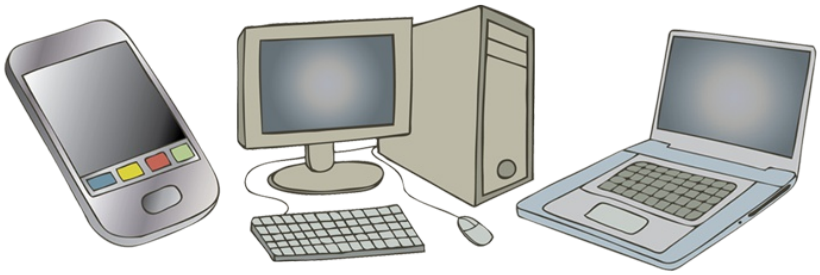
Programming - An Introduction

Many of us today enjoy the benefits of having electronic devices such as computers, smartphones and game consoles. These devices enable us, at the touch of the button, to communicate, search the web, watch videos, take photographs and so much more.



However, apart from using these devices as designed by others, is it possible to use them in other ways, perhaps in ways not yet thought of? Well you can, but only after you have learnt how to communicate with them.

Talking to machines

Some modern computers can execute a billion instructions per second.

Devices like computers can do wonderfully complicated tasks like predicting the weather, controlling the launch of a rocket and locating a position anywhere in the world using GPS. But how do computers and other electronic devices launch a rocket or predict what the weather may be on Tuesday afternoon in Beijing? The truth is, they may seem super intelligent but they are simply following sets of relatively simple instructions at super fast speeds; and importantly those instructions were created and given to those machines by people - programmers.

Computers only really understand instructions encoded as numbers (machine code). A program written in a higher level language is decoded into machine code by a program called a 'compiler' or 'interpreter'.

Programming languages enable programmers to communicate with and control computional devices. Most languages are termed 'high level languages' so that they are easier to write, read and understand. Like the English language, computer languages have their own vocabulary (words), punctuation (syntax) and rules (grammar), which allows programmers to combine those words to express meaning and purpose that machines can understand.

In these examples the word enclosed by double quotes is a data type called a 'string'. The word preceding the '=' sign is a variable and it represents a space in the computer's memory where the data is stored. All of this will be explained later.

Programming languages

There are many different programming languages and a few that you may meet in school are listed below. For each language there is some example code that will print the name 'Blackadder' on the screen. Notice that the first line declares a single piece of data and the second line contains a operation, functions or method, which takes the data and then does something to it, in this case prints the data on the screen.

Microsoft® Small Basic

Small Basic - a small language which is easy to learn. It has an 'intellisense' system which helps the programmer select the correct command, even if spelling is a challenge. Programs made with Small Basic can be shared with friends if they import your published programs. Your creations can even be played in a web-browser, by using the Silverlight plugin.

```
1. name = "Blackadder"
2. TextWindow.WriteLine(name)
```

python™

Python is a fast and powerful language which is relatively easy to use. It is also a good first 'typed' language which has huge possibilities. The same code can run on many different computers. Python has a huge library of existing programs, which enables Python to do almost anything.

```
name = "Blackadder"
print(name)
```

Javascript is the computer language that the web uses to add interactivity. The code runs inside your browser, which makes it a very a portable language. Many modern websites use HTML (see the previous chapter) and Javascript together.

```
1. name = "Blackadder";
2. alert(name);
```

Ruby is an object orientated language (don't worry about what that means) that is powerful and flexible. It is often used in web/internet development. Children can use a version of Ruby called 'KidsRuby', which is a free download.

```
1. name = "Blackadder"
2. puts(name)
```

In all four cases there are special commands which tell the computer to print the data on the screen (WriteLine(), print(), alert() and puts()).

Different computer languages have a huge overlap of common commands, and knowing one language helps you to learn another. Also the words are chosen to be very readable and understandable by humans. For example, can you guess what Clear() in Small Basic tells the computer to do?