

Lecture 14: Introduction to Computer Programming Course - CS1010

DEPARTMENT OF COMPUTER SCIENCE | 03/11/2019



Rensselaer

Announcements

- Homework 6 is due tonight.
- Homework 7 will be posted tonight
- Will be due next Monday
- Exam 2 is scheduled for Thursday (March 21)
- We will review the exam next Monday (March 18)

Goals for today

- For Loops
- Problems on For Loops

Rule to remember

- For all while loops we must:
 - Initialize
 - Give condition (of while)
 - Specify action
 - Increment and/or decrement

For Loops

- For Loop is a counted Loop:
 - We know how many iterations are required to accomplish a task
- Many objects in Python are 'iterable'.
- We can iterate over every element in an object.
- For example we can iterate over every element in a string or a list.
- For loops can be used to execute a block of code for every iteration.

Syntax

- for variable in list/tuple/string:
- block of code

```
for c in ('abcd'):  
    print (c)
```

Output:

```
a  
b  
c  
d
```

Overwriting a Loop

- Changing elements of a List:
- For example triple the elements of a list.

```
Values=[1,2,3,4]
```

```
for v in Values:
```

```
    v=3*v
```

```
    print(v)
```

Values stays the same above

We can do `Values[0]=Values[0]*3`

For this we need to know the right index of each element.

Range for Numbers

- A call to `range(start, stop)` returns a list of integers from start to the first integer before stop.
- A call to `range` with a single argument is equivalent to a call to `range(0, argument)`.
- Q: Produce a list of Leap years in the first half of this century:
- Range (start, stop, step)
 for x in range(2000, 2050, 4):
 print(x, end=" ")
[2000, 2004, 2008, 2012, 2016, 2020, 2024, 2028, 2032, 2036, 2040, 2044, 2048]

Range continued...

- The step size can also be negative, but when it is, the starting index should be larger than the stopping index:
 - `for x in range(2048,2000,-4):`
 - `print(x, end=" ")`
 - `[2048 2044 2040 2036 2032 2028 2024 2020 2016 2012 2008 2004]`
- Example:
 - `values = ['a','b','c']`
 - `len(values)`
 - `list(range(3))`
 - `list(range(len(values)))`
 - Result: `[0,1,2]`

Printing index and values

- For a give list print its values and index

Over-write elements in a list

- Replace a list with a single value
- Replace element of a list with twice its value

Enumerate function

- for x in enumerate('abc'):
- print (x)
- Result:
- (0, 'a')
- (1, 'b')
- (2, 'c')

Enumerate continued

- `values=[1,2,3,4]`

-

Else in For loop

- ```
for x in range(6):
 print(x)
else:
 print("Finally finished!")
```

# Nested for loop

- A nested loop is a loop inside a loop.
  - The "inner loop" will be executed one time for each iteration of the "outer loop":
- `adj = ["red", "big", "tasty"]`
  - `fruits = ["apple", "banana", "cherry"]`
  - `for x in adj:`
  - `for y in fruits:`
  - `print(x, y)`

# Iterate over portion of a string

- To iterate over a portion of string like a sub string , we can use slicing operator to generate a sub string and then iterate over that sub string.
- To generate a slice we will use [] operator i.e.
- **string[start : stop : step size]**



# Iterate over string

- Given a string, iterate over the first 3 elements of the string.

# Iterate

- Over a string by skipping characters
- Over string in backward / reverse direction using slicing

# Problem 1

- Given an integer as input, write a function that finds its factorial.

## Problem 2

- Define a function called `count` that has two arguments called `sequence` and `item`. Return the number of times the item occurs in the list. For example: `count([1,2,1,1], 1)` should return 3 (because 1 appears 3 times in the list).

# Algorithm for Printing Patterns

- We need to use two for loops to print pattern, i.e. nested loops.
- **There is a typical structure to print any pattern, i.e. the number of rows and column in the pattern.**
- Outer loop tells us the number of rows used and inner loop tells us the column used to print pattern.
- Accept the number rows user want to print in the pattern.
- Iterate those number using outer for loop to handle the number of rows.
- Inner loop to handle the number of columns. Inner loop iteration depends on the values of the outer loop.
- Print start, number, asterisk, Pyramid and diamond pattern using the **print()** function.
- Add a new line after each row, i.e. after each iteration of outer for loop so you can display pattern appropriately.

# Problem 3

Write a Python Program (Using for loop) to create the following pattern:

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

# Problem 4

- Write a program using for loops to print Fibonacci series up to a given integer.
- [0,1,1,2,3,5,8,13]

# Problem 5

- Print the given number pattern

- 1

- 2 2

- 3 3 3

- 4 4 4 4

- 5 5 5 5 5

- 6 6 6 6 6 6

- 7 7 7 7 7 7 7

- 8 8 8 8 8 8 8 8

- 9 9 9 9 9 9 9 9 9



# Next Class

More Problems on For Loops

In-Class Exercise