

# WHY RUST?

PEI Developers  
Michael Easter  
Feb. 11, 2021

**About**

@codetojoy

@30\_for\_60

**std::disclaimer**



# THE RUST PROGRAMMING LANGUAGE

STEVE KLABNIK and CAROL NICHOLS,  
with CONTRIBUTIONS from THE RUST COMMUNITY

COVERS  
RUST  
2018

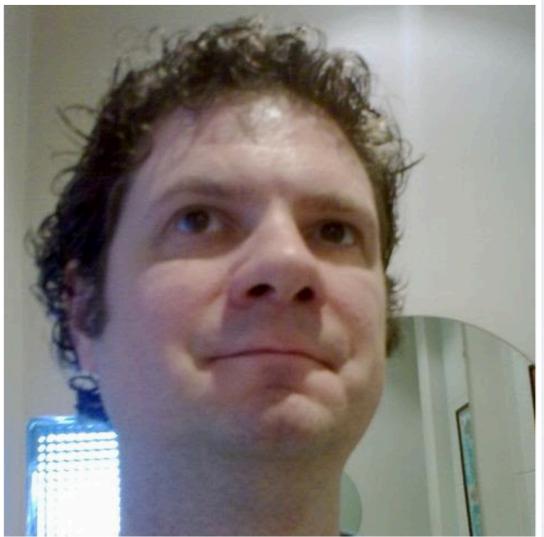


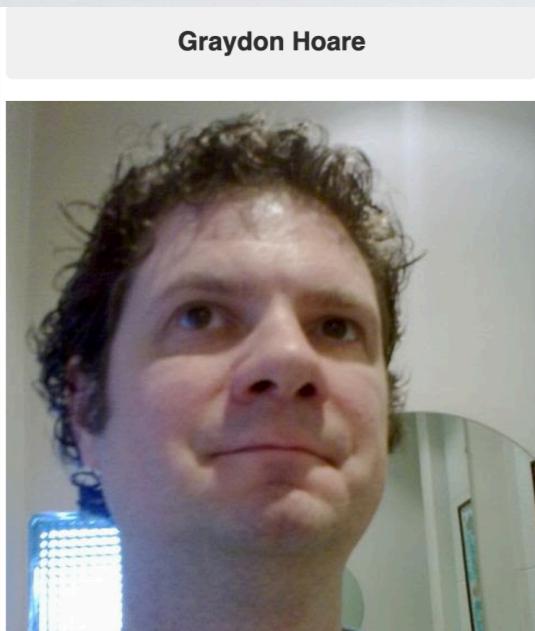
# **Overview**



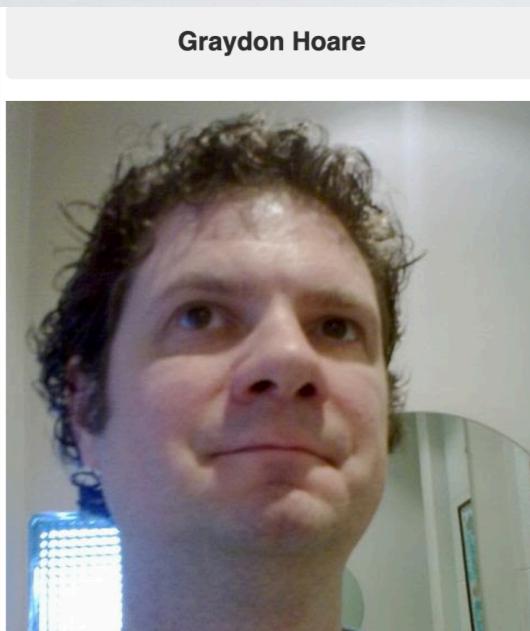


Graydon Hoare

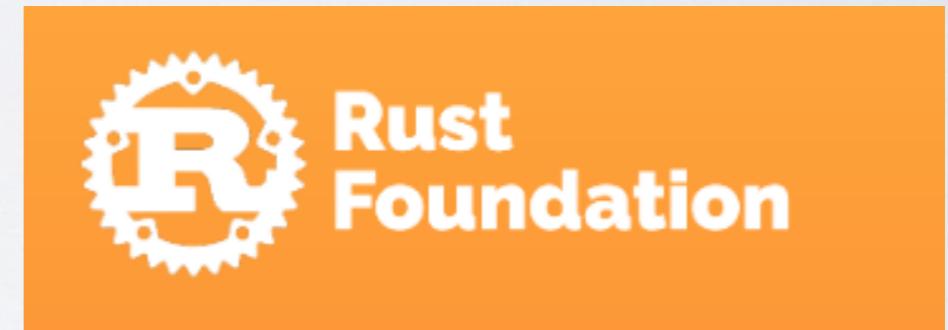


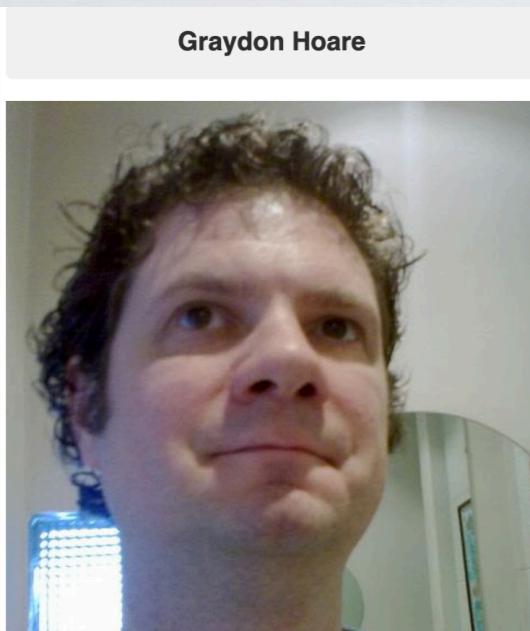


moz://a

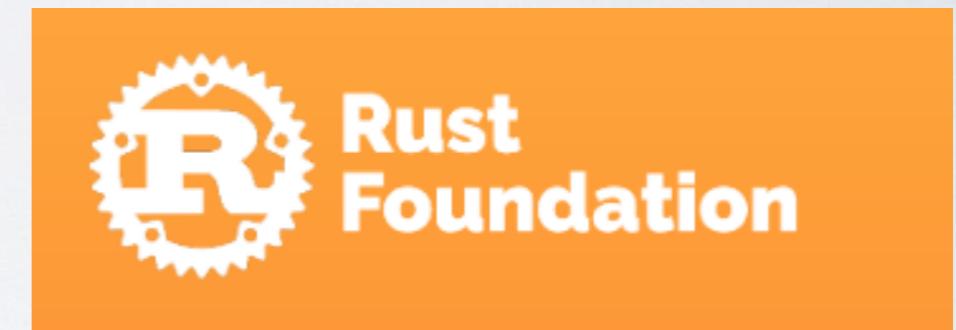


moz://a





moz://a



# Performance

THE  
**C**  
PROGRAMMING  
LANGUAGE



**Reliability**

**moz://a**



**Microsoft**

**System-level**

**Embedded**

**Web Assembly**

**Cloud**

# **Philosophy**

**Static**

**Dynamic**



## No GC



## Garbage Collection



## No GC



## Garbage Collection



**No GC**



c = new

Cat (0xABCD123)

## No GC



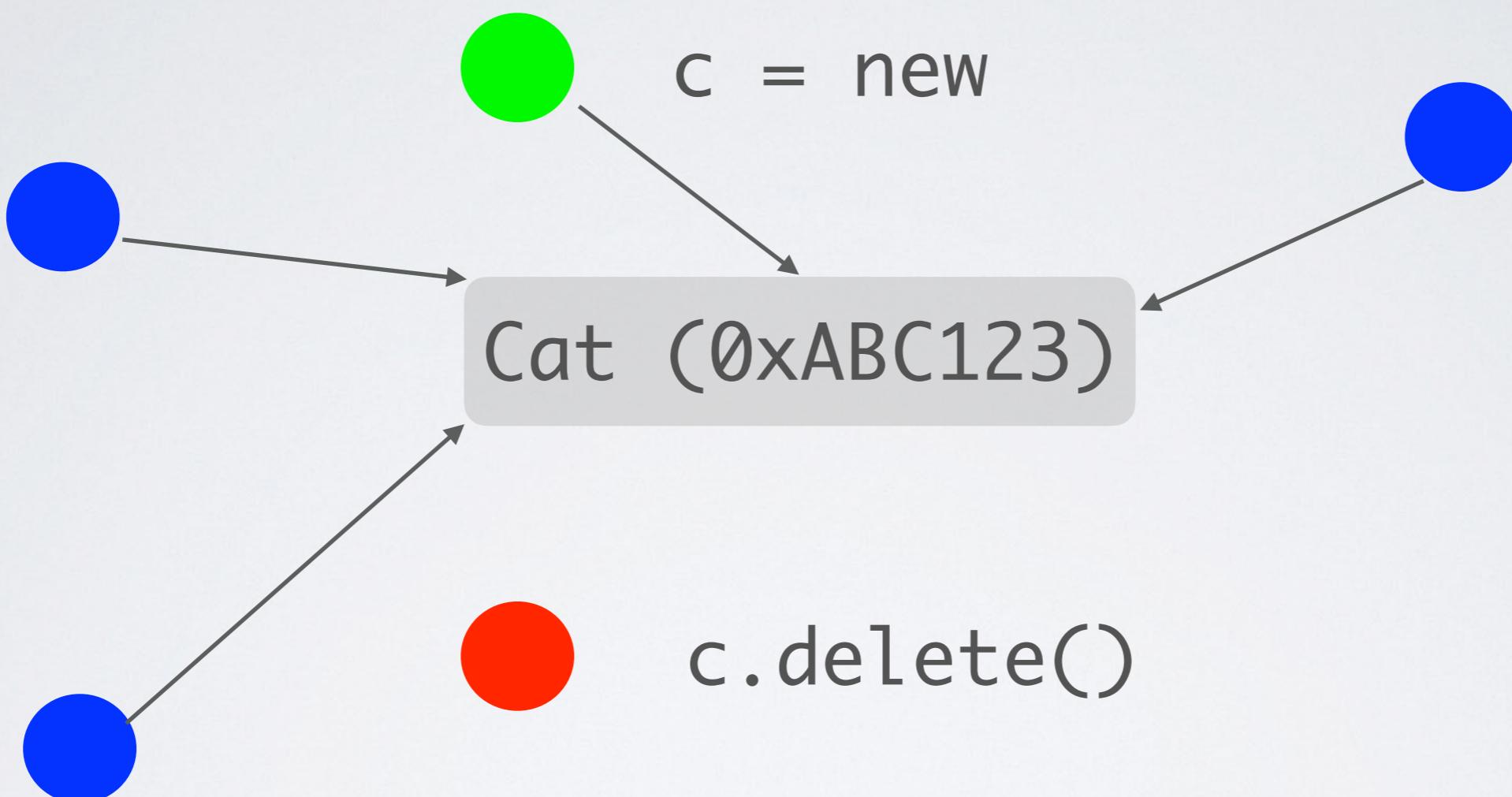
c = new

Cat (0xABCD123)

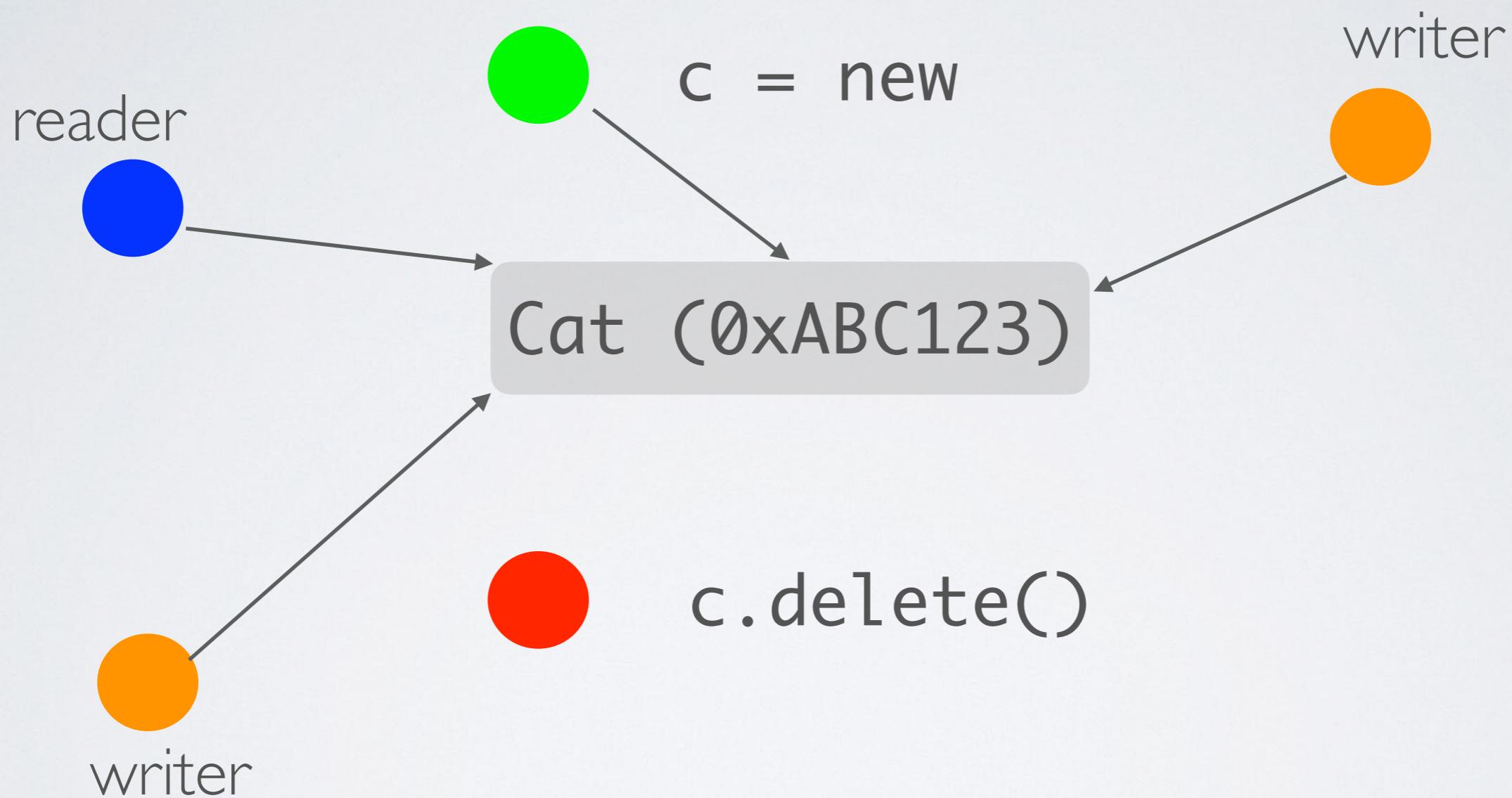


c.delete()

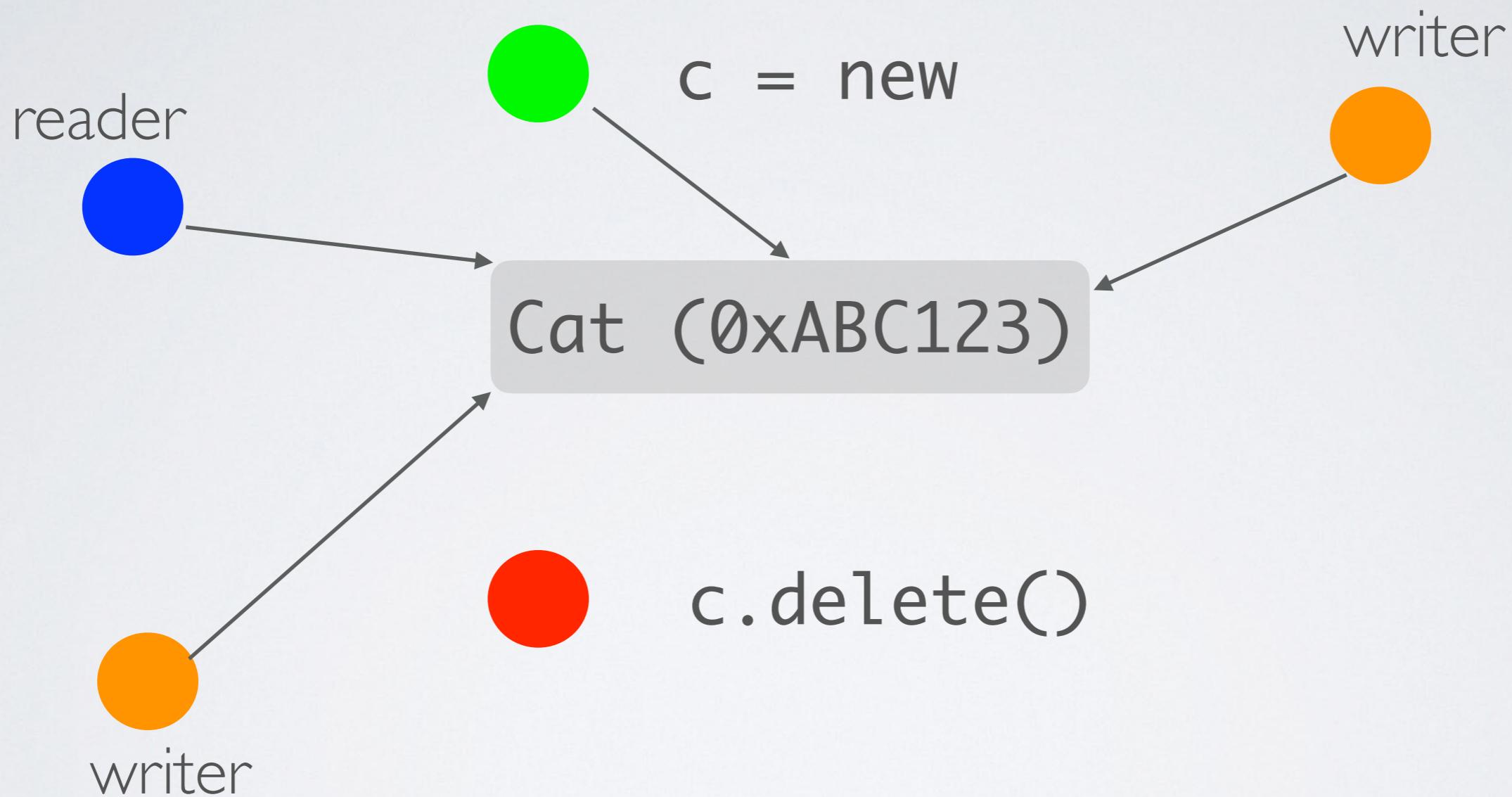
## No GC



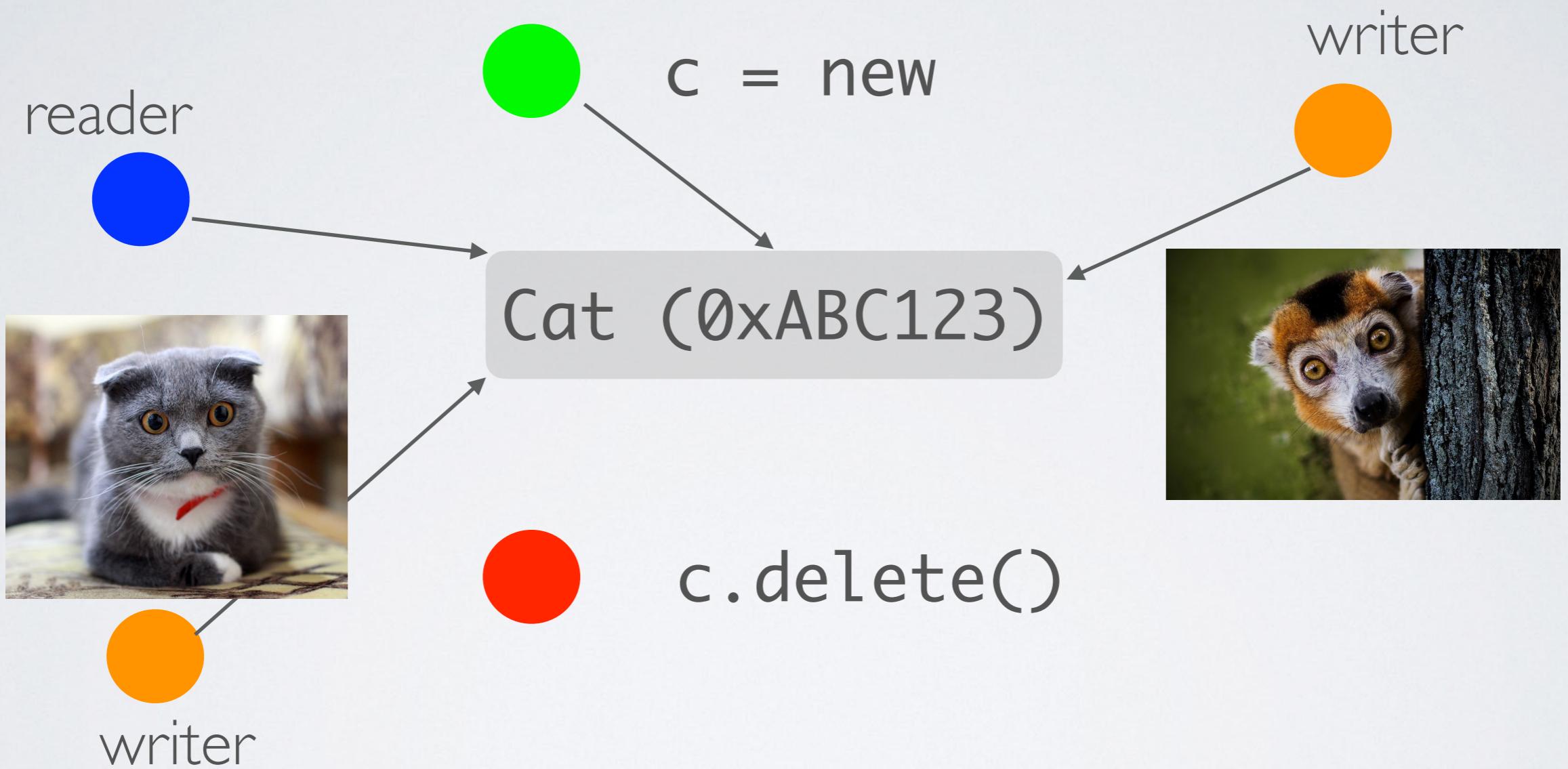
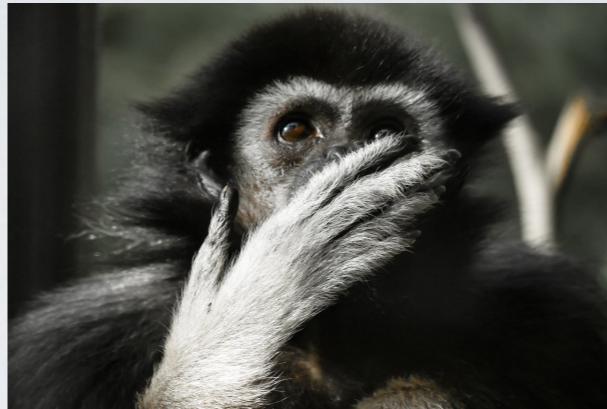
## No GC



## No GC



## No GC



## **Easter's Law of Teams**

A group of N people behaves with a  
collective IQ of C,  
where C approximates ...

## **Easter's Law of Teams**

A group of N people behaves with a collective IQ of C,  
where C approximates ...



## **Easter's Law of Teams**

A group of N people behaves with a collective IQ of C,  
where C approximates ...



## No GC



## Garbage Collection



## No GC



## Garbage Collection



## Ownership



## No GC



## Garbage Collection



## Ownership

**compile time**



**memory safe**

**Borrow Checker**

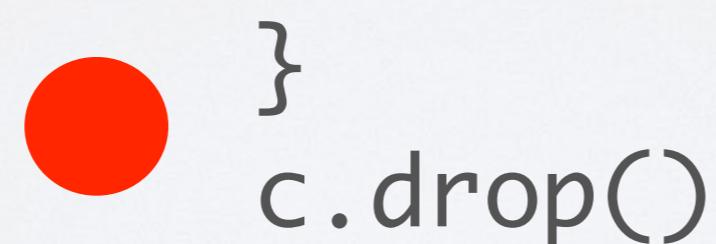
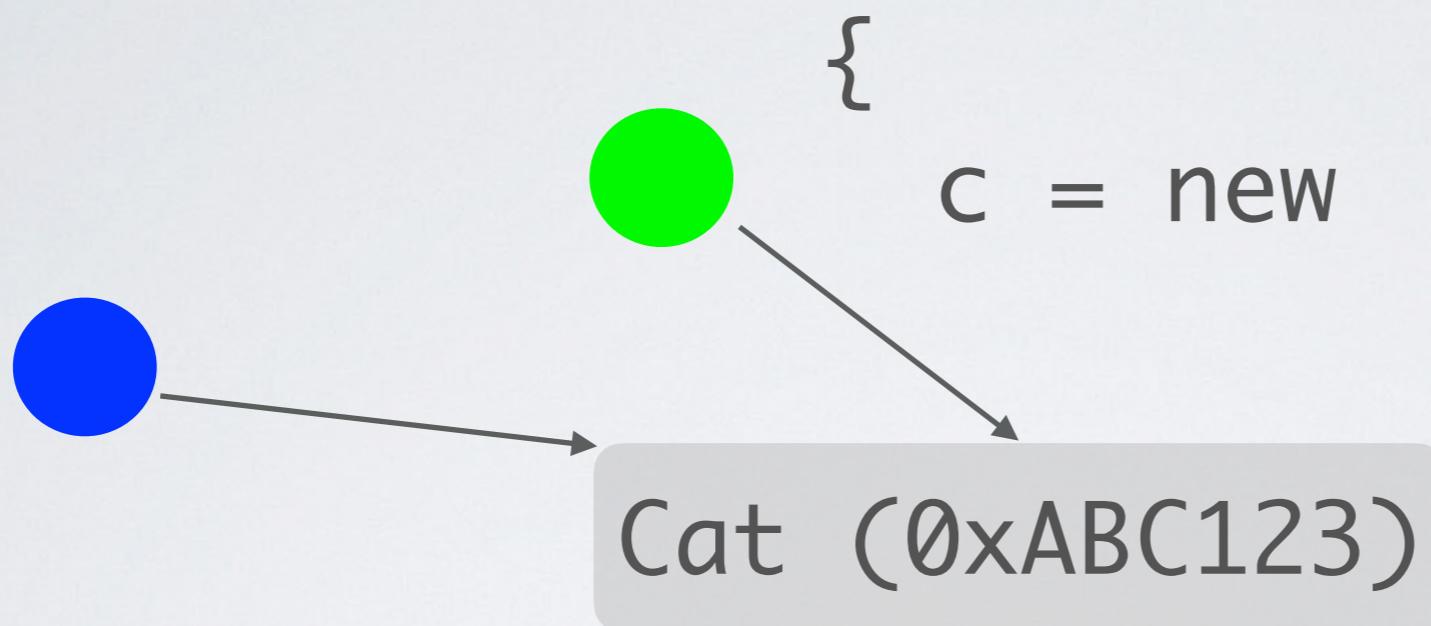
# Rust

```
{  
    c = new
```

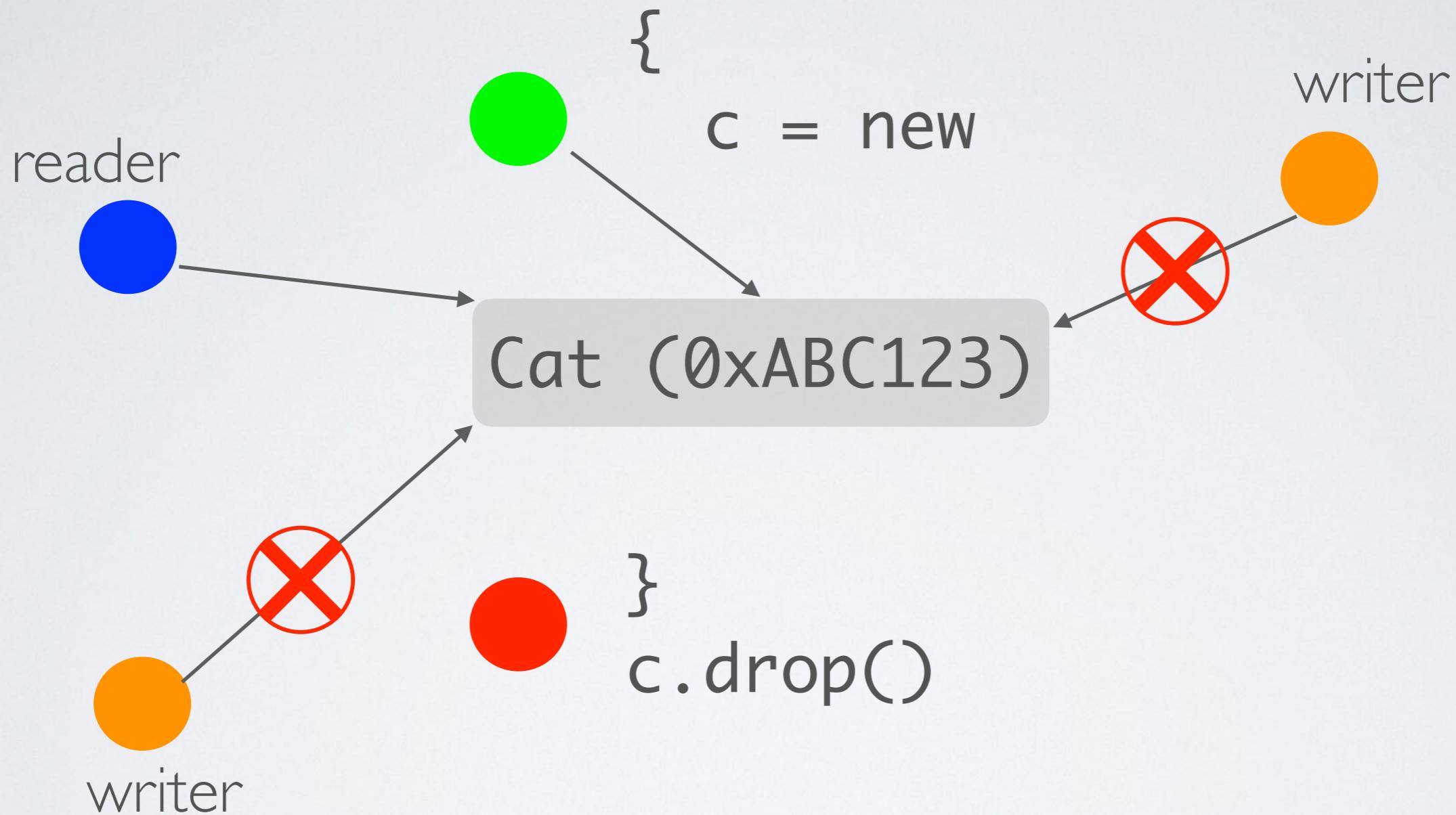
```
Cat(0xABC123)
```

```
}  
c.drop()
```

Rust



# Rust



# Ownership

Each value has an *owner*.

Ownership can be *moved*.

When owner goes out of scope: *dropped*.

# Borrowing

Many *immutable* references.

*Only one* mutable reference.

# Borrow Checker

Pair programmer:  
intense,  
relentless,  
always correct



**Code**

# **Java / Python / Ruby**



# **Java / Python / Ruby**

**C**



# **Rust**

**unsafe**



**Sober Second Thought**

## Pros

Memory safe

Secure

Minimal runtime

Performance

Community

## **Cons**

Learning curve

Borrow Checker

Prototyping

Requirements?





	<b>Build System</b>	<b>Package Manager</b>	<b>Lib</b>	<b>Idiomatic</b>	<b>Fans</b>	<b>Mascot</b>
<b>Rust</b>	Cargo	Cargo	crate	rusty (?)	Rustaceans	Ferris
<b>Java</b>	Gradle	Maven	jar			Duke
<b>Python</b>		pip		pythonic		

# Arbitrary Laundry List

(grotesque anti-pattern)

No null

No inheritance

Types: static, strong, inferred

Traits

Unsafe

# THE RUST PROGRAMMING LANGUAGE

STEVE KLABNIK and CAROL NICHOLS,  
with CONTRIBUTIONS from THE RUST COMMUNITY

COVERS  
RUST  
2018



## No GC



## Garbage Collection



## Ownership

## compile time



## memory safe

## Borrow Checker

شکرا جزیلا

The image features a large, bold red word "thank" positioned above a large, bold red word "you". The entire composition is composed of numerous words from various languages, each representing the phrase "thank you" in its respective tongue. These words are arranged in a dense, overlapping cluster around the central "thank" and "you". The colors of the surrounding words vary widely, creating a vibrant and diverse visual effect. Some of the visible languages include German ("danke"), Chinese ("謝謝"), Spanish ("gracias"), French ("merci"), Italian ("grazie"), Portuguese ("obrigado"), and many others.

# ATTRIBUTION

- Image credits [listed here](#)