**Functions**

Key word DEF define a piece of code that can be stored and reused many times you want.

Two types of functions in Python

* Built-in functions are provided as part of Python and it libraries, like print(), input and int() that we already used.
* Functions that we define ourselves

Taking some arguments as input, functions use that data to compute something and then return a result.

To call the function you use the name, parentheses and arguments in an expression.

Example: getName(id)

**Built-in**

Functions max() and min()

These functions gets the max and min number of a sequence including a scale of how early or late a latter is in the alphabet.

Conversions Functions

Functions built in that gave you the type of a variable - type(), converts numbers to floating point – float(), convert to Integers numbers – int(). Finally, str() converts its argument to a string. To get the size of a string or the number of elements in a list, the length len() function is used, and it returns a integer.

Math Module

A module contains many built-in functions, importing math module you unlock several functions like radians, sin, cos, tan, log and math.pi.

Random numbers

In programing, there are sometimes that you want things different. Let’s the universe get in our code. Deterministic functions as random return a aleatoric number. The random.random()from 0.0 to 1.0, random.randint(5,10) between and integer, random.choice(t) chose randomly a item inside the variable T.

**Building our Own Functions**

Defining

Using the def keyword, naming it with a rememberable link and indenting the code inside it. This code will be storage and remembered wen you call the function by its name.

It’s possible you define a function and never use it, becoming a useless code, but normally you will use the function at least one or the main objective of this structure, more than one time.

Arguments

An Argument is a value, passed as its input when we call the function.

Everything that goes inside a function call is an argument, linked with the variable inside the “argument part” in the function’s definition, this act is called “passed as parameter”. That variable is a parameter.

Return Values

Getting data back out of the function, that the main possibility of returns a such important part of this structure.

A “fruitful” function is one that produces a result (receive a variable, compute its data into something and return the value encountered)

A Void Function has no return, makes whatever you set but don’t return anything. Keep attention to these ones, it could interfere only in parameters inside the function and printing out the values but could not interfere in variables outside the function appearing that nothing happened.

When do we use functions or not?

A clean code is a code organized in “paragraphs” like a text, commented as a football game making it easy to understand even to someone who don’t code.

If you did something big and complex, you could break this into small problems and redesign it, creating functions to be more understandable, clean and even better.

At some point in the future, make a library of common stuff that you do over and over and share it!