



# RUST'S FEARLESS CONCURRENCY — A DEMO



... because Rust is *really* good for creating infrastructure services.



*By leveraging ownership and type checking, many concurrency errors are compile-time errors in Rust rather than runtime errors. [...] We've nicknamed this aspect of Rust **fearless concurrency**.*



# Ownership

- Each value in Rust has a variable that's called its *owner*.
- There can only be one owner at a time.
- When the owner goes out of scope, the value will be dropped.

At any given time, you can have *either* one mutable reference *or* any number of immutable references.



# Expressed through the type system

For example, function declarations:

- take ownership ("move"): **T**
- borrow ("read"): **&T**
- borrow mutably: **&mut T**



# Ownership violation

```
1- fn main() {  
2   let s = String::from("hello");  
3   let t = s;  
4   println!("{}", s, t);  
5 }
```

Execution

Close

Standard Error

```
Compiling playground v0.0.1 (/playground)  
error[E0382]: borrow of moved value: `s`  
--> src/main.rs:4:23  
|  
2 |     let s = String::from("hello");  
|         - move occurs because `s` has type `String`, which does not implement the `Copy` trait  
3 |     let t = s;  
|         - value moved here  
4 |     println!("{}", s, t);  
|                   ^ value borrowed here after move  
  
= note: this error originates in the macro `$crate::format_args_nl` (in Nightly builds, run with -Z macro-backtrace for more info)  
  
For more information about this error, try `rustc --explain E0382`.  
error: could not compile `playground` due to previous error
```



# Borrowing violation

```
1 fn main() {  
2     let mut s = String::from("hello");  
3     let t = &mut s;  
4     println!("{}", s, t);  
5 }
```



Execution

Close

Standard Error

```
Compiling playground v0.0.1 (/playground)  
error[E0502]: cannot borrow `s` as immutable because it is also borrowed as mutable  
--> src/main.rs:4:23  
3 |     let t = &mut s;  
  |           ----- mutable borrow occurs here  
4 |     println!("{}", s, t);  
  |                   ^ - mutable borrow later used here  
  |                   |  
  |                   immutable borrow occurs here  
  
= note: this error originates in the macro `$crate::format_args_nl` (in Nightly builds, run with -Z macro-backtrace for more info)  
  
For more information about this error, try `rustc --explain E0502`.  
error: could not compile `playground` due to previous error
```



## Demo: sum numbers by category

- Numerous files with <category>,<value> lines
- Sum into array where category gives the position
- E.g. ["0,1", "1,1", "0,3"] => [1 + 3, 1] => [4, 1]
- <https://github.com/bittrance/rust-fearless-concurrency-demo>