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():
    print "This room is full of gold.  How much do you take?"

    next = raw_input("> ")
    if "0" in next or "1" in next:
        how_much = int(next)
    else:
        dead("Man, learn to type a number.")

    if how_much < 50:
        print "Nice, you're not greedy, you win!"
        exit(0)
    else:
        dead("You greedy bastard!")


def bear_room():
    print "There is a bear here."
    print "The bear has a bunch of honey."
    print "The fat bear is in front of another door."
    print "How are you going to move the bear?"
    bear_moved = False

    while True:
        next = raw_input("> ")

        if next == "take honey":
            dead("The bear looks at you then slaps your face off.")
        elif next == "taunt bear" and not bear_moved:
            print "The bear has moved from the door. You can go through it now."
            bear_moved = True
        elif next == "taunt bear" and bear_moved:
            dead("The bear gets pissed off and chews your
leg off.")
        elif next == "open door" and bear_moved:
            gold_room()
        else:
            print "I got no idea what that means."


def cthulhu_room():
    print "Here you see the great evil Cthulhu."
    print "He, it, whatever stares at you and you go
insane."
    print "Do you flee for your life or eat your head?"

    next = raw_input("> ")

    if "flee" in next:
        start()
    elif "head" in next:
        dead("Well that was tasty!")
    else:
        cthulhu_room()


def dead(why):
    print why, "Good job!"
    exit(0)

def start():
    print "You are in a dark room."
    print "There is a door to your right and left."
    print "Which one do you take?"

    next = raw_input("> ")

    if next == "left":
        bear_room()
    elif next == "right":
        cthulhu_room()
    else:
        dead("You stumble around the room until you
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.

Full Name

Email Address

● Pay With Credit Card (by Stripe™)

○ Use your PayPal™ account.

Buy Learn Python The Hard Way, 3rd Edition