

**Day 02 - Memory doesn't
grow on Christmas trees 🎄**

Before we start

We'd like to clarify our intention behind [Day 1 of Advent of Rust](#) on [Rustfinitly](#). The challenge involved cloning a string to use it twice, which some people understandably judged too quickly and labeled as "teaching bad practice." We understand why it might have been seen this way, but that was not our goal.

Before we start

The decision to use cloning was intentional, meant for educational purposes and to align with the story's style.

We apologize for any misunderstanding this may have caused. In today's challenge, we'll revisit and demonstrate a better approach, showcasing more optimal Rust practices.

Thank you for your patience and understanding as the story continues to unfold!

Enjoy!

The Story

Santa stormed into the workshop like a grizzly bear that just got paged for an on-call issue at 3 AM. Santa's face twisted into a mix of frustration and disbelief—a dire sign that even his usually jolly demeanor couldn't mask the disappointment.

"Who wrote this monstrosity?!" Santa boomed, holding a printed stack of code like it was his naughty list. "Do you think memory grows on Christmas trees? CLONE?!"

Another allocation for the same string?!"

The Story

The room fell silent. The elves exchanged nervous glances; it seemed like the code you wrote yesterday wasn't up to Santa's standards.

"But Santa," one of the elves began, "we thought cloning was safe! No borrow-checker drama—"

"Safe? Sure. Efficient? NO!" Santa interrupted. "We need to use **references**! Borrow the data, don't hog it! Everything has its own place, and using `.clone()` here is not! This is Rust! Memory efficiency is the whole point!"

The Story

An elfe raised a hand meekly. "But Santa, in JavaScript, you just put it in and it works!"

Santa sighed, pinching the bridge of his nose. "Listen, bros, here's the deal: get this code refactored using references. If I see one more `.clone()` without a good reason, I'm switching to Zig. Now, fix it before I start yapping on stream about incompetent elves!"

The Story

The elves exchanged panicked glances, knowing that if they messed up, Blitzen would be called in to pair-program and nobody wanted that because Blitzen wouldn't stop talking about his neovim macros.

The elves gulped. It was time to work with something that they had never seen before: borrowing and references.

Can you help them fix the code?

Your Task

- Update the `attach_message_to_present` function to accept a reference to a `String` or a `string slice` `str` instead of an owned `String`.
- Update the `main` function to pass a reference to the `gift_message` string instead of cloning it.