

Lecture 06: Lists, Strings, Loops

Swakkhar Shatabda

B.Sc. in Data Science
Department of Computer Science and Engineering
United International University

January 17, 2024

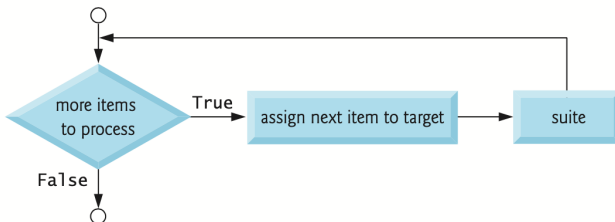


Iterating

- The `for` statement allows you to repeat an action or several actions.
- The `for` statement performs its action(s) for each item in a sequence of items.

```
c = [-45, 6, 0, 72, 1543]
for x in c:
    print(x, end=" ")
```

-45 6 0 72 1543



for loop

```
for character in 'Programming':  
    print(character, end='  ')
```

P r o g r a m m i n g

- display 'Programming' with its characters separated by two spaces



for loop

```
c = [1,4,7,3,5]
for x in c:
    x = 2
    print(x)
print(c)
```

2

2

2

2

2

[1, 4, 7, 3, 5]

- Changing x has got no effect on the list.



range function

- The function call `range(m)` creates an iterable object that represents a sequence of consecutive integer values starting from 0 and continuing up to, but not including, the argument value `m`.

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

0 1 2 3 4 5 6 7 8 9

- It has got a two argument version.

```
for counter in range(2,7):  
    print(counter, end=' ')
```

2 3 4 5 6



range function

- And a 3 argument version as well!!!

```
for counter in range(1,15,3):  
    print(counter, end=' ')
```

1 4 7 10 13

```
for counter in range(100,15,-15):  
    print(counter, end=' ')
```

100 85 70 55 40 25



Augmented Assignments

- Augmented assignments abbreviate assignment expressions in which the same variable name appears on the left and right of the assignment.

```
total = 0
for number in [1, 2, 3, 4, 5]:
    total = total + number
print(total)
```

```
total = 0
for number in [1, 2, 3, 4, 5]:
    total += number    # add number to total
print(total)
```



Augmented Assignments

Augmented assignment	Sample expression	Explanation	Assigns
<i>Assume:</i> $c = 3$, $d = 5$, $e = 4$, $f = 2$, $g = 9$, $h = 12$			
<code>+=</code>	<code>c += 7</code>	<code>c = c + 7</code>	10 to <code>c</code>
<code>-=</code>	<code>d -= 4</code>	<code>d = d - 4</code>	1 to <code>d</code>
<code>*=</code>	<code>e *= 5</code>	<code>e = e * 5</code>	20 to <code>e</code>
<code>**=</code>	<code>f **= 3</code>	<code>f = f ** 3</code>	8 to <code>f</code>
<code>/=</code>	<code>g /= 2</code>	<code>g = g / 2</code>	4.5 to <code>g</code>
<code>//=</code>	<code>g //= 2</code>	<code>g = g // 2</code>	4 to <code>g</code>
<code>%=</code>	<code>h %= 9</code>	<code>h = h % 9</code>	3 to <code>h</code>



Formatted Strings

- The letter f before the string's opening quote indicates it's an f-string. You specify where to insert values by using placeholders delimited by curly braces (and).

```
grade = 13
total = 20
print(f'Your marks is {grade} out of {total}')
print(f'In percentage is {100*grade/total}%')
```

Your marks is 13 out of 20

In percentage is 65.0%



Operators on lists/strings

```
a=[1,2,3]  
b=[4,5]  
c=a+b  
print(c)
```

[1, 2, 3, 4, 5]

```
a="Bangladesh"  
b=" Wins"  
c=a+b  
print(c)
```

Bangladesh Wins



A few string functions

- The `title()` method changes each word to title case, where each word begins with a capital letter.

```
name = "ada lovelace"  
print(name.title())
```

Ada Lovelace

```
name = "Ada Lovelace"  
print(name.upper())  
print(name.lower())
```

ADA LOVELACE

ada lovelace



A few string functions

- To ensure that no whitespace exists at the right side of a string, use the `rstrip()`

```
s = "This string      "  
print(s,end="END\n")  
print(s.rstrip(),end="END\n")  
print(s,end="END\n")
```

This string END

This stringEND

This string END

- You can also strip whitespace from the left side of a string using the `lstrip()` method, or from both sides at once using `strip()`



Prefix and Suffix

```
s = "https://www.uiu.ac.bd/"  
print(s.removeprefix("https://"))  
  
t = "HelloWorld.py"  
print(t.removesuffix(".py"))
```

www.uiu.ac.bd/
HelloWorld



Quotes in quotes

```
s = "my 'world'"
print(s)
print('exactly "Twice" is correct')
```

```
my 'world'
exactly "Twice" is correct
```

Sometimes

```
print('my 'world')
```

File "/Users/swakkhar/helloList.py", line 1

```
print('my 'world')
```

SyntaxError: unterminated string literal (detected at line 1)

While Loop

- The `while` statement allows you to repeat one or more actions while a condition remains True.

```
power = 2
while power <= 50:
    print(power,end=" ")
    power = power * 2
print("")
print("Outside Loop",power)
```

2 4 8 16 32

Outside Loop 64

