Chapter 2: Variables, and Simple I/O: The Useless Trivia Program

Intro Computer Programming
Objectives

In this chapter, you will learn how to:

- Use strings and escape sequences to gain more control over text
- Make your programs do math
- Store data in computer’s memory
- Use variables to access and manipulate data
- Get input from users to create interactive programs
Using Single or Double Quotes to Create Simple Strings

- **Surround the text with identical quotes**
  - “Hello world”
  - ‘Hello world’
    - print “Kent State University”
    - print ‘Intro Computer Programming’

- **What if the text has embedded quote(s)?**
  Example 1 (single quotes): Program ‘game Over’ 2.0
    - “Program ‘game Over’ 2.0”

  Example 2 (double quotes) The “Kent State University” Mission Statement
    - ‘The “Kent State University” Mission Statement’

Quotes delineate the start and end of string
This is how you print a string
Single Quotes inside string
Double Quotes inside string
Continuing a Statement on the Next Line

Usually, We Type one statement per line

What if your statement exceeds one line?

Use the backslash character \ anywhere except inside a string to continue your statement:

\ Represents Line –Continuation Character
Creating Triple Quoted Strings
Convenient way to print more than a few lines (block) of text

Bookend block of text with triple quotes(“““” or ””””)

““““

Block of texts

““““

- Let’s examine the python program for triple quotes on your CD
# Escape Sequences

Allow you to put special characters into your string:

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\</code></td>
<td>Prints one backslash</td>
</tr>
<tr>
<td><code>'</code></td>
<td>Prints a single quote</td>
</tr>
<tr>
<td><code>&quot;</code></td>
<td>Prints a double quote</td>
</tr>
<tr>
<td><code>\a</code></td>
<td>Sounds a system bell (in Script mode only)</td>
</tr>
<tr>
<td><code>\b</code></td>
<td>Move cursor back one space</td>
</tr>
<tr>
<td><code>\n</code></td>
<td>Move cursor to beginning of next line</td>
</tr>
<tr>
<td><code>\t</code></td>
<td>Horizontal tab</td>
</tr>
</tbody>
</table>

The computer interprets the first backslash as the start of an escape sequence.


Concatenating Strings:

‘+’ operator

- How you **join** two or more strings together to make one new string

  - print “cup” + ”cake”  →  **Output**  →  cupcake
  - print “butter” +”fly”  →  butterfly
  - Print “rice “ + “cake”  →  rice cake
  - Print “butter” + “ cup”  →  butter cup

Blank space
Suppressing a New line
Trailing comma

- Add a comma at the end of the print statement:
  print “Kent State University Campus in”,
  print “Kent, OH 44242”

Kent State University Campus in Kent, OH 44242

But:

- print “Kent State University Campus in”
- print “Kent, OH 44242”

Kent State University Campus in Kent, OH 44242

The 2 print Outputs appear on one line:

The 2 print Outputs appear on 2 lines:
Repeating Strings

use the repetition operator: *

- Put the string and the number of repetition together with the repetition operator

print “Pie” * 10

Output:

“PiePiePiePiePiePiePiePiePiePiePiePie”

String “Pie” is repeated 10 times
Working with Numbers

Numeric Types

- **Integers**
  - Whole Numbers (1, 2, 3, 20, 41, 100, 208, ...)
  - Do not have fractional parts

- **Floating-point Numbers (Floats)**
  - Numbers with decimal point (1.0, 2.0, 3.0, 20.0, ...)

Computer Treats Integers and Floats Differently!
As you will see shortly
# Math Operators

## Using Integers

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Multiplication</td>
<td>print 7*3</td>
<td>21</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
<td>print 7/3</td>
<td>2</td>
</tr>
<tr>
<td>%</td>
<td>Modulus (remainder)</td>
<td>print 7%3</td>
<td>1</td>
</tr>
<tr>
<td>+</td>
<td>Addition</td>
<td>print 7+3</td>
<td>10</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
<td>print 7-3</td>
<td>4</td>
</tr>
</tbody>
</table>

When both operands are Integer numbers the output will be integer.

Computer truncates decimal part.
# Math Operators

## Using Floats

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Multiplication</td>
<td>print 7.0*3.0</td>
<td>21.0</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
<td>print 7.0/3.0</td>
<td>2.3333333333</td>
</tr>
<tr>
<td>%</td>
<td>Modulus (remainder)</td>
<td>print 7.0%3.0</td>
<td>1.0</td>
</tr>
<tr>
<td>+</td>
<td>Addition</td>
<td>print 7.0+3.0</td>
<td>10.0</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
<td>print 7.0-3.0</td>
<td>4.0</td>
</tr>
</tbody>
</table>

When both operands are Floats the output will be Float
Math Operators

Using Float & Integer Combinations

print 19.0/4
   Output 4.75
print 19/4.0
   Output 4.75
print 1.0/2
   Output 0.5

But
print 1/2
   Output is 0

When at least one of the numbers is a Float, the output will be Float
**VARIABLES**

- **Concepts**
  - Identify Memory locations/cells
  - Provides a way to store and access information

- **Creating Variables**
  - Must create variables before you can reference it
    - `name = “Larry”`

Memory location identified by “name” contains the string “Larry”
Assignment Statement
- `course = "Intro Computer Programming"`

Referencing variables
- `print "CS-10061 " + course`
- Outputs: CS-10061 Intro Computer Programming

Naming Variables
- Can contain only numbers, letters and underscore
- Cannot start with a number
- Choose Descriptive names
How do you assign a user’s input to a variable?

- Use the `raw_input (…)` statement:
  - `name = raw_input (“Hi. What’s your name?”)`
- Let’s discuss the assignment statement above
- Now Let’s examine the *Greeter Program on the CD*
String Methods
Create New Strings From Old Strings

- Allow you to create new strings from old ones
- Assume the assignment statement
  - course = “Intro computer programming”
- Print in Uppercase (All Caps)
  - print course.upper()
    - Output: INTRO COMPUTER PROGRAMMING
- Print in lowercase
- print course.lower()
  - Output: intro computer programming
Introduction to Programming CS 10061

String Methods

Assume the assignment statement
- course = “Intro computer programming”

- Print as Title
  - print course.title()
    - Output: Intro Computer Programming

- Print with minor replacement
  - print course.replace(“programming”, “science”)
    - Output: Intro computer science

Review String Methods in Table 2.4
Using the Right Data Types

*Integer, Floats, Strings*

- Using Incorrect data types
  - Unpredictable results (*Logical errors*)

---

Let’s examine the Trust Fund Buddy – Bad program
For examples of logical errors

- *Can you find the bug in this program?*

- *How do we fix these bugs?*
  - Convert strings returned by `raw_input()` to values of type `float`
### Convert Values

#### Selected Type Conversion Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>float (x)</td>
<td>Converts x to float value</td>
<td>float(&quot;10.0&quot;)</td>
<td>10.0</td>
</tr>
<tr>
<td>int(x)</td>
<td>Converts x to integer value</td>
<td>int (&quot;10&quot;)</td>
<td>10</td>
</tr>
<tr>
<td>str(x)</td>
<td>Converts x to a string value</td>
<td>str(10)</td>
<td>“10”</td>
</tr>
</tbody>
</table>

*Now, Let’s examine the Trust Fund Buddy – Good program
For proper usage of functions: float(x), int(x) and str(x)*
### Augment Assignment Operators

#### Interesting shortcuts

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example (shortcut)</th>
<th>Is Equivalent To</th>
</tr>
</thead>
<tbody>
<tr>
<td>*=</td>
<td>x *= 5</td>
<td>x = x * 5</td>
</tr>
<tr>
<td>/=</td>
<td>x /= 5</td>
<td>x = x / 5</td>
</tr>
<tr>
<td>%=</td>
<td>x %= 5</td>
<td>x = x % 5</td>
</tr>
<tr>
<td>+=</td>
<td>x += 5</td>
<td>x = x + 5</td>
</tr>
<tr>
<td>-=</td>
<td>x -= 5</td>
<td>x = x - 5</td>
</tr>
</tbody>
</table>
Let’s assume the assignment statement:

```
total = 20
```

What is the output of the following print statement?

```
print "\nGrand Total: ", total
```

Output is

```
Grand Total: 20
```
Let’s Summarize

Go to “Useless Trivia” program

- Printing Lowercase and Uppercase versions of “name”
- Calculating dog years
  - (Assume that: 7 human years are equal to 1 dog year; and dog years are expressed as whole numbers only)
- Calculating Seconds
- Printing “name” Five Times
- Waiting for the User

How do you calculate the average price of six quantities?