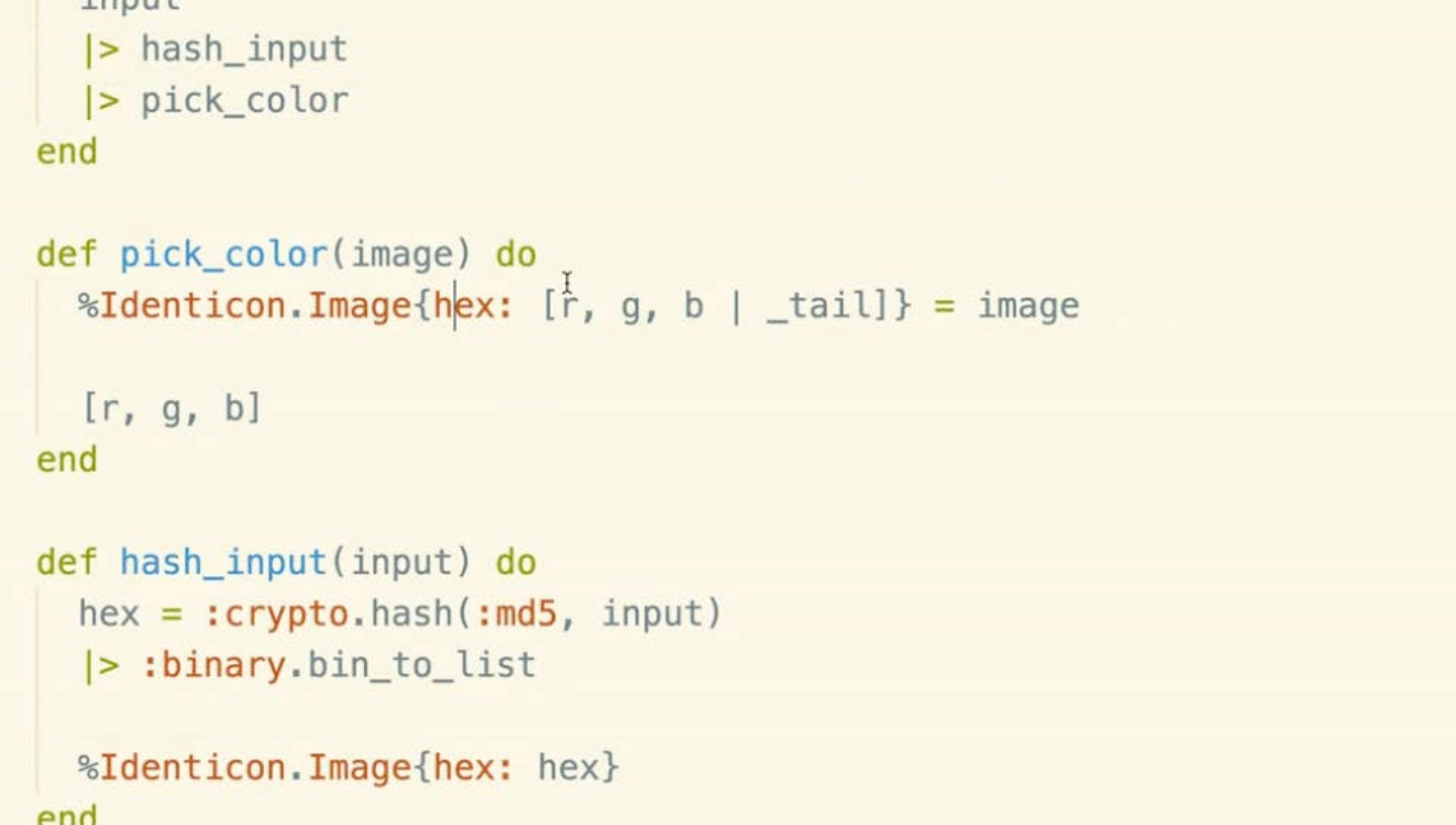


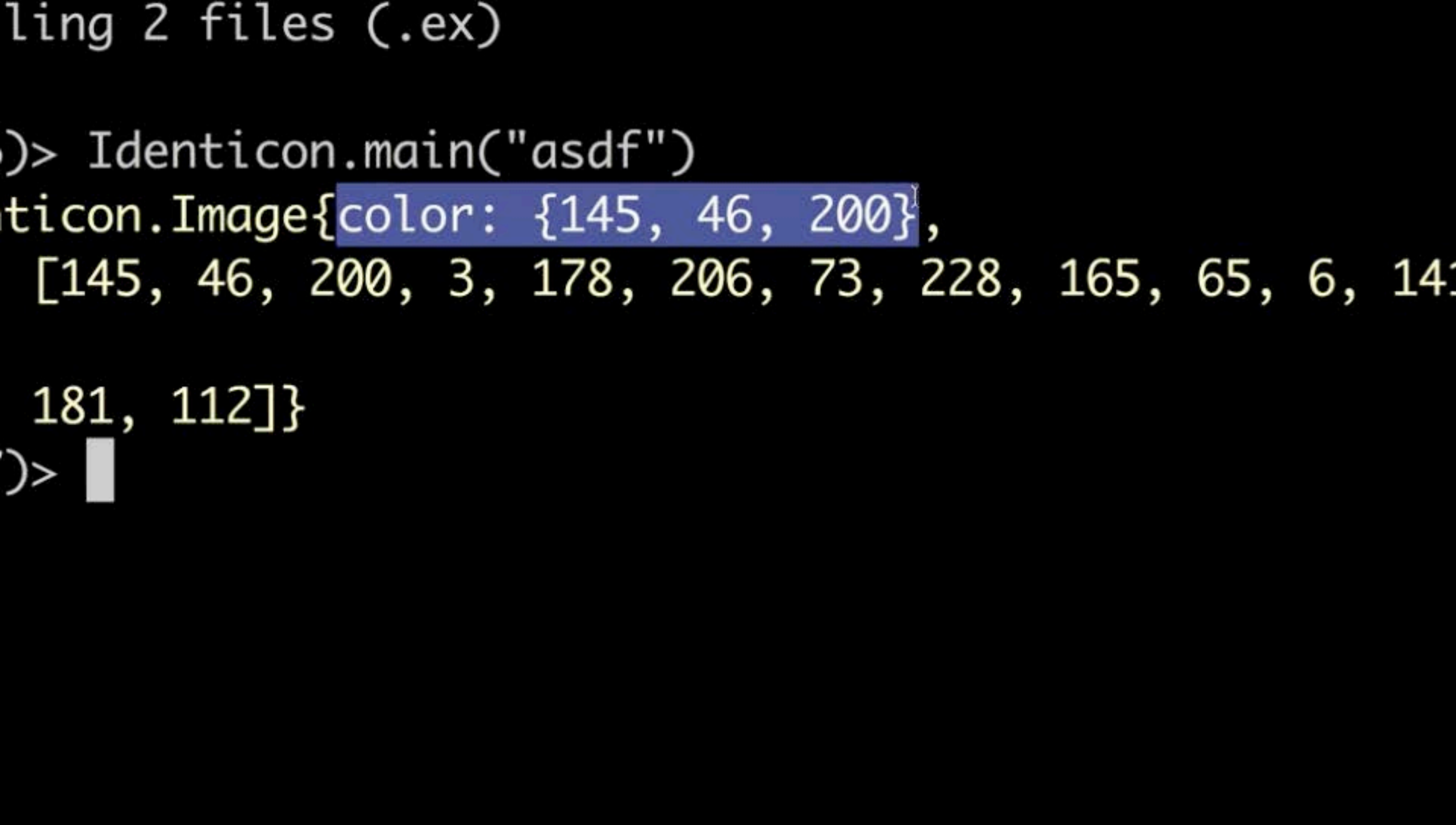
```
    input
    |> hash_input
    |> pick_color
  end

  def pick_color(image) do
    %Identicon.Image{hex: [r, g, b | _tail]} = image

    [r, g, b]
  end

  def hash_input(input) do
    hex = :crypto.hash(:md5, input)
    |> :binary.bin_to_list

    %Identicon.Image{hex: hex}
  end
```

```
 defstruct hex: nil
nd
```

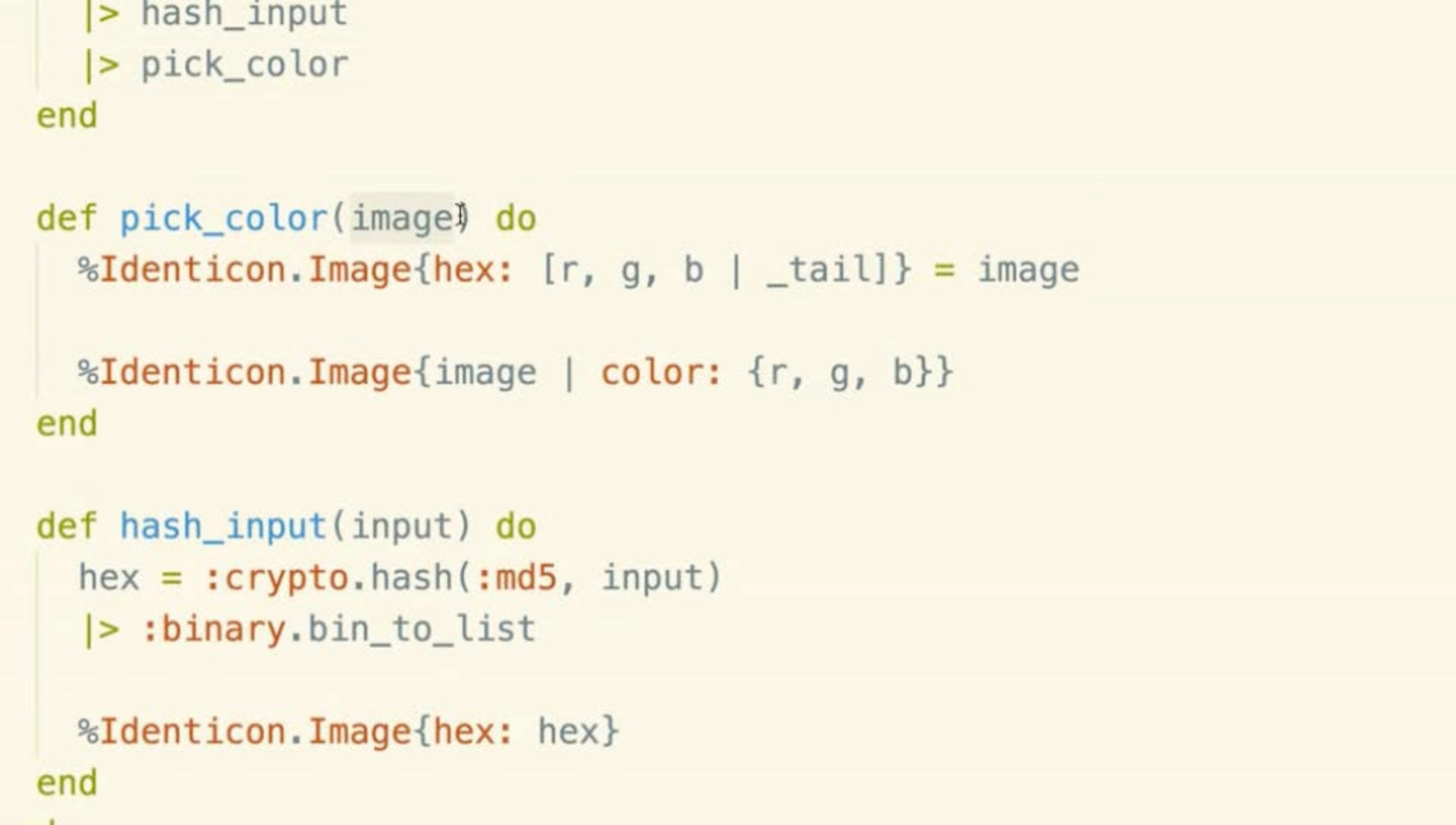
```
 defstruct hex: nil, color: nil
nd
```

```
    |> hash_input
    |> pick_color
  end

  def pick_color(image) do
    %Identicon.Image{hex: [r, g, b | _tail]} = image


  end

  def hash_input(input) do
    hex = :crypto.hash(:md5, input)
    |> :binary.bin_to_list

    %Identicon.Image{hex: hex}
  end
```

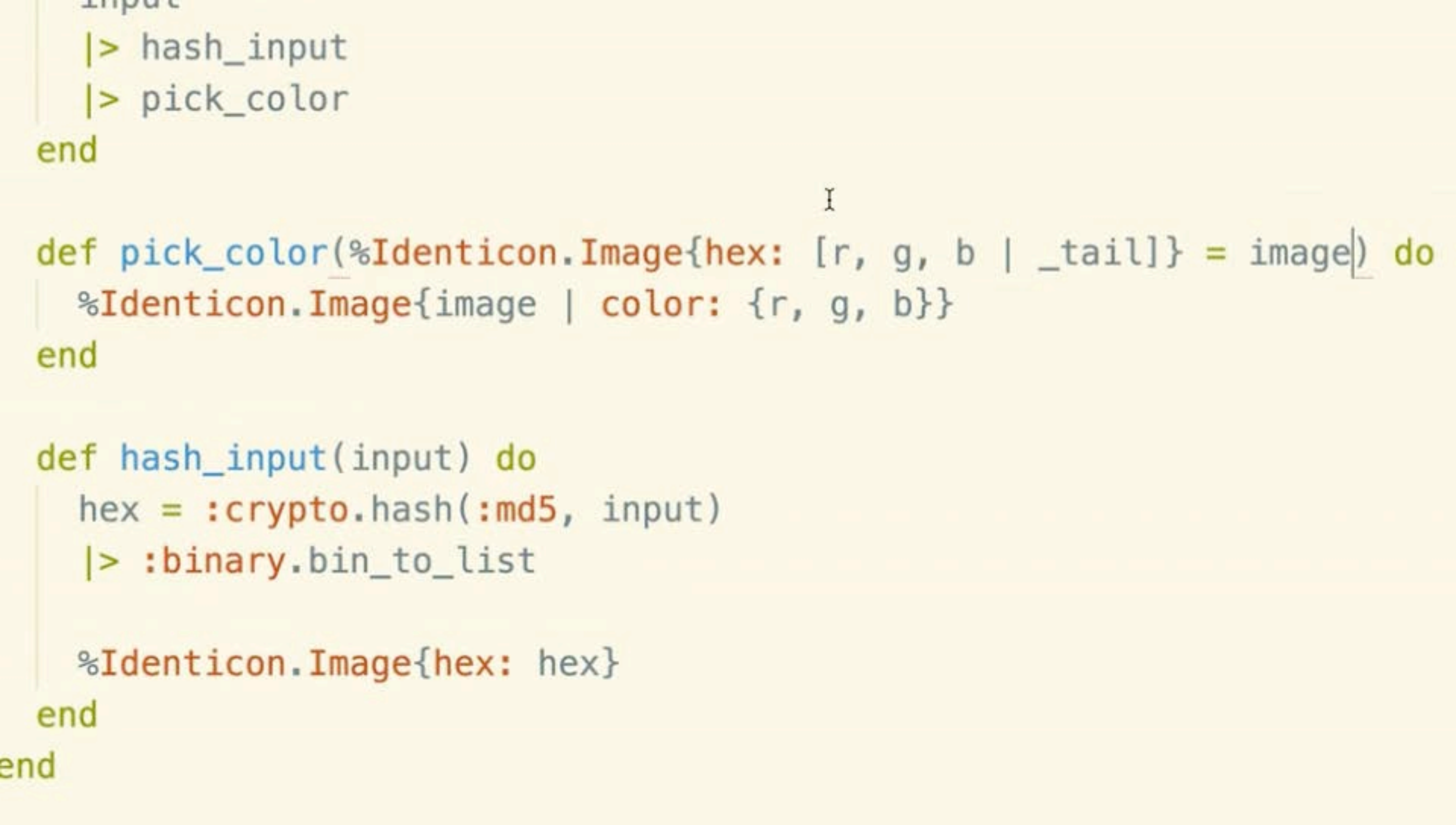
```elixir
    |> hash_input
    |> pick_color
  end

  def pick_color(image) do
    %Identicon.Image{hex: [r, g, b | _tail]} = image

    %Identicon.Image{image | color: {r, g, b}}
  end

  def hash_input(input) do
    hex = :crypto.hash(:md5, input)
    |> :binary.bin_to_list

    %Identicon.Image{hex: hex}
  end
end
```



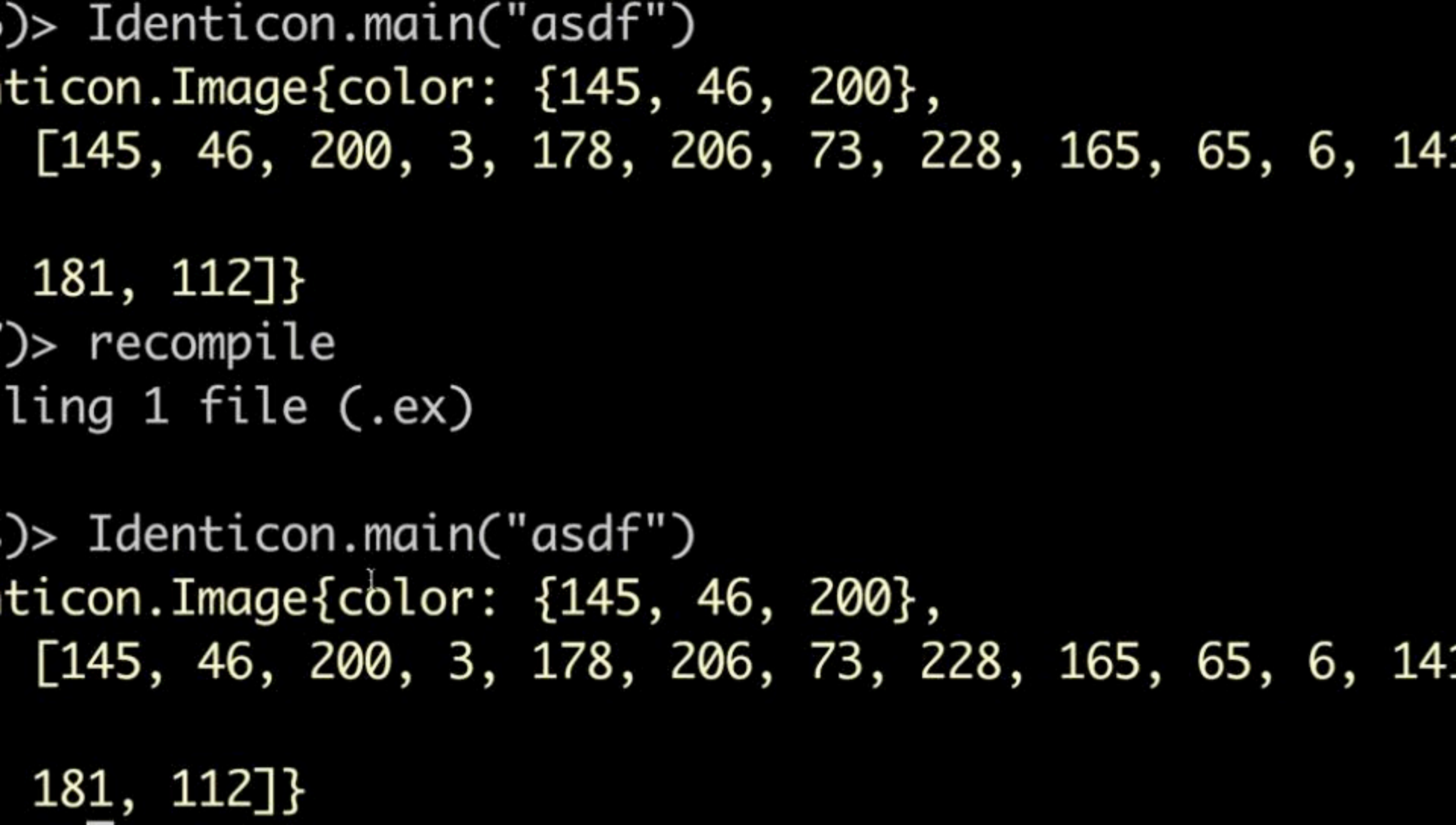
```
ling 2 files (.ex)

)> Identicon.main("asdf")
ticon.Image{color: {145, 46, 200},
 [145, 46, 200, 3, 178, 206, 73, 228, 165, 65, 6, 14


 181, 112]}
)>
```



```elixir
  |> hash_input
  |> pick_color
end

def pick_color(image) do
  %Identicon.Image{hex: [r, g, b | _tail]} = image

  %Identicon.Image{image | color: {r, g, b}}
end

def hash_input(input) do
  hex = :crypto.hash(:md5, input)
  |> :binary.bin_to_list

  %Identicon.Image{hex: hex}
end
```



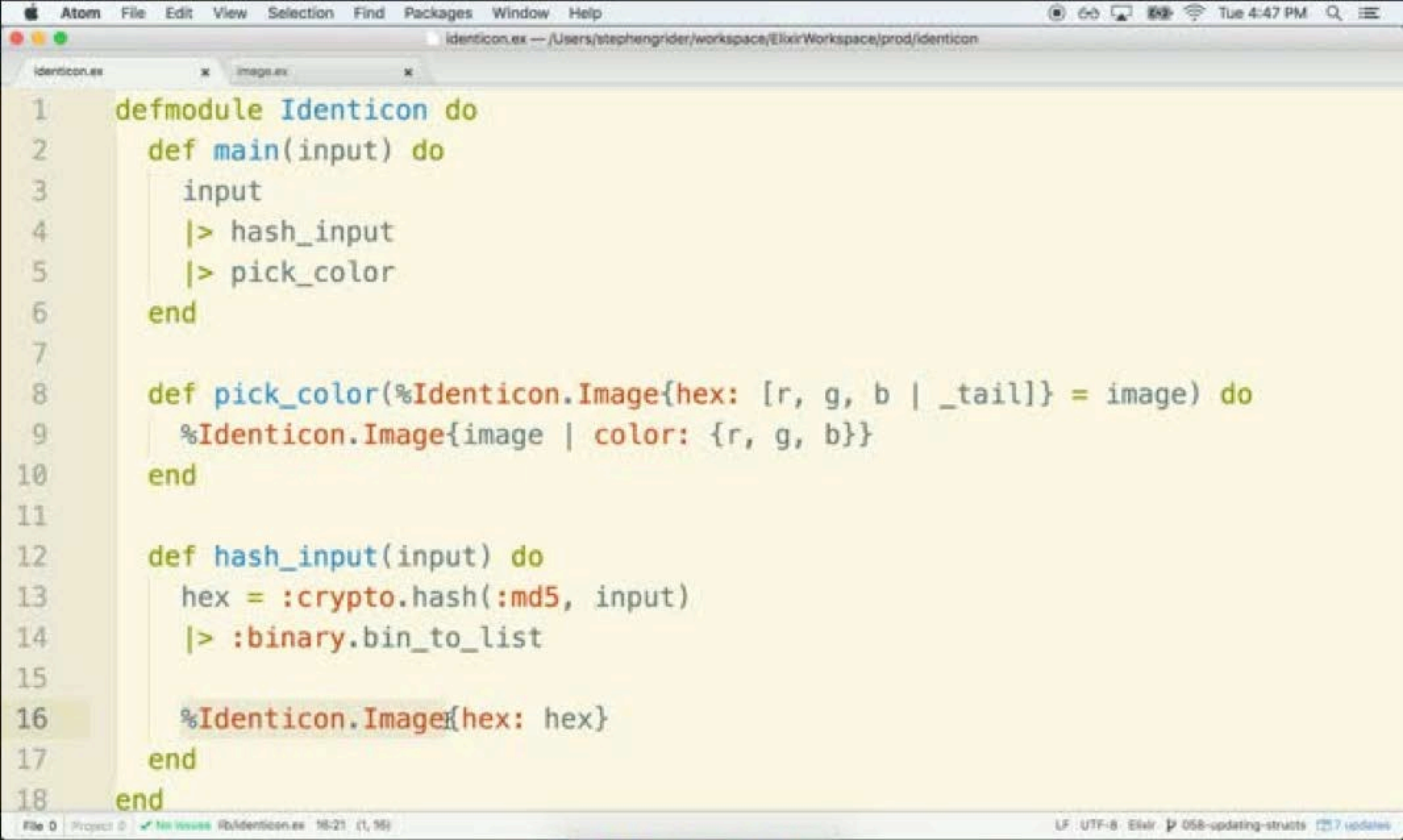
```elixir
    input
    |> hash_input
    |> pick_color
  end

  def pick_color(%Identicon.Image{hex: [r, g, b | _tail]} = image) do
    %Identicon.Image{image | color: {r, g, b}}
  end

  def hash_input(input) do
    hex = :crypto.hash(:md5, input)
    |> :binary.bin_to_list

    %Identicon.Image{hex: hex}
  end
end
```



```
)> Identicon.main("asdf")
ticon.Image{color: {145, 46, 200},
 [145, 46, 200, 3, 178, 206, 73, 228, 165, 65, 6, 14


 181, 112]}
)> recompile
ling 1 file (.ex)


)> Identicon.main("asdf")
ticon.Image{color: {145, 46, 200},
 [145, 46, 200, 3, 178, 206, 73, 228, 165, 65, 6, 14


 181, 112]}
```

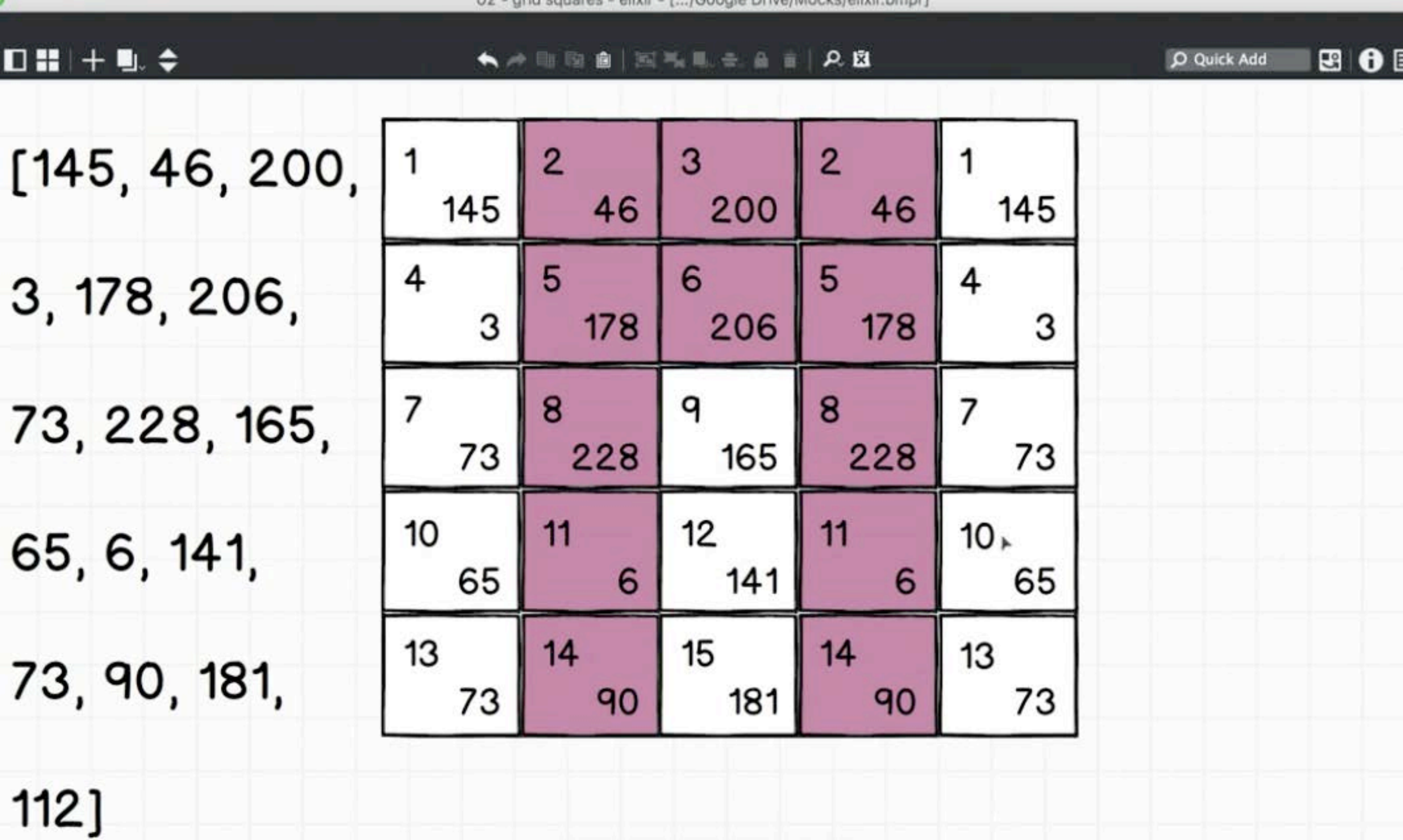
```
end

def pick_color(%Identicon.Image{hex: [r, g, b | _tail]} = image) do
  %Identicon.Image{image | color: {r, g, b}}
end

pick_color: function(image) {
  image.color = {
    r: image.hex[0],
    g: image.hex[1],
    b: image.hex[2]
  };

  return image
}
```

```
  end

  def pick_color(%Identicon.Image{hex: [r, g, b | _tail]} = image) do
    %Identicon.Image{image | color: {r, g, b}}
  end

  def hash_input(input) do
    hex = :crypto.hash(:md5, input)
    |> :binary.bin_to_list

    %Identicon.Image{hex: hex}
  end
end
```


Atom File Edit View Selection Find Packages Window Help
Tue 4:47 PM
identicon.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/identicon
identicon.ex
image.ex

```elixir
defmodule Identicon do
  def main(input) do
    input
    |> hash_input
    |> pick_color
  end

  def pick_color(%Identicon.Image{hex: [r, g, b | _tail]} = image) do
    %Identicon.Image{image | color: {r, g, b}}
  end

  def hash_input(input) do
    hex = :crypto.hash(:md5, input)
    |> :binary.bin_to_list

    %Identicon.Image{hex: hex}
  end
end
```

File 0 Project 0 No Issues lib/identicon.ex 16:21 (1, 16)
LF UTF-8 Elixir 058-updating-structs 7 updates



[145, 46, 200,

3, 178, 206,

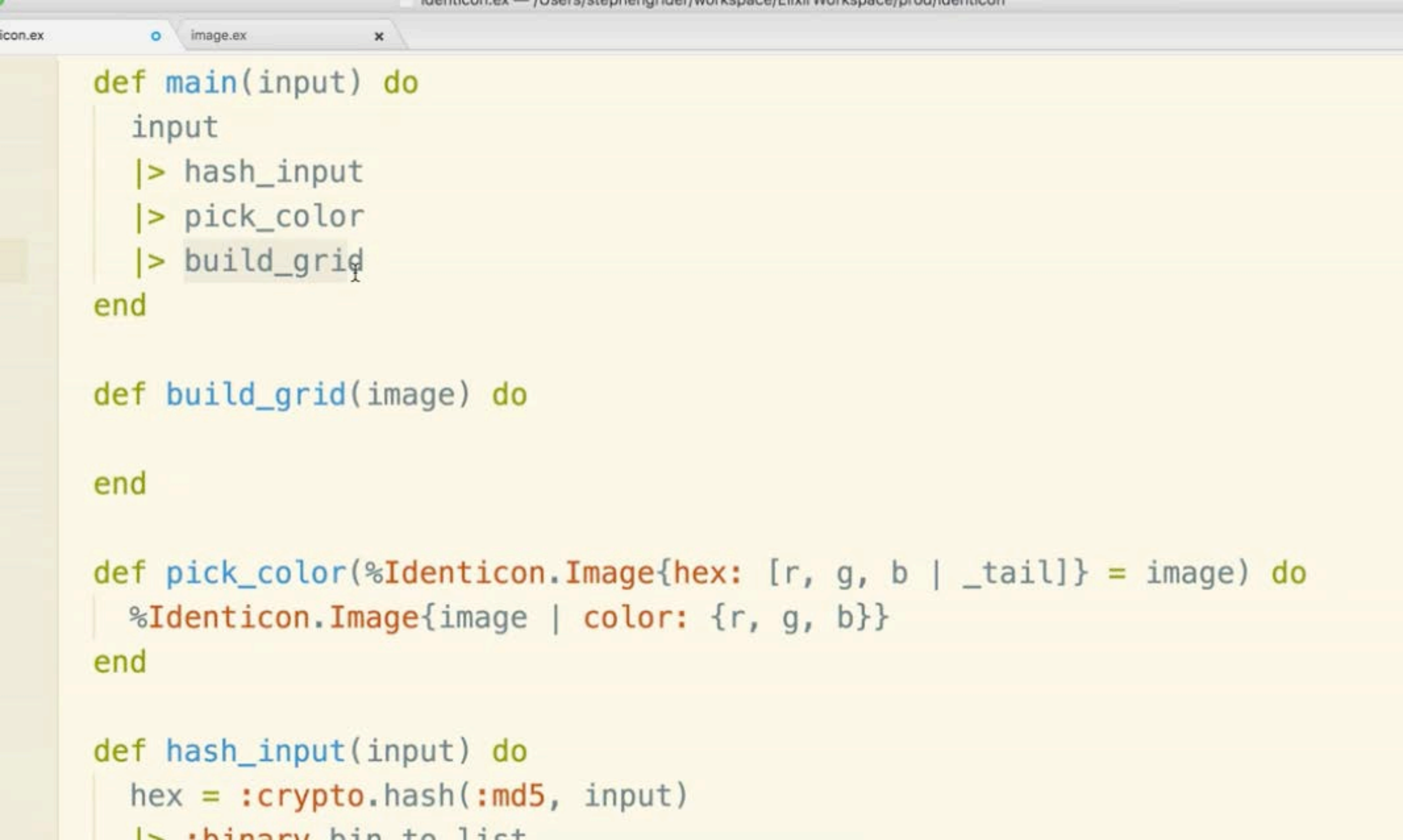
73, 228, 165,

65, 6, 141,

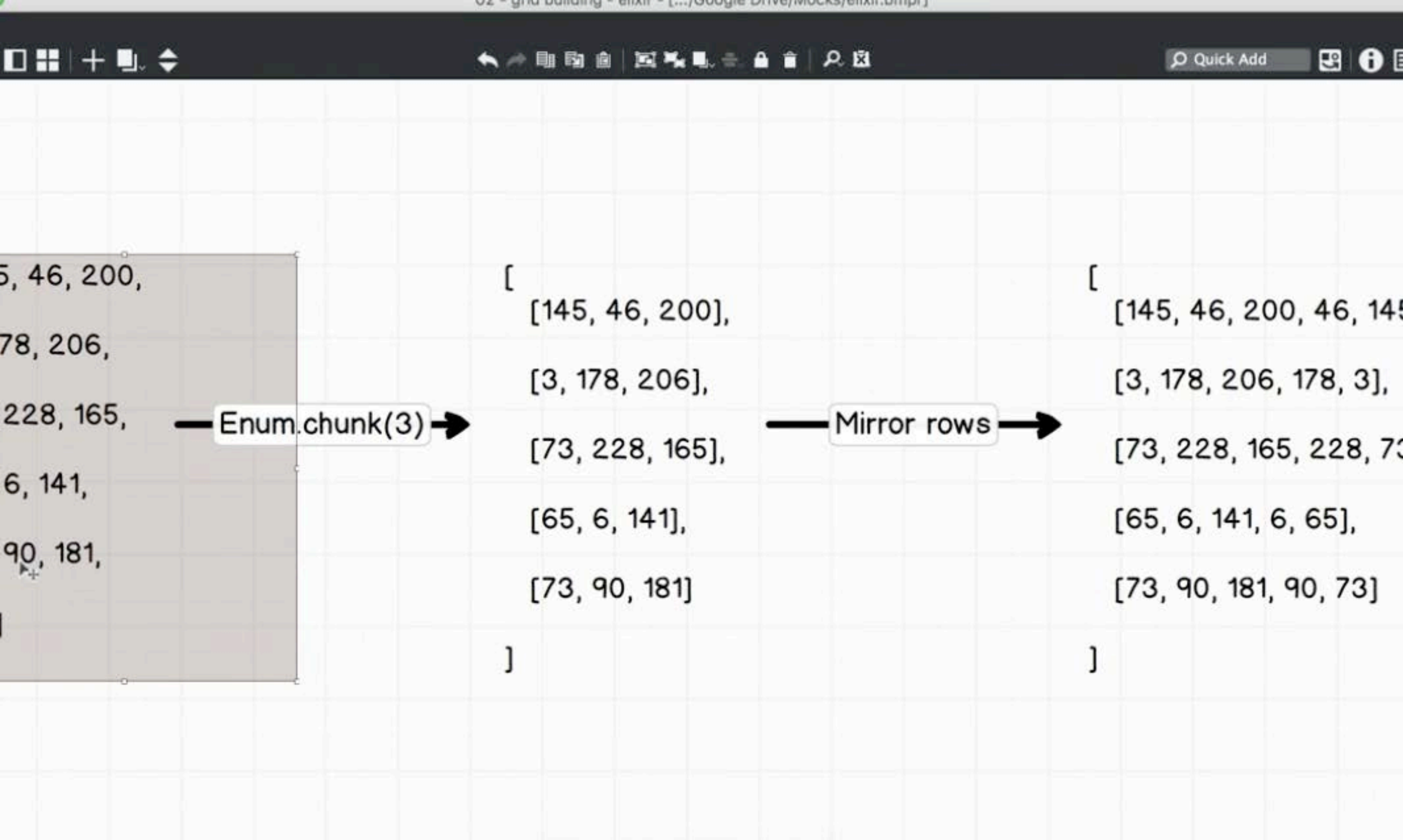
73, 90, 181,

112]

1 145	2 46	3 200	2 46	1 145
4 3	5 178	6 206	5 178	4 3
7 73	8 228	9 165	8 228	7 73
10 65	11 6	12 141	11 6	10 65
13 73	14 90	15 181	14 90	13 73



```elixir
def main(input) do
  input
  |> hash_input
  |> pick_color
  |> build_grid
end

def build_grid(image) do


end

def pick_color(%Identicon.Image{hex: [r, g, b | _tail]} = image) do
  %Identicon.Image{image | color: {r, g, b}}
end

def hash_input(input) do
  hex = :crypto.hash(:md5, input)
  |> :binary.bin_to_list
```



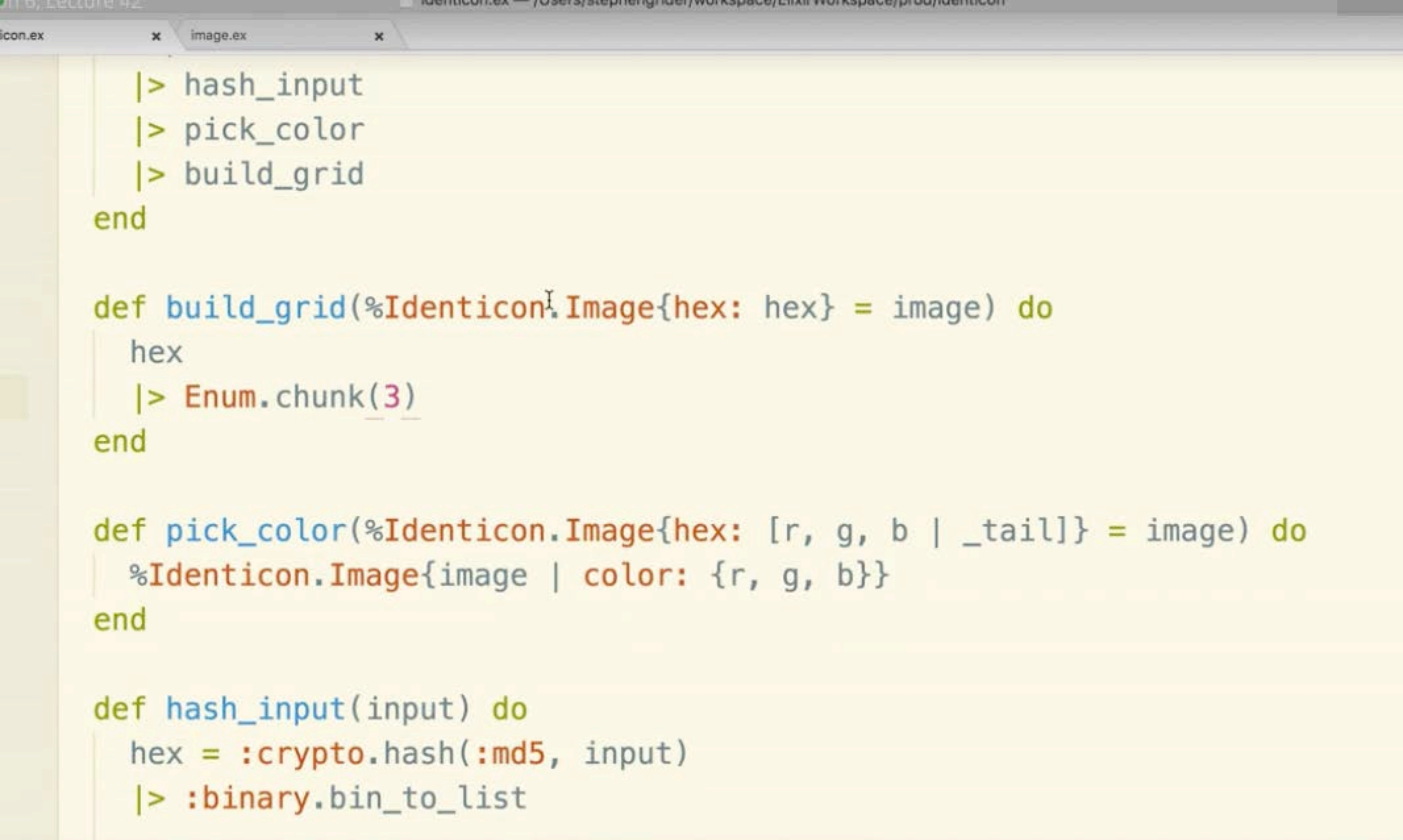
```
5, 46, 200,
78, 206,
 228, 165,
 6, 141,
 90, 181,
```

Enum.chunk(3)

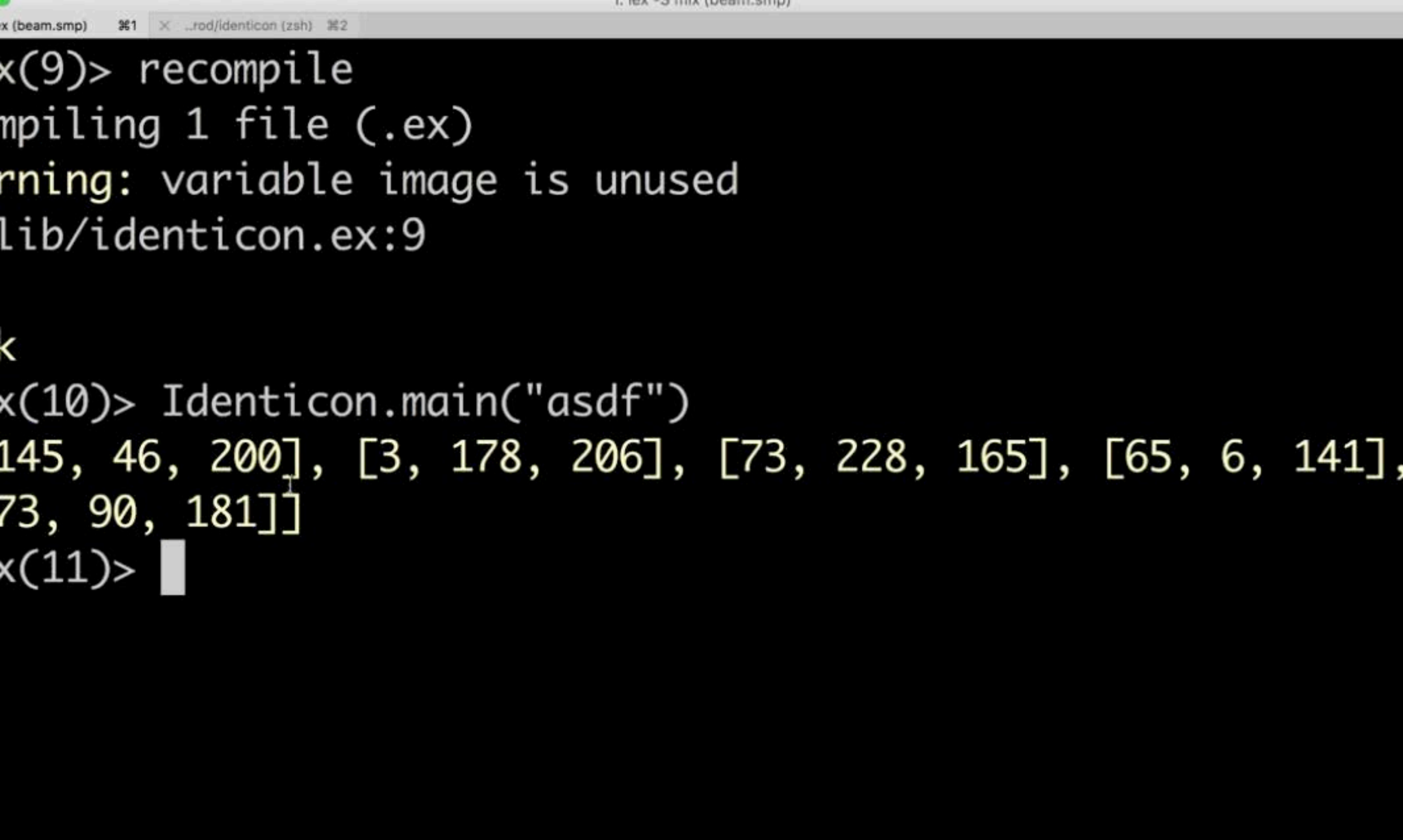
```
[
  [145, 46, 200],
  [3, 178, 206],
  [73, 228, 165],
  [65, 6, 141],
  [73, 90, 181]
]
```

Mirror rows

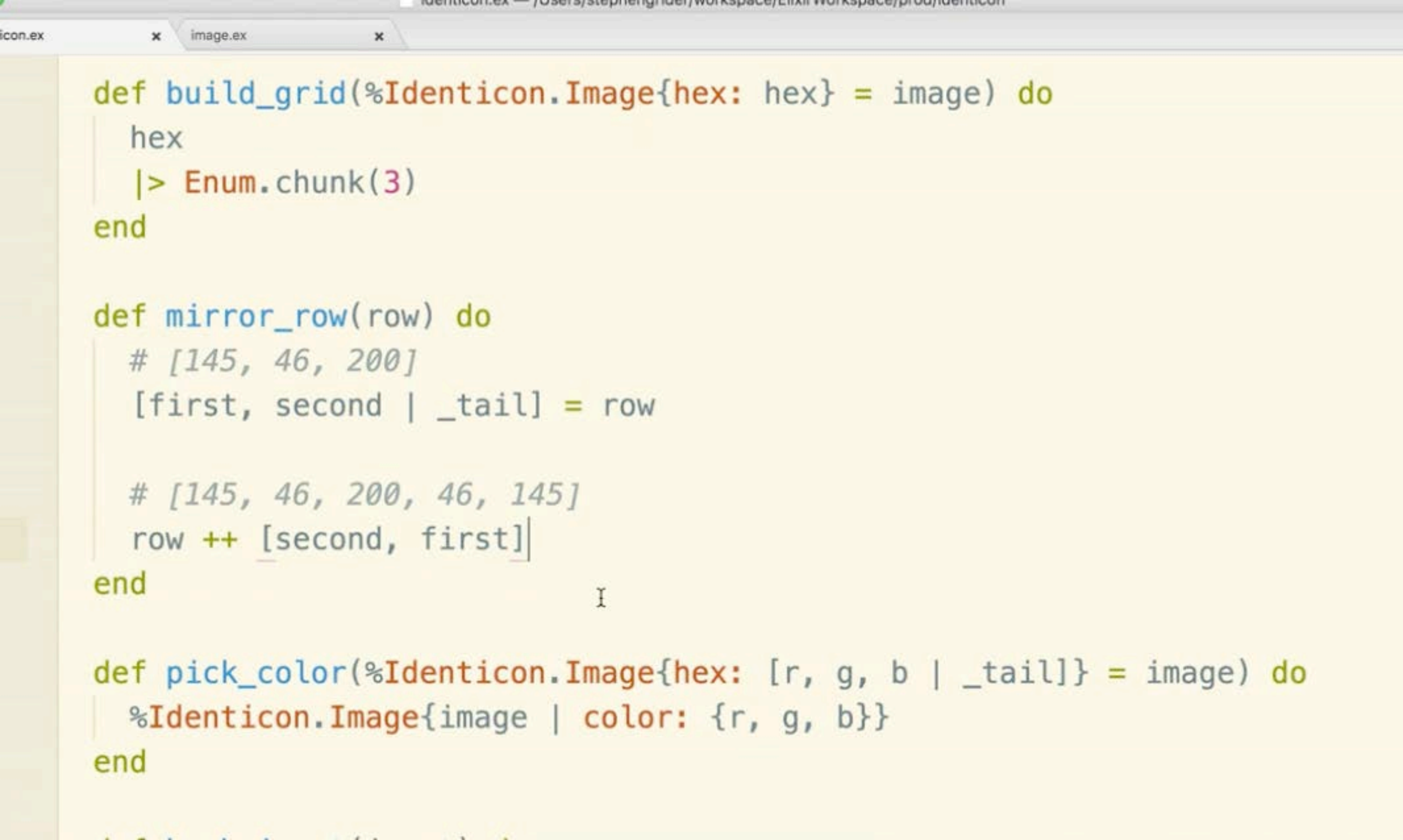
```
[
  [145, 46, 200, 46, 14
  [3, 178, 206, 178, 3],
  [73, 228, 165, 228, 7
  [65, 6, 141, 6, 65],
  [73, 90, 181, 90, 73]
]
```



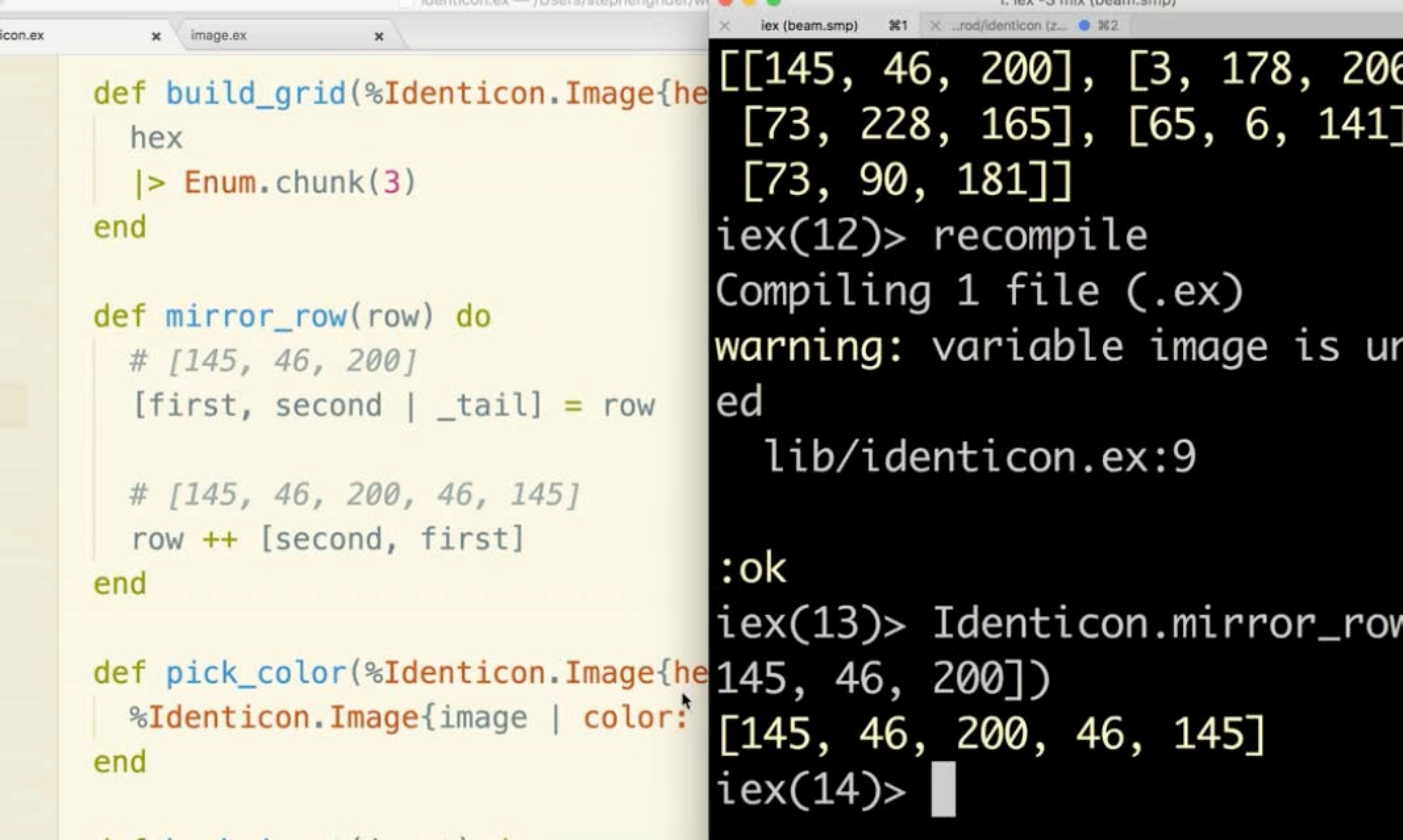
```
    |> hash_input
    |> pick_color
    |> build_grid
  end

  def build_grid(%Identicon.Image{hex: hex} = image) do
    hex
    |> Enum.chunk(3)
  end

  def pick_color(%Identicon.Image{hex: [r, g, b | _tail]} = image) do
    %Identicon.Image{image | color: {r, g, b}}
  end

  def hash_input(input) do
    hex = :crypto.hash(:md5, input)
    |> :binary.bin_to_list
```



```
x(9)> recompile
mpiling 1 file (.ex)
rning: variable image is unused
lib/identicon.ex:9


k
x(10)> Identicon.main("asdf")
145, 46, 200], [3, 178, 206], [73, 228, 165], [65, 6, 141],
73, 90, 181]]
x(11)>
```

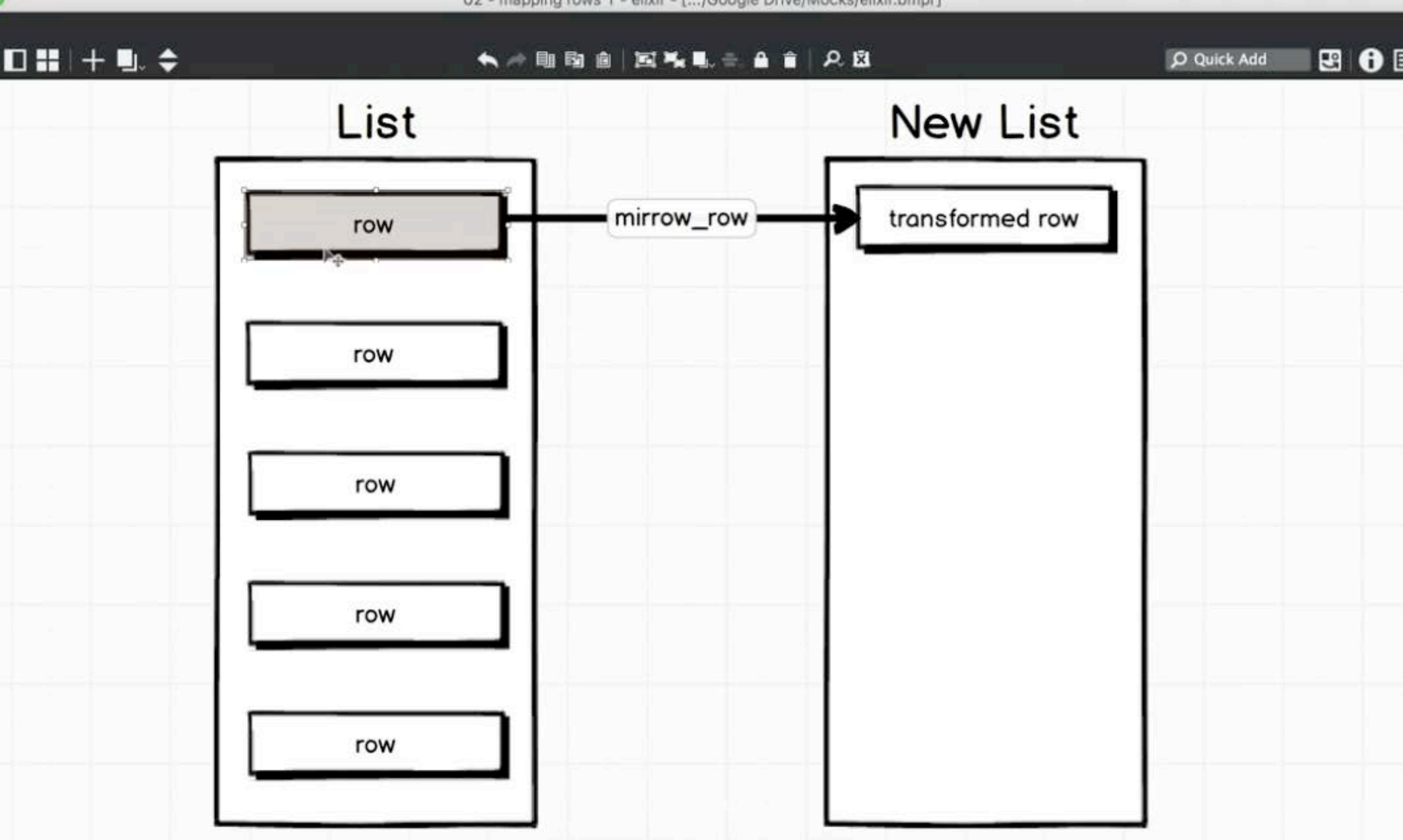


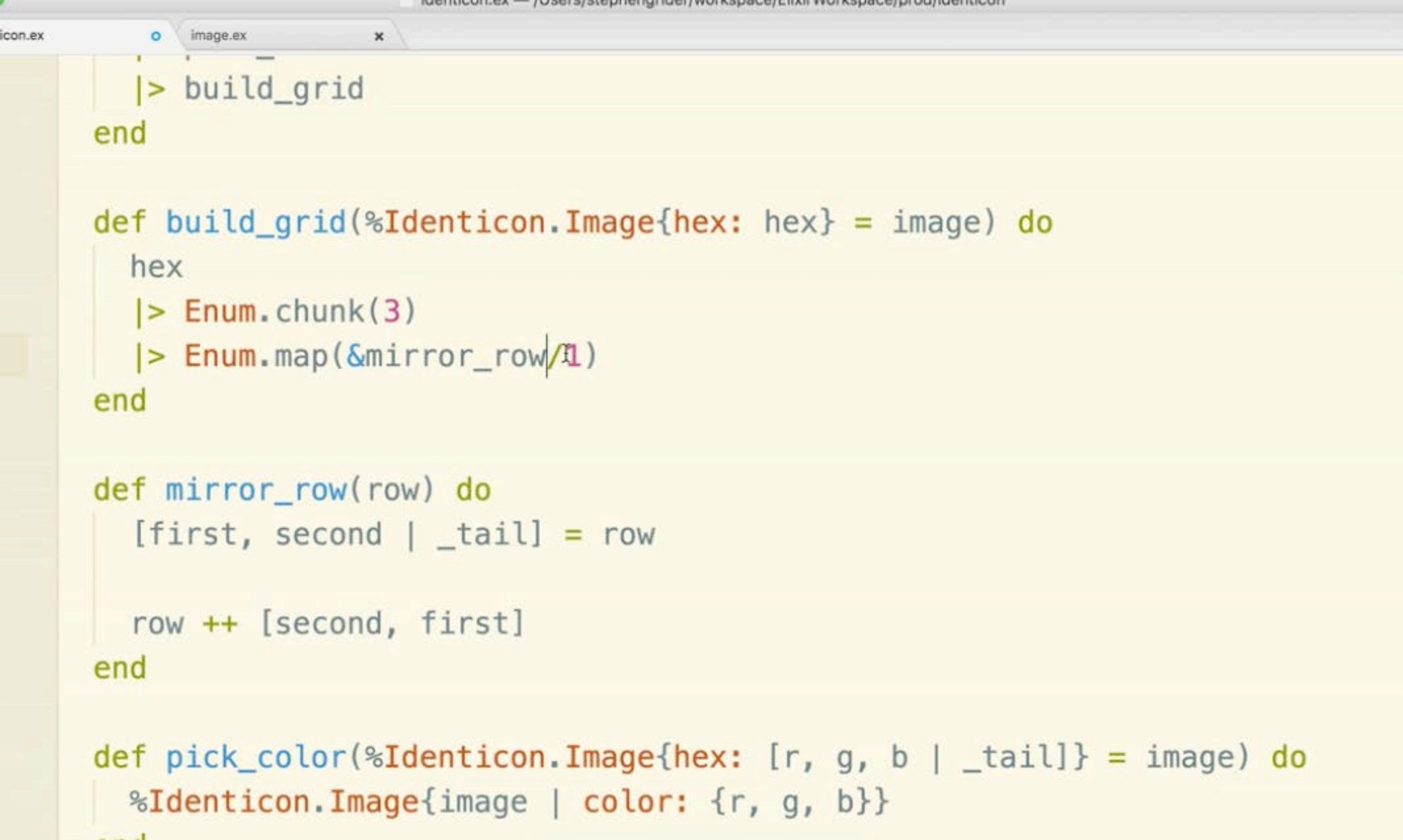
```elixir
def build_grid(%Identicon.Image{hex: hex} = image) do
  hex
  |> Enum.chunk(3)
end

def mirror_row(row) do
  # [145, 46, 200]
  [first, second | _tail] = row

  # [145, 46, 200, 46, 145]
  row ++ [second, first]
end

def pick_color(%Identicon.Image{hex: [r, g, b | _tail]} = image) do
  %Identicon.Image{image | color: {r, g, b}}
end
```



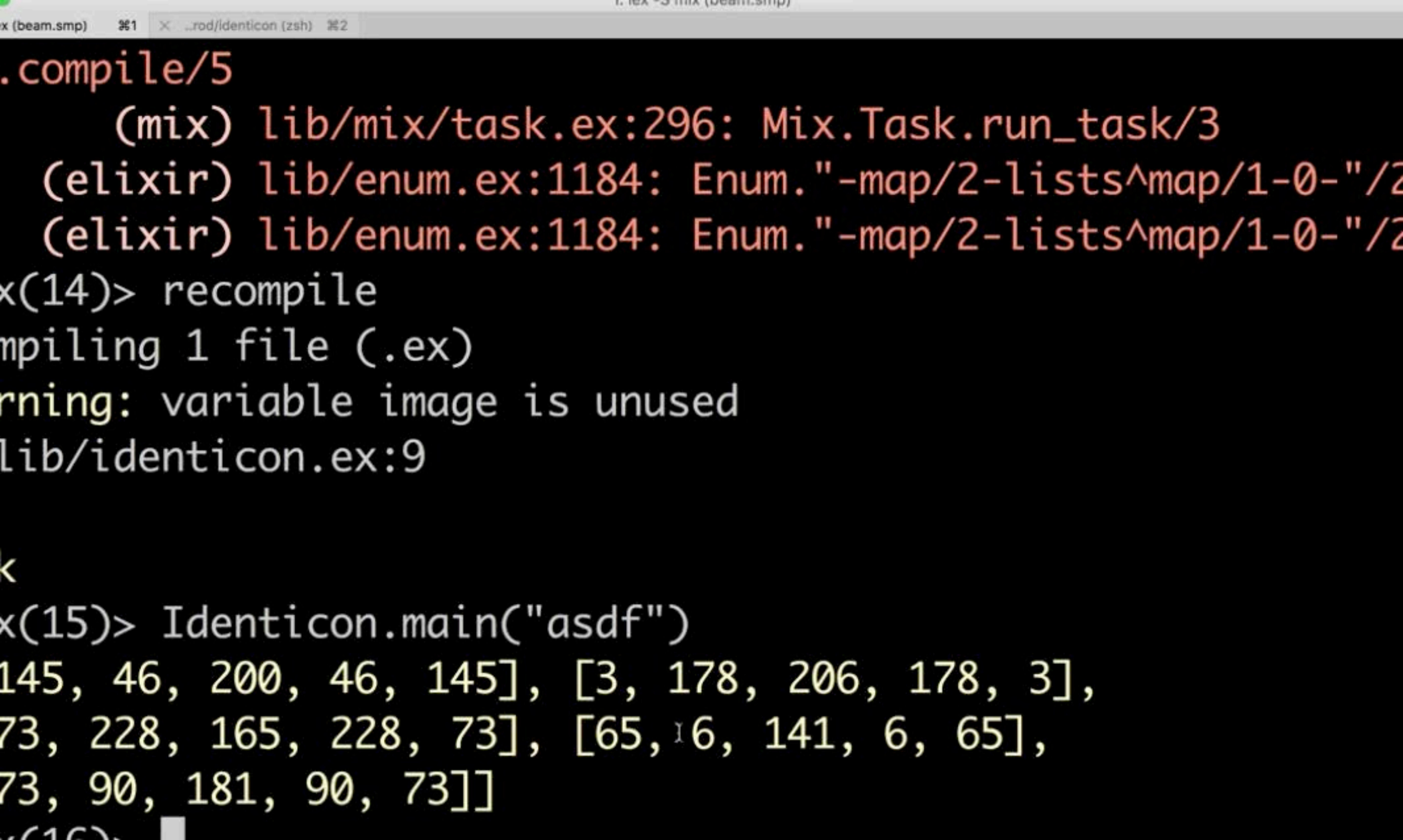
```
def build_grid(%Identicon.Image{he
  hex
  |> Enum.chunk(3)
end

def mirror_row(row) do
  # [145, 46, 200]
  [first, second | _tail] = row

  # [145, 46, 200, 46, 145]
  row ++ [second, first]
end

def pick_color(%Identicon.Image{he
  %Identicon.Image{image | color:
end
```

```
[[145, 46, 200], [3, 178, 206
 [73, 228, 165], [65, 6, 141]
 [73, 90, 181]]
iex(12)> recompile
Compiling 1 file (.ex)
warning: variable image is un
ed
  lib/identicon.ex:9


:ok
iex(13)> Identicon.mirror_row
145, 46, 200])
[145, 46, 200, 46, 145]
iex(14)>
```

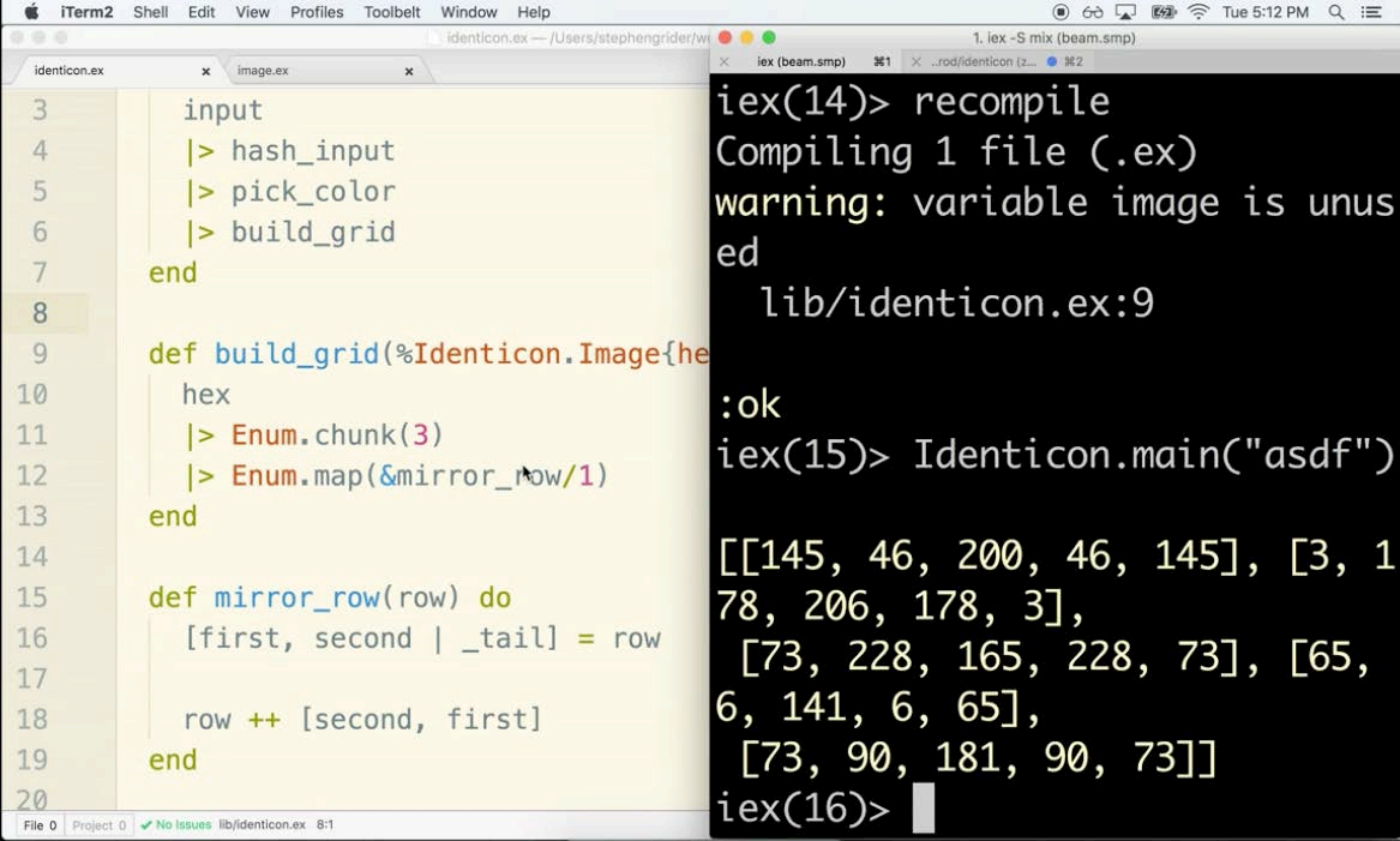

Quick Add
List
New List
row
row
row
row
row
mirrow_row
transformed row



```
  |> build_grid
end

def build_grid(%Identicon.Image{hex: hex} = image) do
  hex
  |> Enum.chunk(3)
  |> Enum.map(&mirror_row/1)
end

def mirror_row(row) do
  [first, second | _tail] = row

  row ++ [second, first]
end

def pick_color(%Identicon.Image{hex: [r, g, b | _tail]} = image) do
  %Identicon.Image{image | color: {r, g, b}}
```

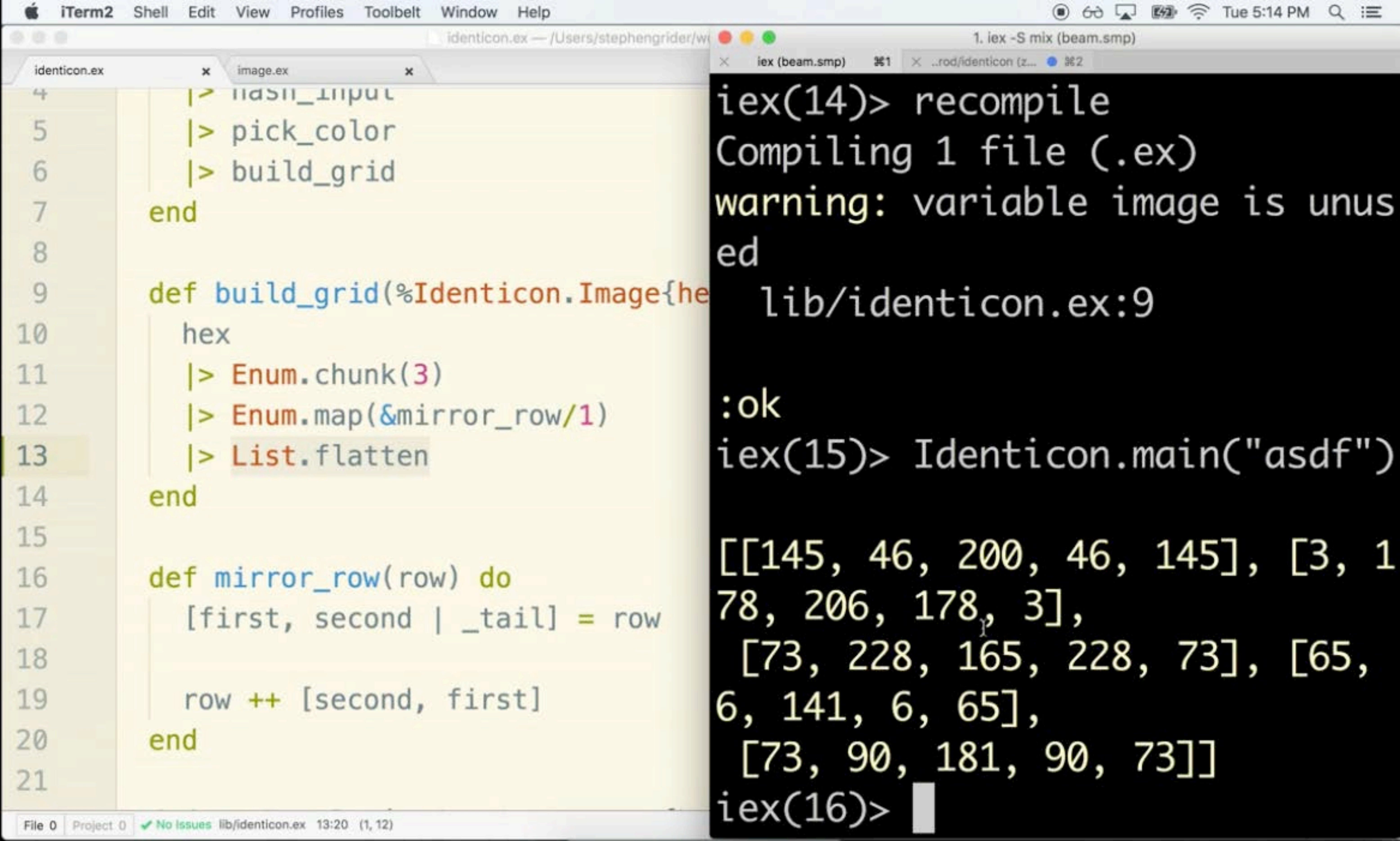


```
.compile/5
     (mix) lib/mix/task.ex:296: Mix.Task.run_task/3
  (elixir) lib/enum.ex:1184: Enum."-map/2-lists^map/1-0-"/2
  (elixir) lib/enum.ex:1184: Enum."-map/2-lists^map/1-0-"/2
x(14)> recompile
mpiling 1 file (.ex)
rning: variable image is unused
lib/identicon.ex:9


k
x(15)> Identicon.main("asdf")
145, 46, 200, 46, 145], [3, 178, 206, 178, 3],
73, 228, 165, 228, 73], [65, 6, 141, 6, 65],
73, 90, 181, 90, 73]]
```



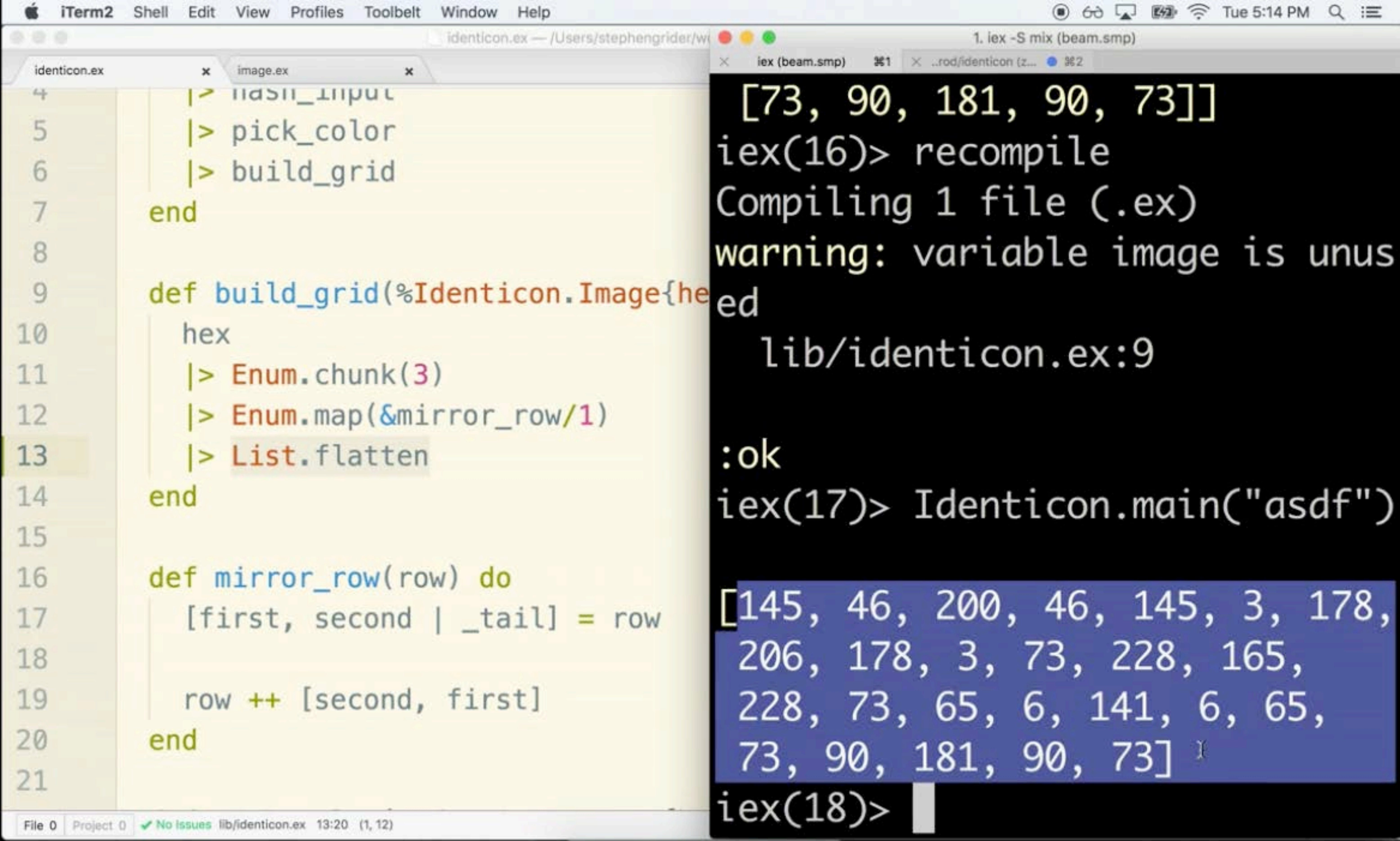
```elixir
    input
    |> hash_input
    |> pick_color
    |> build_grid
  end

  def build_grid(%Identicon.Image{he
    hex
    |> Enum.chunk(3)
    |> Enum.map(&mirror_row/1)
  end

  def mirror_row(row) do
    [first, second | _tail] = row

    row ++ [second, first]
  end
```

```
iex(14)> recompile
Compiling 1 file (.ex)
warning: variable image is unus
ed
  lib/identicon.ex:9

:ok
iex(15)> Identicon.main("asdf")

[[145, 46, 200, 46, 145], [3, 1
78, 206, 178, 3],
 [73, 228, 165, 228, 73], [65,
6, 141, 6, 65],
 [73, 90, 181, 90, 73]]
iex(16)> 
```


iTerm2 Shell Edit View Profiles Toolbelt Window Help
Tue 5:14 PM
identicon.ex — /Users/stephengrider/w
identicon.ex
image.ex
1. iex -S mix (beam.smp)
iex (beam.smp) ⌘1
..rod/identicon (z... ⌘2
File 0 Project 0 No Issues lib/identicon.ex 13:20 (1, 12)

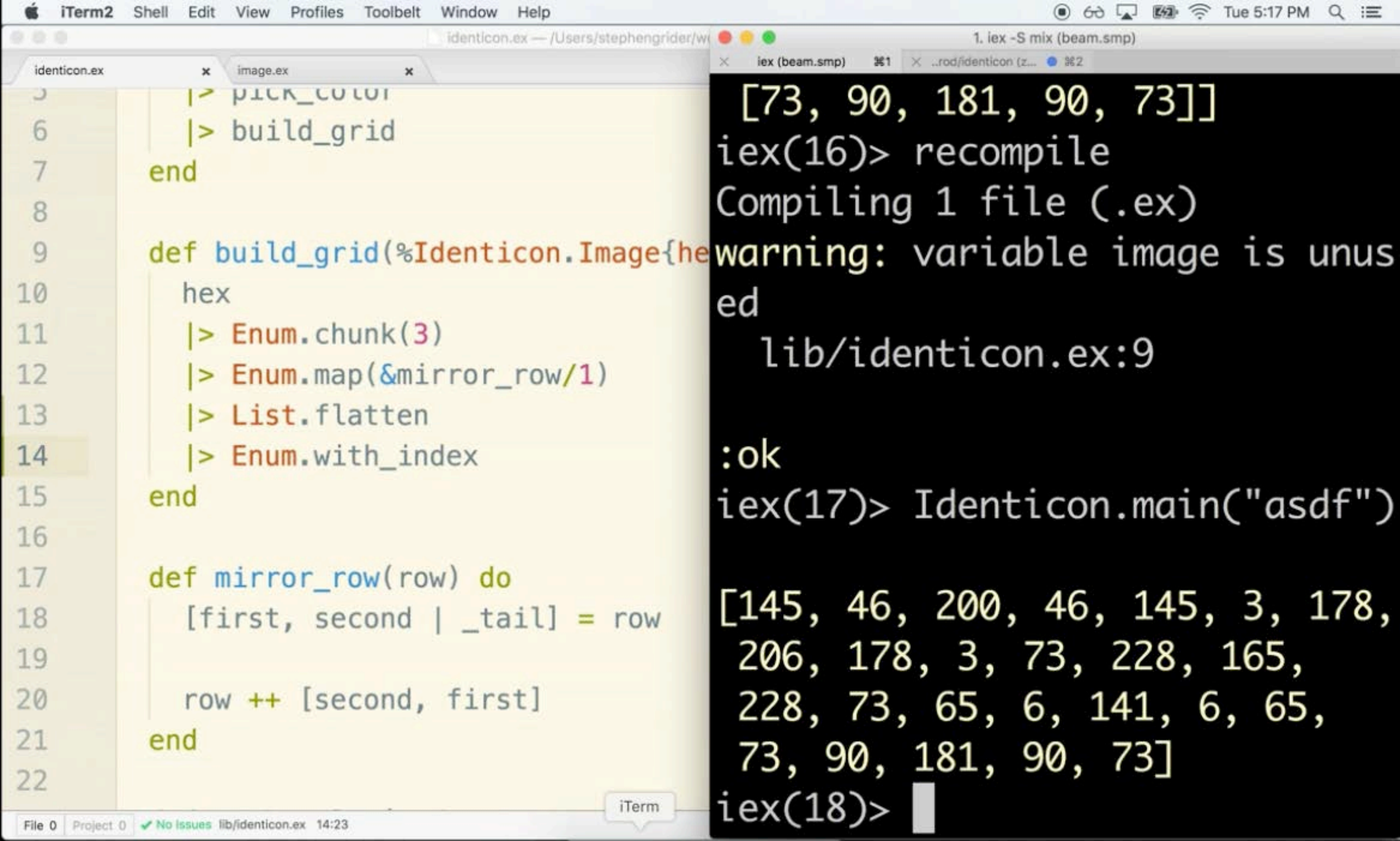
```
    |> hash_input
    |> pick_color
    |> build_grid
  end

  def build_grid(%Identicon.Image{he
    hex
    |> Enum.chunk(3)
    |> Enum.map(&mirror_row/1)
    |> List.flatten
  end

  def mirror_row(row) do
    [first, second | _tail] = row

    row ++ [second, first]
  end

```

```
iex(14)> recompile
Compiling 1 file (.ex)
warning: variable image is unus
ed
  lib/identicon.ex:9

:ok
iex(15)> Identicon.main("asdf")

[[145, 46, 200, 46, 145], [3, 1
78, 206, 178, 3],
 [73, 228, 165, 228, 73], [65,
6, 141, 6, 65],
 [73, 90, 181, 90, 73]]
iex(16)> 
```



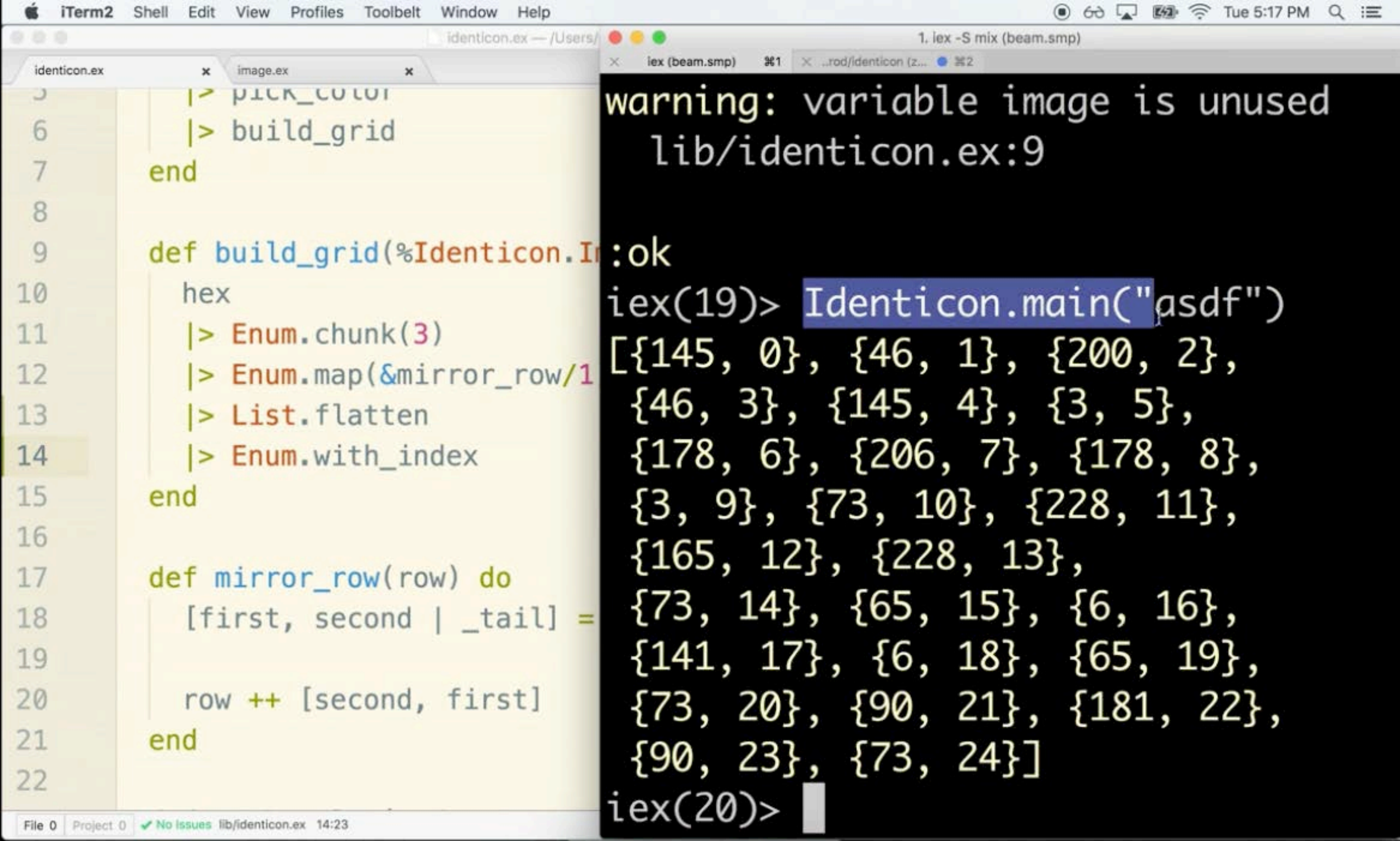
```elixir
    |> hash_input
    |> pick_color
    |> build_grid
  end

  def build_grid(%Identicon.Image{he
    hex
    |> Enum.chunk(3)
    |> Enum.map(&mirror_row/1)
    |> List.flatten
  end

  def mirror_row(row) do
    [first, second | _tail] = row

    row ++ [second, first]
  end
```

```
 [73, 90, 181, 90, 73]]
iex(16)> recompile
Compiling 1 file (.ex)
warning: variable image is unus
ed
  lib/identicon.ex:9

:ok
iex(17)> Identicon.main("asdf")

[145, 46, 200, 46, 145, 3, 178,
 206, 178, 3, 73, 228, 165,
 228, 73, 65, 6, 141, 6, 65,
 73, 90, 181, 90, 73]
iex(18)>
```



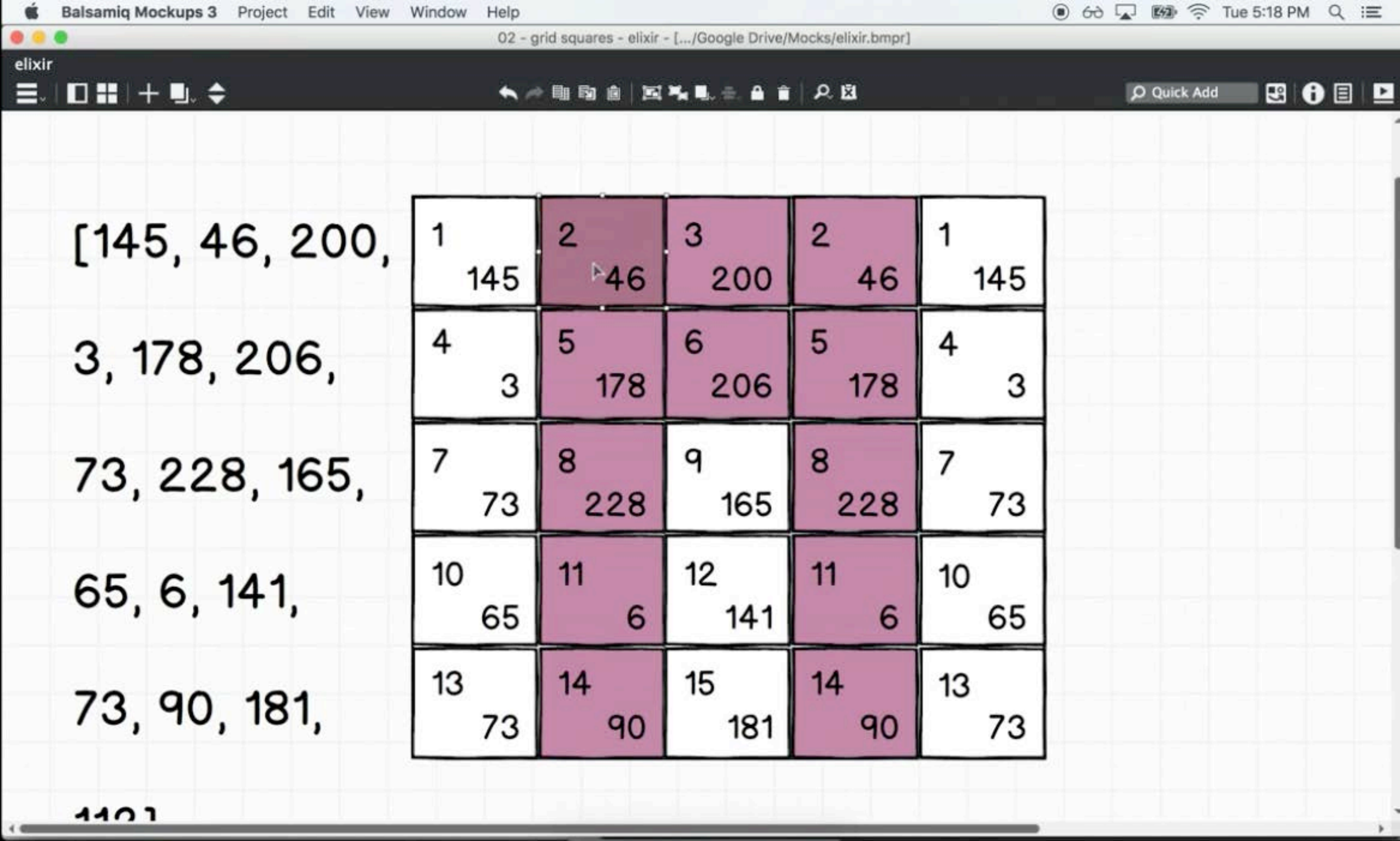
```elixir
    |> pick_color
    |> build_grid
  end

  def build_grid(%Identicon.Image{he
    hex
    |> Enum.chunk(3)
    |> Enum.map(&mirror_row/1)
    |> List.flatten
    |> Enum.with_index
  end

  def mirror_row(row) do
    [first, second | _tail] = row

    row ++ [second, first]
  end
```

```
 [73, 90, 181, 90, 73]]
iex(16)> recompile
Compiling 1 file (.ex)
warning: variable image is unus
ed
  lib/identicon.ex:9

:ok
iex(17)> Identicon.main("asdf")

[145, 46, 200, 46, 145, 3, 178,
 206, 178, 3, 73, 228, 165,
 228, 73, 65, 6, 141, 6, 65,
 73, 90, 181, 90, 73]
iex(18)> 
```



```
    |> pick_color
    |> build_grid
  end

  def build_grid(%Identicon.I
    hex
    |> Enum.chunk(3)
    |> Enum.map(&mirror_row/1
    |> List.flatten
    |> Enum.with_index
  end

  def mirror_row(row) do
    [first, second | _tail] =

    row ++ [second, first]
  end

```

```
warning: variable image is unused
  lib/identicon.ex:9

:ok
iex(19)> Identicon.main("asdf")
[{145, 0}, {46, 1}, {200, 2},
 {46, 3}, {145, 4}, {3, 5},
 {178, 6}, {206, 7}, {178, 8},
 {3, 9}, {73, 10}, {228, 11},
 {165, 12}, {228, 13},
 {73, 14}, {65, 15}, {6, 16},
 {141, 17}, {6, 18}, {65, 19},
 {73, 20}, {90, 21}, {181, 22},
 {90, 23}, {73, 24}]
iex(20)> 
```



[145, 46, 200,

3, 178, 206,

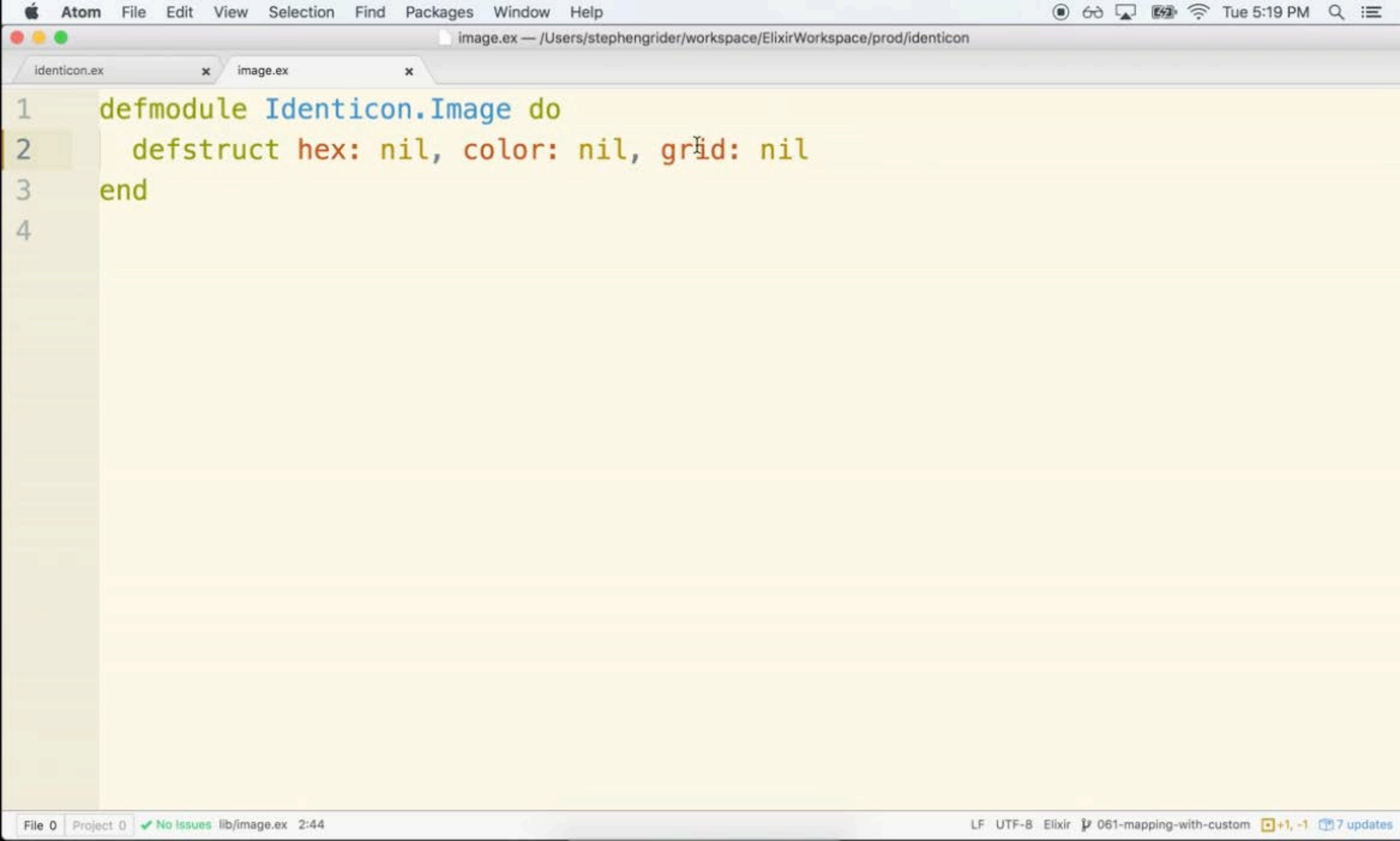
73, 228, 165,

65, 6, 141,

73, 90, 181,

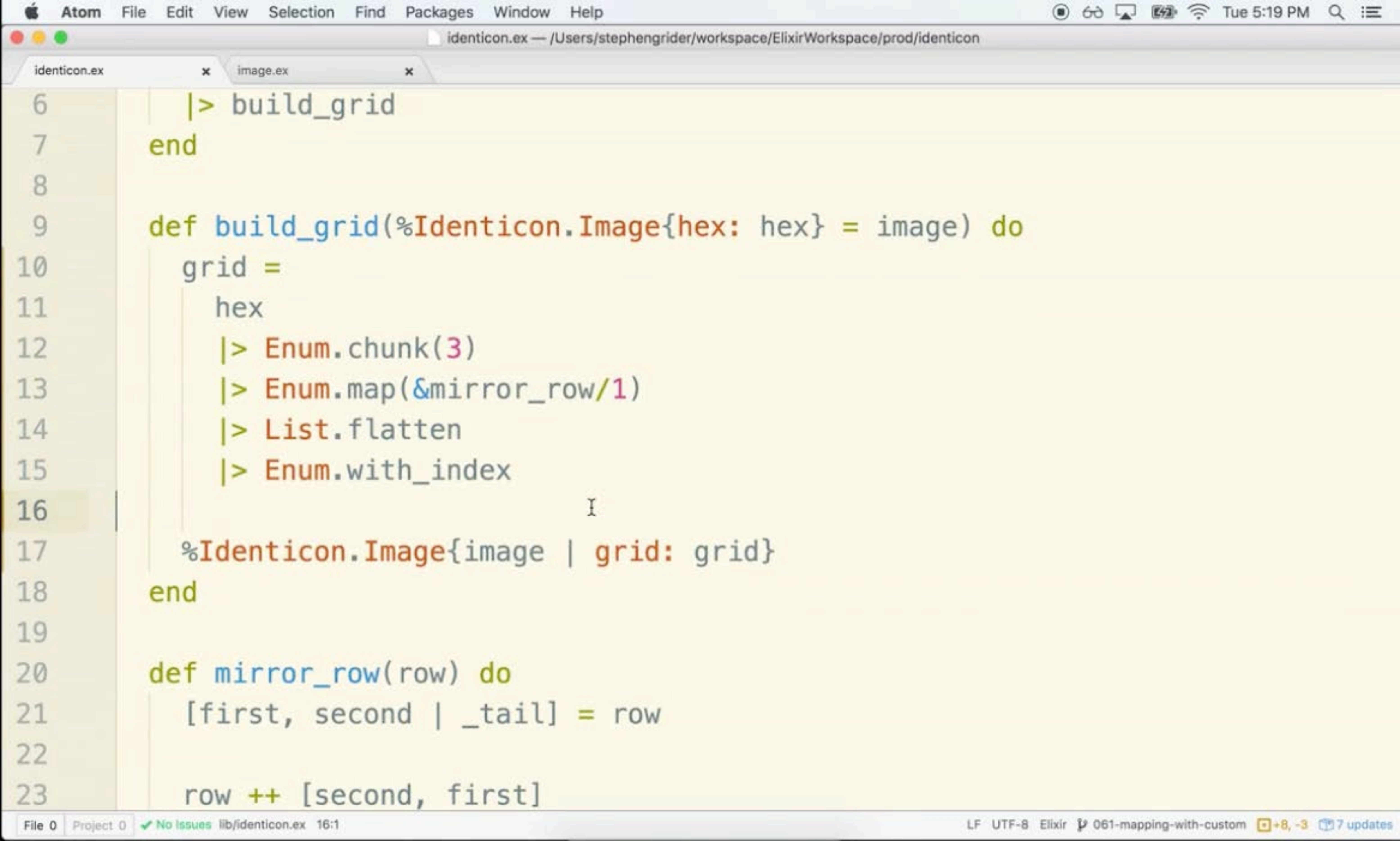
112]

1 145	2 46	3 200	2 46	1 145
4 3	5 178	6 206	5 178	4 3
7 73	8 228	9 165	8 228	7 73
10 65	11 6	12 141	11 6	10 65
13 73	14 90	15 181	14 90	13 73

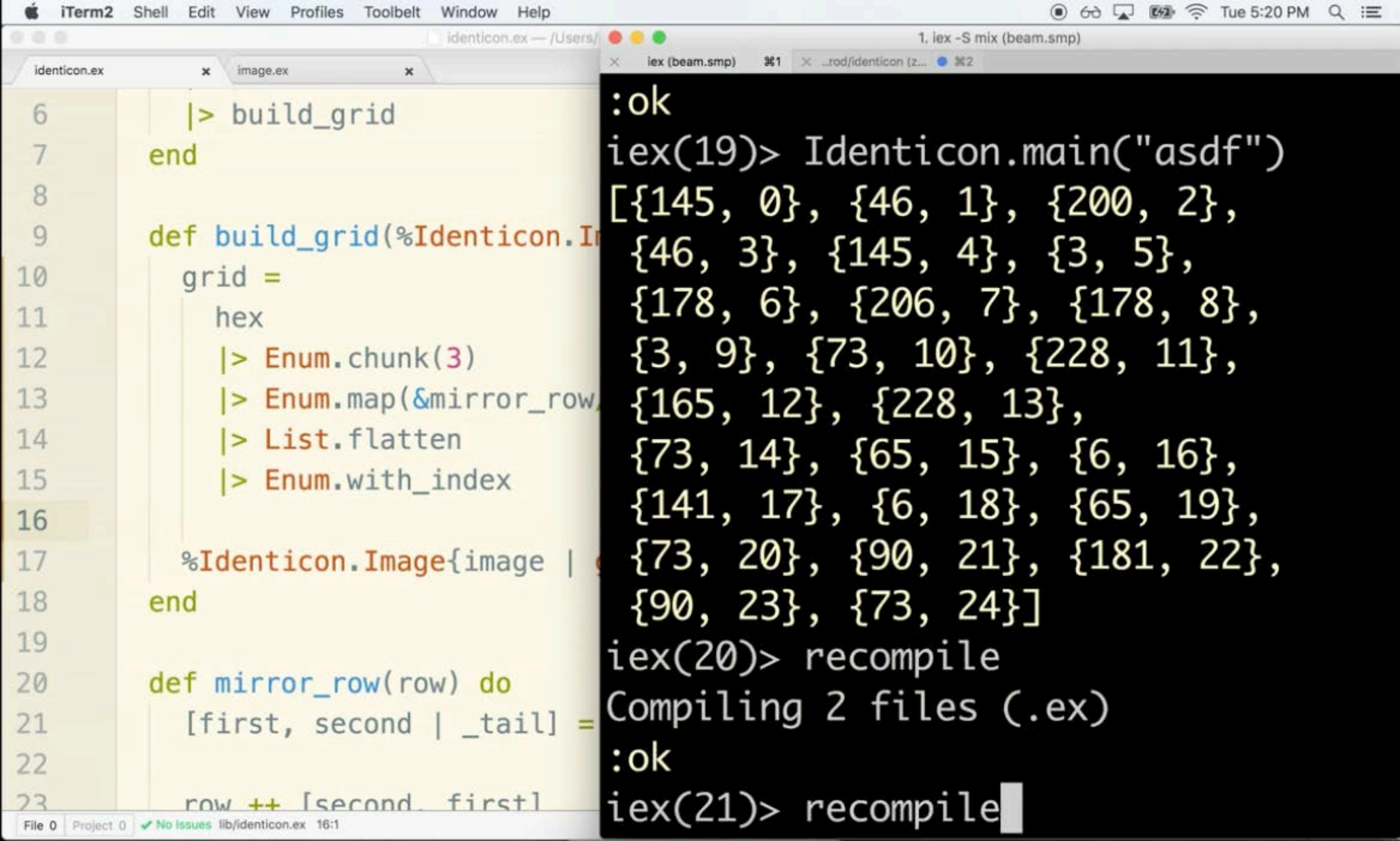

Atom File Edit View Selection Find Packages Window Help
Tue 5:19 PM
image.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/identicon
identicon.ex
image.ex

```elixir
defmodule Identicon.Image do
  defstruct hex: nil, color: nil, grid: nil
end
```

File 0 Project 0 No Issues lib/image.ex 2:44
LF UTF-8 Elixir 061-mapping-with-custom +1, -1 7 updates


Atom File Edit View Selection Find Packages Window Help
Tue 5:19 PM
identicon.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/identicon
identicon.ex
image.ex
File 0 Project 0 No Issues lib/identicon.ex 16:1
LF UTF-8 Elixir 061-mapping-with-custom +8, -3 7 updates

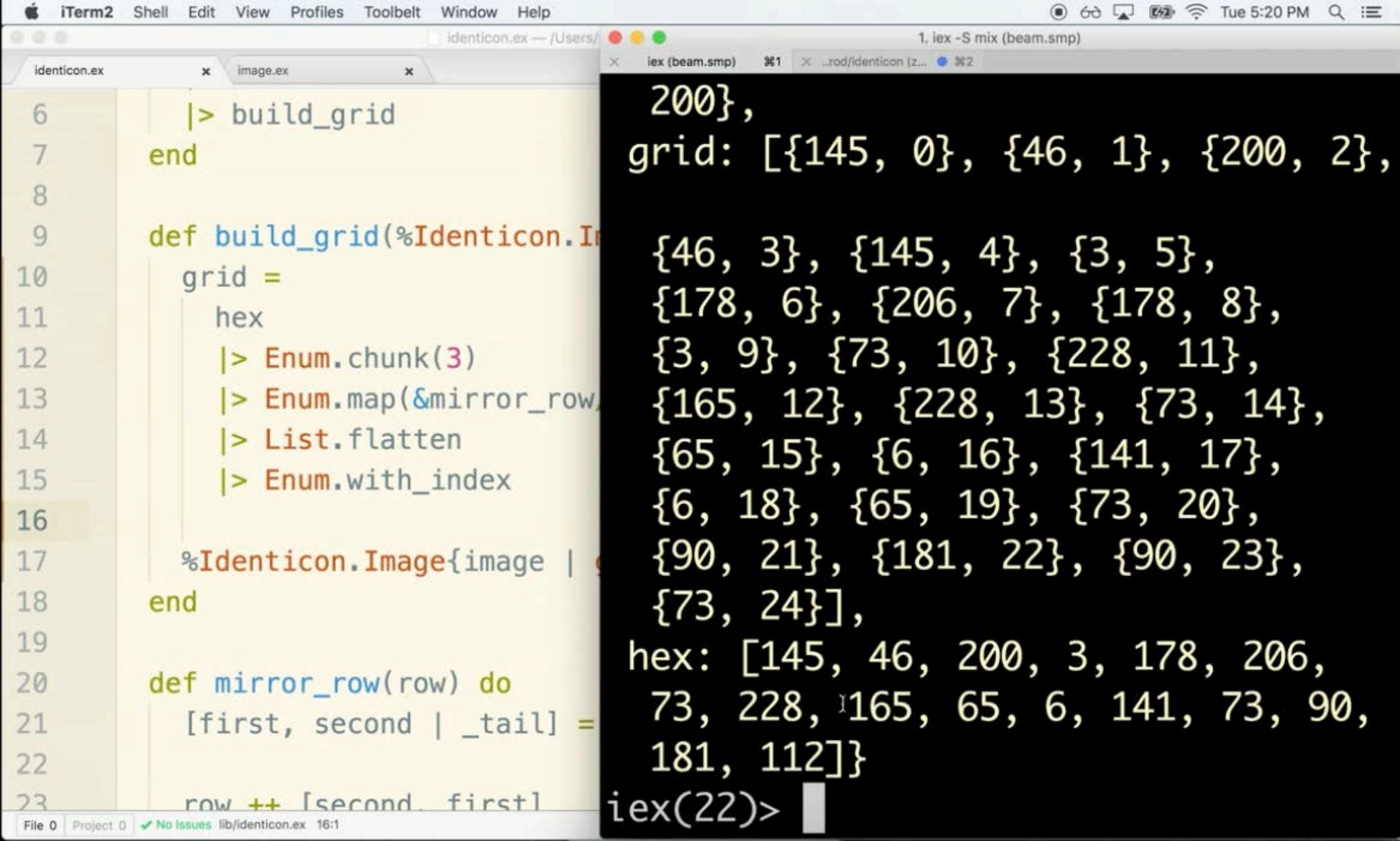
```elixir
    |> build_grid
  end

  def build_grid(%Identicon.Image{hex: hex} = image) do
    grid =
      hex
      |> Enum.chunk(3)
      |> Enum.map(&mirror_row/1)
      |> List.flatten
      |> Enum.with_index

    %Identicon.Image{image | grid: grid}
  end

  def mirror_row(row) do
    [first, second | _tail] = row

    row ++ [second, first]
```



```elixir
    |> build_grid
  end

  def build_grid(%Identicon.I
    grid =
      hex
      |> Enum.chunk(3)
      |> Enum.map(&mirror_row,
      |> List.flatten
      |> Enum.with_index


    %Identicon.Image{image | 
  end

  def mirror_row(row) do
    [first, second | _tail] =

    row ++ [second, first]
```

```
:ok
iex(19)> Identicon.main("asdf")
[{145, 0}, {46, 1}, {200, 2},
 {46, 3}, {145, 4}, {3, 5},
 {178, 6}, {206, 7}, {178, 8},
 {3, 9}, {73, 10}, {228, 11},
 {165, 12}, {228, 13},
 {73, 14}, {65, 15}, {6, 16},
 {141, 17}, {6, 18}, {65, 19},
 {73, 20}, {90, 21}, {181, 22},
 {90, 23}, {73, 24}]
iex(20)> recompile
Compiling 2 files (.ex)
:ok
iex(21)> recompile
```

iTerm2 Shell Edit View Profiles Toolbelt Window Help
Tue 5:20 PM
identicon.ex
image.ex
1. iex -S mix (beam.smp)
iex (beam.smp) ⌘1
..rod/identicon (z... ⌘2
File 0 Project 0 No Issues lib/identicon.ex 16:1



```
    |> build_grid
  end

  def build_grid(%Identicon.I
    grid =
      hex
      |> Enum.chunk(3)
      |> Enum.map(&mirror_row/
      |> List.flatten
      |> Enum.with_index


    %Identicon.Image{image | g
  end

  def mirror_row(row) do
    [first, second | _tail] =

    row ++ [second, first]
```

```
 200},
grid: [{145, 0}, {46, 1}, {200, 2},

 {46, 3}, {145, 4}, {3, 5},
 {178, 6}, {206, 7}, {178, 8},
 {3, 9}, {73, 10}, {228, 11},
 {165, 12}, {228, 13}, {73, 14},
 {65, 15}, {6, 16}, {141, 17},
 {6, 18}, {65, 19}, {73, 20},
 {90, 21}, {181, 22}, {90, 23},
 {73, 24}],
hex: [145, 46, 200, 3, 178, 206,
 73, 228, 165, 65, 6, 141, 73, 90,
 181, 112]}
iex(22)>
```