

All Code Clubs must be registered. Registered clubs appear on the map at codeclubworld.org - if your club is not on the map then visit jumpto.cc/18CpLPy to find out what to do.

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

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:



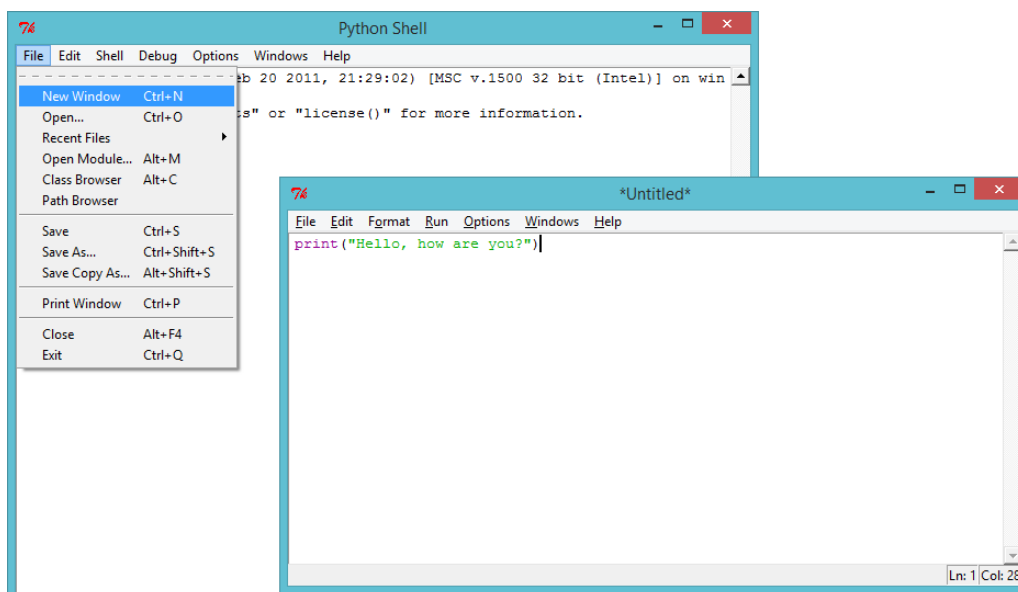
- On Windows, find IDLE in the start menu;
- On Mac, open up Terminal.app and type `idle` and press enter;
- On Linux, open up a Terminal, and type `idle` and press enter.

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:

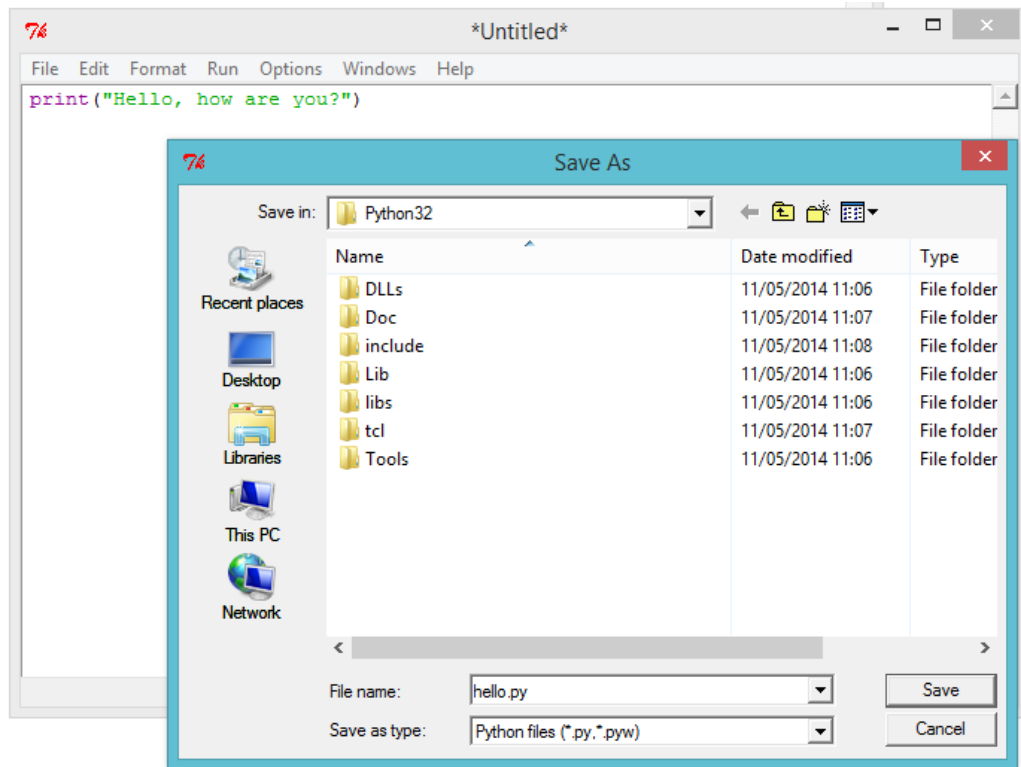


3. Save the file, by clicking `File → Save`, and name the file

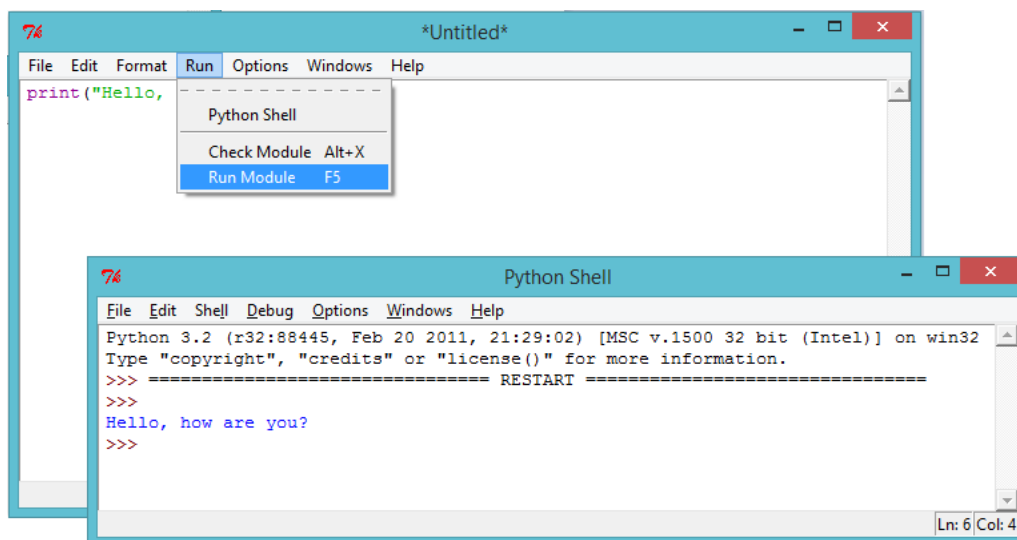


2

`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.

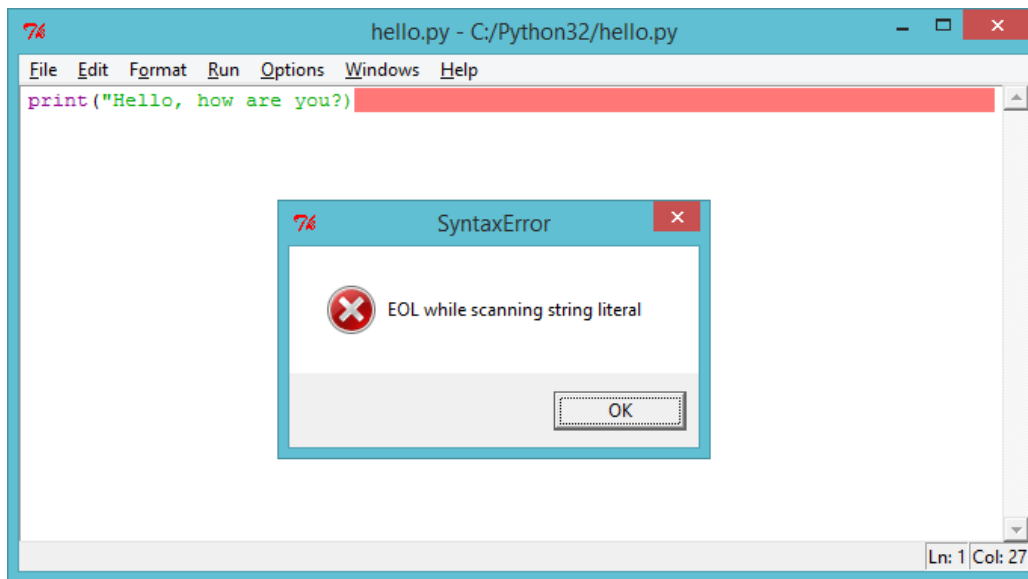


4. Run the file by clicking `Run → Run Module`. You should see another window appear, which is the Python shell. This is the place that your program will run. If everything has worked properly, you should see your text printed to the screen.



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!



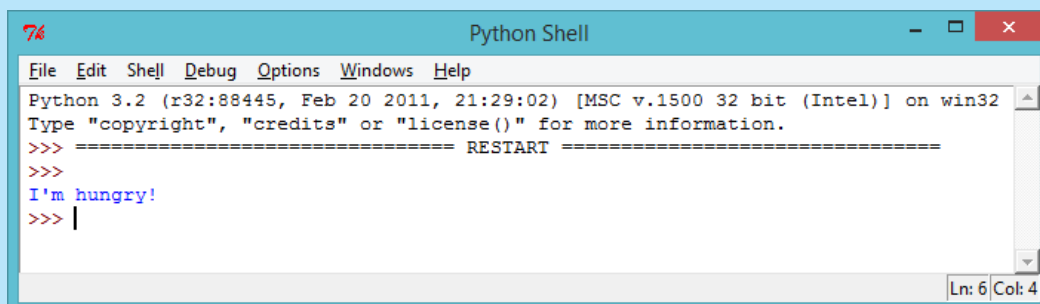
6. Congratulations, you are now officially a Python programmer! Give yourself a pat on the back (or if you're feeling lazy, get someone else to do it for you).



Save Your Project

Challenge: What's on your mind?

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

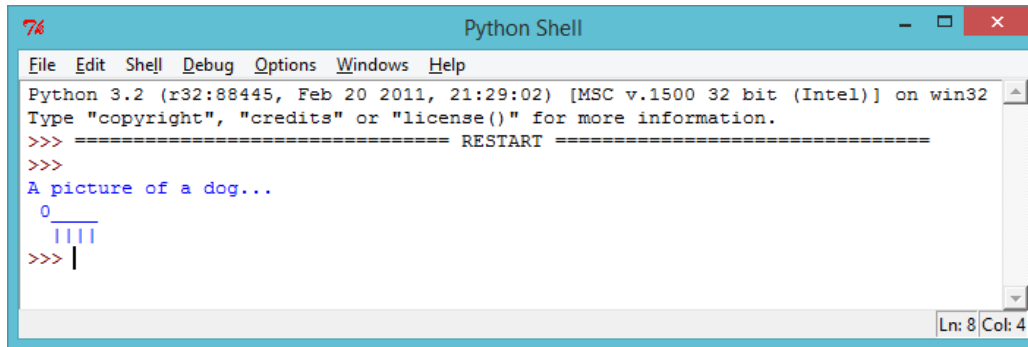


Save Your Project

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!

A screenshot of a Python Shell window titled "Python Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the Python 3.2 prompt and the following text:

```
>>> ===== RESTART =====  
>>>  
A picture of a dog...  
  0____  
  ||||  
>>> |
```

The status bar at the bottom right indicates "Ln: 8 Col: 4".

To make this masterpiece, you can type the following into the IDLE editor and run the program:

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

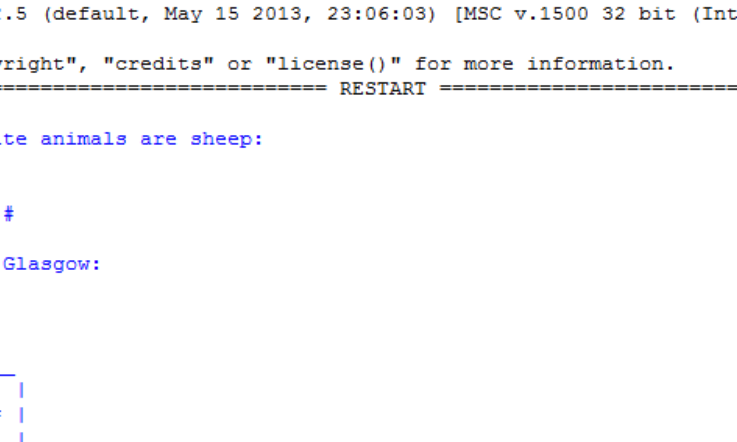
2. If you prefer, you can use 3 single quotes (`'''`) instead of speech marks, which allows you to print multiple lines of text with 1 print statement. Like this: ☐

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

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



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:



The screenshot shows a Python Shell window with the title "Python Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area displays the following content:

```
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:

  _|_
 |#|_|
 |#|#|
 |#|#|
 |#|#|
>>>
```

The drawing consists of a circle for a head, three vertical lines for legs, and a horizontal line for a tail. The sheep is standing on a base of vertical lines, with horizontal lines connecting some of them to form a body.



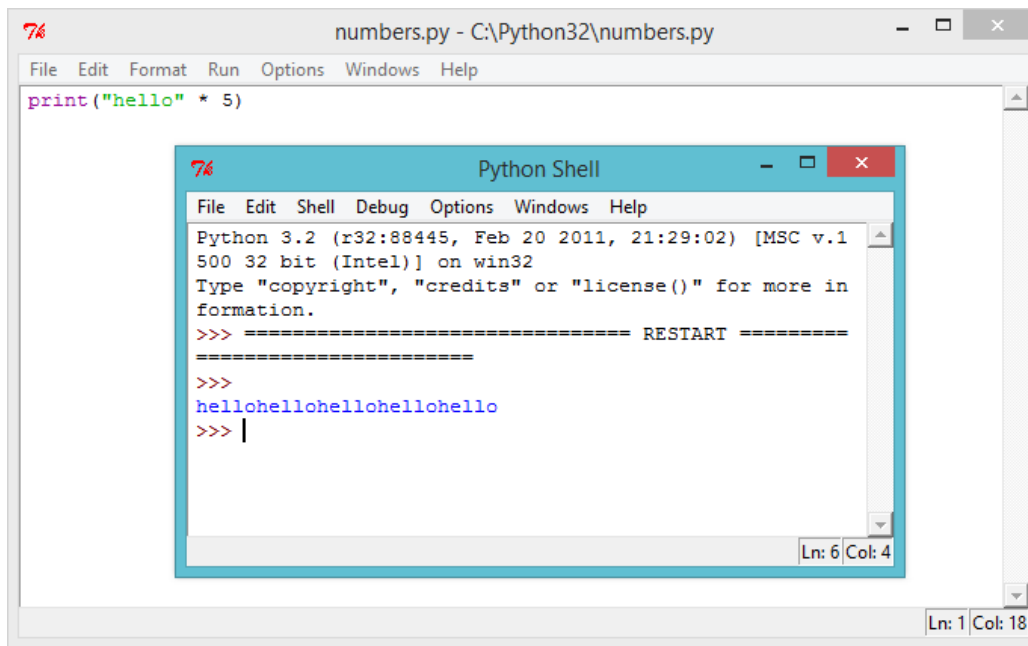
Step 3: Calculating text

Activity Checklist

-

```
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:

```
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?

The screenshot shows a Python IDE window titled 'numbers.py - C:/Python32/numbers.py'. The code in the editor is `print("hello" - 7)`. A 'Python Shell' window is open, displaying the error message: `TypeError: unsupported operand type(s) for -: 'str' and 'int'`. The shell also shows a traceback pointing to line 1 of the module. The status bar at the bottom of the shell indicates 'Ln: 9 Col: 4'.

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")
```

The screenshot shows the same Python IDE window, but the code has been changed to `print("hello " + "world")`. The 'Python Shell' window now displays the output: `hello world`. The status bar at the bottom of the shell indicates 'Ln: 6 Col: 4'.

Does it give you the answer you expected?



Save Your Project

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?



Save Your Project

Step 4: ASCII patterns



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:

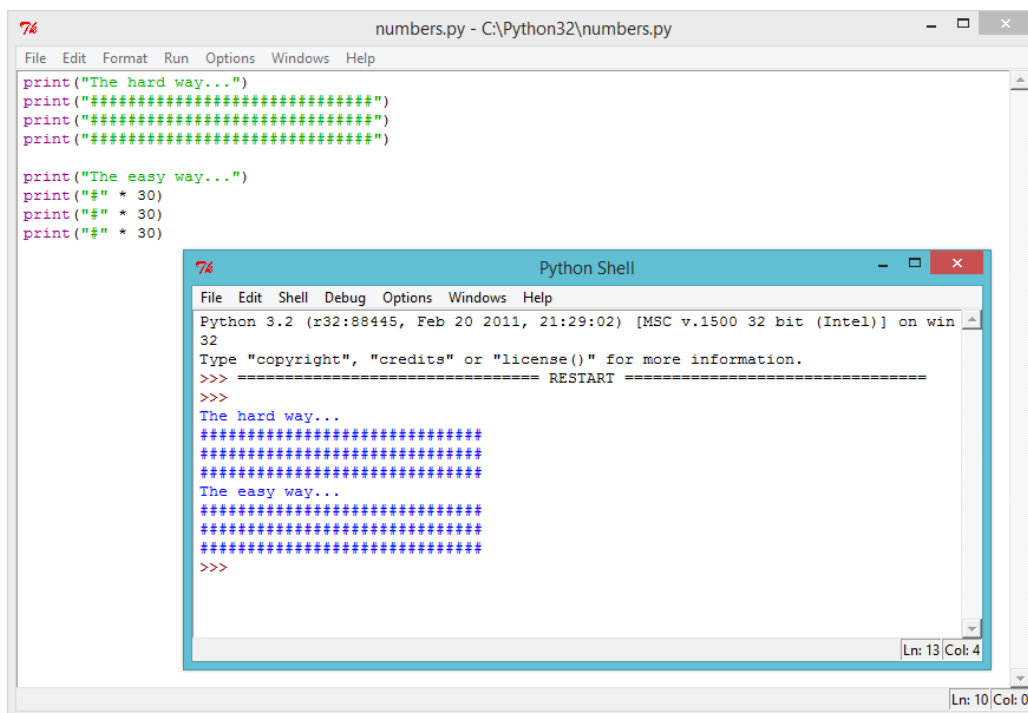


```
print("#####")
print("#####")
print("#####")
```

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

```
print("#" * 30)
print("#" * 30)
print("#" * 30)
```

Both give you exactly the same rectangle printed to the screen:

The screenshot shows a Python IDE window titled 'numbers.py - C:\Python32\numbers.py'. The script contains two sections: 'The hard way...' which prints three lines of 30 '#' characters each, and 'The easy way...' which prints three lines of 30 '#' characters each using string multiplication. Below the script, a 'Python Shell' window shows the execution output. It displays the same three lines of 30 '#' characters for both methods, confirming they produce identical results. The shell window also shows the Python version (3.2) and a 'RESTART' prompt.

2. You could even use calculations to make interesting patterns, like this wave:



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

