

[Skip to main content](#)

[Stack Overflow](#)

1. [About](#)

2. [Products](#)

3. [For Teams](#)

1. [Stack Internal](#) Implement a knowledge platform layer to power your enterprise and AI tools.

2. [Stack Data Licensing](#) Get access to top-class technical expertise with trusted & attributed content.

3. [Stack Ads](#) Connect your brand to the world's most trusted technologist communities.

4. [Releases](#) Keep up-to-date on features we add to Stack Overflow and Stack Internal.

5. [About the company](#) [Visit the blog](#)

Loading...

[current community](#)

[Stack Overflow](#)

[help](#) [chat](#)

- [Meta Stack Overflow](#)

your communities

[Sign up](#) or [log in](#) to customize your list.

[more stack exchange communities](#)

[company blog](#)

3. [Log in](#)

4. [Sign up](#)

[AI Assist is now on Stack Overflow](#). Start a chat to get instant answers from across the network. Sign up to save and share your chats.

[Home](#)

[Questions](#)

[AI Assist](#)

[Tags](#)

[Challenges](#)

7. [Chat](#)

[Articles](#)

[Users](#)

[Companies](#)

[Collectives](#)

Communities for your favorite technologies. [Explore all Collectives](#)

Stack Internal

Stack Overflow for Teams is now called **Stack Internal**. Bring the best of human thought and AI automation together at your work.

[Try for free](#) [Learn more](#)

[Stack Internal](#)

Bring the best of human thought and AI automation together at your work. [Learn more](#)

Collectives™ on Stack Overflow

Find centralized, trusted content and collaborate around the technologies you use most.

[Learn more about Collectives](#)

Stack Internal

Knowledge at work

Bring the best of human thought and AI automation together at your work.

[Explore Stack Internal](#)

[Using for loops to generate `&'static str` from an array](#)

[Ask Question](#)

Asked 2 years ago

Modified [2 years ago](#)

Viewed 111 times

1

I need to know if this is possible in Rust (and if not, how come). I have a function `combined_words` in a trait called `Parent`. The trait contains an associated constant for `SIZE: usize` and `WORDS: [&'static str; SIZE]`.

I want the function to use both known sized associated constants to generate a `&'static str` at compile time.

I'm not sure if this is possible, if so, is there something I'm missing, is there a need to use a macro. Is there a library that can do this.

```
trait Parent {
    const SIZE: usize;

    const WORDS: [&'static str; <X as Parent>::SIZE];

    // this is the function I want to implement. It's just pseudocode for now
    fn combined_words() -> &'static str {
        for i in 0..X::SIZE {
            // concat the words into a single &'static str e.g.
            format!("{}", X::WORDS[i as usize], i);
        }

        // and return it
    }
}

struct X;

impl Parent for X {
    const SIZE: usize = 3;
    const WORDS: [&'static str; <X as Parent>::SIZE] = ["hi", "there", "friend"];
}

fn main() {
    let x: X = X{};
    assert_eq!(X::combined_words(), "hi1there2friend3");
}
```

• [rust](#)

[Share](#)

[Improve this question](#)

Follow

[edited Nov 17, 2023 at 8:00](#)

[Uwe Keim](#)

40.9k6161 gold badges193193 silver badges309309 bronze badges

asked Nov 17, 2023 at 7:40

[martinomburajr](#)

1,27533 gold badges1414 silver badges3131 bronze badges

5

This can't even work currently in Rust for a different reason: [you can't use `Self::SIZE` as the size of a `const` trait member](#).

cdhowie – [cdhowie](#)

2023-11-17 08:06:42 +00:00

Commented Nov 17, 2023 at 8:06

However you can make [Parent trait generic over `const SIZE: usize`](#).

Aleksander Krauze – [Aleksander Krauze](#)

2023-11-17 08:29:07 +00:00

Commented Nov 17, 2023 at 8:29

@AleksanderKrauze that's semantically different, you can implement that trait multiple times with different `SIZES` for the same struct which you couldn't do for OP's variant if it was valid.

cafce25 – [cafce25](#)

2023-11-17 08:31:52 +00:00

Commented Nov 17, 2023 at 8:31

Can you use nightly?

Chayim Friedman – [Chayim Friedman](#)

2023-11-18 18:51:00 +00:00

Commented Nov 18, 2023 at 18:51

Even if this is possible, this will require extreme type metaprogramming. Do you really need it? Why cannot you afford allocating a little `String`?

Chayim Friedman – [Chayim Friedman](#)

2023-11-18 18:53:35 +00:00

Commented Nov 18, 2023 at 18:53

[Add a comment](#) | .

1 Answer

Sorted by: [Reset to default](#)

Highest score (default) Trending (recent votes count more) Date modified (newest first) Date created (oldest first)

-1

No, there can not be one.

`str` is a string slice, i.e. a string located in contiguous memory.

Using `format` (or anything else like that) has to manipulate the string and as such it can not return a reference (there is no place in memory where the concatenated string *already exists* contiguously).

The closest thing you could do is to try the opposite approach, i.e. storing the string in one piece along with the indices of the word boundaries. You could then create a function that gets the *n*th word, just like indexing an array; and getting the entire string is trivial:

```
let string = "Lorem ipsum dolor sit amet";
let indices = [0, 6, 12, 18, 22];

fn get_word<'a>(s: &'a str, indices: &[usize], index: usize) -> &'a str {
    &s[
        indices[index] .. indices.get(index+1).copied().unwrap_or(s.len())
    ]
}

get_word(string, &indices, 3) // &'static "sit "
string // &'static "Lorem ipsum dolor sit amet"
```

Or, you could just go the easy way and use `Strings`.

[Share](#)

[Improve this answer](#)

Follow

answered Nov 17, 2023 at 8:47

[FZs](#)

18.8k1313 gold badges4747 silver badges5656 bronze badges

Sign up to request clarification or add additional context in comments.

Comments

Add a comment

Your Answer

Thanks for contributing an answer to Stack Overflow!

- Please be sure to *answer the question*. Provide details and share your research!

But avoid ...

- Asking for help, clarification, or responding to other answers.
- Making statements based on opinion; back them up with references or personal experience.

To learn more, see our [tips on writing great answers](#).

Sign up or [log in](#)

Sign up using Google

Sign up using Email and Password

Submit

Post as a guest

Name

Email

Required, but never shown

Post as a guest

Name

Email

Required, but never shown

Post Your Answer Discard

By clicking "Post Your Answer", you agree to our [terms of service](#) and acknowledge you have read our [privacy policy](#).

Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

Explore related questions

- [rust](#)

See similar questions with these tags.

- The Overflow Blog
 - [Introducing Stack Overflow AI Assist—a tool for the modern developer](#)
 - [Treating your agents like microservices](#)
- Featured on Meta
 - [Chat room owners can now establish room guidelines](#)
 - [AI Assist is now available on Stack Overflow](#)
 - [Policy: Generative AI \(e.g., ChatGPT\) is banned](#)

Related

- [5](#)
[Is it possible to select the smallest integer type based on a const generic value in Rust?](#)
- [0](#)
[Access impl const without type parameters](#)
- [8](#)
[Using a static integer in the definition of a struct](#)
- [2](#)

[How do I construct static variables with a for loop and store them in a static array?](#)

[1](#)

[BorshDeserialize gives try_from_slice error](#)

[209](#)

[How to convert a String into a &static str](#)

[2](#)

[How to assert that a struct's associated constant satisfies a condition?](#)

[0](#)

[How to return a box to self?](#)

[6](#)

[How to return an anonymous type from a trait method without using Box?](#)

[Hot Network Questions](#)

- [Why does Blender import PLY vertex colors incorrectly \(89 → 0.1 instead of 0.349\)?](#)
- [How do I add a text editor to my \(debian-based\) initramfs?](#)
- [How to prevent a batch file call from escaping a caret character?](#)
- [How to handle Expeditious Retreat with the optional chase rules](#)
- [How many alien worlds could exist within 100 ly without us noticing?](#)
- [Curve has weird bumps when triangulated](#)
- [Displaying features in a layer of geometrytype = GeometryCollection25D](#)
- [What is the difference between an ultraviolet limit and ultraviolet stability?](#)
- [How to use Bézier curves in `TikZ` to draw a quotation mark?](#)
- [Explain this seeming contradiction in Euclid Book 1 Proposition 16](#)
- [PSE Advent Calendar 2025 \(Day 3\): A gift from my travel agent](#)
- ["Creatures your opponents control" after the ability is activated](#)
- [Does short range teleporting end floating disk?](#)
- [Ping sweep of IP subnets](#)
- [Schottky OR-ing diodes passing reverse voltage; is it a faulty diode?](#)
- [A hilarious parody of Western movies where an entire town collapses into tunnels](#)
- [Statistical testing for bimodal/multimodal sample](#)
- [What do the terms postponement, slip and scrub mean in the context of the space shuttle program?](#)
- [Preservation of universal sentences under substructures in higher-order logic](#)
- [Could a world tree have different biomes?](#)
- [Zero Inflated Beta Regression \(or not\)?](#)
- [The fourth moment of the Riemann zeta function without absolute values](#)
- [Why does black allow white to have a protected passed central pawn here?](#)
- [Can I Automatically List Supported Kernel Parameters in GRUB?](#)

[more hot questions](#)

[Question feed](#)

Subscribe to RSS

Question feed

To subscribe to this RSS feed, copy and paste this URL into your RSS reader.

[Stack Overflow](#)

- [Questions](#)
- [Help](#)
- [Chat](#)

[Business](#)

- [Stack Internal](#)
- [Stack Data Licensing](#)
- [Stack Ads](#)

[Company](#)

- [About](#)
- [Press](#)
- [Work Here](#)
- [Legal](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [Contact Us](#)
- [Cookie Settings](#)
- [Cookie Policy](#)

[Stack Exchange Network](#)

- [Technology](#)
- [Culture & recreation](#)
- [Life & arts](#)
- [Science](#)
- [Professional](#)
- [Business](#)
- [API](#)
- [Data](#)
- [Blog](#)
- [Facebook](#)
- [Twitter](#)
- [LinkedIn](#)
- [Instagram](#)

Site design / logo © 2025 Stack Exchange Inc; user contributions licensed under [CC BY-SA](#) . rev 2025.12.4.37651