# **Exercise 35: Branches and Functions**

You have learned to do if-statements, functions, and lists. Now it's time to bend your mind. Type this in, and see if you can figure out what it's doing.

```
from sys import exit
   def gold room():
 1
        print "This room is full of gold. How much do you
   take?"
 3
        next = raw input("> ")
 5
        if "0" in next or "1" in next:
 6
            how much = int(next)
7
        else:
 8
            dead("Man, learn to type a number.")
9
10
        if how much < 50:
11
            print "Nice, you're not greedy, you win!"
12
            exit(0)
13
        else:
14
            dead("You greedy bastard!")
15
16
17
   def bear room():
18
        print "There is a bear here."
19
        print "The bear has a bunch of honey."
20
        print "The fat bear is in front of another door."
21
        print "How are you going to move the bear?"
22
        bear moved = False
23
24
        while True:
25
            next = raw input("> ")
26
27
            if next == "take honey":
28
                dead("The bear looks at you then slaps your
29
   face off.")
30
            elif next == "taunt bear" and not bear moved:
31
                print "The bear has moved from the door. You
32
   can go through it now."
33
                bear moved = True
```

```
elif next == "taunt bear" and bear moved:
34
35
                dead("The bear gets pissed off and chews your
    leg off.")
36
37
            elif next == "open door" and bear moved:
38
                gold room()
39
            else:
40
                print "I got no idea what that means."
41
42
43
    def cthulhu room():
44
        print "Here you see the great evil Cthulhu."
45
        print "He, it, whatever stares at you and you go
46
47
        print "Do you flee for your life or eat your head?"
48
49
        next = raw_input("> ")
50
51
        if "flee" in next:
52
            start()
53
        elif "head" in next:
54
            dead("Well that was tasty!")
55
        else:
56
            cthulhu room()
57
58
59
    def dead(why):
        print why, "Good job!"
60
61
        exit(0)
62
    def start():
63
64
        print "You are in a dark room."
        print "There is a door to your right and left."
65
        print "Which one do you take?"
66
67
68
        next = raw_input("> ")
69
70
        if next == "left":
71
            bear room()
        elif next == "right":
72
73
            cthulhu room()
74
75
            dead("You stumble around the room until you
76
    starve.")
    start()
```

# What You Should See

Here's me playing the game:

```
$ python ex35.py
You are in a dark room.
There is a door to your right and left.
Which one do you take?
> left
There is a bear here.
The bear has a bunch of honey.
The fat bear is in front of another door.
How are you going to move the bear?
> taunt bear
The bear has moved from the door. You can go through it now.
> open door
This room is full of gold. How much do you take?
> 1000
You greedy bastard! Good job!
```

# **Study Drills**

- 1. Draw a map of the game and how you flow through it.
- 2. Fix all of your mistakes, including spelling mistakes.
- 3. Write comments for the functions you do not understand. Remember doc comments?
- 4. Add more to the game. What can you do to both simplify and expand it?
- 5. The gold\_room has a weird way of getting you to type a number. What are all the bugs in this way of doing it? Can you make it better than just checking if "1" or "0" are in the number? Look at how int() works for clues.

# **Common Student Questions**

## Help! How does this program work!?

Any time you get stuck understanding a piece of software, simply write an English comment above *every* line explaining what it does. As you go through doing this, correct comments that aren't right based on new information. Then when you're done try to either diagram how it works, or write a paragraph or two describing it. If you do that you'll get it.

#### Why are you doing while True:?

That makes an infinite loop.

#### What does exit(0) do?

On many operating systems a program can abort with exit(0), and the number passed in will indicate an error or not. If you do exit(1) then it will be an error, but exit(0) will be a good exit. The reason it's backward from normal boolean logic (with 0==False is that you can use different numbers to indicate different error results. You can do exit(100) for a different error result than exit(2) or exit(1).

#### Why is raw\_input() sometimes written as raw\_input('> ')?

The parameter to raw\_input is a string that it should print as a prompt before getting the user's input.

## **Purchase The Videos For \$29.59**

For just \$29.59 you can get access to all the videos for Learn Python The Hard Way, **plus** a PDF of the book and no more popups all in this one location. For \$29.59 you get:

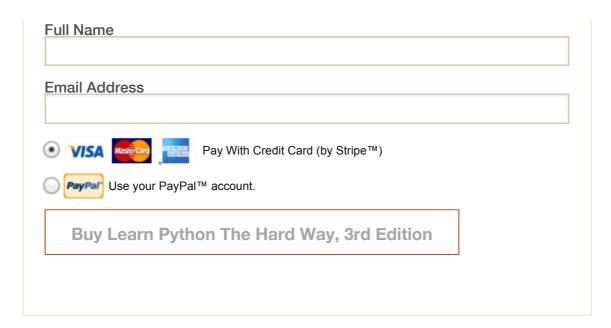
- All 52 videos, 1 per exercise, almost 2G of video.
- A PDF of the book.
- Email help from the author.
- See a list of everything you get before you buy.

When you buy the videos they will immediately show up **right here** without any hassles.

### Already Paid? Reactivate Your Purchase Right Now!

## **Buying Is Easy**

Buying is easy. Just fill out the form below and we'll get started.











Copyright (C) 2010 Zed. A. Shaw