

Exercise 36: Designing and Debugging

Now that you know `if`-statements, I'm going to give you some rules for `for`-loops and `while`-loops that will keep you out of trouble. I'm also going to give you some tips on debugging so that you can figure out problems with your program. Finally, you are going to design a similar little game as in the last exercise but with a slight twist.

Rules For If-Statements

1. Every `if`-statement must have an `else`.
2. If this `else` should never be run because it doesn't make sense, then you must use a `die` function in the `else` that prints out an error message and dies, just like we did in the last exercise. This will find *many* errors.
3. Never nest `if`-statements more than two deep and always try to do them one deep. This means if you put an `if` in an `if` then you should be looking to move that second `if` into another function.
4. Treat `if`-statements like paragraphs, where each `if elif else` grouping is like a set of sentences. Put blank lines before and after.
5. Your boolean tests should be simple. If they are complex, move their calculations to variables earlier in your function and use a good name for the variable.

If you follow these simple rules, you will start writing better code than most programmers. Go back to the last exercise and see if I followed all of these rules. If not, fix it.

Warning

Never be a slave to the rules in real life. For training purposes you need to follow these rules to make your mind strong, but in real life sometimes these rules are just stupid. If you think a rule is stupid, try not using it.

Rules For Loops

1. Use a `while`-loop only to loop forever, and that means probably never. This only applies to Python; other languages are different.
2. Use a `for`-loop for all other kinds of looping, especially if there is a fixed or limited number of things to loop over.

Tips For Debugging

1. Do not use a "debugger." A debugger is like doing a full-body scan on a sick person. You do not get any specific useful information, and you find a whole lot of information that doesn't help and is just confusing.
2. The best way to debug a program is to use `print` to print out the values of variables at points in the program to see where they go wrong.
3. Make sure parts of your programs work as you work on them. Do not write massive files of code before you try to run them. Code a little, run a little, fix a little.

Homework

Now write a similar game to the one that I created in the last exercise. It can be any kind of game you want in the same flavor. Spend a week on it making it as interesting as possible. For Study Drills, use lists, functions, and modules (remember those from Exercise 13?) as much as possible, and find as many new pieces of Python as you can to make the game work.

However, before you start coding you must write up a map for your game.

Create the rooms, monsters, and traps that the player must go through on paper before you code.

Once you have your map, try to code it up. If you find problems with the map then adjust it and make the code match.

One final word of advice: Every programmer becomes paralyzed by irrational fear starting a new large project. They then use procrastination to avoid confronting this fear and end up not getting their program working or even started. I do this. Everyone does this. The best way to avoid this is to make a list of things you should do and then do them one at a time.

Just start doing it, do a small version, make it bigger, keep a list of things to do, and do them.

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.

Full Name

Email Address



Pay With Credit Card (by Stripe™)



Use your PayPal™ account.

Buy Learn Python The Hard Way, 3rd Edition



Copyright (C) 2010 Zed. A. Shaw