

ASSIGNMENT #5 : The Turtle artist (with for loops and functions)

**This assignment may be team work (of up to 2 people each) – SUBMITTED VIA CANVAS
YOU NEED TO JOIN A PREEXISTING CANVAS GROUP TO BE ABLE TO SUBMIT AS A GROUP
See further submission details below**

DEADLINE : Monday March 9, 11:59 PM



Write a program which does a drawing with turtle built-in functions so that the program:

- Includes some color, fills in some area with some color, changes the pen color in some parts
- the turtle does at least three right or left turns
- the pen changes from being up to down at least a couple times
- there are at least three for-loops
- you create at least three void (or non-fruitful) functions (i.e. not returning any value), where the functions do drawings and/or pen movements, but do not return any values
- You create at least two functions which have some parameters (they may be productive or void functions, they may be functions that count for the previous requirement)
- The defined functions are called (used and passed arguments to the parameters as needed)
- The program generates some random values and uses them (such as random colors)

Organize your program as follows: **place the functions definitions first and after including all the definitions place the top or main level.** Call the functions from the top/main level or also, functions may call other functions (as long as the called functions are placed earlier in the python file). The drawings you obtain are completely up to you. Enjoy the creative process!

Submit Assignment

WHAT TO SUBMIT

- If you work as a team** (of up to 2 people) **you need to join a pre-existing Canvas group associated to this assignment (Group gturtle).** Go to People → Groups / Groups are called “gturtle– n” (where n is your group number). **Both team members should join the same group. DO NOT CREATE THE GROUPS, THE GROUPS EXIST AND YOU NEED TO JOIN THEM.** The two partners need to join the same group.
- ONE PARTNER ONLY SUBMITS: the turtle program (the .py file) via Canvas**
Assignments → Assignment #3 Turtle artist → Click the blue button at the top right.

3. **Everyone (regardless if you work on your own or with a partner)** has to **ALSO** submit a separate text file where you include the names and student numbers of both team-mates and you **describe how you distributed your work**. Also inform the time that it took you to do this assignment, include the time of both teammates in case of working in pairs. **If you worked on your own** you should name this file as **individually.txt**, if you worked in a team name the file **team.txt**.
4. **Additionally, if you will want to show your drawing in class (optional)**, submit your file in the Assignment "Showing your solutions in class".

EXAMPLES of built-in Turtle functions and the definition of our own functions

1. Check labs exercises and examples from labs weeks #5 and week #6 (A link to the example files from those labs are also provided with this assignment description)
2. Check Canvas Home → More LINKS for Turtle Built-in functions and **tables for color coding**

CLARIFICATIONS – READ. about importing modules and about defining your functions:

1. In your programs, place the *import* statements of any modules that you may need (such as *import turtle* or *import turtle as t*), **outside the functions** you define, that is, place them at the "top level" of the program.

By doing so, the module name (*turtle* or *t* depending on how you imported) is *global*, and thus you will be able to use the module (*turtle* or *t*) everywhere (i.e. *t.forward(...)*) : BOTH at the top level or main program and also inside any functions you define (with no need to include the module name as parameter to the functions).
2. You may use turtle objects (as described in the interactive text) or you may use the functions directly using the turtle module (as shown in the examples provided with the lab exercises).
3. Using the "import xxx as" format vs. "import xxx" format is up to you, the first format allows you to abbreviate typing when using the dot notation.
4. Your own functions need to be defined (placed in the .py file, with **def**) **before they are used** (i.e. before they are called or invoked and passed arguments) . They may be all called from the top level, and only if a function needs to use another function, you may call them from another function.
5. YOU ARE ENCOURAGED TO NOT CALL a function from another function when the first function (e.g. *funA*) ends, and you want to continue execution in the other (*funB*) function. RATHER, to accomplish that two functions are executed one after the other, the top level (or whichever level needs to execute the two functions) should call the two functions, in this example, first call *funA* and then call *funB*.
6. Name your functions and variables reasonably (to suggest their purpose).

End of Assignment #5 description, Turtle artist