TCSS 142 – Introduction to Programming

Autumn 2014
Day 06

Day 6 Overview

- for loop
  - Generating sequences
  - Incorporating I/O
  - Iterating over strings
  - ifs and sequential fors

Loops

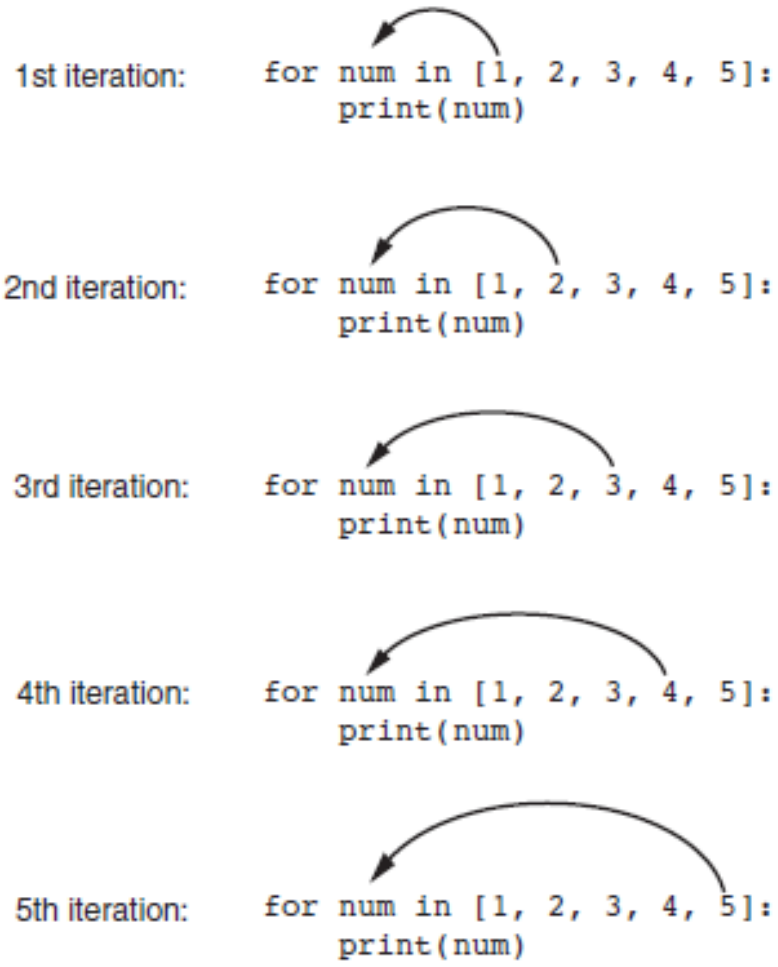
- A for loop is an example of a repetition control structure.
  - it causes a single statement or block to be executed repeatedly

- A for loop is a count-controlled loop
  - repeats a specified number of times
  - there is a variable that controls the loop (called a counter) that gets incremented or decremented each time a loop executes
  - the counter can be used in the body of the loop
  - designed to work with sequence of data items
    - Iterates once for each item in the sequence
  - general format:

```
for variable in sequence:
        statements
```

Visualizing the for loop

Figure 4-4 The `for` loop



Generating a sequence

- To generate a numeric sequence, use function range:
  - One argument: used as ending limit
  - Two arguments: starting value and ending limit
  - Three arguments: third argument is step value

- Open a new file and call it `forTests.py`
  - Let's start with a piece of code that will generate the message

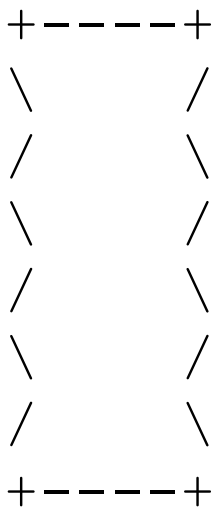
    1 potato

    2 potato

    3 potato
  - What if we want to print number 4 below it? How can we do it?

Multiple Statements in `for`

- Let's write a piece of code that will generate the following output:



Exercise

- Create a Python file called `pattern.py` which contains the code that produces the following pattern using sequential for loops
  - The first loop is to print the first row
  - The second loop is to print the second row
  - The third loop is to print the third row

```
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
||||||||||||||||||||||||||||||||||||||
\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
```

I/O to control a for loop

- User input may control the loop
- Read in interactive input into variables and use variables to control the loop
  - you may also use expressions to control range
- Go back to `pattern.py` and ask for the length of the pattern, then use it to control your loops

Generating a sequence

- Go back to `forTests.py` and add the for loops that prints the following sequences (each sequence on one line with a blank between the numbers):
  - 2 7 12 17 22
  - Even numbers from 14 to 24 inclusive
  - Multiples of 3 from 6 – 30 inclusive
  - Every other number from 7 to -7 inclusive

User I/O

- Change the first loop to use variables entered by the user

```
beginning, stop, step
```

- Will it work for all of the sequences? Even decreasing ones?
  - Solution?

for loop and strings

- You can use for loops to iterate over strings
  - Each time through the loop the iterator is matched with each string character

- Create a new file `stringFor.py` and enter the following code

```
for ch in  'I am a string":
    print(ch)
```

Loops

- Loops are often used to
  - count data values
  - sum data values (shown in the book)
  - count special values

- Let's adjust `stringFor.py` to count the number of times letter i appeared in our string

- Add interactive input so that a user enters a string to be processed

- Add another counter that calculates the length of the entered string

Transposition Cipher

	0123..
Original:	It was a dark and stormy night
Even/Odd:	It_was_a_dark_and_stormy_night
Message:	twsandr_n_tryngtI_a__akadsom_ih

Algorithm

Set up a string (`odds`) that will contain odd characters

Set up another string (`evens`) that will contain even characters

Process each character in plaintext

For each character determine, if it is odd or even and add it to the appropriate substring

Concatenate the strings (`cipher = odds + evens`)

Divide and Conquer

- Based on our discussion so far, how can we figure out whether a character is in an odd or even position?
- Create a new file `transposition.py` and let's just write the code for determining the odd vs even position
- Once we have it, let's add interactive input
- Finally, we can set up appropriate strings for odds and evens.

Gangsta Name

Write a program `gangsta.py` that based on the person's first and last names outputs a person's "gangsta name." Assume the first and last names are stored in a string and are separated by a blank character, e.g. "Mallory Kane"

- Use the following algorithm to change the name
  - first initial followed by a period
  - *Diddy*
  - last name
  - first name
  - -izzle

  Example: Mallory Kane becomes M. Diddy Kane Mallory-izzle

Divide and Conquer

- How can we determine it is a first character?
  - Let's solve for `M. Diddy`

- How do we know we are dealing with the first vs last name?
  - we can use a flag – a Boolean variable used to signal an error or completion.

- When using a flag
  - initialize a flag (to true or false) before the body of a loop
  - use a meaningful name for the flag
  - a condition in the loop body changes the value of the flag
  - optional: test for the flag in the loop test expression

Last Slide ☺

- Class ends at 17:10

More Exercises

- Textbook p. 161, ex. 5

Write a loop that calculates and prints the total of the following series of numbers:

1/30 + 2/29 + 3/28 + … + 30/1

- Textbook p. 161, ex. 1

A bug collector collects bugs every day for five days. Write a program that keeps a running total of the number of bugs collected during the five days. The loop should ask for the number of bugs collected for each day, and when the loop is finished, the program should display the total number of bugs collected.

More Exercises

- Textbook p. 162, ex. 7

Write a program that calculates the amount of money a person would earn over a period of time if his or her salary is one penny the first day, two pennies the second day, and continues to double each day. The program should ask the user for the number of days.
Display a table showing what the salary was for each day, and then show the total pay at the end of the period. The output should be displayed in a dollar amount, not the number of pennies.

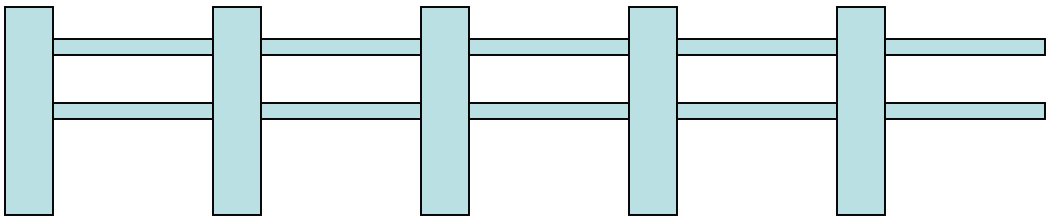
Fence post

- Suppose you were to print the sequences with comas in between, e.g.

  2 , 7, 12, 17, 22

- Replace end = ' ' with end = ', ' in the code for that sequence – what happens?

- Similar to building a fence with wires separated by posts:
  - If we use a flawed algorithm that repeatedly places a post + wire, the last post will have an extra dangling wire.



Possible Solutions

for length of fence - 1:

place a post.

place some wire.

place a post.

---

place a post.

for length of fence ***- 1***

place some wire.

place a post.

for length of fence:

place a post.

if not the last post:

place some wire.

---

if length of fence > 0:

for loop solutions from the left