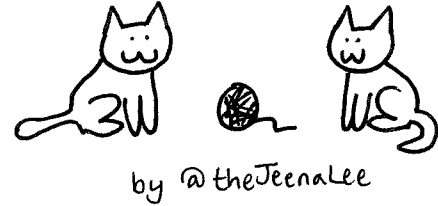


If you only have a reference to a string, but you want the string capabilities, you need to:

- clone the string
- or, take ownership

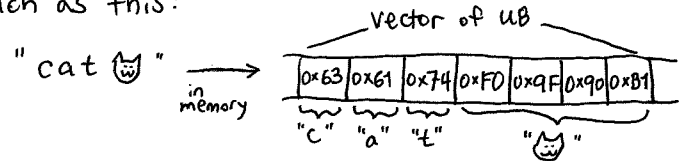


## String and &str in Rust



String is a `Vec<u8>`, a vector of unsigned 8-bit integers.

Such as this!



`&str` is a shared reference to the String!  
The borrower does not own the string.  
`&str` is a wrapper around `&[u8]` ("and slice of u8").

When do you want to use `&str` or `String`?

Typically, you would want to use `&str`, if you can!

But if you can't, use `String`!

Some of the use cases are...

### When to use `&str`

• reading an existing string



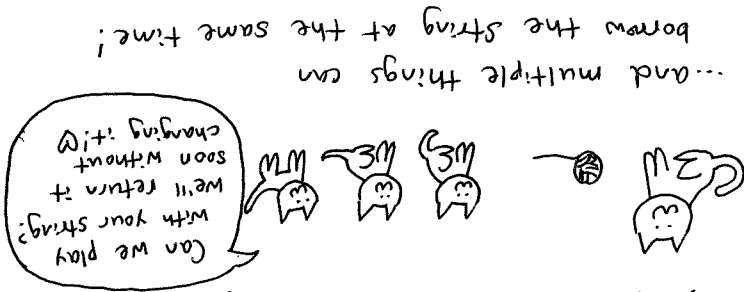
• parsing



• iterating characters



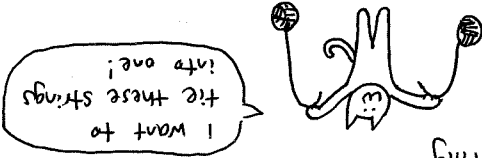
• any operation that is temporary!



...and multiple things can borrow the string at the same time!

### When to use String

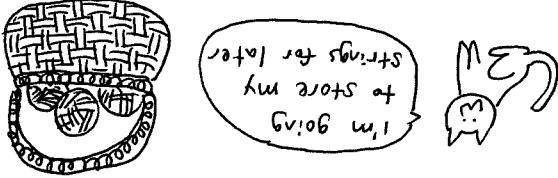
• concatenating



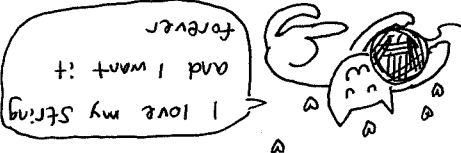
• string-ifying a value (like a number)



• saving owned copies for later



• any operation that is long-term



...there can only be a single owner for a String!