Program in Python at home

If you've enjoyed programming in Python and you have access to a computer at home, why not install Python so you can do some programming while you're missing school over the long, boring Summer Holiday.

You can install and use Python *completely free of charge* and there's lots of stuff on the web to help you, including lots that's written specifically for children.

Installing Python

First, make sure you have permission from an adult and/or the owner of the computer.

Go to www.python.org and click on "Downloads". Now you need to know what sort of machine you're going to install on:

- Windows
- Linux/UNIX
- Mac OSX

Follow the appropriate link. Next you need to choose between "Latest Python 2 Release" and "Latest Python 3 release". Choose Latest Python **3** release. Finally, for Windows, you need to know whether you are running a 32-bit or 64-bit version of windows (if you have to guess, guess "32-bit" if you are running Windows XP and "64-bit" otherwise. If you guess wrong, you will get an error when you try to install and you can simply try again with the other version).

What to do next

Once you have installed Python successfully you should see a menu entry for IDLE and it should start up just as it does when you use it at school. Remember:

- The first window that opens, with the ">>>" prompt, is the *python shell*
- Use the `File->New Window` menu option to open an *editor window*
- Choose somewhere (like My Documents) to save your programs, and save them all in the same place so that you can find them again later

Before the end of term

Bring in a memory stick and take a copy of the programs you have written in code club. You can use these as a starting point for experiments or for writing new programs.

Also, you can get PDF copies of any of the code club projects we have done.

More turtle graphics

A nice tutorial in using Python turtle graphics:

http://openbookproject.net/thinkcs/python/english3e/hello_little_turtles.html

A presentation by the person who wrote the turtle graphics for Python:

https://code.google.com/p/python-turtle-demo/downloads/list

The presentation is quite technical but has some great pictures made using Python Turtle. You can download TurtleDemo-1.0.0-py3.x.zip with *all of the demonstration programs and more*, and use them to modify and to learn from.

Invent your own computer games with Python

This is a very nice book which teaches Python programming through the development of a set of games. The author has made the whole book available free-of-charge so that you can read it and follow along by typing in the programs (or you can cheat and download the programs)

http://inventwithpython.com/

If you work your way through all the examples in this book you will have a pretty good grounding in Python and you should be able to use what you've learned to use your own ideas to design and write programs.

Built-in help

From the Python Shell use the `Help ->Python Docs` menu option to get access to almost everything there is to know about Python. Remember to look for Python 3 stuff, *not* Python 2 stuff. There are links to a tutorial and a beginner's guide and a Quick Search box

Elsewhere on the Internet

If you get stuck and you decide to use a search engine, remember to use the Internet safely, as you have been taught in school, and ask an adult to help you if you get stuck or confused about something you've found.

If you are searching for something python-related, always include the word "python" in your search. For example, if you are looking for something to do with turtle graphics you need to search for something like "python turtle graphics".

If you get stuck

It's very rare that you (or anyone) will write a program of more than 5 or so lines and have it work perfectly the first time. You will get two sorts of errors:

- You use the language wrongly – these are called "syntax errors" and the computer will detect them and refuse to run your program. It will try to help you by telling you the line number where it thinks the error occurred.
- Your program does what you said but not what you wanted – these are called "semantic errors". The computer cannot help you. Stare at your program and try to see where it is starting to go wrong. Try breaking the program into smaller parts, adding print statements to report what's going on, slowing down the turtle (if it's a turtle program) so that you can follow what it's doing step by step.