

# CONCURRENT PHILOSOPHERS

Programming II - Spring 2018



# Philosophical Ideas

**Hypatia**

**Elisabeth**

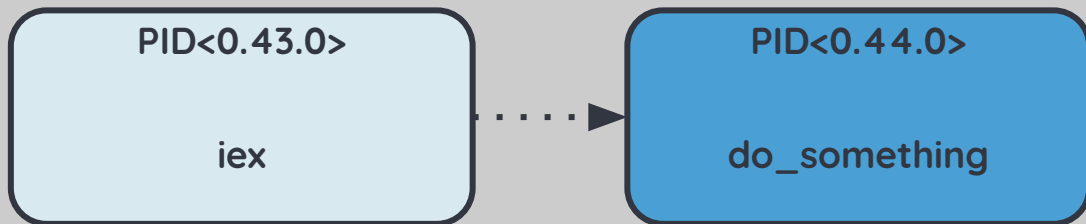
**Simone**

**Ayn**

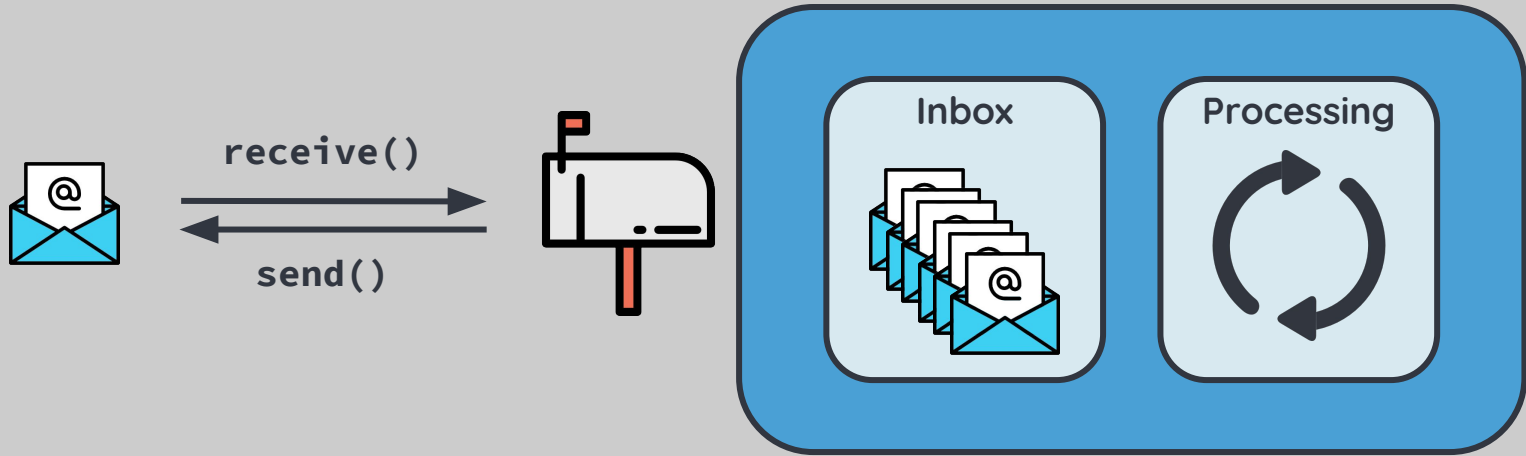
**Arendt**

# A Process

```
pid = spawn(fn -> do_something end)
```



# A Bunch of Messages



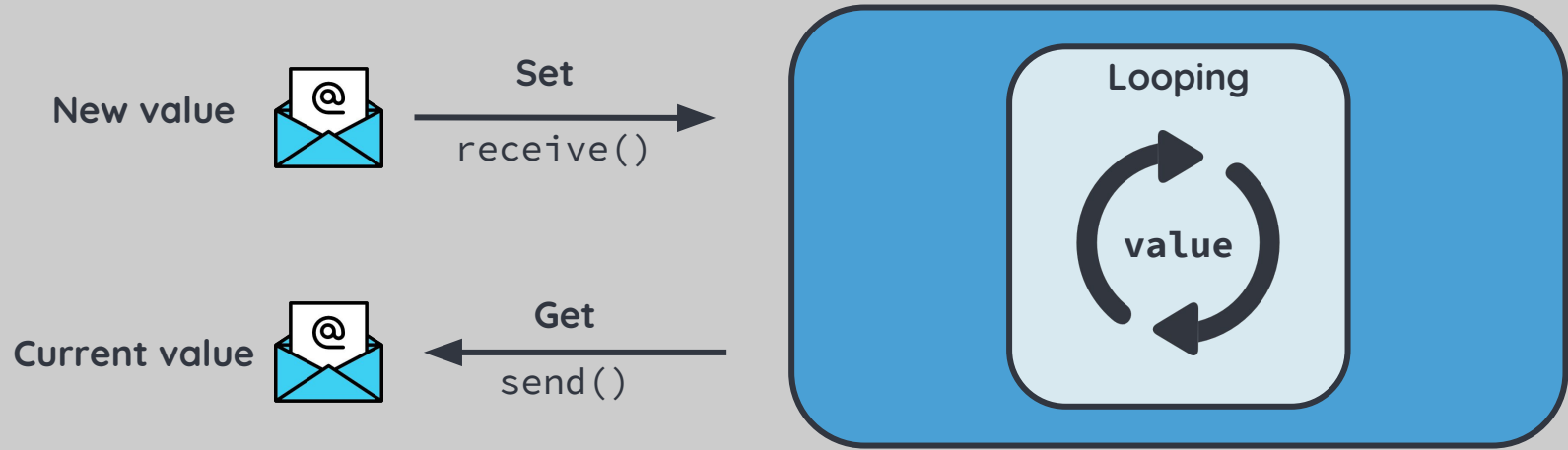
# All together

```
iex> parent = self()  
#PID<0.41.0>
```

```
iex> spawn fn -> send(parent, {:hello, self()}) end  
#PID<0.48.0>
```

```
iex> receive do  
...>   {:hello, pid} -> "Got hello from #{inspect pid}"  
...> end  
"Got hello from #PID<0.48.0>"
```

# Keeping the State

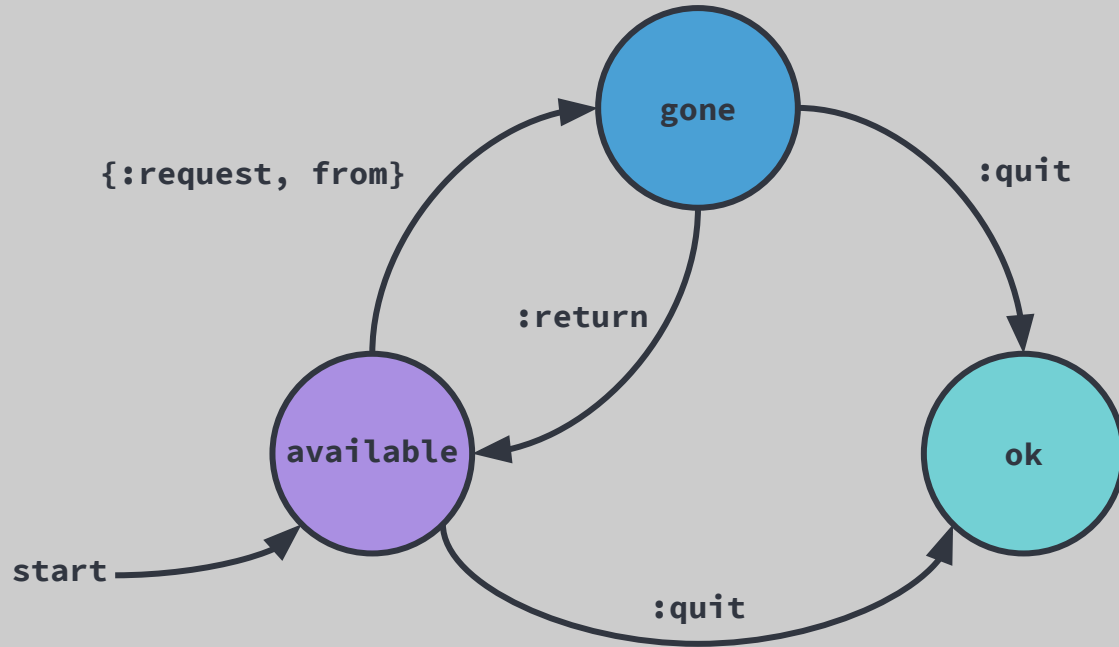


# Memory Cell

```
def start do
  spawn_link(fn -> loop(%{}) end)
end

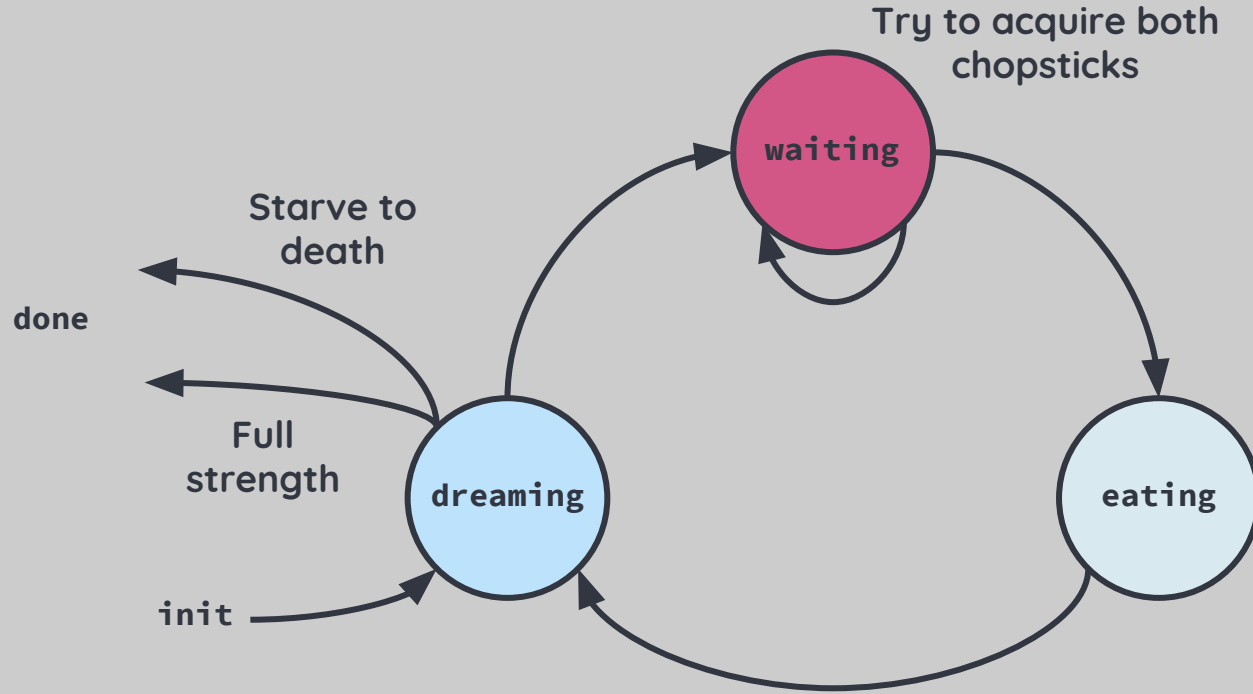
defp loop(map) do
  receive do
    {:get, key, caller} ->
      send(caller, Map.get(map, key))
      loop(map)
    {:put, key, value} ->
      loop(Map.put(map, key, value))
  end
end
```

# A Chopstick

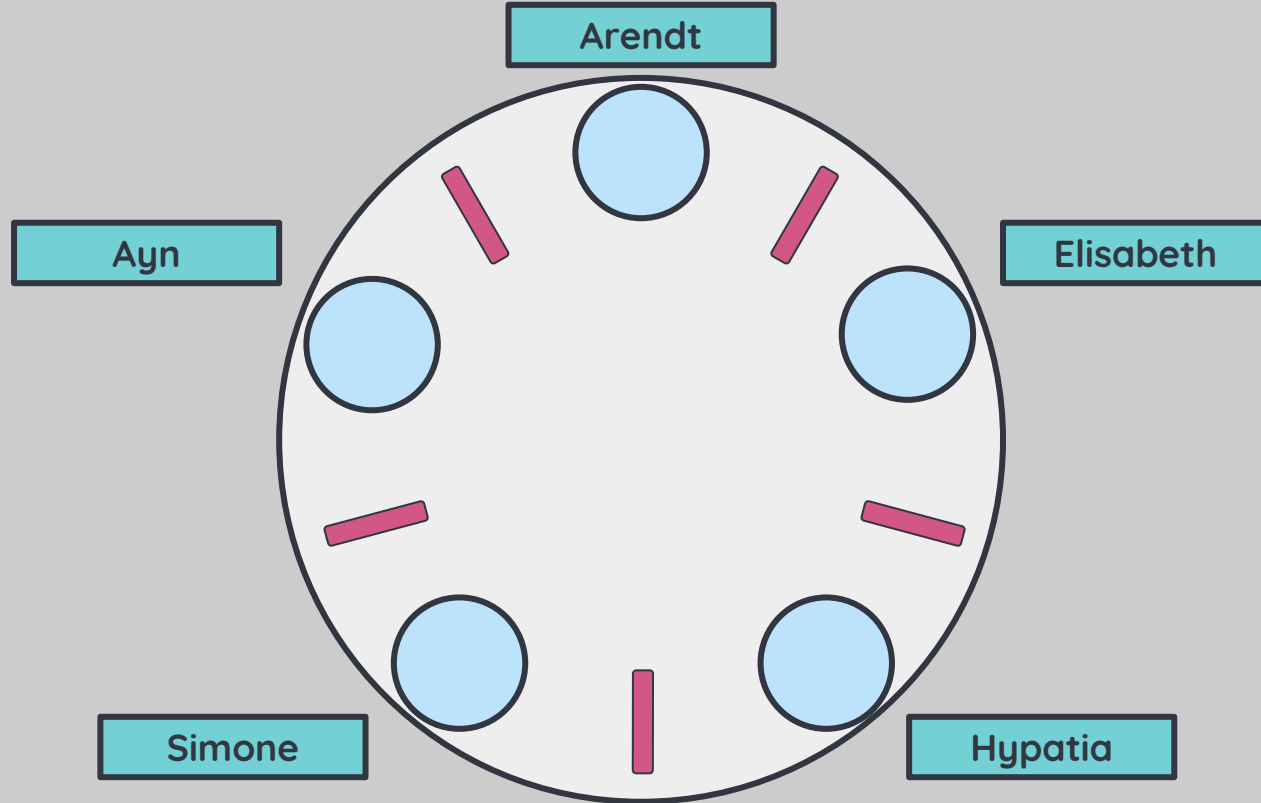




# A Philosopher

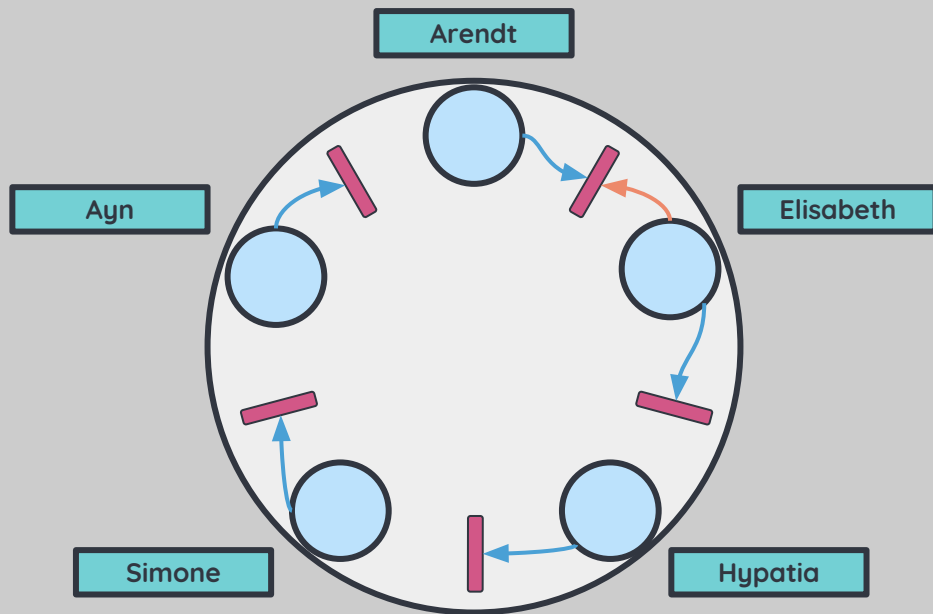


# 5 Philosophers at a Table



# Left and Then Right

First grab the left chopstick, then the right one.  
*What happens if everyone does it at the same time?*



*Good luck!*