(Beta) ASCII Art

{ codeclub world.org }

Introduction:

Python allows you to turn a series of instructions into useful programs and fun games! In this project you'll learn how to run a Python program, and how to print text to the screen.

Activity Checklist   Follow these **INSTRUCTIONS** one by one

Test your Project   Click on the green flag to **TEST** your code

**Save your Project**   Make sure to **SAVE** your work now

1

Step 1: Saying hello

✅ Activity Checklist

1. Let's start by writing a very simple program, just so that you know how to get a Python program running. Open the IDLE program editor:

   - On Windows, find IDLE in the start menu;
   - On Mac, open up Terminal.app and type `idle` and press enter;
   - On Linux, open up a Terminal, and type `idle` and press enter.

2. Click `File → New Window`, and type the following into the window that appears:

   ```
   print("Hello, how are you?")
   ```

   This program will print some text to the screen. Notice that the text you want to print is surrounded by speech marks (`"`). Here's an image showing what you need to do:
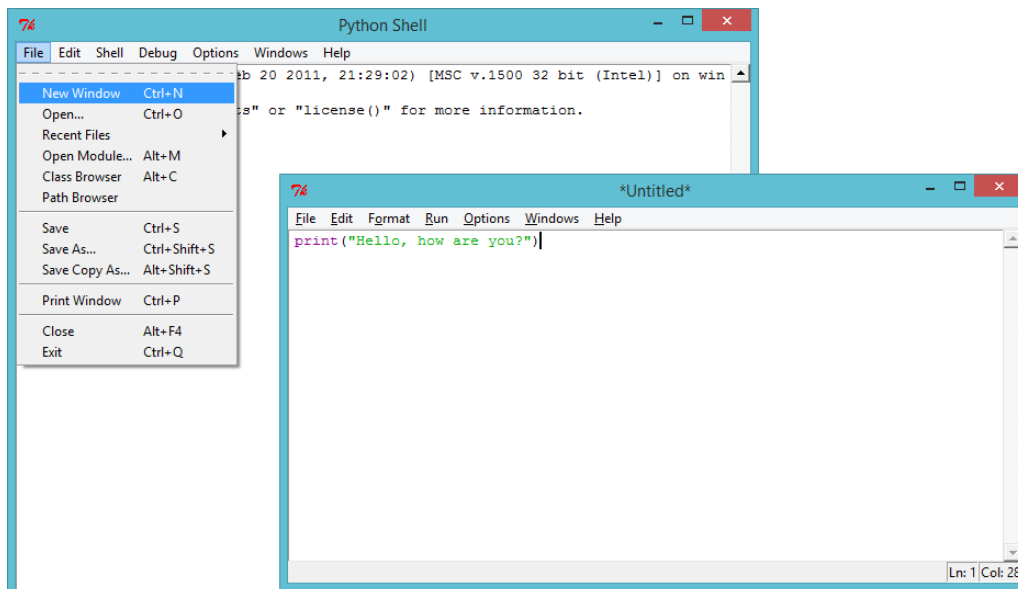

3. Save the file, by clicking `File → Save`, and name the file `hello.py` or something similar. Don't forget to type the `.py` bit at the end, which tells the computer that it's a Python file. Without it, your program won't be colour coded, which can be

really helpful.
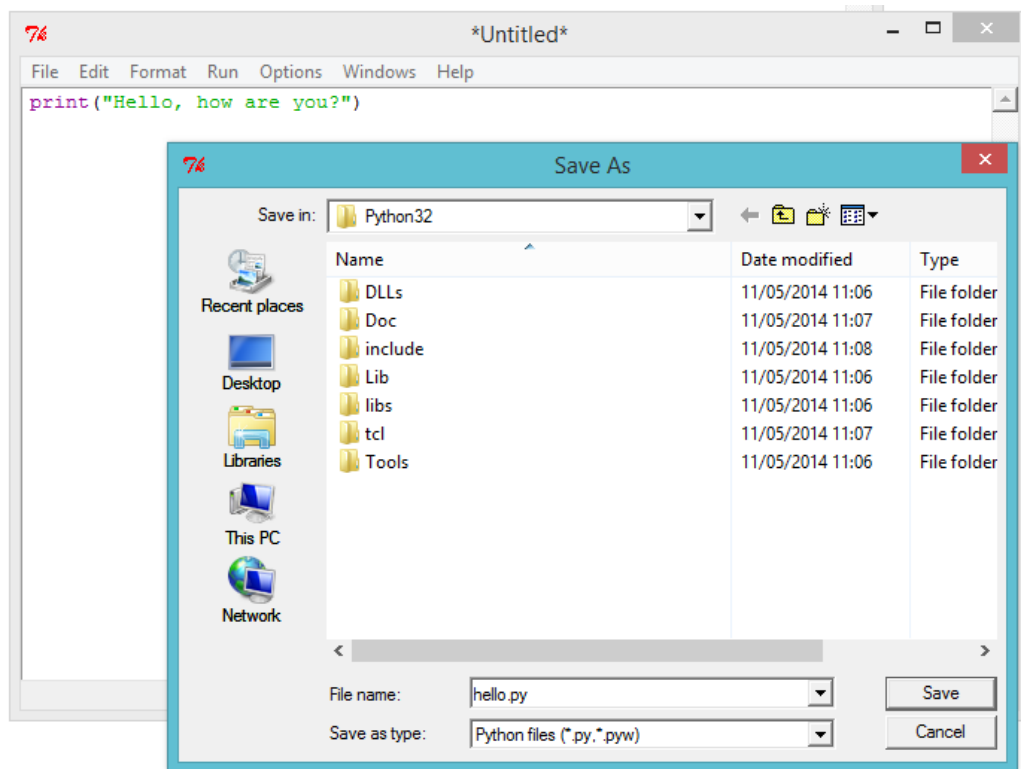
4. Run the file by clicking `Run → Run Module`. You should see another window appear, which is the Python shell. This is the place that your program will run. If everything has worked properly, you should see your text printed to the screen.
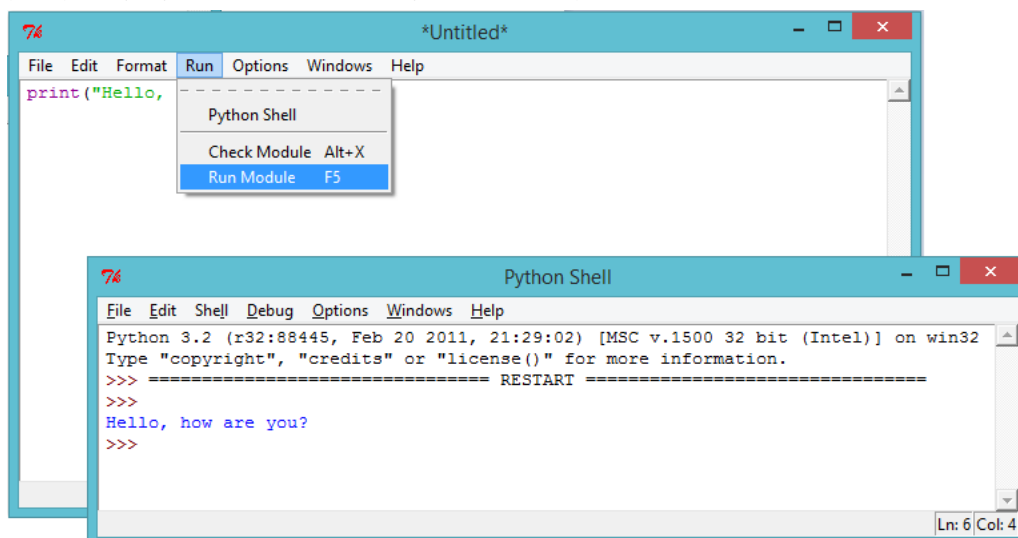
5. If you've made a mistake, for example missing out a speech mark (`"`), then you'll get an error message instead, telling you what went wrong! Try it!
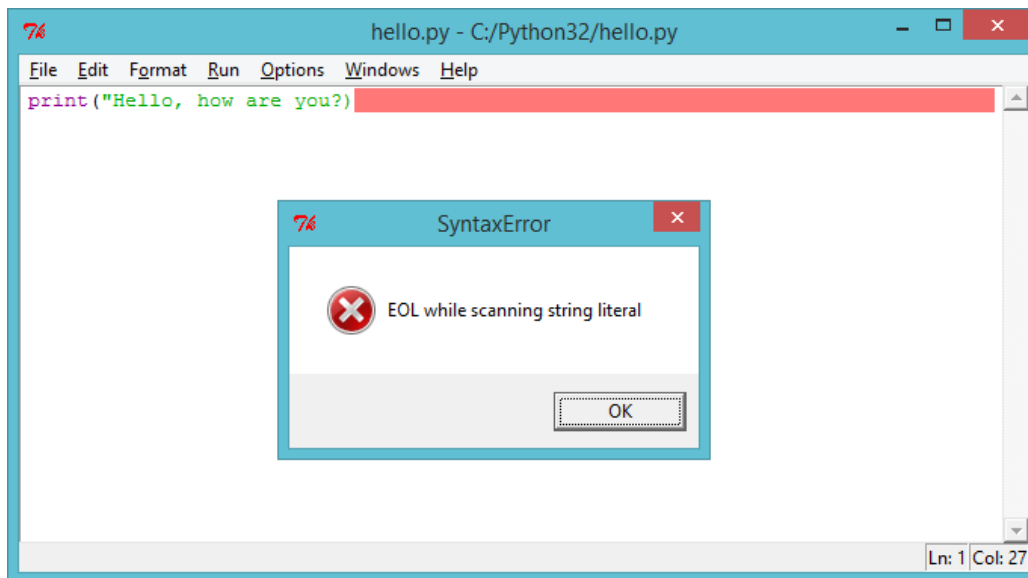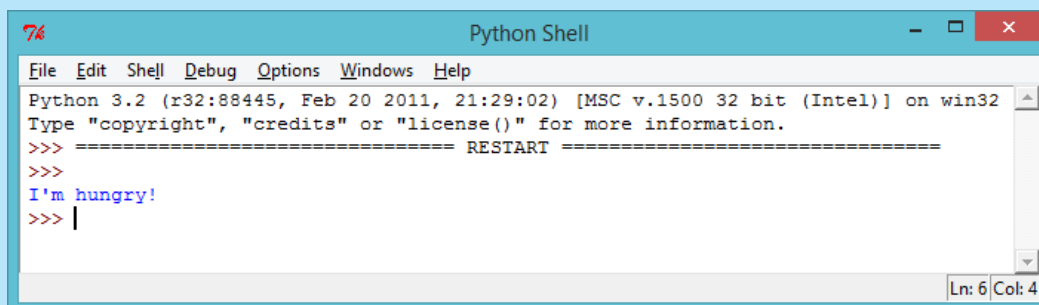
6. Congratulations, you are now officially a Python programmer! Give yourself a pat on the back (or if you're feeling lazy, get someone else to do it for you).
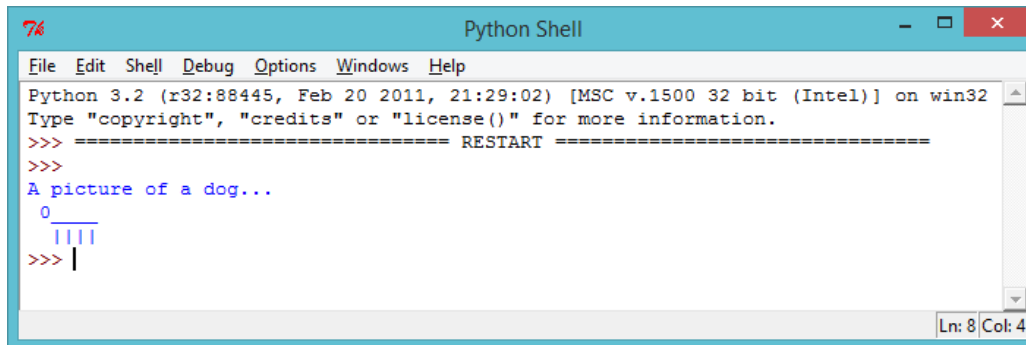
Challenge: What's on your mind?

Change the program above to print something more interesting to the screen!

Step 2: About yourself

✅ Activity Checklist

1. Let's print something much more fun than text... ASCII art! ASCII art is creating pictures out of text. Here's an example - it's meant to be a dog!

```
Python Shell                                          _  □  ✕
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ============================== RESTART ==============================
>>>
A picture of a dog...
 0____
  ||||
>>> |
                                                             Ln: 8 Col: 4
```

To make this masterpiece, you can type the following into the IDLE editor and run the program:
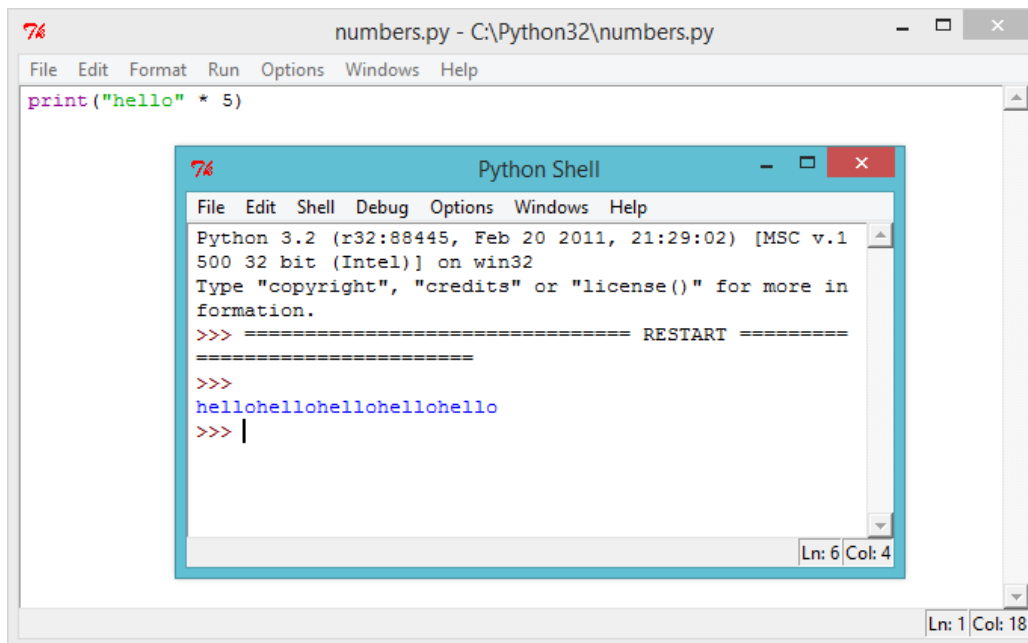
```python
print("A picture of a dog...")
print(" 0____  ")
print("  ||||  ")
```

2. If you prefer, you can use 3 single quotes (``` ''' ```) instead of speech marks, which allows you to print multiple lines of text with 1 print statement. Like this:

```python
print('''
A picture of a dog...
 0____
  ||||
''')
```

If you run this program, you'll see it prints the same dog as before.

⑤

Challenge: About yourself

Write a Python program to tell others about yourself, by using text and ASCII art. You can create images of your hobbies, friends, family... anything you want! Here's an example:

```
7₄                          Python Shell                      _  □  ×

File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2.5 (default, May 15 2013, 23:06:03) [MSC v.1500 32 bit (Intel)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> =============================== RESTART ===============================
>>>
My favourite animals are sheep:

  o-###-
   | |   #

I live in Glasgow:

    _|_
   |   |
   |#  |___
   |   |   |
   |  #|  # |
   |   | #  |
>>>
                                                              Ln: 9 Col: 0
```

Step 3: Calculating text

✓ Activity Checklist

1. Python can also do calculations using text! What do you get if you multiply `"hello"` by 5? Let's ask Python, by running this program: ☐

```
print("hello" * 5)
```

The star `*` in the program above is a multiply sign. Run the program above, and you should see the answer:

2. You can make the printed text above easier to read, by putting a space after the word `"hello"` in your program:

```python
print("hello " * 5)
```

Run this program and you'll see that the output is a little easier to read than before.

3. If `"hello "` multiplied by 5 is `"hello hello hello hello hello "`, then what is `"hello" - 7`? Does this calculation even make sense?
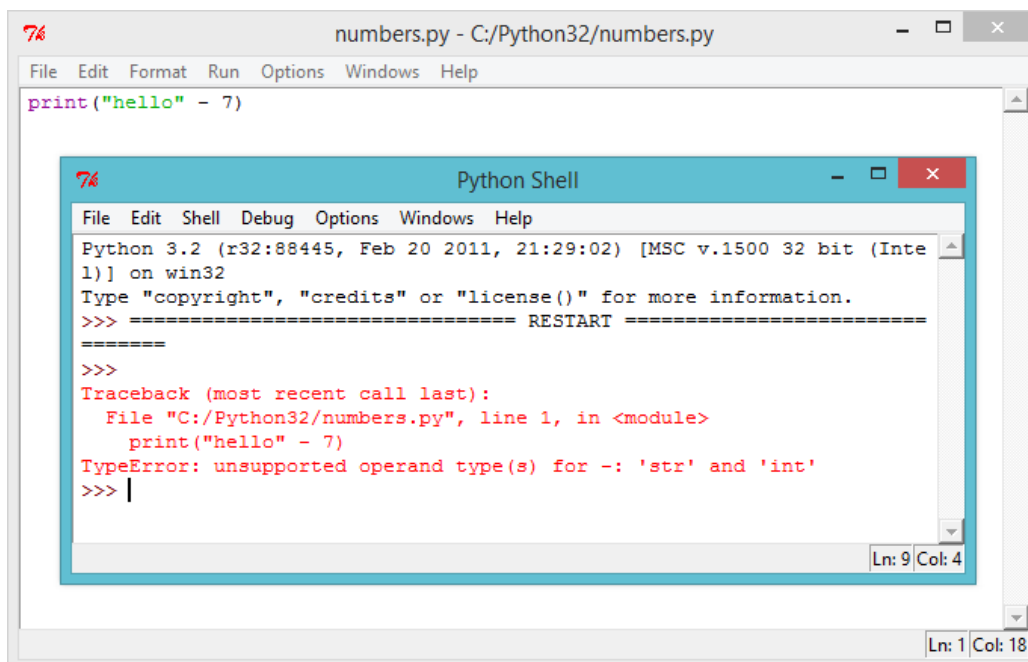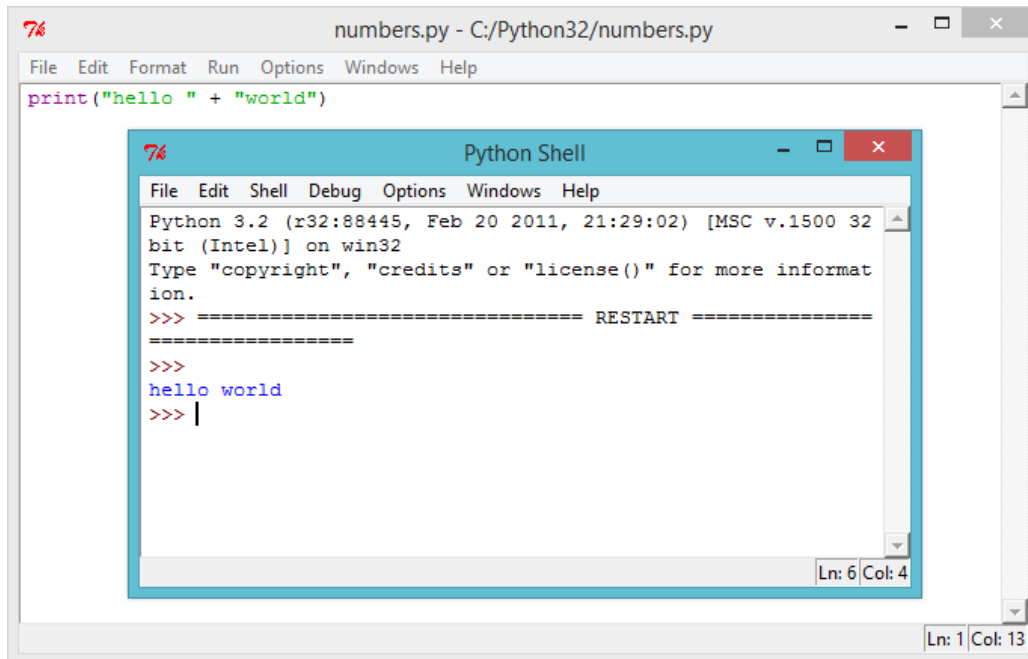

Oops, you've broken it! Instead of an answer, we get an error

message. It looks like that calculation doesn't make sense in Python!

4. How about addition? What answer do you think `"hello "` + `"world` " would give? Try it out, by running the following program:

```
print("hello " + "world")
```

```
numbers.py - C:/Python32/numbers.py
File  Edit  Format  Run  Options  Windows  Help
print("hello " + "world")
```
```
Python Shell
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informat
ion.
>>> ================================ RESTART ===============
==================
>>>
hello world
>>> |
                                                      Ln: 6 Col: 4
```
```
                                                      Ln: 1 Col: 13
```

Does it give you the answer you expected?

Challenge: Words and numbers

What does the following program print to the screen? See if you can guess correctly before running the program.

```
print("ha "*4)
print("ba" + "na"*2)
print("He" + "l"*2 + "o" + "!"*10)
```

Can you make up any words of your own?

✅ Activity Checklist
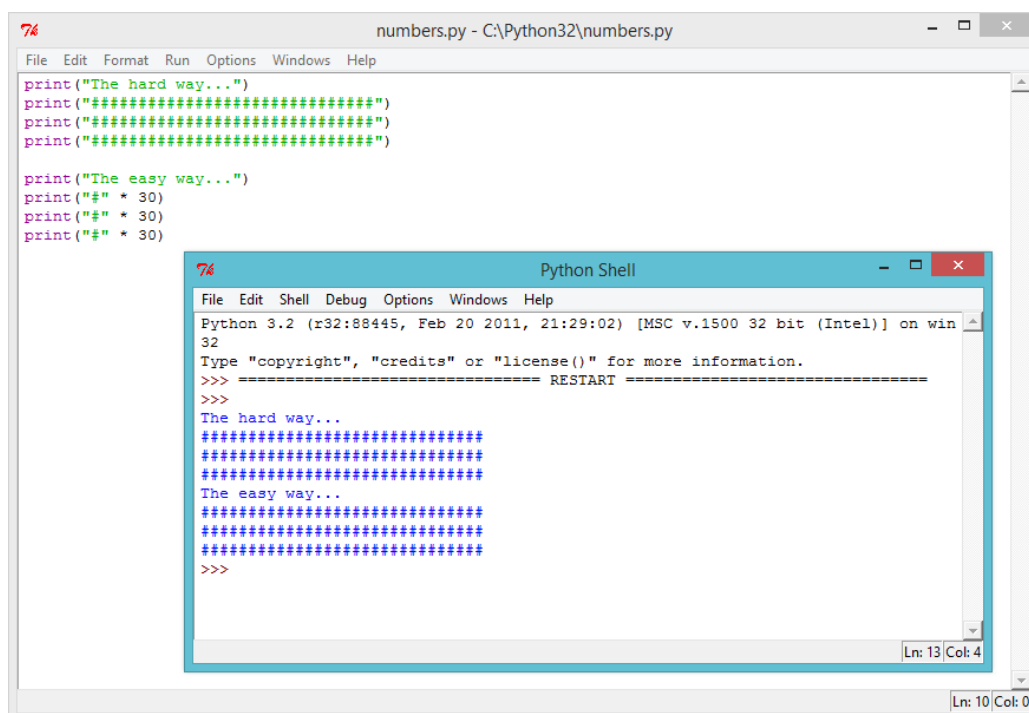
1. Now that you know how to do calculations on text, now what? Why is it useful? Well, let's say you wanted to draw an ASCII art rectangle that is 30 characters long and 3 characters high. You could either draw it the hard way, like this: ☐

```python
print("##############################")
print("##############################")
print("##############################")
```

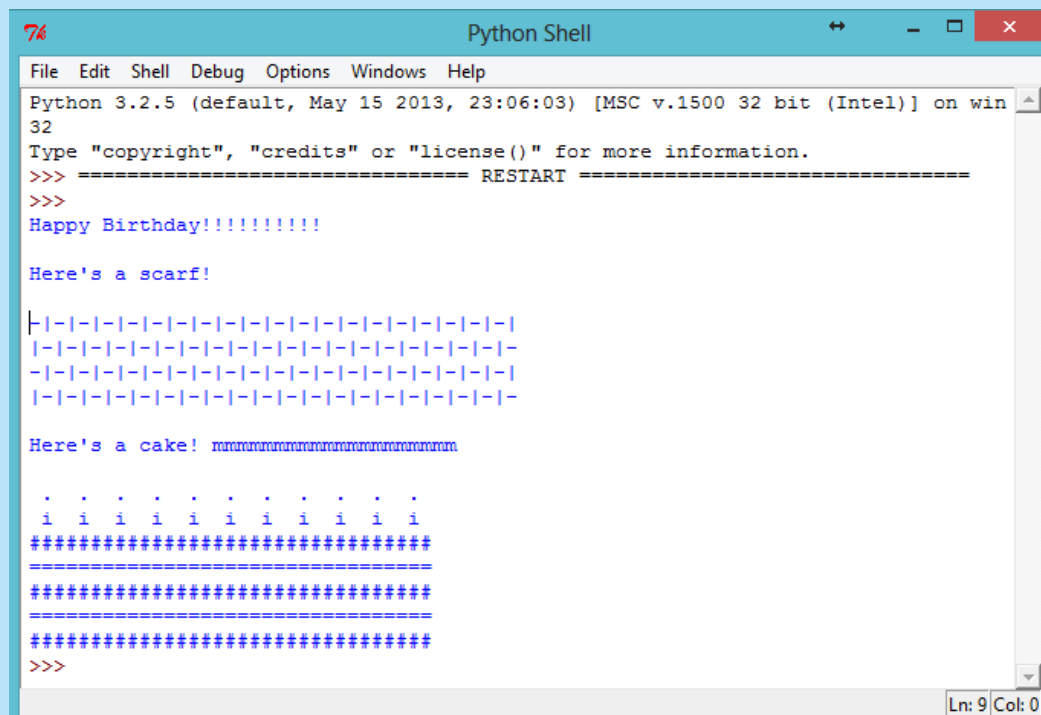Or you could save time and draw it the easy way, like this:

```python
print("#" * 30)
print("#" * 30)
print("#" * 30)
```
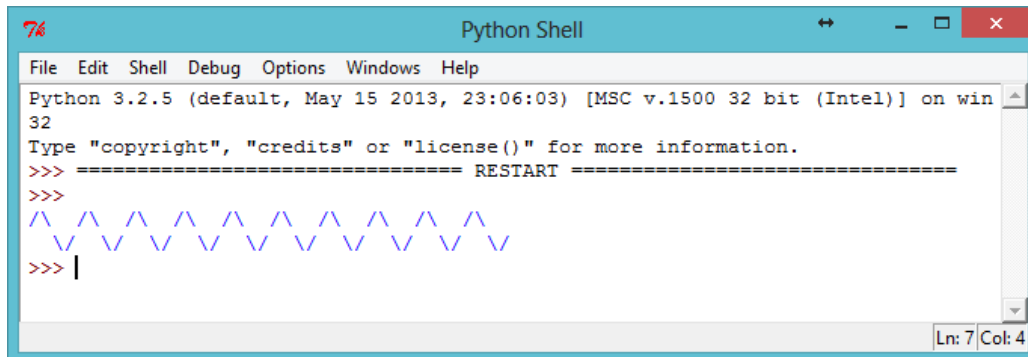
Both give you exactly the same rectangle printed to the screen:

2. You could even use calculations to make interesting patterns, ☐

like this wave:

```
print("/\  "*10)
print("  \/"*10)
```


Challenge: Code a scarf

Your best friend is having an 11th birthday party, and as a gift you've decided to code them a scarf! Use calculations wherever possible to make your own scarf pattern.

If you're feeling generous, you could even code them a cake (including 11 candles) to go with it!