

# Chapter 1

## Python, IDLE and your first program

In this chapter you are going to:

- learn about computer programming and the different languages that you can use
- meet the Python programming language
- learn how to use IDLE, which will help organise your programs and allow you to run them easily
- check that your computer has been set up correctly
- write and run your first program.

I am never IDLE, although I do occasionally get a bit sleepy...zzzz.



# Coding

Coding is writing instructions for a computer to perform a task. This code has to be in a form that the computer can understand. This is more formally known as computer programming.

Computers and coding have not been around for a long time but they have sure packed in some interesting history in a short space of time. The first machine that stored instructions in a way that future computers could take advantage of was the Jacquard loom that used holes punched in cards and was invented in 1801. Charles Babbage is often credited with inventing the first computer which he described in 1837 but was not built until 100 years later. In 1989 Guido van Rossum started to create the Python programming language which he named after Monty Python's Flying Circus, a BBC comedy sketch show.

## Programming languages

There are many programming languages currently used by coders around the world. Some are best in one situation, others in another.

- HTML is good for producing web pages.
- SQL is great at making databases do what you want.
- Python is brilliant for writing quick applications, running programming experiments and for building larger applications, including games.

If you have previously programmed in Scratch (produced by MIT) you will find you can pick up Python very quickly. Scratch is great for learning how to think like a programmer and is very good for making games. If you have not tried Scratch before, you might enjoy trying that next because the ability to learn a new programming language is an important skill for coders. You will find it is a lot easier than learning a new human language.

Once you have learned one modern programming language, you can quickly learn others. You simply have to find out how your new language handles variables, loops, etc. (You will know what these are by the end of the book.)

## Python

Python is a typed computer language. This makes writing short programs very fast and you can produce almost anything you can imagine.

Python is a powerful, modern programming language used by many famous organisations such as YouTube and NASA. It is one of three programming languages that can be used to write Google Apps. Python is a great language. Enjoy!

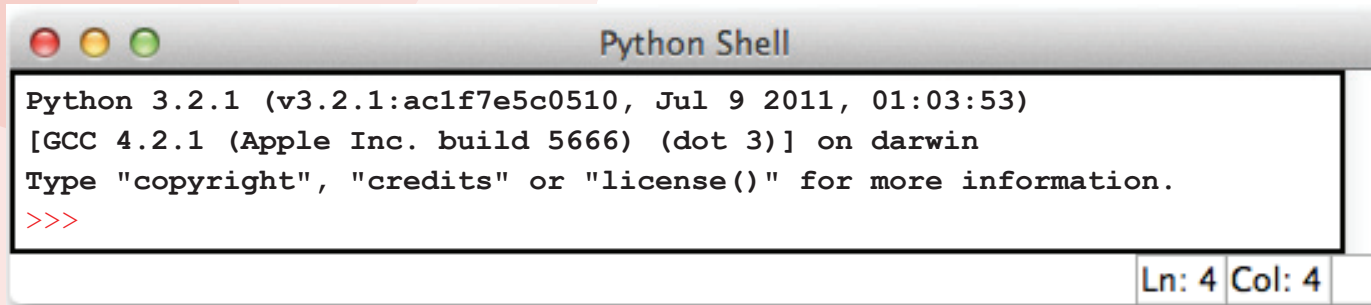
## IDLE

You will start programming in IDLE which comes with Python. IDLE is a special text editor like Microsoft Word, except it understands Python and helps you get your code right. IDLE is itself, a Python application.

Python is also one of the languages used by the European Particle Accelerator organisation, CERN.



Let's look at IDLE:



```
Python 3.2.1 (v3.2.1:ac1f7e5c0510, Jul 9 2011, 01:03:53)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
```

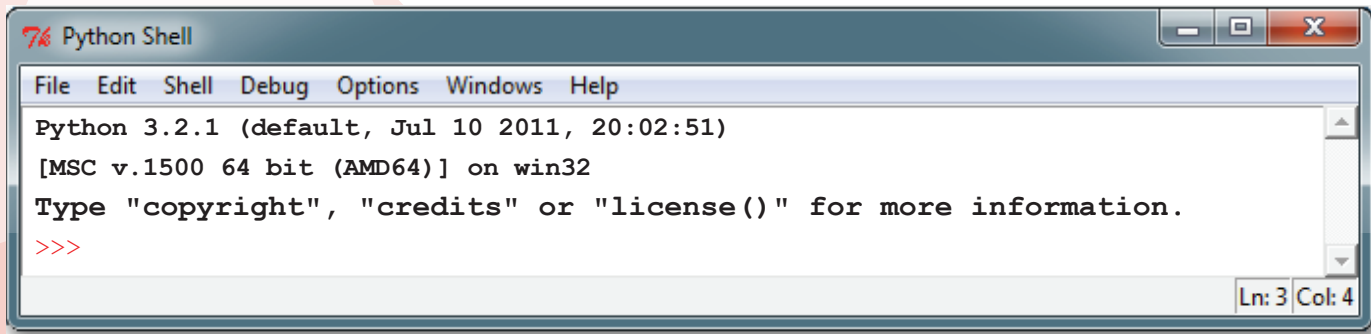
IDLE when started on an Apple Mac.

The code you want to run is typed after the special entry prompt:

```
>>> my code goes here
```

To run the code we press the return key. This is how Python runs in IDLE's **interactive mode**. Python can run files as well but to start with, this is all we need.

Let's see how IDLE looks on a Windows PC:



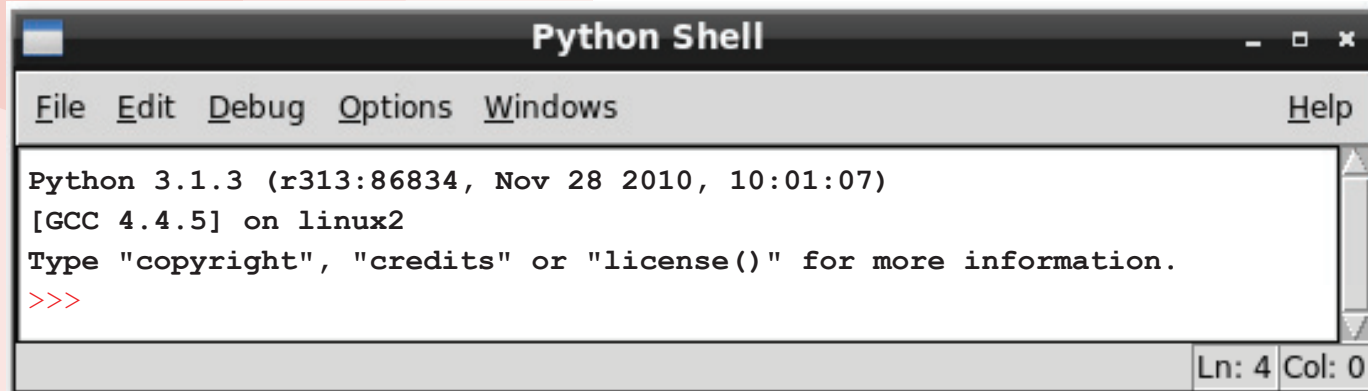
```
Python 3.2.1 (default, Jul 10 2011, 20:02:51)
[MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

IDLE when started on Windows 7.

It is important when learning a programming language to learn the special vocabulary that goes with it. This is because when you want to try to find something out, you know which keywords to search for. This is why new 'computer speak' words appear in **bold**. This means that they will be explained in the glossary at the end of the book. Obviously **bold** will not appear in the glossary!



And finally, how IDLE looks on a Linux computer:



```
Python Shell
File Edit Debug Options Windows Help
Python 3.1.3 (r313:86834, Nov 28 2010, 10:01:07)
[GCC 4.4.5] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>
Ln: 4 Col: 0
```

IDLE when started up on the Raspberry Pi computer.

A great reason for learning Python and using IDLE as our **IDE** (Integrated Development Environment) is that it is very similar on all the different types of computers available.

The text before the `>>>` prompt is unimportant at the moment. However, it is always useful to know what version of Python you are using.

## Hello World!

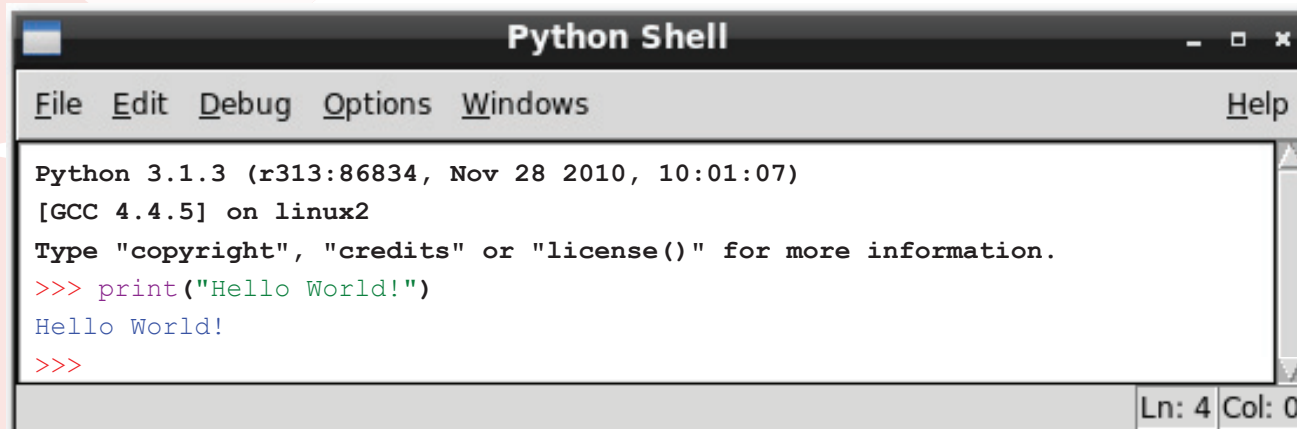
Since the dawn of programming, when the first cave-coders booted up their cave-computers, it has been a tradition that your first program when learning a new language is 'Hello World'. The aim is to try to make the computer say 'hello' to the world. If you can do this you will have tested whether everything that was set up for you is working properly.

- If it is not already started, start up IDLE.
- After the `>>>` prompt write in the code from Code Box 1.1 and then press your return key to run the program.

### Code Box 1.1

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

If all is well, you should get something like this:



```
Python Shell
File Edit Debug Options Windows Help
Python 3.1.3 (r313:86834, Nov 28 2010, 10:01:07)
[GCC 4.4.5] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> print("Hello World!")
Hello World!
>>>
```

Hello World!

Python has followed your instruction and **output** 'Hello World!'

You have written your first computer program. Well done!

This might not seem much to you, but you have given your computer a direct instruction that you have written yourself and it has carried out your instruction. From this small seed, great applications will grow!



# Making mistakes

## Did you get a syntax error?

**Syntax errors** are very common when typing in code (as are other errors). If you make one or two it is not your fault. It is because although computers are fast, they can also be a bit stupid. If there are any tiny mistakes in your code, they panic and produce error messages. These messages try to explain to you what the problem is but they are often difficult to understand.

Colons, brackets, speech marks, apostrophes and spelling of Python words have to be just right. Although we can read imperfect sentences, computers cannot.

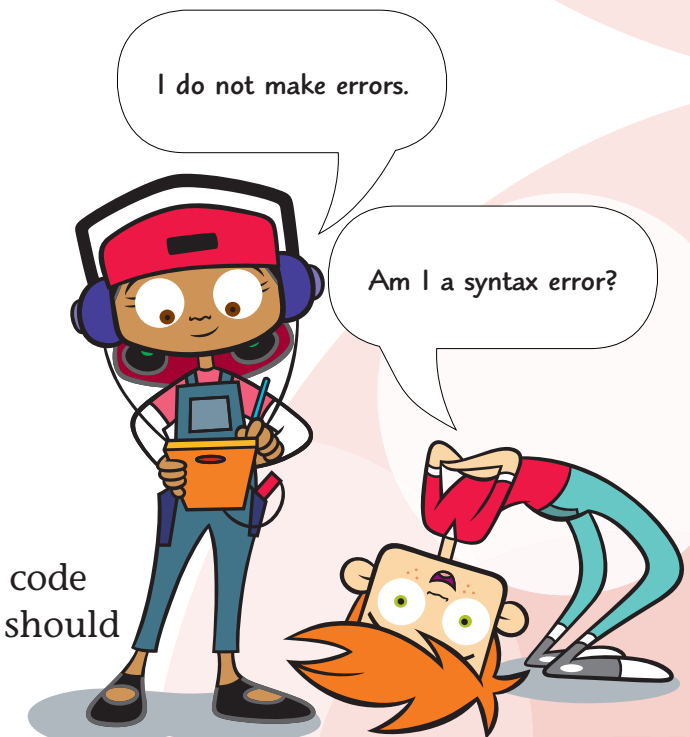
Whether or not you got any errors, try this Quick Quiz.

### ? Quick Quiz 1

Which of these lines of code are correct?

- 1 `Print("Hello world!")`
- 2 `print("Hello world!")`
- 3 `print(Hello world!)`
- 4 `print "Hello world!"`

Notice how the coloured text helps you spot code that is not going to work. All the code listings in this book use the same coloured text as in IDLE's standard display. This should help you to spot **bugs** in your code.



# Chapter summary

In this chapter you have learned:

- that programming is writing instructions for computers
- that there are many different computer languages
- why Python is a great language to learn
- how to use IDLE in interactive mode
- how to write and run a simple program
- that the `print()` command means 'show on the screen' not 'send to the printer'.

## Idea 1

- 1 Write some new code so that a short message is displayed that says thank you to whoever got everything ready for you.
- 2 Run your new code to display the message.
- 3 Now show them your message. This will make them happy.

## Idea 2

- 1 Write some code so that the computer will show the text for a joke.

```
>>> print("Question: What goes clip?")
Question: What goes clip?
>>> print("Answer: A one legged horse")
Answer: A one legged horse
>>>
```

