Syntax and logic errors | Part A

Even when the most experienced programmers write programs, they always make errors. This might mean that an error message is displayed on the screen, or it may result in the program doing something unexpected. Making errors is a normal part of programming, but good programmers learn to recognise where errors have occurred in their program so that they can fix them. This process is called debugging.
One type of error that you are likely to encounter quite quickly as a programmer is called a syntax error. The syntax of a program is the set of rules which must be followed to create a valid program.

For example you can only use certain commands such as “print”. If you spell any of these words wrong, this will result in a syntax error.

prnit(“Hello”)

You can’t invent your own commands either, so this will also cause a syntax error.”

display(“Hello”)
Just like in English and all of the other human languages, the order of the words in a programming language makes a difference. You wouldn’t say:

cheese me give please some

… because that doesn’t make any sense – the words are in the wrong order! The computer expects to see commands or values in the order that makes sense, for example when we are defining a variable we can’t put

age 5 =

…because the computer expects to see

<name of variable> <equals sign> <value of variable>

.. in exactly that order!
Probably one of the most common syntax errors you will make is forgetting to close quotes or brackets in your program. For example:

print(“this is a long sentence)

Syntax errors are showstoppers because if you have made even one of these errors, the program cannot run at all. The computer will try to be as helpful as possible and tell you which line of the program the error was found on.

If you are writing a long program and you receive this kind of error, the problem is often on the previous line to the one reported as having an error. This is because the program hasn’t realised you meant to close a quote or bracket and thinks you’re still writing the previous command!

Sometimes we might write a program which has no syntax errors, but it still doesn’t work properly. This may be due to a logical error and these errors are generally more difficult to spot! A logical error occurs when the programmer writes code which will run, but doesn’t do what he or she intended it to do.