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

Home
Questions
AI Assist
Tags
Challenges
7. Chat
Articles
Users
Companies
Collectives

Communities for your favorite technologies. Explore all Collectives

Why does a for loop not require a mutable iterator?

Ask Question

Asked 10 years, 6 months ago

Modified 7 years, 7 months ago

Viewed 905 times

16

If I want to consume an iterator by hand, it has to be mutable:

```
let test = vec![1, 2, 3];
let mut test_mut = test.iter();
while let Some(val) = test_mut.next() {
   println!("{:?}", val);
}
```

But I can happily consume it with a `for` loop, even if it's immutable.

```
let test = vec![1, 2, 3];
let test_imm = test.iter();
for val in test_imm {
   println!("{:?}", val);
}
```

I *think* this works because `test_imm` is moved into the for loop's block, so `test_imm` can't be used by the outer block any more and is (from the point of view of the outer block) immutable up until the for loop, and then it's inaccessible, so it's okay.

Is that right? Is there more to be explained?

- rust

Share

Improve this question

Follow

edited Apr 29, 2018 at 17:30

Shepmaster

439k116116 gold badges1.3k1.3k silver badges1.5k1.5k bronze badges

asked May 20, 2015 at 20:59

Roy

31511 silver badge66 bronze badges

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)

14

That's exactly right. Since it's moved to the for loop, the for loop now owns it and can do whatever it wants with it, including "making it" mutable. Consider this analogous example, where we appear to be mutating `xs` despite it being immutable, but really it's because we're moving it, so the new owner is free to do with it whatever it wants, including re-binding it as mutable:

```
let xs: Vec<i32> = vec![1, 2, 3];

fn append(v: Vec<i32>, x: i32) -> Vec<i32> {
   let mut my_v = v;
   my_v.push(x);
   my_v
}

let appended = append(xs, 4);
```

playground

Note that the function can be made shorter using the `mut` parameter convenience syntax:

```
fn append(mut v: Vec<i32>, x: i32) -> Vec<i32> {
   v.push(x);
   v
}
```

This is more or less explained in the `iter` module's documentation.

Share

Improve this answer

Follow

edited Apr 29, 2018 at 17:37

Shepmaster

439k116116 gold badges1.3k1.3k silver badges1.5k1.5k bronze badges

answered May 20, 2015 at 21:45

Jorge Israel Peña

39k1616 gold badges9696 silver badges126126 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

Linked

3
Why can I loop over an immutable iterator with for construct?
1
Why does a for loop not require the iterator to be mutable?

Related

9
Why are iterators not copyable?
9
Why does the compiler not complain that an iterator moved to a for loop is immutable?
3
Why does Iterator::all need the iterator to be mutable?
3
Why are mutable values from an iterator created with cycle() not updated, leading to an infinite loop despite a stop condition?
1
Different approach to call a mutable method inside a for iterator
3
Why must a mutable reference be used in the for loop here?
6
Why does adding mut to passed Iterator reference solve this?
1
Mutating elements inside iterator
2
Why does an iterator variable need to be mutable
3
Use mutable iterator twice

Hot Network Questions

- Why is "cmd | xargs" to join lines of cmd with spaces wrong
- Curve has weird bumps when triangulated
- The fourth moment of the Riemann zeta function without absolute values
- Who is Patrick and why is he referred to at anti-AfD demonstrations?
- How to handle Expeditious Retreat with the optional chase rules
- Calculating the Resistance in an Induction Heater
- How do I add a text editor to my (debian-based) initramfs?
- Is there any point in sharing a scientific discovery as a "nobody"?

- Explain this seeming contradiction in Euclid Book 1 Proposition 16
- Is the phrase "murder will out" from Chaucer?
- How many alien worlds could exist within 100 ly without us noticing?
- Zero Inflated Beta Regression (or not)?
- Could a world tree have different biomes?
- "Us" in Hegel's "Greater Logic"
- What may cause any version of Microsoft Outlook to run slowly?
- PSE Advent Calendar 2025 (Day 5): Cinnamon overdose
- is it possible to keep the space when using url in latex
- TV program where real actors interact with 2-Dimensional characters
- Preservation of universal sentences under substructures in higher-order logic
- Why does Mtt 13:34 say that Jesus always spoke in parables?
- How do Buddhists interpret the Buddha's explanation of earthquakes in AN 8.70?
- Can enemies damage each other in Breath of the Wild?
- "Imagine if..." vs. "Imagine..."
- When should I use tied notes vs. the whole note duration?

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