

Learning Objectives

After completing this unit, you should be able to:

- Identify whether a sequence of steps is an algorithm in the strict sense.
- Explain the purpose/function of the editor and shell windows in IDLE.
- Trace Python programs involving variables, operators, input, and print.
- Summarize problem-solving strategies for “getting a foot in the door”.
- Define an algorithm for solving decimal and binary arithmetic puzzles.
- Write a Python function that implements the steps of a given algorithm.
- Identify the three components of loop control (initialize, test, modify).

Textbook Sections

- 5.1 The Concept of an Algorithm
- 5.2 Algorithm Representation
- 5.3 Algorithm Discovery
- 5.4 Iterative Structures

Video Lectures

- Intro to Python
- What Most Schools Don't Teach

Assignments

Act07 Hello, Python!; Chapter 5 Problems

Lab07 Codecademy (1 & 2); Intro to Python and IDLE

Unit 7 Checklist: Oct 14 – Oct 20


Before Friday	Date Completed
FINISH models 1 and 2 of Hello Python activity	
READ textbook 5.1 The Concept of an Algorithm (take notes) ANSWER questions 1 and 4 in your notes	
READ textbook 5.2 Algorithm Representation (take notes) ANSWER questions 2 and 4 in your notes	
WATCH video lecture: Intro to Python (take notes)	
WATCH Code.org video: What Most Schools... (take notes)	
READ textbook 5.3 Algorithm Discovery (take notes) ANSWER questions 3 and 4 in your notes	
READ textbook 5.4 Iterative Structures (take notes) (Up to the point described in the video)	
DO tutorial: Codecademy (1. Python Syntax)	
DO tutorial: Codecademy (2. Strings and Console Output)	
START Lab07: Intro to Python and IDLE	(10 pts)
START Act07 exercises (complete at least 75%)	(15 pts)
Before Monday	Date Completed
COMPARE your Lab07 and Act07 with the solutions in Canvas	
SUBMIT Quiz07 – 1st attempt closed: see what you don't know	
STUDY your notes, ask questions on Piazza, meet with the TAs	
SUBMIT Quiz07 – 2nd attempt open: try to get the full 10 points	(10 pts)
TAKE Exam07	(40 pts)

Activity 7: Hello, Python!

“By the way, the language is named after the BBC show ‘Monty Python’s Flying Circus’ and has nothing to do with reptiles. Making references to Monty Python skits in documentation is not only allowed, it is encouraged!” (Source: <https://docs.python.org/2/tutorial/appetite.html>)

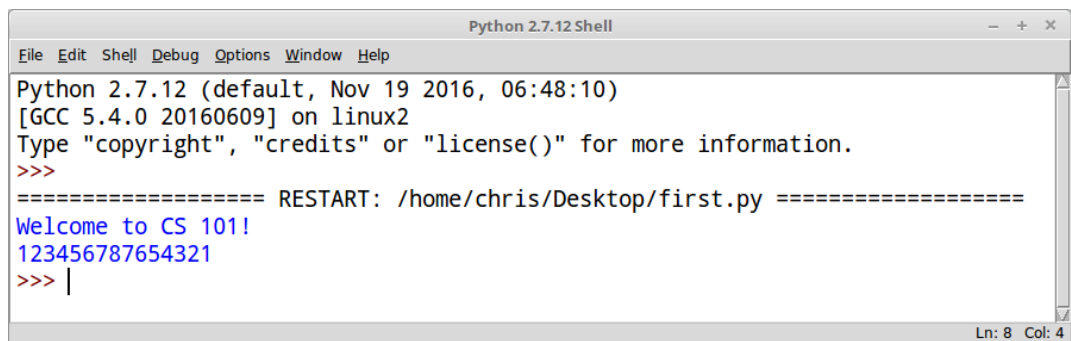
Model 1 Using IDLE

“IDLE is Python’s Integrated Development and Learning Environment. It has two main window types: the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously.” (Source: <https://docs.python.org/2/library/idle.html>)



```
first.py - /home/chris/Desktop/first.py (2.7.12)
File Edit Format Run Options Window Help
# display a welcome message
print "Welcome to CS 101!"

# multiplying ones always gives you a palindrome
x = 11111111
print x * x
|
Ln: 7 Col: 0
```



```
Python 2.7.12 Shell
File Edit Shell Debug Options Window Help
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/chris/Desktop/first.py =====
Welcome to CS 101!
123456787654321
>>> |
Ln: 8 Col: 4
```

Questions (15 min)

Start time: _____

1. Which of the two screenshots in Model 1 is the Shell window? Which is the Editor window?
2. Explain the terms “Editor” and “Shell” based on what you learned previously in the course.

3. What is the name of the file in the editor? What directory is it saved in?

4. Explain the Python code in the editor window. What does each line do?

5. What is the output of the program? Where should you look for output?

6. Predict the output of each line below. Then type each line into the Shell window (one at a time) and check your answers.
 - a) `print 2 * 5`
 - b) `print 2 + 5`
 - c) `print "2 + 5"`
 - d) `print CS rocks!`
 - e) `print 2 # 5`

7. Explain the difference between b) and c) in the last question. Why are the results different?

8. What is wrong with the code in d)? Explain the error message. How do you fix the error?

Model 2 Guessing Game

Create a new file named `guess.py` and enter the following code. Replace the name in Line 2 with your own name. Be careful to type the code *exactly* as shown.

```
1 name = raw_input("What is your name? ")
2 if name == "Taylor":
3     print name, "is a great name!"
4 else:
5     print name, "is an okay name."
```

Note: `raw_input` is a **function** that displays a **prompt** on the screen and reads a line from the keyboard. In this program, the result of `raw_input` is stored in the **variable** `name`.

Questions (15 min)

Start time: _____

9. What is the prompt? Why is there a space at the end of it?

10. Run the program a few times, entering a different name each time. Feel free to modify the messages as you see fit.

11. Enter each of these lines into the IDLE shell, and explain where the syntax error occurs.
 - a) `name? = raw_input("What is your name?")`
 - b) `your name = raw_input("What is your name?")`
 - c) `1st_name = raw_input("What is your name?")`
 - d) `from = raw_input("Where were you born?")`

12. Based on the errors in the previous question and the following correct examples, describe three rules that need to be followed when naming a variable.

```
name2 = raw_input("What is your name?")
your_name = raw_input("What is your name?")
firstName = raw_input("What is your name?")
```

13. At the end of your `guess.py` program, create two new variables named `number` and `guess`. Set the value of `number` to be an integer between 1 and 100 (of your choice). Ask the user to guess your number, and store the result in `guess`. When asking for numbers, use `input` instead of `raw_input`. Write your two statements in the space below.

14. Add the following logic to your program: If the guess is too high, display the message "Too high!"; if the guess is too low, display the message "Too low!"; if the user guessed the number, display "You got it!". Write your statements in the space below.

15. What is the difference between `=` and `==` in the programs you have written today?

16. At this point, you should have a program that allows the user to make only one guess. Rather than run this program over and over again, you can use a `while` loop to make it repeat the guessing part. Insert the following two lines before the `input` line you wrote in #13.

```
guess = -1
while guess != number:
```

17. What did you have to do after inserting the `while` loop to make it work? In other words, how did you make the `input` and `if` statements part of the `while` loop?

18. Rather than guess the same number every time, you can have the computer select a random number for you:

- At the top of your program, add the line `import random` (without the quotes).
- Then change the line where you set value of `number` to use this example instead:

```
number = random.randint(1, 100)
```

Chapter 5: Algorithms

Complete the following Chapter Review Problems on pages 251–252.

#5 (algorithm in the strict sense) – *also list four requirements of algorithms*

#12 (“day of week” algorithm) – *just describe the steps, don’t write any code*

#13 and #14 (language vs pseudocode, syntax vs semantics)

#15 (decimal arithmetic) – *give the answer, and describe how you found it*

#17 (binary arithmetic) – *just describe the algorithm, don’t write any code*