Do Parsers Dream of Electric Guitars?!

!!Con(**@arkh4m**)

Hello! 

My name is Ju 

I like to play the 

But I'm not very good at it 

So I have to look up songs on the 

But those  are dark and full of 

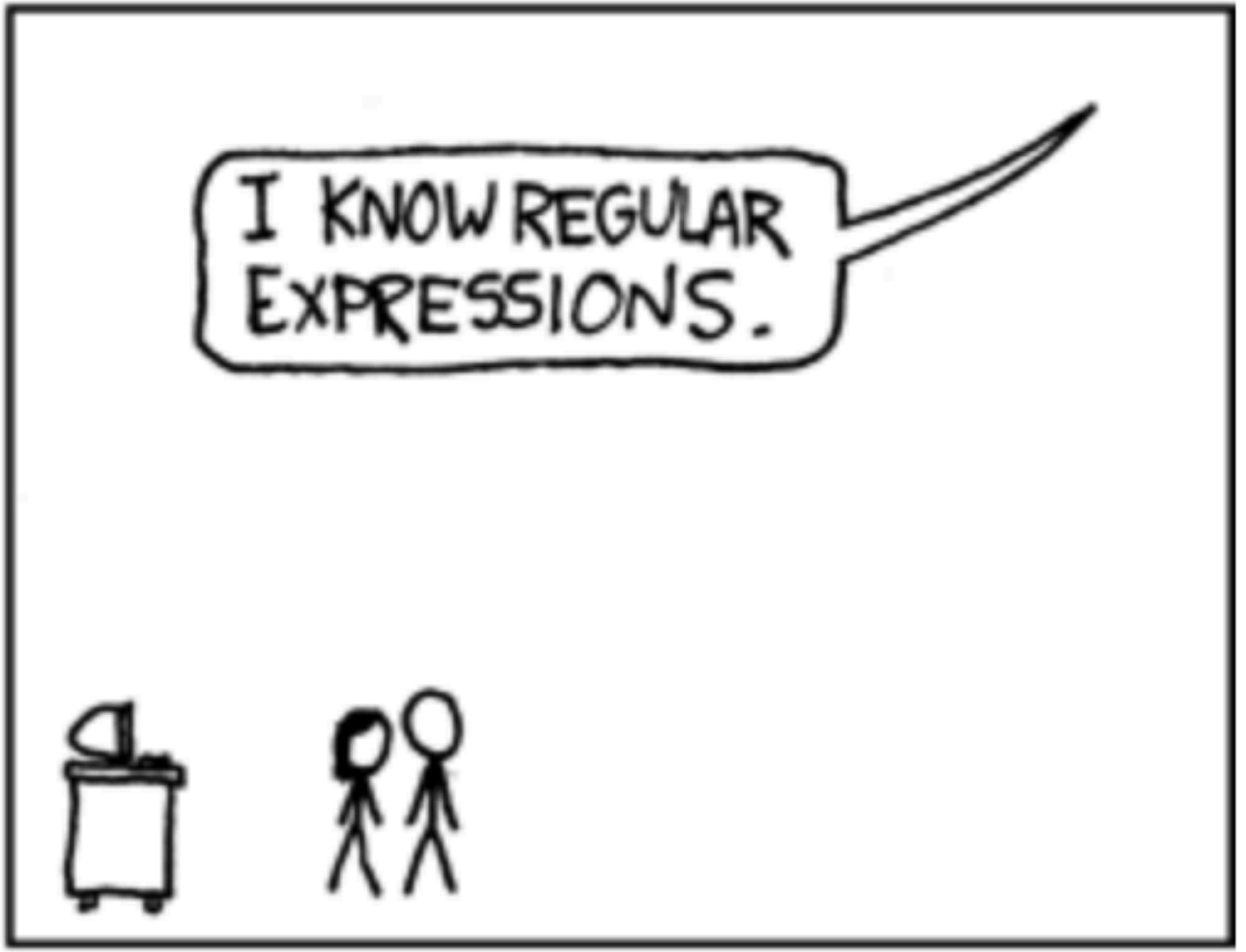
[Am] Hello darkness, my old **[G]**friend,
I've come to talk with you **[Am]**again,
Because a regex soft**[F]**ly cree**[C]**ping,
Left its seeds while I **[F]**was slee**[C]**ping,
The **[F]**revision that was planted in
[C]git blame, still remains**[C] [Am]**
[C]Within the **[G]**sound of **[Am]**silence.


ARE YOU SMARTER THAN
JUSTIN BIEBER?
His IQ is above 102! Most People
CANNOT beat him. Can you beat him?
Connect with Facebook
Are You Smart Enough!?
Conncet with your Facebook Account
to see if you're SMARTER than Justin Bieber?
CLICK!
You've won!
account instantly.
Call any family and friends you may have
in a two block radius. The card is good
for six minutes after clicking here.

Fine, I'll do it myself

[Am] Hello darkness, my old **[G]**friend,
I've come to talk with you **[Am]**again,
Because a regex soft**[F]**ly cree**[C]**ping,
Left its seeds while I **[F]**was slee**[C]**ping,
The **[F]**revision that was planted in
[C]git blame, still remains**[C] [Am]**
[C]Within the **[G]**sound of **[Am]**silence.

Parsing a blurb of text?



xkcd.com/208

/\[[ABCDEFG]m?\]/

This regex finds chords

But it only understands text

It doesn't *really* understands chords

String -> [Chord]

String -> [String]

* disappointed_but_relieved_face

Let's use Elm and elm/parser

Wishful Programming™•

• Not actually trademark

!!Con(**@arkh4m**)

A Note

```
type Note
    = A
    | C
    | F
    | G
```

A Chord

```
type Chord =
    Chord Note
```

A Chord, Pt II

```
type Quality
     = Major
     | Minor
     | Dominant7

type Chord
     = Chord Note Quality
```

But we still need to parse the text!

Let's build a parser

```
import Parser exposing (Parser)

chordParser : Parser ???
chordParser =
    ???
```

```
import Parser exposing (Parser)

chordParser : Parser Chord
chordParser =
    ???
```

Wishful Programming™

!!Con(**@arkh4m**)

```
import Parser exposing (Parser, (|=), (|.))

chordParser : Parser Chord
chordParser =
    Parser.succeed (\note quality -> Chord note quality)
    |= noteParser
    |= qualityParser
```

```
noteParser : Parser Note
noteParser =
    Parser.oneOf
        [ Parser.succeed A
            |. symbol "A"
        , Parser.succeed F
            |. symbol "F"
        , Parser.succeed G
            |. symbol "G"
        , Parser.succeed C
            |. symbol "C"
        ]
```

```
qualityParser : Parser Quality
qualityParser =
    Parser.oneOf
        [ Parser.succeed Minor
            |. symbol "m"
        , Parser.succeed Dominant7
            |. symbol "7"
        , Parser.succeed Major
        ]
```

Tada 🎉

```
import Parser exposing (Parser, (|=), (|.))

chordParser : Parser Chord
chordParser =
    Parser.succeed (\note quality -> Chord note quality)
    |= noteParser
    |= qualityParser
```

Tada 🎉

```
import Parser exposing (Parser, (|=), (|.))

chordParser : Parser Chord
chordParser =
    Parser.succeed (\note quality -> Chord note quality)
    |. symbol "["
    |= noteParser
    |= qualityParser
    |. symbol "]"
```

Now we really know what the chords are!

What can we do with that

→ Figure out which notes compose a chord

→ Build a guitar fretboard in Elm

→ Choose chords that are nice to play

→ Build SVG charts of such chords

→ All of this on the fly! Nothing hardcoded!

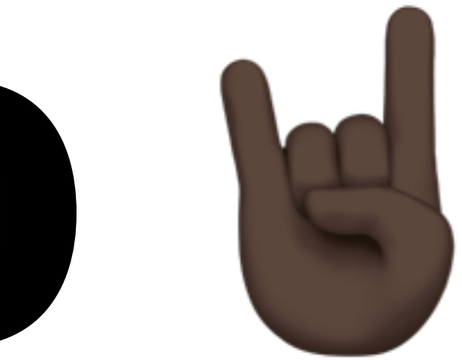
DEMO 

```
type Quality
    = Fifth
    | Tertian TertianQuality
    | Sus2 TertianQuality
    | Sus4 TertianQuality
    | Add9 TertianQuality
    | Add11 TertianQuality
    | NewRoot Note TertianQuality

type TertianQuality
    = Major -- Root, Major Third, Perfect Fifth
    | Minor -- Root, Minor Third, Perfect Fifth
    | Augmented -- Root, Major Third, Augmented Fifth
    | Diminished -- Root, Minor Third, Diminished Fifth
    | Dominant7 -- Root, Major Third, Perfect Fifth, Minor Seventh
    | Major7 -- Root, Major Third, Perfect Fifth, Major Seventh
    | Minor7 -- Root, Minor Third, Perfect Fifth, Major Seventh
    | AugmentedDominant7 -- Root, Major Third, Augmented Fifth, Minor Seventh
    | AugmentedMajor7 -- Root, Major Third, Augmented Fifth, Major Seventh
    | Diminished7 -- Root, Minor Third, Diminished Fifth, Diminished Seventh
    | Major6 -- Root, Major Third, Diminished Fifth, Major Sixth
    | Minor6 -- Root, Minor Third, Diminished Fifth, Major Sixth
    | Dominant9 -- Root, Major Third, Perfect Fifth, Minor Seventh, Major Ninth
    | Major9 -- Root, Major Third, Perfect Fifth, Major Seventh, Major Ninth
    | Minor9 -- Root, Minor Third, Perfect Fifth, Minor Seventh, Major Ninth
```

Thanks! 

One more thing...

LIVE DEMO 

Thank you!

CODE

github.com/Arkham/elm-chords

DEMO

ellie-app.com/5wDzdJyjQ6Ca1

LIVE DEMO

ellie-app.com/5wKq8hvfXxBa1