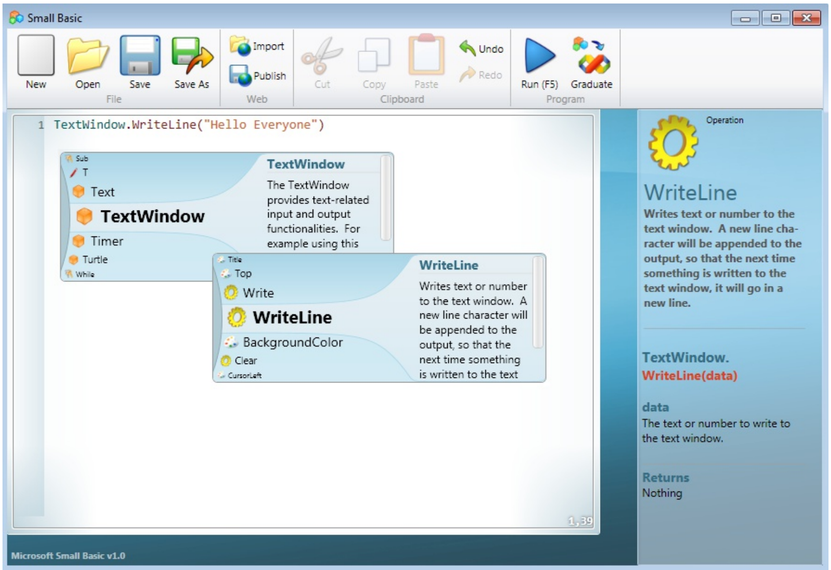
The intellisense system enables programmers to select rather than type words. This is particularly useful because it allows exploration of the language. It also helps children who find spelling a challenge.

Writing a computer program

Let's write our first program using Small Basic. It will be very simple and will cause the words 'Hello Everyone' to be printed on the computer screen. In fact they will be printed in a small window called a 'terminal emulator'.

Use the intellisense system to select the object 'TextWindow', then put a fullstop. Once you have put a fullstop the intellisense window will pop up again and you should select 'WriteLine'. Then inside the opened and closed brackets type "Hello Everyone". Don't forget to type the speech marks!

```
TextWindow.WriteLine("Hello Everyone")
```



```
TextWindow.WriteLine("Hello Everyone")
```

TextWindow

The TextWindow provides text-related input and output functionalities. For example using this

WriteLine

Writes text or number to the text window. A new line character will be appended to the output, so that the next time something is written to the text

WriteLine

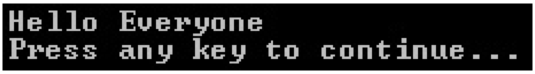
Writes text or number to the text window. A new line character will be appended to the output, so that the next time something is written to the text window, it will go in a new line.

TextWindow.
WriteLine(data)

data
The text or number to write to the text window.

Returns
Nothing

When you run the program (F5 or press the blue run arrow) you should see the following output.

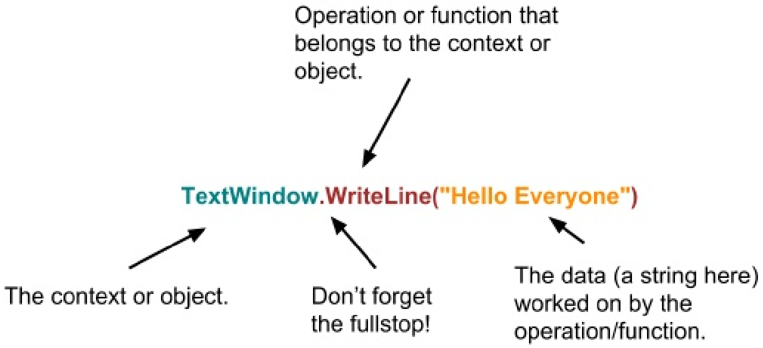


```
Hello Everyone
Press any key to continue...
```

Literals are constant data values written directly into your programming code. They often appear on the right-hand side of an assignment operator (=).

What just happened?

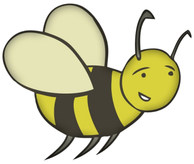
WriteLine() is an operation that will display whatever data is placed inside the brackets. "Hello Everyone" is the data or stuff that WriteLine() operates on. Small Basic knows that anything between speech marks is a string literal (a line of characters or letters) data type.



```
TextWindow.WriteLine("Hello Everyone")
```

Operation or function that belongs to the context or object.
The context or object.
Don't forget the fullstop!
The data (a string here) worked on by the operation/function.

TextWindow is the console window that you saw pop up when the program ran. WriteLine() only works with the TextWindow, infact it is an operation that belongs to the TextWindow. It gives the WriteLine ("Hello Everyone") operation a context to work in - a place where the data can be printed. In Small Basic almost every operation needs a context placed before it followed by a full stop, because every operation belongs to a particular context.



Activity 1: Textwindow & WriteLine

So you can get the hang of this, try writing out other words to the screen. Write more than one statement say three or more. Try to create imaginary conversations.

Try these questions

1. What happens if your add two words together like this WriteLine("Fish"+"Chips")
2. What happens if you put a number between the WriteLine brackets - WriteLine(23)?
3. What happens if you put add two numbers like this WriteLine(23+44)?

Simple calculations

1. Given that * means times, / means divide and - means subtract, can you use Small Basic as a simple calculator?
2. What is the answer to 22 divided by 7?

Further exercise

1. Try breaking it! When does TextWindow.WriteLine() stop working?
2. Try using the function Write() rather than WriteLine. How are they different?