Concurrent Programming in Erlang

In this exercise we'll look at a few ways in which the frequency server can be modified.

As usual, please do share your thoughts on the exercise, and your solutions, using the comments on this step.

There is a supporting file, `frequency.erl`, available (as a zip file) under 'Downloads' below.

Naming the server

As introduced in the previous step and in the module `frequency.erl`, the frequency server is spawned and then accessed through its `Pid`. Modify this so that it is spawned and then named, so that messages to it can be sent to the named process. The convention in this situation is to give the server the same name as the module in which it is defined: `frequency`.

Test your re-definition from the shell.

Improving the functionality of the server

The current implementation is rather permissive:

- it allows a client to hold more than one frequency; and,
- it allows a client to deallocate a frequency that it is not currently using.

Modify the definitions of `allocate` and `deallocate` to prevent these two things happening.

In doing this you may need to think about the reply that you send to the client in the case that an allocation or deallocation has not been possible.
