Small Basic

Syntax

Syntax, in programming terms, means spelling and punctuation. Some things to remember are:

- Only one statement per line
- Most commands are properties or methods of TextWindow or GraphicsWindow. Use a dot (.) to find out what is available (a list will show up in a box and on the right hand side)
  - **Methods** may need arguments, for example:
    
    ```
    GraphicsWindow.DrawRectangle(10,20,50,50)
    ```
  - **Methods** may return a value, for example:
    
    ```
    n=TextWindow.ReadNumber()
    ```
  - **Properties** are assigned a value using the equal sign (=), for example:
    
    ```
    TextWindow.BackgroundColor = "red"
    GraphicsWindow.title="This is a graphics window"
    ```

Small Basic provides lots of help for syntax rules, for example you can use the mouse wheel or the up-down keyboard arrows to see all of the methods and properties that can be used with a particular object.

If you do get a syntax rule wrong, the program will not run until you correct the error. Look in the panel below your code to see the line number and a hint for fixing the problem. The message below indicates there is an error on line 17.

The error is that the statement has not been finished – this is a property and should be used to assign a value such as:

```
TextWindow.Title="Guessing Game"
```
Variables

What is a variable?
Variables are used to store data such as a number or a word – this is the ‘short term memory’ of the computer. In the same way, we might write a phone number on a sticky note, or place a bookmark in a book, so we can find it later.

Variables must have a unique name (no spaces) and are assigned a data type. Small Basic does this ‘behind the scenes’ (ie you do not have to specific the data type), but they should ALWAYS be assigned a value (either directly, or through an input statement) before they are used.

A variable can be assigned a value, from a constant, user input, or a calculation. For example:

```
payRate = 25.40  'payRate is assigned a decimal value
payRate = TextWindow.Read()  'user will enter the payRate at run-time
payRate = LastPay * 1.2  'the value can come from other variables or from a calculation
```

Operators
Operators modify values of variables.

Arithmetic
The simplest operators are those which perform basic maths.

**Addition** is done using the plus (+) sign and **subtraction** is done using the minus (-) sign. For example:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>7 + 2</td>
<td>9</td>
</tr>
<tr>
<td>Addition</td>
<td>3.4 + 8.1</td>
<td>11.5</td>
</tr>
<tr>
<td>Subtraction</td>
<td>6 - 4</td>
<td>2</td>
</tr>
<tr>
<td>Subtraction</td>
<td>11.1 – 7.6</td>
<td>3.5</td>
</tr>
</tbody>
</table>

**Multiplication** is done using the asterisk (*) and **division** is done using the slash (/). For example:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplication</td>
<td>8 * 4</td>
<td>32</td>
</tr>
<tr>
<td>Multiplication</td>
<td>2.3 * 12.2</td>
<td>28.06</td>
</tr>
<tr>
<td>Division</td>
<td>12 / 2</td>
<td>6</td>
</tr>
<tr>
<td>Division</td>
<td>45.26 / 6.2</td>
<td>7.3</td>
</tr>
</tbody>
</table>

The mathematical operators have the following **precedence** indicating the order they are evaluated without specific groupings:
1. Multiplication (*) and division (/)
2. Addition (+) and subtraction (-)

If multiplications and divisions or additions and subtractions are in the same expression, they are performed in left-to-right order. Use **parentheses** around expressions to force precedence.
Comparison and Logical Operators

There are six comparison operators in Small Basic used to compare the value of two expressions (the expressions must be of the same data type). These are the basis for making decisions:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>=</td>
<td>Equal to</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Not equal to</td>
</tr>
</tbody>
</table>

Logical operators operate on Boolean data types, providing a Boolean result. They are also used in decision making. We will use two logical operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>And</td>
<td>Logical And</td>
</tr>
<tr>
<td>Or</td>
<td>Logical Or</td>
</tr>
</tbody>
</table>

Concatenation Operator

To concatenate two string data types (tie them together), use the + symbol, the string concatenation operators:

```
CurrentTime = "The current time is " + TimeNow
SampleText = "Hook this " + "to this"
```

Other functions

There are also a number of Math functions – such as sin, cos, tan, square root, etc. These are methods of the Math object. Type Math to see a list, for example:

```
Math.
Max
Min
NaturalLog
Pi
Power
Remainder
Round
```

Exercises

1. Ask the user to enter a number, and store this number as a variable. Display the square and cube of the number.
2. Ask the user for two numbers and use the arithmetic operators to display the sum, product, quotient and difference of the two numbers (+, -, *, /)
3. Explore the methods and properties of the TextWindow and GraphicsWindow. Use the help and intellisense (the window above) to work out how to use the commands.