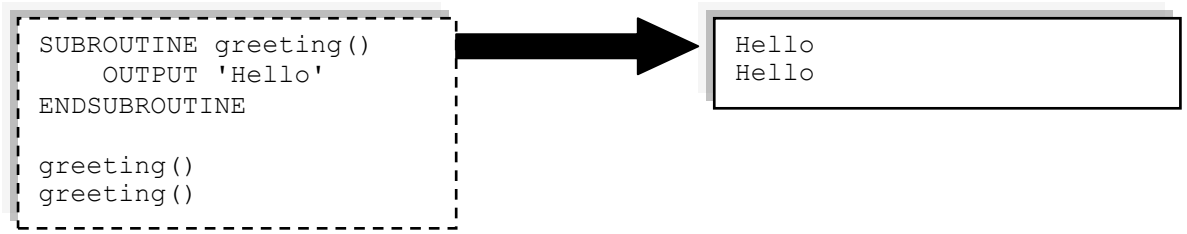
We often want to carry out many instructions again and again in programs. Whilst loops are good at repeating the same instructions over and over, sometimes we need to execute a set of instructions at various points in a program. We therefore use **subroutines**, the simplest of which is a **procedure**.

A **procedure** is a way of giving a sequence of instructions **one identifier** which can then be **called** from anywhere in the program. We can also give **inputs** into the **procedure** when it is called – these are known as **parameters**.

To make a procedure which outputs 'hello' on the screen we would use the following:

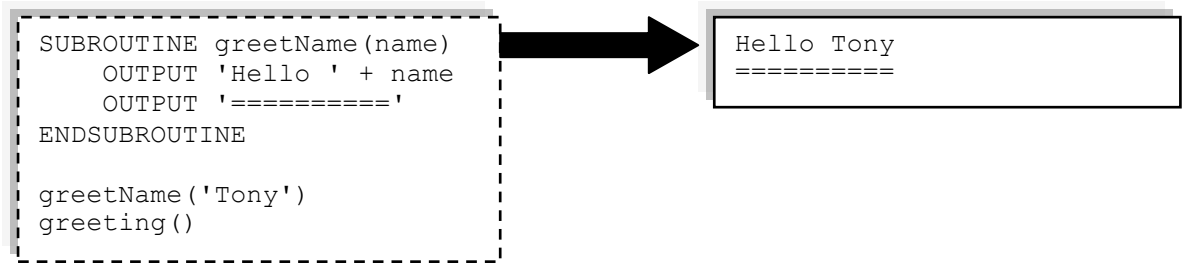


The above code will first create a procedure named *greeting*. There is nothing between the brackets so there are no inputs for this procedure. The procedure contains one line of code which outputs the text "Hello" to the screen. The procedure is then **called** twice. This results in the text "Hello" being output on the screen twice.

The general syntax for writing a procedure is:

```
SUBROUTINE procedureName(parameter1, parameter2, parameter3...)
       //instructions for the procedure go here
ENDSUBROUTINE
```

The following procedure will take someone's name as an argument and then print it on the screen.



Local and global variables

When using a subroutine such as a procedure, you can use **variables** inside the procedure itself. If a variable is created inside the procedure then it will only be available for use in that procedure. It is known as a **local variable** and will be **destroyed** once the procedure ends.

If a variable is created outside of any other subroutines or structures then it is a **global variable**. These can be used anywhere in the program, but in general we try to create local variables wherever possible.