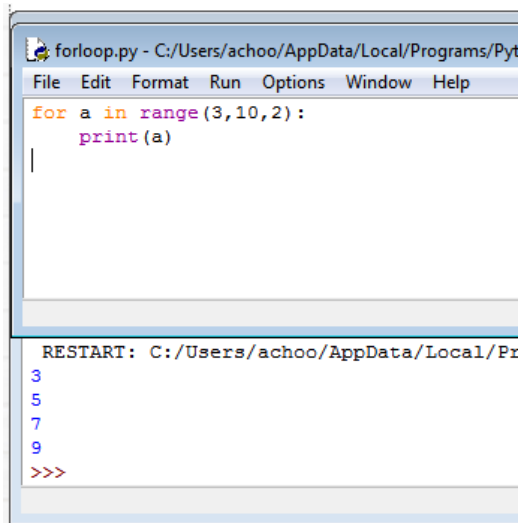
Python for Loop and range() function

- range(a,b, c): Generates a sequence of number from a to b excluding b, incrementing by c.
- Syntax:

```
for<variable> in range(<"start_number", "end_number", "increase_number">):
```

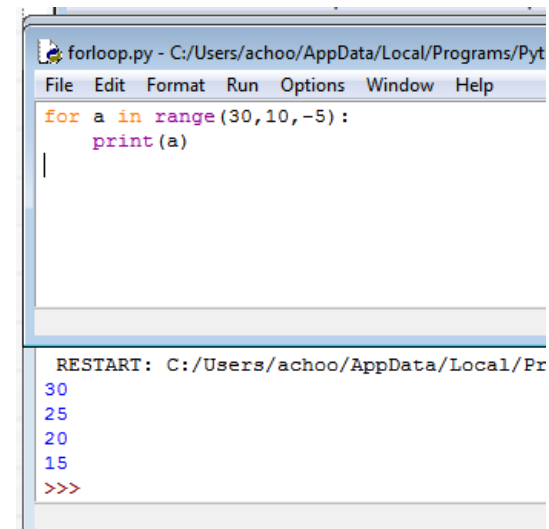
Python for Loop and range() function

- Increase loop
- Decrease loop



```
forloop.py - C:/Users/achoo/AppData/Local/Programs/Pyt
File Edit Format Run Options Window Help
for a in range(3,10,2):
    print(a)

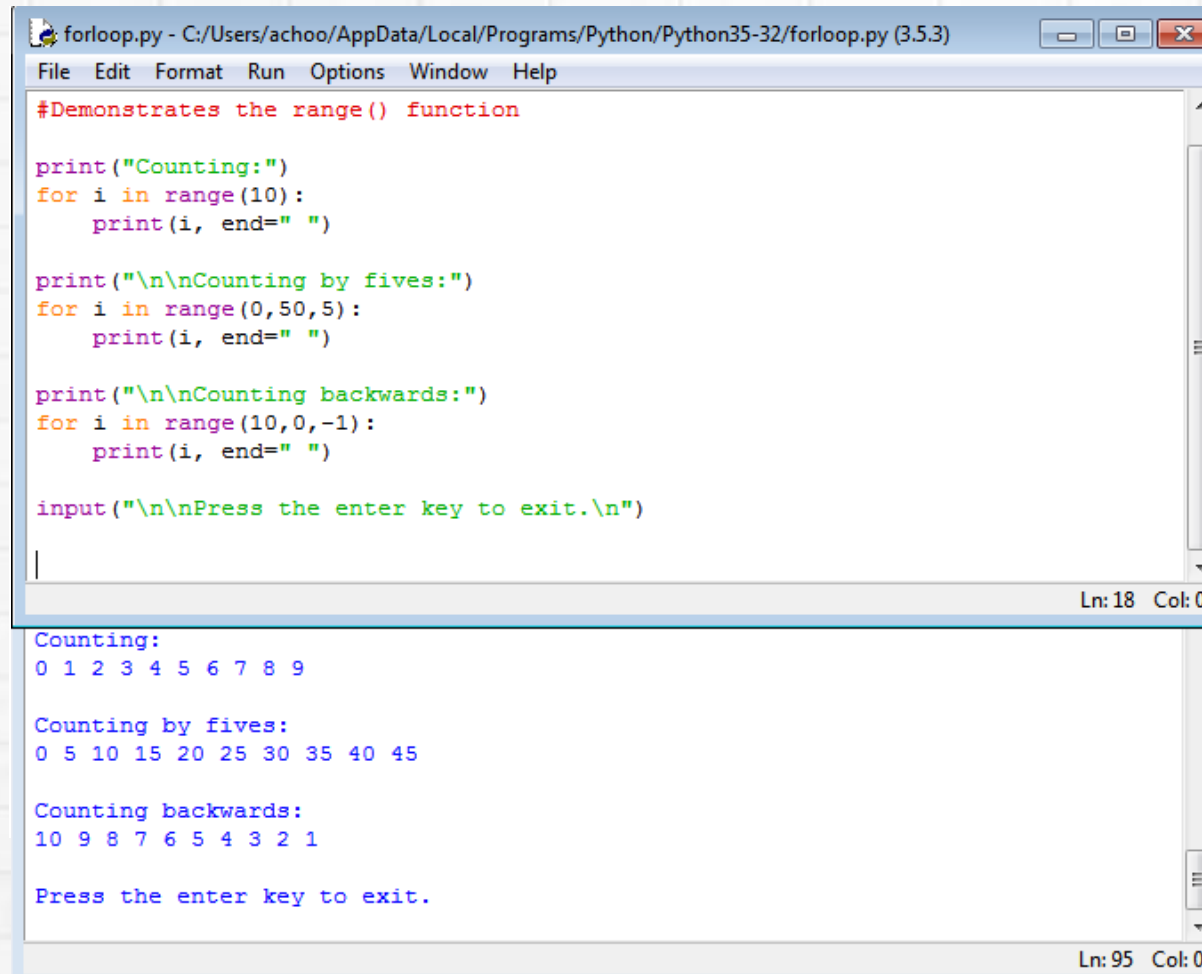
RESTART: C:/Users/achoo/AppData/Local/Pr
3
5
7
9
>>>
```



```
forloop.py - C:/Users/achoo/AppData/Local/Programs/Pyt
File Edit Format Run Options Window Help
for a in range(30,10,-5):
    print(a)

RESTART: C:/Users/achoo/AppData/Local/Pr
30
25
20
15
>>>
```

Demonstrates the range() function



```
forloop.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py (3.5.3)
File Edit Format Run Options Window Help
#Demonstrates the range() function

print("Counting:")
for i in range(10):
    print(i, end=" ")

print("\n\nCounting by fives:")
for i in range(0,50,5):
    print(i, end=" ")

print("\n\nCounting backwards:")
for i in range(10,0,-1):
    print(i, end=" ")

input("\n\nPress the enter key to exit.\n")

|
```

Ln: 18 Col: 0

```
Counting:
0 1 2 3 4 5 6 7 8 9

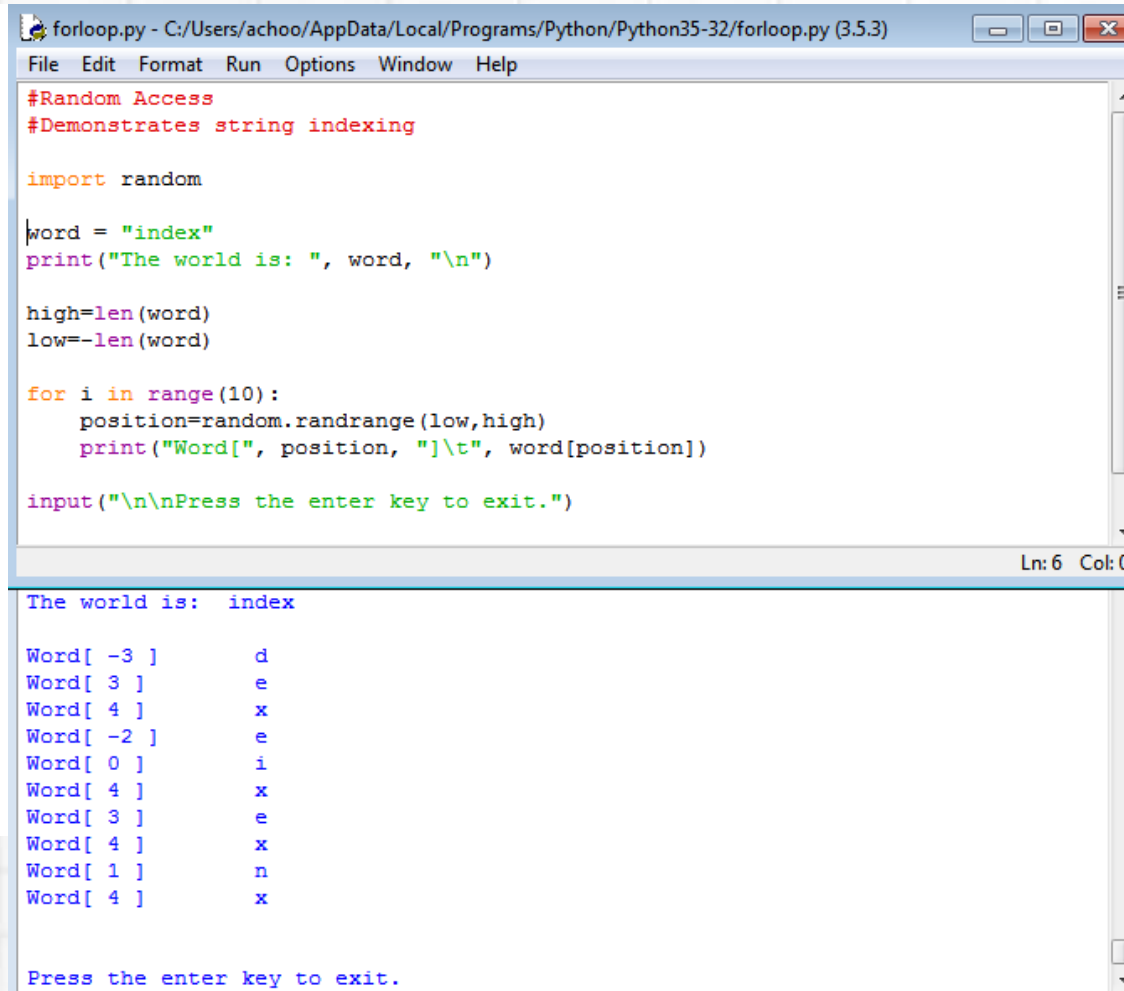
Counting by fives:
0 5 10 15 20 25 30 35 40 45

Counting backwards:
10 9 8 7 6 5 4 3 2 1

Press the enter key to exit.
```

Ln: 95 Col: 0

Random Access with len() function



```
forloop.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py (3.5.3)
File Edit Format Run Options Window Help

#Random Access
#Demonstrates string indexing

import random

word = "index"
print("The world is: ", word, "\n")

high=len(word)
low=-len(word)

for i in range(10):
    position=random.randrange(low,high)
    print("Word[", position, "]\t", word[position])

input("\n\nPress the enter key to exit.")

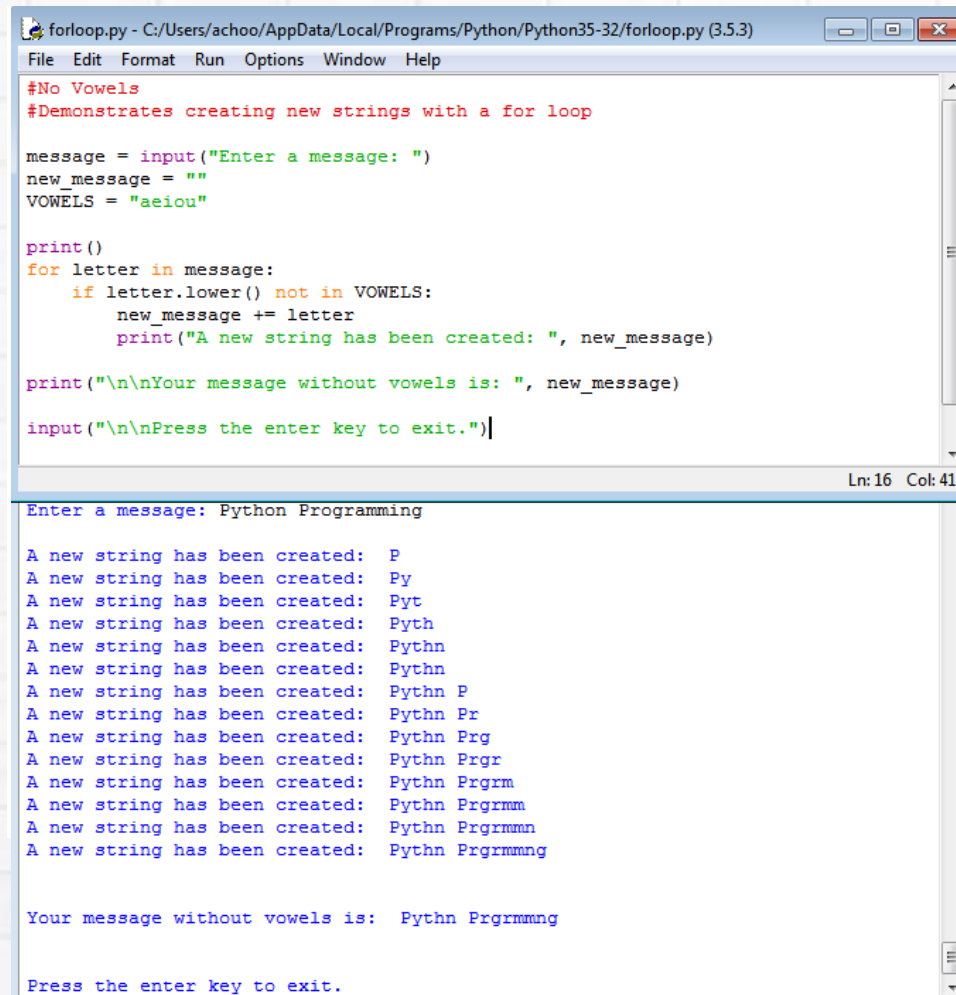
Ln: 6 Col: 0

The world is:  index

Word[ -3 ]      d
Word[ 3 ]       e
Word[ 4 ]       x
Word[ -2 ]      e
Word[ 0 ]       i
Word[ 4 ]       x
Word[ 3 ]       e
Word[ 4 ]       x
Word[ 1 ]       n
Word[ 4 ]       x

Press the enter key to exit.
```

New strings with a for Loop



```
forloop.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py (3.5.3)
File Edit Format Run Options Window Help

#No Vowels
#Demonstrates creating new strings with a for loop

message = input("Enter a message: ")
new_message = ""
VOWELS = "aeiou"

print()
for letter in message:
    if letter.lower() not in VOWELS:
        new_message += letter
        print("A new string has been created: ", new_message)

print("\n\nYour message without vowels is: ", new_message)

input("\n\nPress the enter key to exit.")
```

Ln: 16 Col: 41

Enter a message: Python Programming

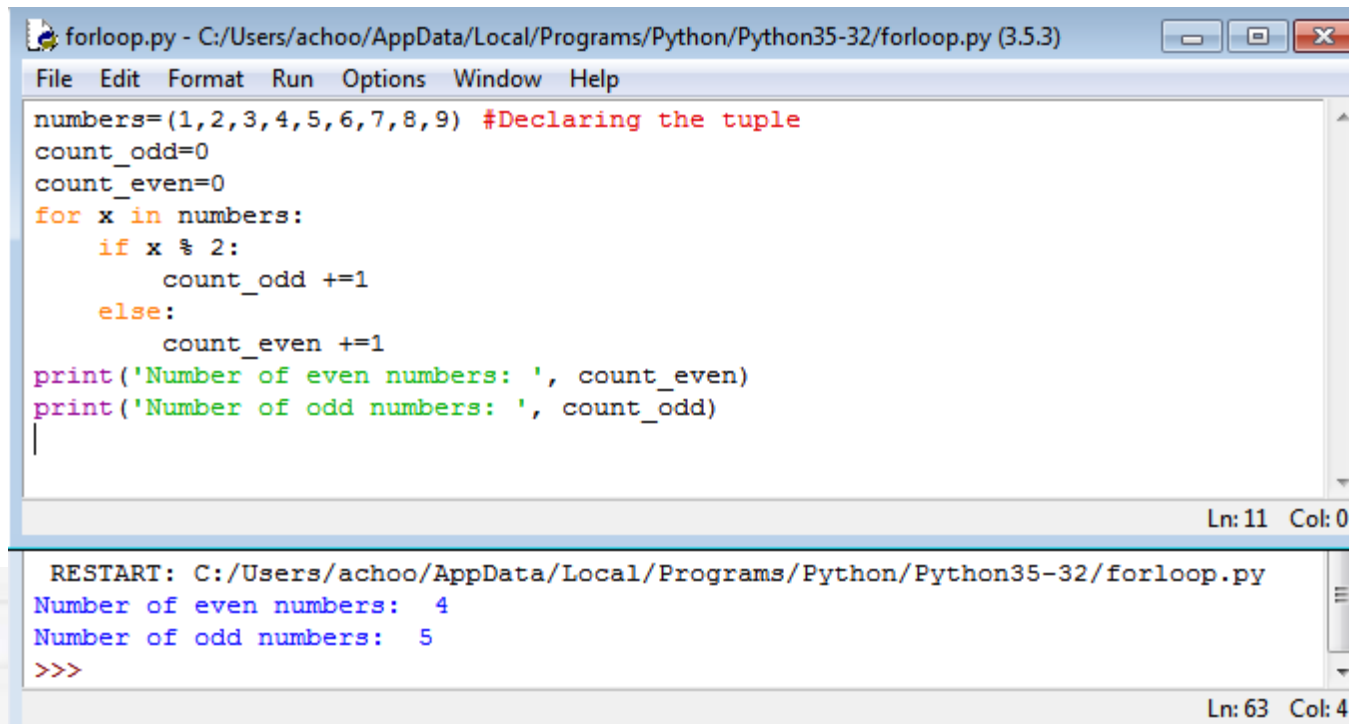
A new string has been created: P
A new string has been created: Py
A new string has been created: Pyt
A new string has been created: Pyth
A new string has been created: Pythn
A new string has been created: Pythn
A new string has been created: Pythn P
A new string has been created: Pythn Pr
A new string has been created: Pythn Prg
A new string has been created: Pythn Prgr
A new string has been created: Pythn Prgrm
A new string has been created: Pythn Prgrmm
A new string has been created: Pythn Prgrmmn
A new string has been created: Pythn Prgrmmng

Your message without vowels is: Pythn Prgrmmng

Press the enter key to exit.

Python for loop: Iterating over tuple, list, dictionary

- Iterating over tuple
- The following example counts the number of even and odd numbers from a series of numbers.

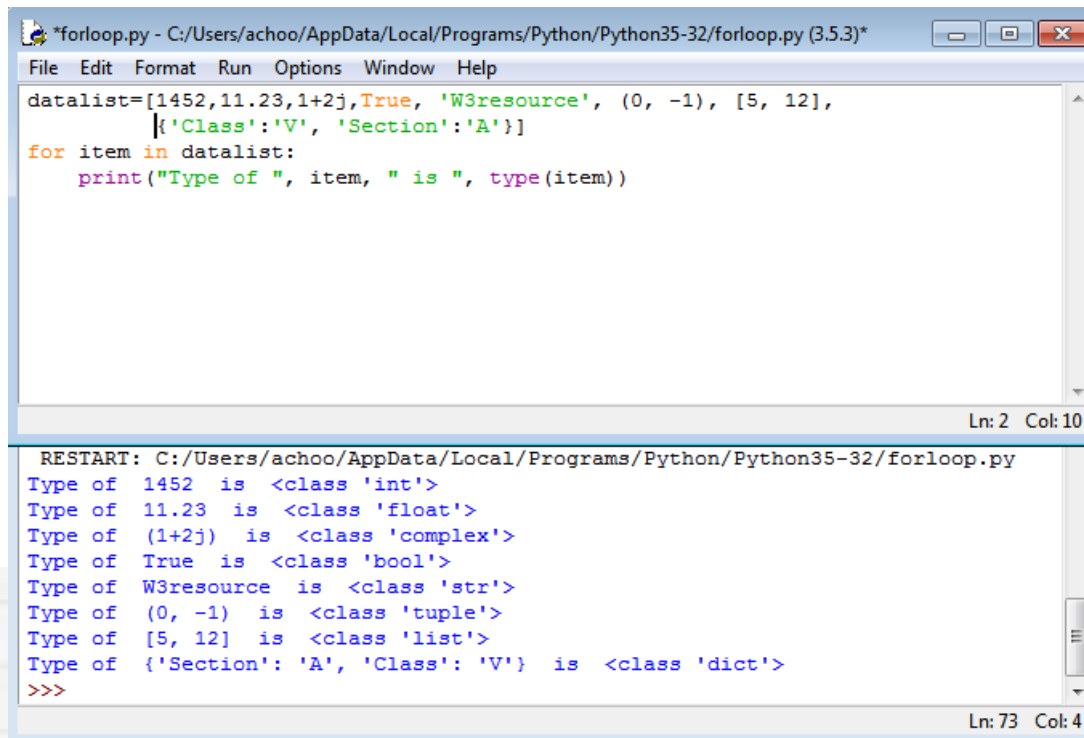


```
forloop.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py (3.5.3)
File Edit Format Run Options Window Help
numbers=(1,2,3,4,5,6,7,8,9) #Declaring the tuple
count_odd=0
count_even=0
for x in numbers:
    if x % 2:
        count_odd +=1
    else:
        count_even +=1
print('Number of even numbers: ', count_even)
print('Number of odd numbers: ', count_odd)
|
Ln: 11 Col: 0

RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py
Number of even numbers:  4
Number of odd numbers:  5
>>>
Ln: 63 Col: 4
```

Python for loop: Iterating over tuple, list, dictionary

- Iterating over list
- The following example for loop iterates through the list “datalist” and prints each item and its corresponding.



```
*forloop.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py (3.5.3)*
File Edit Format Run Options Window Help
datalist=[1452,11.23,1+2j,True, 'W3resource', (0, -1), [5, 12],
          {'Class':'V', 'Section':'A'}]
for item in datalist:
    print("Type of ", item, " is ", type(item))

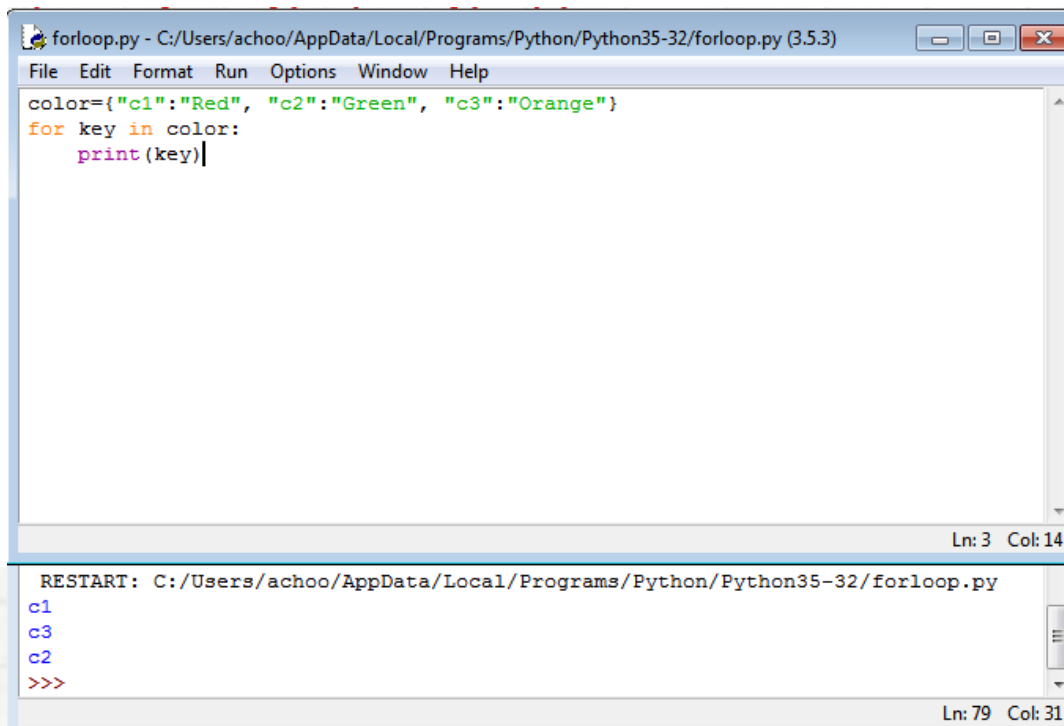
Ln: 2 Col: 10

RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py
Type of 1452 is <class 'int'>
Type of 11.23 is <class 'float'>
Type of (1+2j) is <class 'complex'>
Type of True is <class 'bool'>
Type of W3resource is <class 'str'>
Type of (0, -1) is <class 'tuple'>
Type of [5, 12] is <class 'list'>
Type of {'Section': 'A', 'Class': 'V'} is <class 'dict'>
>>>

Ln: 73 Col: 4
```


Python for loop: Iterating over tuple, list, dictionary

- Iterating over dictionary
- The following example for loop iterates through the dictionary "color" through its keys and prints each key.



```
forloop.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py (3.5.3)
File Edit Format Run Options Window Help
color={"c1":"Red", "c2":"Green", "c3":"Orange"}
for key in color:
    print(key)

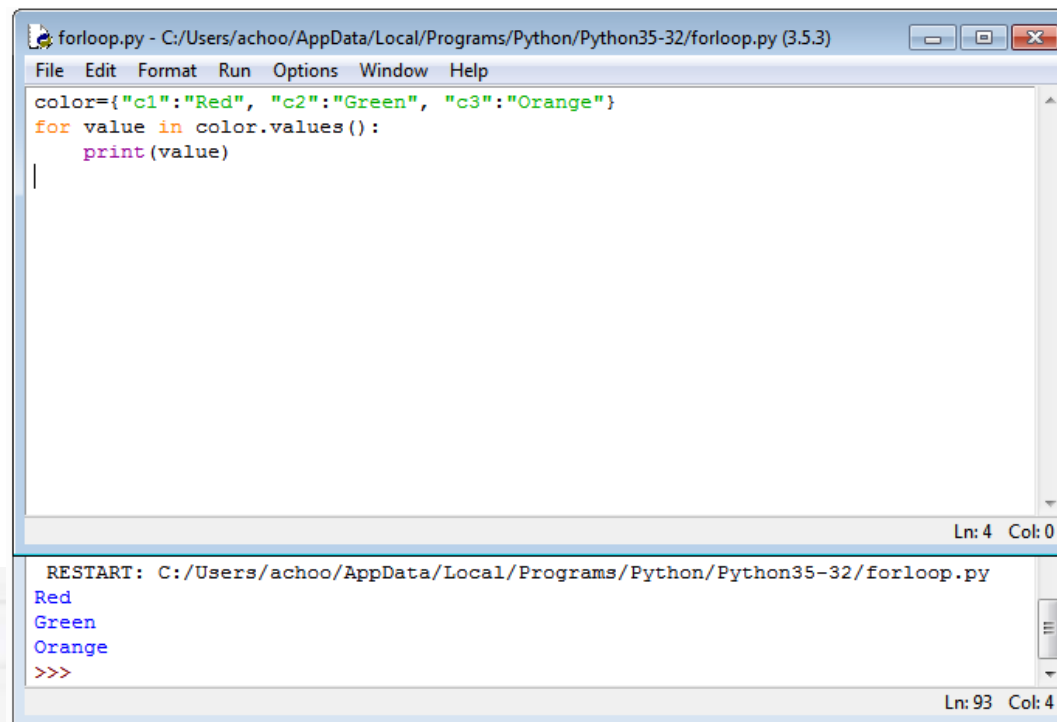
Ln: 3 Col: 14

RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py
c1
c3
c2
>>>

Ln: 79 Col: 31
```

Python for loop: Iterating over tuple, list, dictionary

- Iterating over dictionary
- The following example for loop iterates through the dictionary "color" through its values and prints each value.



The screenshot shows a Python IDE window titled "forloop.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py (3.5.3)". The code in the editor is:

```
color={"c1":"Red", "c2":"Green", "c3":"Orange"}
for value in color.values():
    print(value)
```

The output console at the bottom shows the execution results:

```
RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/forloop.py
Red
Green
Orange
>>>
```