Game Programming

The L Line

The Express Line to Learning

© Wiley Publishing. 2007. All Rights Reserved.

CHAPTER

1

Writing Your First Program

Stations Along the Way

- Installing Python
- Interacting with the Python console
- Building a program that greets the user
- Getting basic text input from the user

CHAPTER

1

Writing Your First Program (continued)

Stations Along the Way

- •Building a string variable with an appropriate name
- •Outputting the value of a string variable to the user
- •Creating subsets of a string with slicing
- •Using string interpolation for complex output

Why Use Python?

- ✓Freely available
- ✓Platform-independent
- ✓Easy to learn
- ✓Powerful
- ✓Extensible
- ✓Transferable

Installing Python

- ✓ Download binaries from `www.python.org`.
- ✓ Run the installer with default parameters.
- ✓ Mac or Linux users, follow the Python Web site instructions.
- ✓ Also possible to use a “*Sand Box*”
  - https://repl.it/languages/python

Starting the Engine

- ✓ Run Python in the command-line console.
- ✓ Windows:
  - Run→cmd
- ✓ Mac/Linux:
  - Start terminal console
- ✓ Move to the Python directory, if necessary.
- ✓ Type `python` to begin your session.

Interacting with the Console

- ✓The `>>>` symbol is Python’s prompt.
- ✓Type `print (‘Hello, there!’)`
- ✓View immediate results.

Using Python as a Simple Calculator

- ✓ Type a simple math problem (`4 + 3`) at the `>>>` prompt.
- ✓ View the immediate response.
- ✓ Try other operations:
  - Multiply = *
  - Divide = /
  - Try parentheses

Storing Information in Variables

- ✓Type the following on the console:

```
answer = 5 + 3
```

- ✓Retrieve the answer with this code:

```
print (answer)
```

- ✓*Variables* are locations in memory designated to hold a piece of information.

Using IDLE

- ✓ IDLE is an Integrated Development Environment for Python.
- ✓ It comes standard with most versions of Python.
- ✓ It's a text editor specialized for creating and testing Python programs.

IDLE's Two Modes

- ✓ If you type `idle` into the command line, the IDLE window shows the `>>>` prompt.
- ✓ This is *interactive mode.* You can type instructions directly.
- ✓ File-new calls up a new IDLE window that acts more like a text editor.
- ✓ Note the menus are slightly different in the two modes.

Storing Code in a File

1. Open a new IDLE window.
2. Note the different menus.
3. Continue writing code (nothing happens immediately).
4. Save your file with a `.py` extension.
5. Run the program (F5).

Your First Interactive Game

✓Type this code into IDLE:

```
"""Cheese Shop
    cheeseShop.py
    demonstrate comments, raw input, and
string variables
    from Game Programming - L-line, Andy
Harris
    28/09/17
# Modified by G2 for Python 3.6
#tell the user something
print ("Welcome to the cheese shop!")
#get information from the user
cheeseType = input("What kind of cheese would
you like? ")
#we don't have that kind...
print ("Sorry, We're all out of")
print (cheeseType)
```

Using docstrings

✓ The triple-quoted string contains special comments about the program:

- Program name
- Author
- Date
- Filename

Printing Output

- ✓`print` prints output to the screen.
- ✓Values in quotes are printed exactly.
- ✓The value of a variable is printed.

Getting Input from the User

- ✓ input gets data from the screen.
- ✓ It prints a prompt.
- ✓ It retrieves text data.
- ✓ It expects a variable in which to store the result.

Variable Naming Conventions

- ✓ Descriptive
- ✓ No spaces
- ✓ Case-sensitive
- ✓ Manageable length

Introducing Strings

- ✓ Programmers call text *strings.*
- ✓ The storage mechanism uses a sequence of memory cells.
- ✓ This reminded early programmers of beads on a string.
- ✓ Variables that contain text are called *string variables.*

Building a Basic String

- ✓Type string assignment in the console (the >>> will already be there):

```
>>> playerName = "Princess Oogieboogie"
```

- ✓Output the value of the string:

```
>>> print (playerName)
Princess OoogieBoogie
```

- ✓Storing a string value into a variable automatically creates a string variable.

Introducing Methods

- ✓ Python uses Object-Oriented Programming (OOP).
- ✓ All entities are objects.
- ✓ Objects have *methods* (things they can do).
- ✓ Strings have a bunch of interesting methods.

Discovering String Methods

- ✓ Python has a very rich built-in help system you can use to learn about objects.
- ✓ Type `help("str")` at the console to learn about the str (string) object.
- ✓ Note: `help("string")` produces results too, but these are older functions that have been replaced by str.

Exploring String Methods

```
""" nameGame.py
    illustrate basic string functions
    Andy Harris- Modified by GG for Python 3.6
    28/09/17"""
userName = input("Please tell me your name: ")
print ("I will shout your name: ", userName.upper())
print ("Now all in lowercase: ", userName.lower())
print ("How about inverting the case? ",
userName.swapcase())
numChars = len(userName)
print ("Your name has", numChars, "characters")
print ("Now I'll pronounce your name like a cartoon
character:")
userName = userName.upper()
userName = userName.replace("R", "W")
userName = userName.title()
print (userName)
```

Selected String Methods

String Method	Description
*stringName.*upper()	Converts *stringName* into all uppercase
*stringName.*lower()	Converts *stringName* into all lowercase
*stringName.*swapcase()	Converts uppercase to lowercase, lowercase to uppercase
stringName.replace(*old*, *new*)	Looks in the string for the value *old* and replaces it with the value *new*
*stringName.*title()	Capitalizes each word in the string
len*(string)*	Returns the length of the string

Making the Cartoon Version

- ✓ The "cartoon voice" requires a couple of steps.
- ✓ Convert the string to uppercase.
- ✓ Replace "R" with "W."
- ✓ Convert back to title case.
- ✓ Program catches uppercase and lowercase "R."

Slicing Strings

- ✓ You can extract parts of a string.
- ✓ This technique is called *slicing.*
- ✓ String has positions *between* characters:
  - `0 1 2 3 4 5 6`
  - `|s|a|l|a|m|i|`
- ✓ Please view `salamiSlice.py`.

String Slicing Example

✓Guide:

- 0 1 2 3 4 5 6
- |s|a|l|a|m|i|

```
>>> meat = "salami"
>>> print (meat[2:5])
'lam'
>>> print (meat[0: 3])
'sal'
Print (meat[4:6])
'mi'
```

More String Slicing

- ` 0 1 2 3 4 5 6`
- `|s|a|l|a|m|i|`

```
>>> meat = "salami"
>>> print (meat[0:3])
'sal'
print meat[:3]
'sal'
print meat[4:6]
'mi'
print meat[4:]
'mi'
print meat[-3:]
'ami'
Print meat[4]
'm'
```

String Interpolation

- ✓ Sometimes, you want to combine variables and literal values.
- ✓ Python has a nice technique called *string interpolation:*

```
>>> userName = "Benjamin"
>>> print ("Hi there, %s!" % userName)
Hi there, Benjamin!
```

- ✓ %s indicates a *string placeholder.*
- ✓ The second % indicates variable to stuff into string.

Interpolating Numbers

- ✓Use %s to embed a string value.
- ✓Use %d to embed an integer (a number without decimal values).
- ✓Use %f to embed a real number (with decimal values).
- ✓Use %.2f to embed a real number to two places.
- ✓You can find more on numbers in Chapter 2.

Interpolating Multiple Values

- ✓A string interpolation can include multiple values:

```
Print ("%s is %d years old today." % (name, age))
```

- ✓Use a placeholder for each value.
- ✓Use parentheses to make a list of variables.
- ✓Please view `interpolation.py`.