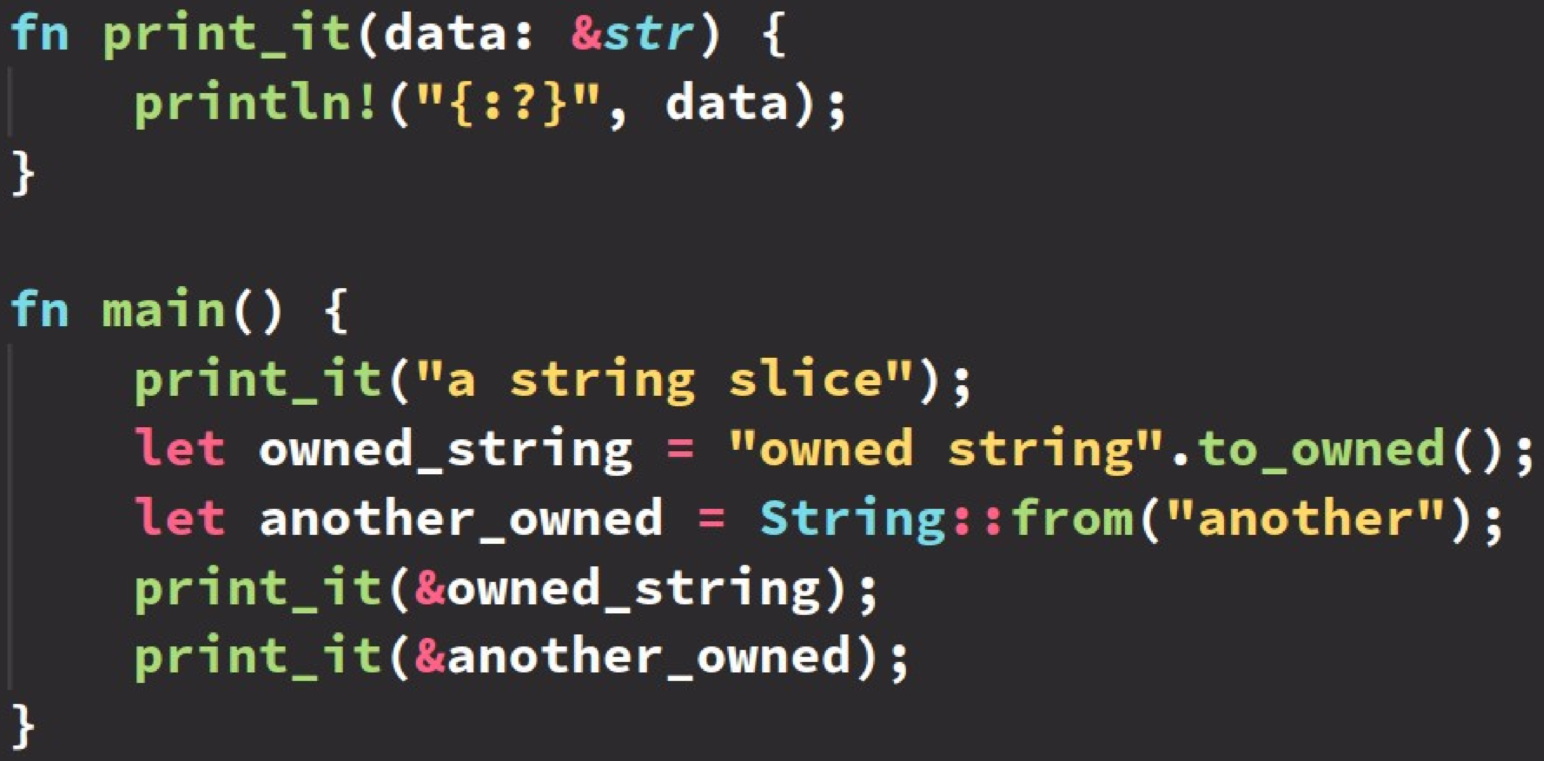
# Data Types | Strings

# String and &str

- Two commonly used types of strings
  - *String* – owned
  - *&str* – borrowed *String* slice
- Must use an owned *String* to store in a *struct*
- Use *&str* when passing to a function

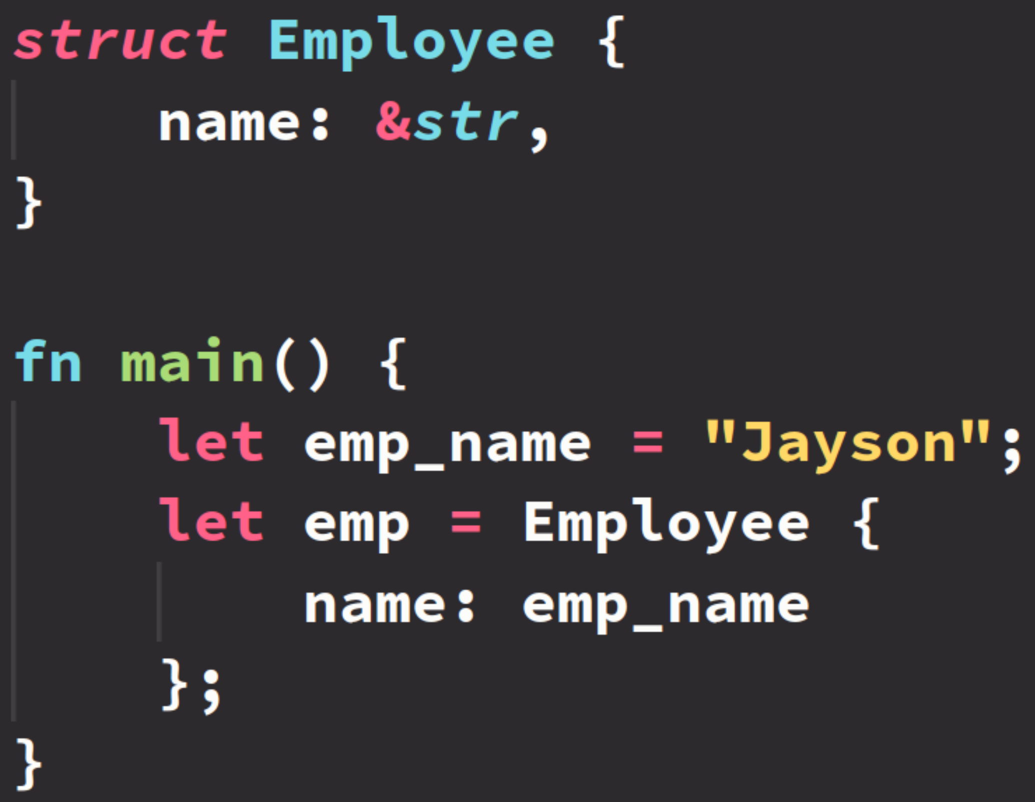
# Example - Pass to function



```rust
fn print_it(data: &str) {
    println!("{:?}", data);
}

fn main() {
    print_it("a string slice");
    let owned_string = "owned string".to_owned();
    let another_owned = String::from("another");
    print_it(&owned_string);
    print_it(&another_owned);
}
```

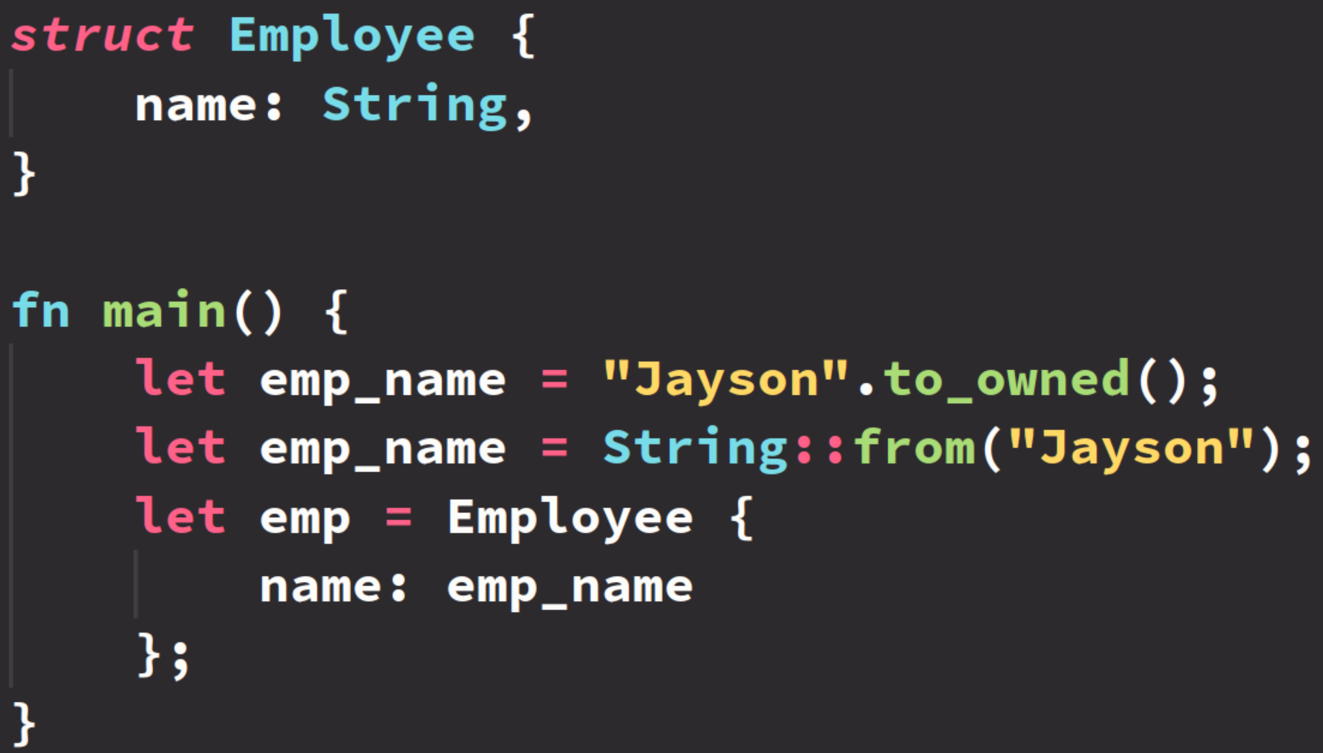
# Example – Will not work



```rust
struct Employee {
    name: &str,
}

fn main() {
    let emp_name = "Jayson";
    let emp = Employee {
        name: emp_name
    };
}
```

# Example - Works!



```rust
struct Employee {
    name: String,
}

fn main() {
    let emp_name = "Jayson".to_owned();
    let emp_name = String::from("Jayson");
    let emp = Employee {
        name: emp_name
    };
}
```

# Recap

- Strings are automatically borrowed
- Use *.to_owned()* or *String::from()* to create an owned copy of a string slice
- Use an owned *String* when storing in a struct