

Exercise 10: What Was That?

In Exercise 9 I threw you some new stuff, just to keep you on your toes. I showed you two ways to make a string that goes across multiple lines. In the first way, I put the characters `\n` (back-slash `n`) between the names of the months. What these two characters do is put a `new line character` into the string at that point.

This use of the `\` (backslash) character is a way we can put difficult-to-type characters into a string. There are plenty of these "escape sequences" available for different characters you might want to put in, but there's a special one, the `double backslash`, which is just two of them `\\`. These two characters will print just one backslash. We'll try a few of these sequences so you can see what I mean.

Another important escape sequence is to escape a single-quote `'` or double-quote `"`. Imagine you have a string that uses double-quotes and you want to put a double-quote in for the output. If you do this `"I understand" joe."` then Python will get confused since it will think the `"` around `"understand"` actually *ends* the string. You need a way to tell Python that the `"` inside the string isn't a *real* double-quote.

To solve this problem you *escape* double-quotes and single-quotes so Python knows to include in the string. Here's an example:

```
"I am 6'2\" tall." # escape double-quote inside string
'I am 6\'2" tall.' # escape single-quote inside string
```

The second way is by using triple-quotes, which is just `"""` and works like a string, but you also can put as many lines of text as you want until you type `"""` again. We'll also play with these.

```

1 tabby_cat = "\tI'm tabbed in."
2 persian_cat = "I'm split\non a line."
3 backslash_cat = "I'm \\ a \\ cat."
4
5 fat_cat = """
6 I'll do a list:
7 \t* Cat food
8 \t* Fishies
9 \t* Catnip\n\t* Grass
10 """
11
12 print tabby_cat
13 print persian_cat
14 print backslash_cat
15 print fat_cat

```

What You Should See

Look for the tab characters that you made. In this exercise the spacing is important to get right.

```

$ python ex10.py
    I'm tabbed in.
I'm split
on a line.
I'm \ a \ cat.

```

```

I'll do a list:
    * Cat food
    * Fishies
    * Catnip
    * Grass

```

Escape Sequences

This is the list of all of the escape sequences Python supports. You may not use many of these, but memorize their format and what they do anyway. Also try them out in some strings to see if you can make them work.

Escape	What it does.
\\	Backslash ()

<code>\'</code>	Single-quote (')
<code>\"</code>	Double-quote (")
<code>\a</code>	ASCII bell (BEL)
<code>\b</code>	ASCII backspace (BS)
<code>\f</code>	ASCII formfeed (FF)
<code>\n</code>	ASCII linefeed (LF)
<code>\N{name}</code>	Character named name in the Unicode database (Unicode only)
<code>\r ASCII</code>	Carriage Return (CR)
<code>\t ASCII</code>	Horizontal Tab (TAB)
<code>\uxxxx</code>	Character with 16-bit hex value xxxx (Unicode only)
<code>\Uxxxxxxxx</code>	Character with 32-bit hex value xxxxxxxx (Unicode only)
<code>\v</code>	ASCII vertical tab (VT)
<code>\ooo</code>	Character with octal value ooo
<code>\xhh</code>	Character with hex value hh

Here's a tiny piece of fun code to try out:

```
while True:
    for i in ["/", "-", "|", "\\", "|"]:
        print "%s\r" % i,
```

Study Drills

1. Memorize all the escape sequences by putting them on flash cards.
2. Use `'''` (triple-single-quote) instead. Can you see why you might use that instead of `"""`?
3. Combine escape sequences and format strings to create a more complex format.
4. Remember the `%r` format? Combine `%r` with double-quote and single-quote escapes and print them out. Compare `%r` with `%s`. Notice how

`%r` prints it the way you'd write it in your file, but `%s` prints it the way you'd like to see it?

Common Student Questions

I still haven't completely figured out the last exercise. Should I continue?

Yes, keep going and instead of stopping take notes listing things you don't understand for each exercise. Periodically go through your notes and see if you can figure these things out after you've completed more exercises. Sometimes though you may need to go back a few exercises and go through them again.

What makes `\\` special compared to the other ones?

It's simply the way you would write out one backslash (`\`) character. Think about why you would need this.

When I write `//` or `/n` it doesn't work.

That's because you are using a forward-slash `/` and not a backslash `\`. They are different characters that do very different things.

When I use a `%r` format none of the escape sequences work.

That's because `%r` is printing out the raw representation of what you typed, which is going to include the original escape sequences. Use `%s` instead. Always remember this: `%r` is for debugging, `%s` is for displaying.

I don't get Study Drills #3. What do you mean by "combine" escapes and formats?

One of the things I try to get you to understand is that each of these exercises can be combined to solve problems. Take what you know about format sequences and write some new code that uses those *and* the escapes from this exercise.

What's better, `' '` or `" "`?

It's entirely based on style. Go with the ''' (triple-single-quote) style for now but be ready to use either depending on what feels best or what everyone else is doing.

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



VISA



Pay With Credit Card (by Stripe™)



Use your PayPal™ account.

Buy Learn Python The Hard Way, 3rd Edition



Copyright (C) 2010 Zed. A. Shaw