

More on the Print Command

The Point of Print

The `print` command is probably one the most important commands, as it allows the computer to send messages to your screen, so everything from “Hello, World!” to all of your emails, webpages, and documents (including this file) are all created using the `print` command. Some version of this command is used to display every piece of text you see on a computer screen. Here’s some terminology:

<code>Hello, World!</code>	This is the message to be printed.
<code>"Hello, World!"</code>	Any message enclosed in inverted commas is called a <i>String</i> (or a <i>String of Characters</i>).
<code>print("Hello, World!")</code>	The <code>print</code> command takes in a String and puts it on the screen. (A fancy way of saying this is “The <code>print</code> function takes in a String as a parameter and puts it on the screen.”)

And as we saw before, when we run the `print` command above we get:

```
Hello, World!
```

If you want to leave a blank line after the message, Python has a special character sequence called the *newline* character (`\n`) to make that happen. All you have to do is add this into the String. So we can add it in at the end of the String, as follows:

```
print("Hello, World!\n")
```

And we will see the following written on the screen:

```
Hello, World!
```

Alternatively we can add the *newline* character to the start of the String:

```
print("\nHello, World!")
```

And we will see the following written on the screen:

```
Hello, World!
```

If we put the *newline* character in the middle of the String:

```
print("Hello,\nWorld!")
```

And we will see the following written on the screen:

```
Hello,  
World!
```

So remember that we can put the *newline* character anywhere in the String, but it is also worth noting that we can get the same output without using the *newline* character by doing the following:

```
print("Hello,")  
print("World!")
```

And this is a crucial point to remember; with most computer programs you write, there are multiple ways to achieve the same output, and it’s up to you as the programmer to choose which approach you take. You have the power!