Let’s Play Fizz Buzz

Sample Programs

There are several sample Small Basic programs for you to refer to. They include enough information for you to figure out how to solve the main challenge. The programs are:
01Welcome.sb: shows how to display details on the screen.
02DataTypes.sb: reminds you about different data types.
03SimpleMaths.sb: shows ways to find a remainder which might be useful!
04BlastOff.sb: demonstrates both types of loop.
Remember, Intellisense will help you if you are unsure what an object does.

Fizz Buzz

You should know how to play Fizz Buzz, but how will we code it? The key is to **DECOMPOSE** the problem into smaller parts. We’ll start with a game that goes up to fifty.

Challenge 1: Decomposing the problem

In the box below outline a series of smaller steps to build a working program. The first step is given to get you started.

Design a program to count up to 50, displaying each number on a new line on screen.

Challenge 2: Implementing a solution in stages

Now write and test a program that implements each step you outlined above. Do not move on to the next step until the program works.

Challenge 3: Making the game more flexible

You should now have a working program that outputs the correct FizzBuzz sequence up to 50. Amend your program so the user can input the number the sequence should end at.

Challenge 4: Controlling the display

Once you have a working program that can play FizzBuzz up to large numbers it is difficult to know if the earier values are correct. Alter your program to include a pause so you could see the early numbers, rather than have them scroll straight past when it runs.

Challenge 5: Fizz, Buzz and Boing

Alter your program so that it includes ‘Boing’ if the number divides by 7. So 21, which divides by 3 and 7 would be replaced with ‘Fizz Boing’. 35 would output ‘Buzz Boing’ and 105, which divides by 3, 5 and 7 so would output ‘Fizz Buzz Boing’.