* *Why should we use functions at all?*

In Python, a function is a named sequence of statements that belong together. Their primary purpose is to help us organize programs into chunks that match how we think about the problem.

* *How to define/declare a function?*

The syntax for afunctiondeﬁnitionis: def NAME( PARAMETERS ): STATEMENTS

* *How to call/use a function?*

There can be any number of statements inside the function, but they have to be indented from the def.

Function deﬁnitions are the second of several compoundstatements we will see, all of which have the same pattern:

1. A header line which begins with a keyword and ends with a colon.

2. Abodyconsisting ofone or more Pythonstatements, each indented the sameamount — the Python style guide recommends 4 spaces — from the header line. We’ve already seen the for loop which follows this pattern.

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

The return statement is followed an expression (a in this case). This expression will be evaluated and returned to the caller as the “fruit” of calling this function.

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

All the functions in the previous section return values.

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

Parameters are also local, and act like local variables. So it is not possible for a function to set some local variable to a value, complete its execution, and then when it is called again next time, recover the local variable. Each call of the function creates new local variables, and their lifetimes expire when the function returns to the caller.

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

Additionally, a only exists while the function is being executed — we call this its lifetime. When the execution of the function terminates, the local variables are destroyed.