

[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](#)

[How to use a String instead of &str in iterator, if/else block](#)

[Ask Question](#)

Asked 10 years, 3 months ago

Modified [10 years, 3 months ago](#)

Viewed 223 times

3

Would someone be kind enough to explain to me why using a `String` in this script does not work but `&str` does. Additionally how can I modify it so that it will work with `String`'s? [version 1.2]

```
use std::collections::{HashMap};

fn main() {

    let mut hash = HashMap:::<&str, &str>::new();
    hash.insert("this", "value");
    let l: &str = "this is a borrowed string reference";
    // If the above line was defined as:
    //let l: String = "this is a string".to_string();

    let mut all = l.split(" ");
    let name: &str = all.next().unwrap();

    if hash.contains_key(name) == true {
        hash.remove(name);
    } else {
        hash.insert(name, "stuff");
    }
}
```

- [rust](#)

[Share](#)

[Improve this question](#)

Follow

[edited Aug 20, 2015 at 13:58](#)

asked Aug 20, 2015 at 13:44

[kezzos](#)

3,24133 gold badges2525 silver badges4040 bronze badges

1

3

`String` *where*? There's four different places `&str` could be replaced with `String`, all with different reasons why it may or may not work. Do you mean *just* changing the type of `l`?

DK. – [DK.](#)

2015-08-20 13:53:12 +00:00

Commented Aug 20, 2015 at 13:53

[Add a comment](#) | .

3 Answers

Sorted by: [Reset to default](#)

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

5

Ok, let's boil this down to just the necessities:

```
use std::collections::HashMap;

fn main() {
    let mut hash = HashMap:::<&str, &str>::new();
    hash.insert("this", "value");
    let l: String = "this is a borrowed string reference".to_string();
    hash.insert(&l, "stuff");
}
```

Compiling this gives us:

```

<anon>:7:18: 7:19 error: `l` does not live long enough
<anon>:7      hash.insert(&l, "stuff");
              ^

<anon>:4:49: 8:2 note: reference must be valid for the block suffix following statement 0 at 4:48...
<anon>:4      let mut hash = HashMap::<&str, &str>::new();
<anon>:5      hash.insert("this", "value");
<anon>:6      let l: String = "this is a borrowed string reference".to_string();
<anon>:7      hash.insert(&l, "stuff");
<anon>:8  }

<anon>:6:71: 8:2 note: ...but borrowed value is only valid for the block suffix following statement 2 at 6:70
<anon>:6      let l: String = "this is a borrowed string reference".to_string();
<anon>:7      hash.insert(&l, "stuff");
<anon>:8  }

```

Which tells you more or less *exactly* what it won't work. You try to insert a borrowed pointer to the `String l` into the hashmap. However, the `String` simply doesn't live long enough.

Specifically, values are destroyed by Rust in reverse lexical order. So, when execution gets to the end of the function, Rust will deallocate `l` *first*, followed by `hash`. This is a problem: it means that there is a window during which `hash` contains pointers to destroyed data, which Rust absolutely *will not allow*.

The reason this works with a `&str` is that a string literal "like this" is not merely `&str`; it's actually `&'static str`. This means that a string literal "lives" for the entire duration of the program: it *never* gets destroyed, thus it's safe for the hashmap to hold a pointer to it.

The solution is to ensure the `String` will outlive the `HashMap`:

```

use std::collections::HashMap;

fn main() {
    // Declare `l` here ...
    let l;
    let mut hash = HashMap::<&str, &str>::new();
    hash.insert("this", "value");

    // ... but initialise it here.
    l = "this is a borrowed string reference".to_string();
    hash.insert(&l, "stuff");
}

```

Now, `hash` gets destroyed *first*, followed by `l`. It's fine to leave a variable uninitialised, so long as you *do* initialise it prior to reading or using it.

[Share](#)

[Improve this answer](#)

Follow

answered Aug 20, 2015 at 14:06

[DK.](#)

59.7k77 gold badges210210 silver badges171171 bronze badges

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

3 Comments

Add a comment

kezzos

[kezzos Over a year ago](#)

Thanks I was really stumped there for a while, that makes it very clear!

2015-08-20T14:27:27.963Z+00:00

0

Reply

- Copy link

flogram_dev

[flogram_dev Over a year ago](#)

"values are destroyed by Rust in reverse lexical order" – Umm no, they're destroyed in reverse order of *declaration*: [is.gd/LEBmOi](#)

2015-08-21T14:04:11.057Z+00:00

0

Reply

- Copy link

DK.

[DK, Over a year ago](#)

"lexical" as in the order they appear lexically; *i.e.* in the text.

2015-08-21T17:35:57.29Z+00:00

0

Reply

- Copy link

Add a comment

2

If you change `l` to `String` you get:

```
error: l does not live long enough
```

Which is true:

`name` is a reference to one of the substrings of `l`. You're possibly inserting the `name` reference to `hash`, but the lifetime of `hash` is longer than the lifetime of `l`. Therefore when the lifetime of `l` ends, `hash` would contain invalid references. This is not allowed. F.ex. if you remove the `insert` line, Rust is happy.

There are multiple ways to fix it, depending on your needs. One of them would be to make the lifetime of `hash` shorter than the lifetime of `l`, by instantiating `hash` after `l`:

```
use std::collections::HashMap;

fn main() {
    // let l: &str = "this is a borrowed string reference";
    // If the above line was defined as:
    let l: String = "this is a string".to_string();

    let mut hash = HashMap::new();
    hash.insert("this", "value");

    let mut all = l.split(" ");
    let name: &str = all.next().unwrap();

    if hash.contains_key(name) == true {
        hash.remove(name);
    } else {
        hash.insert(name, "stuff");
    }
}
```

Or, you could store copies of the strings in the map.

[Share](#)

[Improve this answer](#)

Follow

[edited Aug 20, 2015 at 14:22](#)

answered Aug 20, 2015 at 14:07

[flogram_dev](#)

43k1111 gold badges109109 silver badges161161 bronze badges

Comments

Add a comment

1

You have a lifetime issue.

Any object inserted into `hash` must only reference things that outlive `hash` (or not reference anything at all).

However, here, you define `l` after `hash` and therefore `l` has a shorter lifetime. In turn this means that `name` which references `l`'s buffer has a shorter lifetime than `hash` and thus `name` is not suitable to be inserted into `hash` (though it can be used to search/remove).

Switching the order in which `l` and `hash` are defined make it work:

```
use std::collections::HashMap;

fn main() {
```

```
let l: String = "this is a string".to_string();

let mut hash = HashMap::(&str, &str)::new();
hash.insert("this", "value");

let mut all = l.split(" ");
let name: &str = all.next().unwrap();

if hash.contains_key(name) == true {
    hash.remove(name);
} else {
    hash.insert(name, "stuff");
}
}
```

If this is not possible, then use a `HashMap<String, String>` to avoid lifetime issues.

[Share](#)

[Improve this answer](#)

Follow

answered Aug 20, 2015 at 14:08

[Matthieu M.](#)

303k5858 gold badges495495 silver badges754754 bronze badges

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](#)

Linked

[8](#)

[How do I transfer ownership of strings?](#)

Related

[0](#)

[Unable to coerce &String to &str](#)

[12](#)

[Why can't I use `&Iterator<Item = &String>` as an iterator?](#)

[65](#)

[How to accept &str, String and &String in a single function?](#)

[7](#)

[How to pass Iterator<String> as Iterator<&str>?](#)

[1](#)

[How to index an &str and push the chosen element to String?](#)

[2](#)

[Getting an Iterator<Item=str> from an array of &str?](#)

[2](#)

[How to return an &str when strings are dynamically generated?](#)

[0](#)

[error\[E0277\]: can't compare `str` with `&str`—what's the right way to do that?](#)

[1](#)

[Concatenate a slice of `&str`s into an owned `String`](#)

[2](#)

[How to succinctly convert an iterator over &str to a collection of String](#)

Hot Network Questions

- [PSE Advent Calendar 2025 \(Day 5\): Cinnamon overdose](#)
- [Is Brian Ripley's Pattern Recognition and NN book still an insightful read?](#)

- ["Us" in Hegel's "Greater Logic"](#)
- [Is it possible to use LaTeX to create RPG-related PDFs, such as character sheets or adventure modules?](#)
- [Storm Sphere and Obstruction](#)
- [When should I use tied notes vs. the whole note duration?](#)
- [Recent time traveling TV show. The lead might have been Asian\(?\), set in San Francisco\(?\)](#)
- [When the fuel governor fails, why is autorotation the only option for landing?](#)
- [Is there any point in sharing a scientific discovery as a "nobody"?](#)
- [How to check if two tables are identical?](#)
- [How can I determine whether a complex rational equation has real solutions?](#)
- [Displaying features in a layer of geometrytype = GeometryCollection25D](#)
- [How do Buddhists interpret the Buddha's explanation of earthquakes in AN 8.70?](#)
- [What does Felix mean by telling Bond to pick a contact point while "standing up" in Diamonds Are Forever?](#)
- [Polygons \(and some stars\)](#)
- [Schottky OR-ing diodes passing reverse voltage; is it a faulty diode?](#)
- [Is there a term for forming emotional or relational bonds with AI chatbots? If not, is "cyberpomorphic" a valid neologism?](#)
- ["Imagine if..." vs. "Imagine..."](#)
- [Correct website for Thailand ETA application form](#)
- [How do I add a text editor to my \(debian-based\) initramfs?](#)
- [Do something and "win"](#)
- [I can't add a loop cut on one face](#)
- [Ping sweep of IP subnets](#)
- [2 ways of resolution of vectors](#)

[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