1. Why should we use functions at all?

* We use functions to organize our programs so that it matches the way we think about the programs. Functions also make fixing bugs and programming easier and faster, as we only need one line of code to call functions every time we need it, instead of a lot of lines of codes just for the use of a function.

1. How to define/declare a function?

* We use this syntax for functions definition/declaration:

def function\_name(arguments):

the\_codes\_inside\_the\_function

1. How to call/use a function?

* We call out the name of the function, followed by the value of its arguments.

Ex: function\_name(values)

1. What is return, why and how do we use it?

* Return is the thing that lots of functions do: Returning a value that can be used later.
* We use return when we need some of the value from functions, values that we will use later in our program.
* We use the return statement, followed by the expression (values) that will be returned.

1. Do we have to use return in **every** function?

* No. The functions that don’t return anything are called void functions, which means they exist.

1. What are function arguments/parameters, why and how we use it?

* They are special variables used in functions, as the actual input expression given to functions.
* We use them when we need our functions to run with different values that we can change later.
* We define them along with the function, and we call their values along with the call of the function that we use.

1. How to use function from a different file other than our currently working file?

* We import them from their home file (that is in the same folder with our current file), then use them normally.

Ex:1. from fire\_style import fireball\_technique

fireball\_technique()

2. import inner\_gates

inner\_gates.gate\_open(8)