

Now for a server. Putting the pieces together.



I mean I've tried to fit in I've really tried. I even tried learning what Functional Programming was.

```

1 defmodule NameServer1 do
2   @moduledoc """
3   Very simple name server supporting two methods:
4   add: Add a name and a place.
5   find: Given a name, return the place or nil
6   """

```

```

8   def start(name) do
9     Process.register(spawn(NameServer1, :loop, [name, %{}]), name)
10    :ok
11  end

```

Constructor.

Spawn a new process, running loop() and register its PID with a name.

```

13  def rpc(name, request) do
14    send(name, {self(), request})
15
16    receive do
17      {_name, response} -> response
18    end
19  end

```

Method interface.

Send a message to the server, requesting it to perform an action.

```

21  def loop(name, state) do
22    receive do
23      {from, request} ->
24        {response, newState} = handleRequest(request, state)
25        send(from, {name, response})
26        loop(name, newState)
27    end
28  end

```

The server.

Preserves the server data (State), and calls the methods and returns the result.

```

30  defp handleRequest({:add, name, place}, state) do
31    newState = Map.put(state, name, place)
32    {:ok, newState}
33  end
34
35  defp handleRequest({:find, name}, state) do
36    {state[name], state}
37  end

```

Methods.

The functions that implement the server actions.

```

38 end

```