Functions

- Functions are similar to subroutines in that they allow you to create your own procedures.
  (in most other languages, functions + subroutines are the same thing).
- Functions also:
  1) Are used differently from subroutines
  2) are constructed slightly differently
  3) have some "extra baggage" when using them

- Motivation
  - Functions are more flexible in certain cases.
  - The output of a function can be sent directly as the input to another function or dummy argument to a subroutine.
    (can be used in line).

Differences

Subroutine

- can change none, some or all of the dummy arguements
- outputs can be any type and be scalars or arrays. Can have multiple outputs.
- invoked by the CALL statement

Function

- can't change any of the dummy arguement lie, the all have INTENT(IN
- can only have a single scalar output
- invoked by an assignment (=)

Subroutine	function
- Does not have a type	- must have a type (integer, double precision)
- Doesn't have to be declared	- must be declared by any routine that references it directly
	These are responsible for the "extra baggage" in using a function
- any procedure can always be programmed as a subroutine	(*) - only processes that return a single scalar can be programmed as a function

- If your procedure satisfies (*) and does not change any dummy arguements, you can elect to program it as a function instead of a subroutine.

- See factfun and binfun programs on website for more information on using / constructing functions).