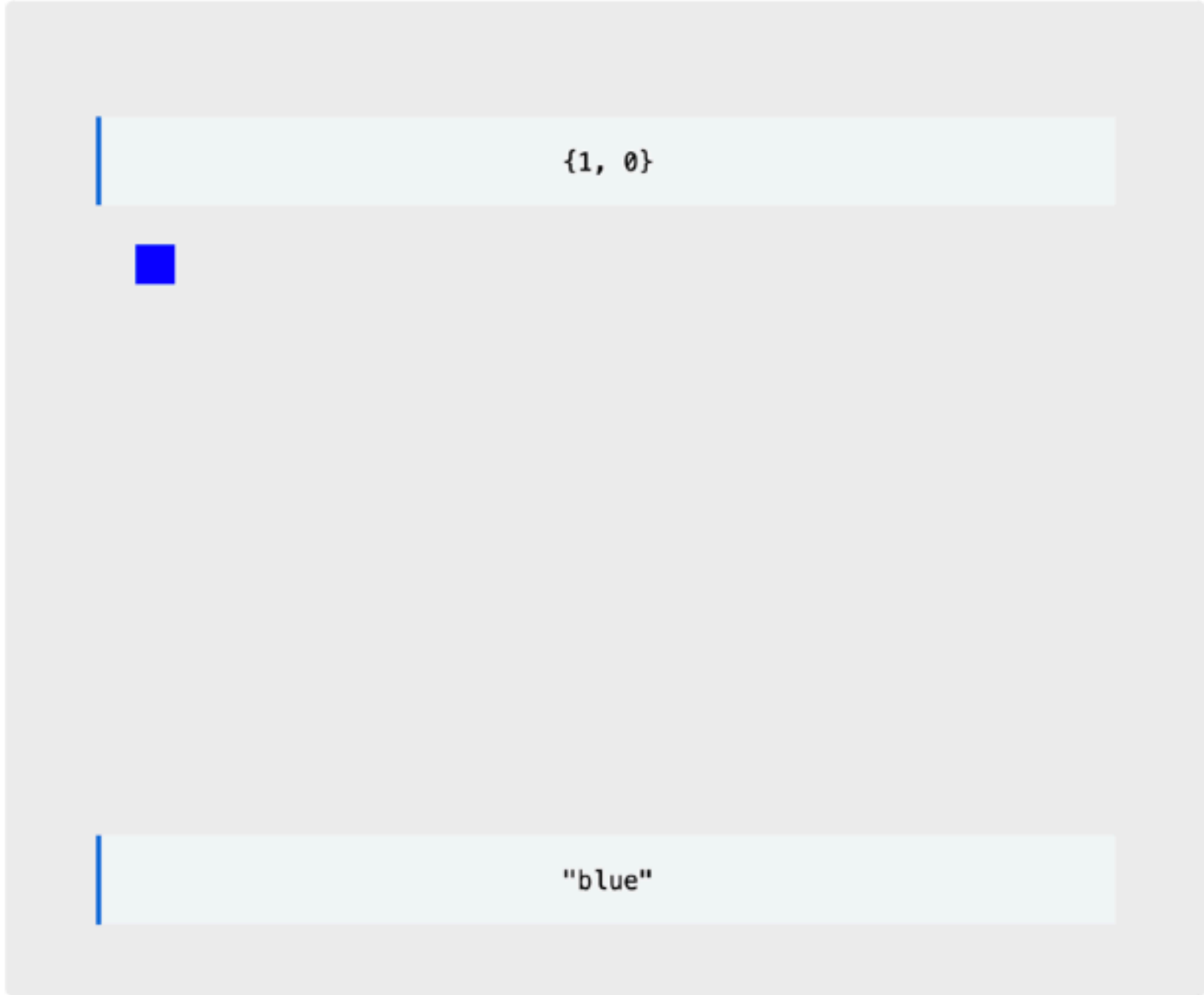




Handle

```
def handle_event("keydown", %{"key" => "ArrowRight"}, socket) do
  {:noreply, update(socket, :x, &(&1 + 1))}
end
```



{1, 0}

"blue"

```
def handle_info(:tick, socket) do
  {:noreply, assign(socket, color: Enum.random(@colors))}
end

def handle_event("keydown", %{"key" => "ArrowRight"}, socket) do
  {:noreply, update(socket, :x, &(&1 + 1))}
end

def handle_event("keydown", %{"key" => "ArrowLeft"}, socket) do
  {:noreply, update(socket, :x, &(&1 - 1))}
end

def handle_event("keydown", %{"key" => "ArrowUp"}, socket) do
  {:noreply, update(socket, :y, &(&1 - 1))}
end

def handle_event("keydown", %{"key" => "ArrowDown"}, socket) do
  {:noreply, update(socket, :y, &(&1 + 1))}
end

def handle_event("keydown", _key, socket) do
  {:noreply, socket}
end
```