Day 06

Blitzen and Snowball's Unproductive Debate

The Story

The North Pole Dev room was quiet—too quiet. Santa was still away and Blitzen was still in charge for the day, the elves didn't like that, some wished Santa's unhinged management style was back.

Blitzen leaned back in his reindeer chair knowing he's the boss now, sipping his coffee. "Hey, Snowball, did you know a function in Rust can return a reference?"

The Story

Snowball was a junior developer, he didn't know anything about Rust especially references, it seemed to confuse him, he looked up from their keyboard, skeptical. "That's absurd. Functions return values, not references. You can't return a borrowed value, Blitzen. It'll dangle."

"Oh, you poor, naive elf," Blitzen said with a smug grin. "Behold the power of lifetimes!" He started scribbling on the whiteboard.

The Story

"Okay, but why do we even need this?" Snowball asked, raising an eyebrow. "What's the use case?"

"We need to avoid unnecessary re-allocations, Snowball. It's more efficient this way. Remember day 2 when Santa was mad at us for a simple clone on a damn `String` ? It wasn't even that big of a deal!"

"Fine! You're right, Santa hates clones."

The Story

"I challenge you, Snowball. Write a function that returns the longer string without any re-allocation. Trim the strings, compare their lengths, and make sure it doesn't involve cloning or creating new allocations."

The two bickered about ownership, lifetimes, and why Snowball wasn't using Arch Linux for the next hour.

The Story

Now it's your turn. Can you help Snowball write the function and put Blitzen in his place? Show that junior developers can handle lifetimes too! Try to finish the function `longer_wish`.

Requirements

- If `s1` is longer than `s2`, return a reference to `s1` otherwise return a reference to `s2` inside a Some variant.
- If both slices have the same length, return `None`.
- Any white spaces in the beginning or end of the string should be trimmed.

Good Luck!

Hints

- Use the `trim()` method to remove any white spaces from the beginning or end of the string. e.g. `s1.trim()`.

- Use the `chars()` method to get the UTF-8 characters in the string slice. e.g. `s1.chars()`.

Hints

- Use the `count()` method to get the number of chracters in the string slice. e.g. `s1.chars().count()`.
- To return a `Some` variant, you can use the `Some(s1)` syntax.