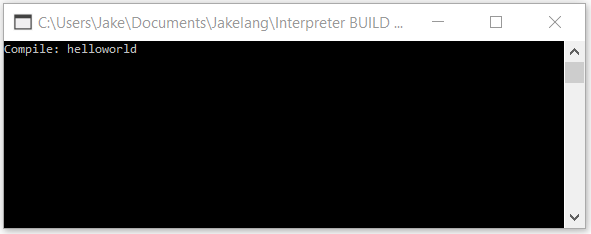
**Lesson 1 – Running a Program**

Jakelang is an interpreted language. To run a Jakelang program, it must be compiled on the interpreted and run on the command line.

All Jakelang files are named “<name>.jakelang”. It is important that the name of a Jakelang file does not have any spaces so that the interpreter can find the program.



When the interpreter is run, you should enter the name of the Jakelang file to compile and press enter. This is case sensitive. You may enter the file name with or without the .jakelang extension.

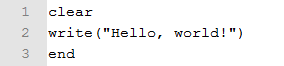
After pressing enter, the file will be compiled and the program will be executed immediately.

**Lesson 2 – Hello World**

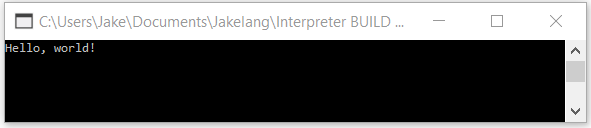
Jakelang files can be written in any text editor such as Notepad++. To start, open a file and name is “helloworld.jakelang”.

Every program should begin with a *clear* statement. This clears the console from clutter and allows the user to only see the program on the command line.

On the final line of the program, an *end* statement should be added. In simple programs this will not be noticeable however in a more complex program may crash if an end statement is not added.

Now, we want to make the console write “Hello, world!”. To write a line to the console, we will use a *write(“”)* statement where the text we want to print goes between the speech marks.

Save the file and run it on the interpreter. If everything is correct, it should print “Hello, world!” to the console.



Congratulations on writing your first Jakelang program!

**Lesson 3 – A personalised greeting using custom string input**

In Jakelang, there are two main types of input: integer and string. An integer input can be used in mathematical functions such as *add(,)*. A string input, however, can hold letters which can be returned to the user on the console.

In this lesson, we will build a program which asks for your name and greets the user based on the name they input.

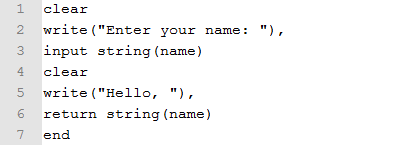
To start, we will add our *clear* and *end* statements.

Next, we want to ask the user their name. To tell the user what to input, we will write text to the console before taking an input. Since we do not want to start a new line at the end of the *write* statement, we will use *write(“”),*.

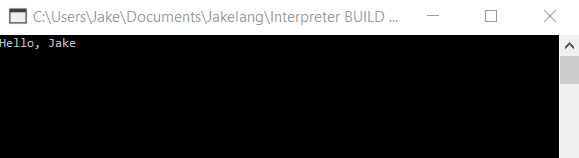
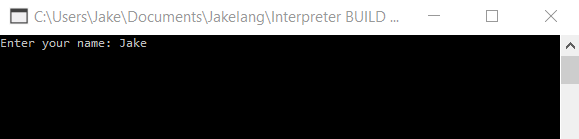
Next we want to take a string input from the user in the form of their name. To do this we will use the *input type(var)* statement. The type, as we have decided, will be “string” and the variable name can be set to “name”.

Now we want to greet the user using their name however we want to clear the console first so they do not see the console asking for the name. We can use another *clear* statement to do this. Next we will write to the console “Hello, “ using a *write* statement with a comma to indicate the same line.

To return the value of the string to the user, we will use a *return type(var)* statement. This will return the value of the specified variable. Our variable name is “name” and our variable type is “string”.



Now we want to run our file on the Jakelang interpreter after saving it as a .jakelang file.



**Lesson 4 – Mathematical Functions**

Jakelang has four common built in mathematical functions.