|  |  |
| --- | --- |
|  | Chapter 5 |
| Jokes |

Topics Covered In This Chapter:

* Escape characters
* Using single quotes and double quotes for strings
* Using print()’s end keyword argument to skip newlines

## Making the Most of print()

Most of the games in this book will have simple text for input and output. The input is typed by the user on the keyboard. The output is the text displayed on the screen. In Python, the print() function displays textual output on the screen. But there’s more to learn about how strings and print() work in Python.

This chapter’s program tells a few different jokes to the user, and demonstrates advanced string and print() code.

## Sample Run of Jokes

What do you get when you cross a snowman with a vampire?

Frostbite!

What do dentists call an astronaut's cavity?

A black hole!

Knock knock.

Who's there?

Interrupting cow.

Interrupting cow wh-MOO!

## Source Code of Jokes

Open a new file editor window by clicking on the **File** ► **New Window**. In the blank window that appears type in the source code and save it as jokes.py. Then run the program by pressing **F5**.

**IMPORTANT NOTE!** The programs in this book will only run on Python 3, not Python 2. When the IDLE window starts, it will say something like “Python 3.4.2” at the top. If you have Python 2 installed, you can have Python 3 installed at the same time. To download Python 3, go to https://python.org/download/.

If you get errors after typing this code in, compare the code you typed to the book’s code with the online diff tool at http://invpy.com/diff/jokes.

jokes.py

1. print('What do you get when you cross a snowman with a vampire?')

2. input()

3. print('Frostbite!')

4. print()

5. print('What do dentists call a astronaut\'s cavity?')

6. input()

7. print('A black hole!')

8. print()

9. print('Knock knock.')

10. input()

11. print("Who's there?")

12. input()

13. print('Interrupting cow.')

14. input()

15. print('Interrupting cow wh', end='')

16. print('-MOO!')

How the Code Works

1. print('What do you get when you cross a snowman with a vampire?')

2. input()

3. print('Frostbite!')

4. print()

Lines 1 to 4 have three print() function calls. You don’t want the player to immediately read the joke’s punch line, so there’s a call to the input() function after the first print(). The player can read the joke, press enter, and then read the punch line.

The user can still type in a string and hit enter, but this returned string isn’t being stored in any variable. The program will just forget about it and move to the next line of code.

The last print() function call has no string argument. This tells the program to just print a blank line. Blank lines are useful to keep the text from being bunched up.

## Escape Characters

5. print('What do dentists call a astronaut\'s cavity?')

6. input()

7. print('A black hole!')

8. print()

On line 5, there’s a backslash right before the single quote: \'. Note that \ is a backslash, and / is a forward slash. This backslash tells you that the letter right after it is an escape character. An escape character lets you print characters that are hard to enter into the source code. On line 5 the escape character is the single quote.

The single quote escape character is there because otherwise Python would think the quote meant the end of the string. But this quote needs to be a part of the string. The escaped single quote tells Python that the single quote is literally a part of the string rather than marking the end of the string value.

Some Other Escape Characters

What if you really want to display a backslash? This instruction would not work:

>>> print('They flew away in a green\teal helicopter.')

They flew away in a green eal helicopter.

This is because the “t” in “teal” was seen as an escape character since it came after a backslash. The escape character t simulates pushing the Tab key on your keyboard. Instead, try this line:

>>> print('They flew away in a green\\teal helicopter.')

They flew away in a green\teal helicopter.

Table 5-1 is a list of escape characters in Python.

Table 5-1: Escape Characters

|  |  |
| --- | --- |
| Escape Character | What Is Actually Printed |
| \\ | Backslash (\) |
| \' | Single quote (') |
| \" | Double quote (") |
| \n | Newline |
| \t | Tab |

## Quotes and Double Quotes

Strings don’t always have to be between single quotes in Python. You can also put them between double quotes. These two lines print the same thing:

>>> print('Hello world')

Hello world

>>> print("Hello world")

Hello world

But you cannot mix quotes. This line will give you an error if you try to use them:

>>> print('Hello world")

SyntaxError: EOL while scanning single-quoted string

I like to use single quotes so I don’t have to hold down the shift key to type them. It’s easier to type, and Python doesn’t care either way.

Just like you need the escape character \' to have a single quote in a string surrounded by single quotes, you need the escape character \" to have a double quote in a string surrounded by double quotes. For example, look at these two lines:

>>> print('I asked to borrow Abe\'s car for a week. He said, "Sure."')

I asked to borrow Abe's car for a week. He said, "Sure."

>>> print("She said, \"I can’t believe you let them borrow your car.\"")

She said, "I can’t believe you let them borrow your car."

In the single quote strings you don’t need to escape double quotes, and in the double quote strings you don’t need to escape single quotes: "astronaut's". The Python interpreter is smart enough to know that if a string starts with one type of quote, the other type of quote doesn’t mean the string is ending.

## print()’s end Keyword Argument

9. print('Knock knock.')

10. input()

11. print("Who's there?")

12. input()

13. print('Interrupting cow.')

14. input()

15. print('Interrupting cow wh', end='')

16. print('-MOO!')

Did you notice the second parameter on line 15's print()? Normally, print() adds a newline character to the end of the string it prints. This is why a blank print() function will just print a newline. But the print() function can optionally have a second parameter (which has the name end.)

The blank string passed is called a keyword argument. The end parameter has a specific name, and to pass a keyword argument to this specific parameter you must type end= before it.

By passing a blank string for end, the print() function won’t add a newline at the end of the string, but instead add a blank string. This is why '-MOO!' appears next to the previous line, instead of on its own new line. There was no newline after the 'Interrupting cow wh' string was printed.

Summary

This chapter explores the different ways you can use the print() function. Escape characters are used for characters that are difficult or impossible to type into the code with the keyboard. Escape characters are typed into strings beginning with a backslash \ followed by a single letter for the escape character. For example, \n would be a newline. To include a backslash in a string, you would use the escape character \\.

The print() function automatically appends a newline character to the end of the string passed it to be displayed on the screen. Most of the time, this is a helpful shortcut. But sometimes you don’t want a newline character at the end. To change this, you can pass the end keyword argument with a blank string. For example, to print “spam” to the screen without a newline character, you would call print('spam', end='').

Python provides many flexible ways to display text on the screen.