Exercise 4: The return of printing

By: Rahul Sharma 16 April, 2018

Welcome to another week of learning python from basics. And well, no matter you were extremely tired of printing the statements including variables and format strings, because this lecture is also about printing. You can rejoice because this might be the last lecture on exclusively printing stuff on your terminal screen. So, bear with me and complete the exercises in time. This lecture also includes explanation on some new things which are called *escape sequences*. So let's get started.

This time we will do an exercise first and move on with the lecture.

Exercise 1: Printing in separate lines

1. Create a file named **ex4_1.py** and type in the following code:

```
# Here's some new strange stuff, remember type it exactly.
days = "Mon Tue Wed Thu Fri Sat Sun".
months = "Jan\nFeb\nMar\nApr\nMay\nJun\nJul\nAug"
print "Here are the days: ", days
print "Here are the months: ", months
print """
There's something going on here.
With the three double- quotes.
We'll be able to type as much as we like.
Even 4 lines if we want, or 5, or 6.
"""
```

2. Run the program and write a comment above each line explaining what you saw after running the python code. Write down your mistakes and don't repeat them again. It is recommended to run the first program before you move on with reading this section. Complete the program, notice what happened and then proceed with reading the next paragraph.

Okay, so what happened in the previous code? You must have seen the days getting printed in one line and months in separate lines. The last print statement also shows the sentences in different lines. It's nothing, these are just two ways of printing the strings in different lines.The first way uses something called an **escape sequence.** Escape sequences are nothing but ways to print special characters inside a string. Or i'd rather say they are used to print characters which are difficult to print with normal strings. Try to print the double-quotes using the print statement. Just try it then move on. You saw nothing right? Now try the following code:

```
print "\"\""
```

Got it? Well, in the Exercise 1 code you used the escape sequence "\n" (new-line) which creates a new line every time python sees the characters \n. There are a number of escape sequences, an interesting one is "\\" which is used to print a "backslash (\ ). This was the first way to put strings in new lines. The second way is to use """ """ (triple-quotes). When printing a string using triple-quotes, you can write as much you want inside the quotes. It also handles the working of escape sequences automatically. Let's move on to the next exercise.

Exercise 2: Printing and escape sequences

1. Create a file named **ex4_2.py** and write the following code in it:

```
tabby_cat = "\tI'm tabbed in."
persian_cat = "I'm split\non a line."
backslash_cat = "I'm \\ a \\ cat."
fat_cat = """
I'll do a list:
\t* Cat food
\t* Fishies
\t* Catnip\n\t* Grass
"""
print tabby_cat
print persian_cat
print backslash_cat
print fat_cat
```

2. Run the code and write a comment before each line telling what is it doing.

BONUS: try to write a program using all the escape sequences listed below. Name the file bonus4.py

Here is the list of all the escape sequences supported by python. You may not use all of them, but should know what they are doing.

ESCAPE	WHAT IT DOES
\\	backlash (\ )
\'	Single-quote(')
\"	Double-quote(")
\a	ASCII bel (BEL)
\b	ASCII backspace (BS)
\f	ASCII formfeed (FF)
\n	ASCII linefeed (LF)
\N{name}	Character name in Unicode database (unicode)
\r	ASCII carriage return (CR)
\t	ASCII horizontal tab (TAB)
\uxxxx	Character with 16- bit hex value xxxx (unicode)
\uxxxxxxxx	Character with 32- bit hex value xxxxxxxx (uni)
\v	ASCII vertical tab (VT)
\ooo	Character with octal value oo
\xhh	Character with hex value hh