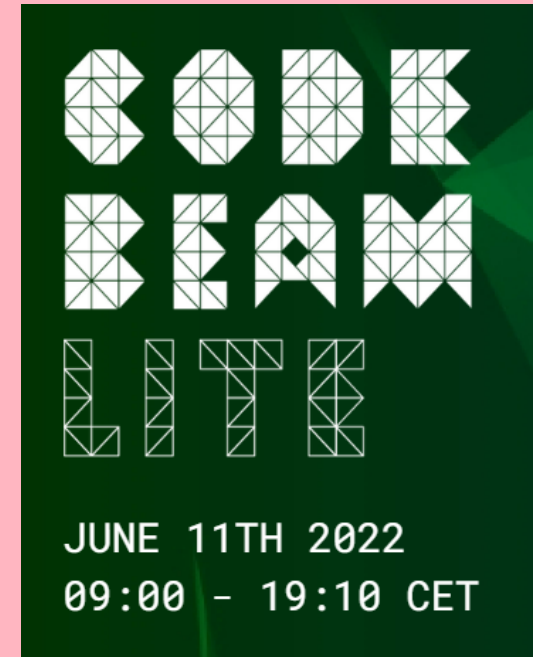


Learning Elixir and Erlang with Advent of Code and Exercism: advantages and challenges

Adolfo Neto [@adolfont](#) on Twitter

adolfont.github.io



Presentation script

- Who am I?
- What is Advent of Code?
- What is Exercism?
- What is good/bad in AoC/Exercism?
- **How to use AoC and/or Exercism to learn Erlang and/or Elixir?**

Who am I?

- Professor at the [Federal University of Technology – Parana \(UTFPR\)](#)



Who am I? (part 2)

- Co-host of [Elixir em Foco \(Elixir in Focus\)](#) podcast (in Brazilian Portuguese)
- Member of the [Education, Training, & Adoption Working Group of the Erlang Ecosystem Foundation](#)
- [YouTube EN](#), [YouTube PT](#), [Instagram](#), [Telegram](#), [GitHub](#), [Twitch](#), [TikTok](#)...

Which are the difficulties in learning Elixir and Erlang?

- Common programming pitfalls
- Erlang/Elixir pitfalls
 - Functional programming concepts
 - Syntax details
 - Not knowing the best practices of the communities
 - Concurrency/OTP (difficult by itself)



What helps? (in my opinion)

- Reading [books](#) or well-structured materials ([Elixir School](#))
- Having a mentor
- Chatting with other people
- Participating in the community
 - [Elixir Forum](#)
 - [Erlang Forums](#)
 - [#MyElixirStatus](#), [#ElixirLang](#), [#WeBEAMtogether](#), [#Erlang](#)
 - [Elixir World](#)
- Listening to [podcasts](#)
- What else?



What helps? (in my opinion)

- Putting knowledge into practice - just a little bit above your current level
- Write code idiomatically, the way more experienced people in the community write

Which is most used? I don't know, but...

"Have you ever solved an Advent of Code or an Exercism challenge?"

	A	B	C	D	E	F
1		Erlang Forums	Elixir Forum	Twitter	Sum	%
2	Total	10	26	135	171	
3	Exercism	4	19	85	108	63,16%
4	Advent of Code	7	17	32	56	32,75%
5	Nenhum dos dois	1	4	42	47	27,49%
6						



What is the Advent of Code?

- Website: <https://adventofcode.com/>
- Wikipedia: https://en.wikipedia.org/wiki/Advent_of_Code
- Creator: [Eric Wastl](#)
- Annual “event” that goes from December 1st to December 25th
 - It's a kind of competition
 - There are private leaderboards
 - What counts for the ranking is who finds the answer, usually a number, first
- So the time zone counts.

Me in Advent of Code

- 2021: 1 to 10, 13 and 14 - total $12 \times 2 = 24$
- 2020: 1 to 9 and part 1 of day 10: total 19
- 2019: 1 and 2. Total: 4

Where I am saving my solutions as of 2021

<https://github.com/adolfont/pensandoemelixir/tree/main/adventofcode/>



Important!

- Relatively long description (631, 577, 1002)
- Pay close attention to the wording!
 - [Day 2 Part 1 2019](#)
 - *"To do this, before running the program, replace position 1 with the value 12 and replace position 2 with the value 2."*
- Undisclosed difficulty
- Ad-hoc solutions



Exercism



exercism.io—programming skills practice and crowd-sourced mentorship.

[Exercism](#) is a place where people can improve their programming skills by solving toy problems and then having friendly and rich discussions about the ways to make the code better. The discussions often revolve around idioms and trade-offs, readability and optimization, and (of course) personal preferences.

Being able to effectively critique code or perform a code review means recognizing code smells and being able to articulate the pain associated with it.

I started building exercism as an experiment in workflow optimization back in 2013. Since then over 10,000 programmers have participated in the discussions. When we launched the exercises were all in Ruby, but exercism.io now has exercises in over 20 different languages.

exercism.io—programming skills practice and crowd-sourced mentorship.

[Exercism](#) is a place where people can improve their programming skills by solving toy problems and then having friendly and rich discussions about the ways to make the code better. The discussions often revolve around idioms and trade-offs, readability and optimization, and (of course) personal preferences.

Being able to effectively critique code or perform a code review means recognizing code smells and being able to articulate the pain associated with it.

I started building exercism as an experiment in workflow optimization back in 2013. Since then over 10,000 programmers have participated in the discussions. When we launched the exercises were all in Ruby, but exercism.io now has exercises in over 20 different languages.

What is Exercism?

- Created by [Katrina Owen](#)
 - read <https://www.kytrinyx.com/exercism/> !!!
 - and <https://en.wikipedia.org/wiki/Exercism>
- Exercises + free mentoring
- It gives you short descriptions (72, 96, 172)
- It provides you with tests for your programming language
- Current version: version 3






Access [https://exercism.org/https://exercism.org/](https://exercism.org/)


Mentoring example

Private link:

<https://exercism.org/mentoring/discussions/ad416ef10f1e42b4b97445f2ee672f53>



**jeanrangel**
on Freelancer Rates in Elixir

 **Ended**


Iteration 3


Latest

Published

Submitted 2 months ago

Passed





lib/freelancer_rates.ex

Instructions

Tests

```
1 defmodule FreelancerRates do
2   def daily_rate(hourly_rate) do
3     hourly_rate * 8.0
4   end
5
6   def apply_discount(before_discount, discount) do
7     before_discount - (before_discount * (discount/100))
8   end
9
10  def monthly_rate(hourly_rate, discount) do
11    hourly_rate
12    |> daily_rate
13    |> apply_discount(discount)
14    |> Kernel.*(22)
15    |> ceil
16  end
17
18  def divide_numbers(number1, number2) do
19    number2/number1
20  end
21
22  def days_in_budget(budget, hourly_rate, discount) do
23    hourly_rate
24    |> daily_rate
25    |> apply_discount(discount)
26    |> divide_numbers(budget)
```

1

2


4

1

3

Previous

Next




Discussion


Scratchpad


Guidance



Who you're mentoring


jeanrangel  1


jeanrangel

 **Favorited**

 **Iteration 1** was submitted 2mo ago

 Student received automated feedback 


 **jeanrangel** 2mo ago
This is a private review session


 **adolfont** 2mo ago
Please remove the (now useless) comments (lies 8, 13, 18).

Line 4: why do you need "/" and the corresponding parentheses?

Lines 14 and 19 could use more pipes (|>).

`hourly_rate |> daily_rate` does not seem to be an idiomatic use of pipe.

 **Iteration 2** was submitted 2mo ago

 **jeanrangel** 2mo ago
Please remove the (now useless) comments (lies 8, 13, 18).

ok

Line 4: why do you need "/" and the corresponding parentheses?

How can I change INT to FLOAT in Elixir?

Lines 14 and 19 could use more pipes (|>)

This discussion has ended. Have more to say? You can [still post](#).

Adolfo Neto (UTFFPR)

16



adolfont

2mo ago

Please remove the (now useless) comments (lies 8, 13, 18).

Line 4: why do you need `"/1"` and the corresponding parentheses?

Lines 14 and 19 could use more pipes (`|>`).

`hourly_rate |> daily_rate` does not seem to be an idiomatic use of pipe.



adolfont

2mo ago

How can I change INT to FLOAT in Elixir?

You could just do:

```
hourly_rate * 8.0
```

How can a function in pipe receive more than 1 argument? For example: (return_number1 & return_number2) |> multiply_two_numbers?

It would be:

```
first_argument  
|> your_function(second_argument)
```

José Valim and the Advent of Code

- José Valim's Twitch: 2018 and 2021
- Sekun's YouTube playlist with shorter, edited versions of Valim's videos
 - Support SEKUN



sekun | @sekun@mastodon.social

@sekunho_

I DID IT!!!! I GOT AN OFFER
YEAAAAHHHHHHHHHHHHHHH!!!!!!!!!!!!!!!!!!!!

11:39 AM · May 18, 2022 · Twitter for iPhone

19 Likes

Good/bad in AoC

- What's good about AoC?
 - Community (Twitter, Elixir Forum and Erlang Forums)
- What is "bad" in AoC?
 - Weekends! *But [SpawnFest](#) too :)*
 - December! *Spring/Summer in the Southern Hemisphere*
 - Some challenges require (a lot of) prior knowledge (programming competitions)
 - Is there a stimulus to write unreadable code?

Good/bad in Exercism

- What is good about Exercism?
 - mentoring
 - tests
 - automatic feedback
- What is "bad" in Exercism?
 - the tests are already there
 - It is impossible to do TDD
 - v3 kind of discourages mentoring (debatable)

What have I learned? 1/3

- Use of Enum (Valim) #aoc #elixir
 - *There's an Enum function for that!*
- Lots of pipes!
- Functions with the same name but different arity, one being public and the other private #erlang #exercism
- Various uses of **list comprehension**:

```
[command(S) || S <- string:split(String, [$\n], all), S != ""].
```



What have I learned? 2/3

- [case](#) [#erlang](#) [#aoc](#)

```
process(String, H, V, V2, A) ->
  case string:split(String, " ") of
    ["forward", Value] ->
      process(H + int(Value), V, V2 + A * int(Value), A);
    ["down", Value] ->
      process(H, V + int(Value), V2, A + int(Value));
    ["up", Value] ->
      process(H, V - int(Value), V2, A - int(Value))
  end.
```



- [read, understand and run code from others](#) [#erlang](#) [#aoc](#)

What have I learned? 2/3

- `re:split`

```
243 parse_file({ok, RawData}) ->
244     [Template | Rules] = re:split(RawData, "[\n\n]"),
245     {unicode:characters_to_list(Template), parse_rules(Rules)};
246 parse_file({error, _}) ->
247     "No file with that name!".
248
249 parse_rules(List) ->
250     [parse_rule(Rule) || Rule <- List, Rule != <<>>].
251
252 parse_rule(Rule) ->
253     [Left, Right | _Tail] = re:split(Rule, "[\->]+"),
254     {string:trim(
255         unicode:characters_to_list(Left)),
256      string:trim(
257         unicode:characters_to_list(Right))}.
```



How to use AoC to learn Erlang and/or Elixir?

- During December:
 - Set a time limit
 - Try to do both tasks for the day
 - If you can, read others' solutions (on Elixir Forum or Erlang Forums)
 - If not, then either give up or try to complete it another day

How to use Exercism to learn Erlang and/or Elixir?

- Join the two tracks
- Ask for mentorship whenever possible
 - You can do that with the link
- Look at other people's solutions
- Share your solution on the platform

Final Tips

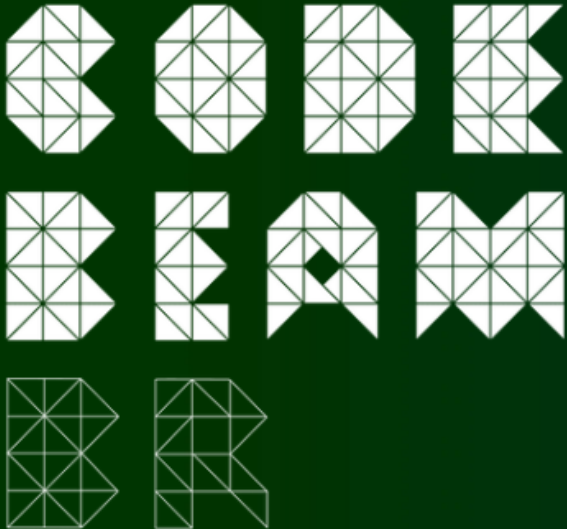
- "Sharpen the Saw" (Habit 7 - Sharpen the Saw)
- Exercise to stay in shape

Announcements

Code BEAM Brasil 2022

Probably in September. Online.

<https://www.codebeambr.com/>



O PRINCIPAL EVENTO DO ECOSSISTEMA
ERLANG ESTÁ DE VOLTA AO BRASIL

O país de origem do Elixir se prepara para a segunda edição da Code BEAM BR.

Devido à situação da COVID 19 no país, o evento será 100% online mais uma vez, possibilitando a criação de um espaço seguro e mais acessível, com a presença de pessoas de diferentes cidades do Brasil e do mundo.

Nos dias 5 e 6 de agosto de 2021, voltaremos a reunir os principais nomes da comunidade Elixir e Erlang, com a maior parte das palestras em português.

A nossa venda de ingressos já está aberta! Fiquem de olho nas nossas redes sociais. ;)

5 E 6 DE AGOSTO DE 2021
09H - 14H GMT-3

Elixir Brasil 2022

Probably in September. ???

<https://elixirbrasil.com/>



10 Years of Elixir

Learning Elixir and Erlang with Advent of Code and Exercism: advantages and challenges

You can watch the recording!

Laura Castro gave a testimonial!

<https://bit.ly/10YearsOfElixir>



Adolfo Neto @adolfont on Twitter

adolfont.github.io



[Slides](#)