**Modules**

* A file containing Python definitions and statements intended for use in other Python programs
* Modular design is a design approach that subdivides a system into smaller parts called modules, that can be independently created and then used in different system.
* Functional partitioning into discrete scalable, reusable modules.
* Turtle:
  + import turtle *# allows us to use the turtle library*
  + wn = turtle.Screen() *# creates a graphics window*
  + alex = turtle.Turtle() *# create a turtle named alex, data object*
  + alex.forward(150) *# tell alex to move forward by 150 units*
  + alex.left(90) *# turn by 90 degrees*
  + alex.forward(75) *# complete the second side of a rectangle*
  + wn.exitonclick()
* Random Generator: pseudo-random generators
  + random.randrange(1,101) *# gives 1-100*
* import
  + from math import \*

**Functions**

* A named sequence of statements that belong together, to help organise programs into chunks. Consists of header, body, parameters.
* Docstrings: triple-quoted string that is the first thing after the function header to provide information about the function
  + <function\_name>.\_\_doc\_\_
* Fruitful: functions that return values. E.g. max, abs, range, int
* Procedure: functions that execute a sequence of step only, no return
* Composition: call a function from within a function
* Python functions return None unless specified otherwise
* Unit testing:
  + test.testEqual( ‘item1’, ‘item2’) # test module
* Name variables carefully
  + If input is Boolean, can name it as is\_leap\_year
  + Write a comment about parameters of function below main definition

**Recursive Functions**

* Functions that call themselves
* Base case: ‘simplest case’ that does not require recursion
* General case: cases that requires recursion to bring the computation ‘closer’ to the base case