

# HOW DARE YOU?

Praanto Samadder

March 13, 2024

## 1 The Elixir way



Our morse code can have three characters: ‘?.’, ‘?-’ and ‘? ’. The period representing a short beep, take right, the dash representing a long beep, take left, the whitespace representing end of character, repeat.

Therefore, we can expand our `decode/5` method to account for a period

```
1 def decode([? . | rest], { :node, char, left, _ }, root, _, msg) do
2   decode(rest, left, root, char, msg)
3 end
```

and a dash

```
1 def decode([? - | rest], { :node, char, _, right }, root, _, msg) do
2   decode(rest, right, root, char, msg)
3 end
```

and a whitespace

```
1 def decode([? | rest], { :node, char, _, _ }, root, _, msg) do
2   msg = msg <> <<char::utf8>>
3   decode(rest, root, root, " ", msg)
4 end
```

When we encounter a whitespace, of course we have to append our character to the message. We then keep passing this message around as it travels through the recursive path of hell.

However, we do encounter a small edge case: when our ‘child’ character is `:na`. This, as mentioned in the instructions, is a dummy character and thus, we discard. Commence Elixir pattern matching

```
1 def decode([32 | rest], {:_node, :na, _, _}, root, _, message) do
2   decode(rest, root, root, "", message)
3 end
```

## 2 The encoder

The encoder is very trivial to implement. Recursively loop through the plain-text and replace each character with it’s morse code equivalent followed by a whitespace.

## 3 Final answers

The final two texts are

all your bases are belong to us

and

<https://www.youtube.com/watch?v=d%51w4w9%57g%58c%51>

which is a link Rick Astley’s hit song Never Gonna Give You Up (how dare you rickroll us like that?!).

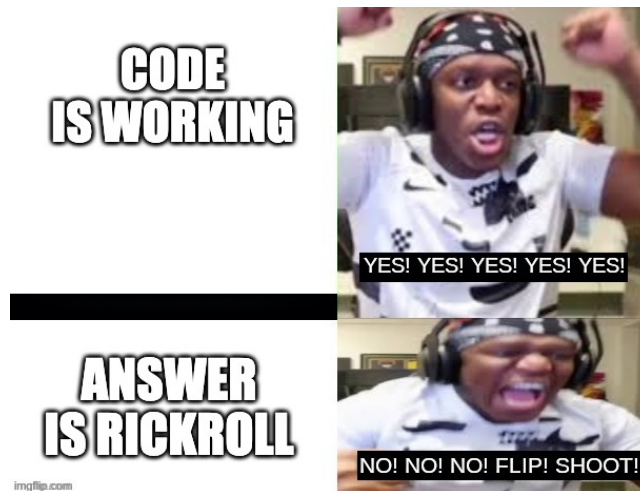


Figure 1: My reaction

## 4 Final thoughts

Finally, as a personal note to Montelius: I had used functional programming before but I used it in multi-paradigm languages which means I got to cheat my way out whenever I needed to. I used to hate recursive programming and think you needed to be gifted or be insane to even start thinking recursively. This course, with Elixir however, has changed all of those views. And I am thankful for that.

Now, I shall begin on a two week rehab period to alleviate myself from functional-programming-withdrawal-syndrome as I go back to working on my personal projects, all of which were written with procedural programming.