Functions and Structs

This lesson will get you acquainted with how to use structs while passing them in functions.

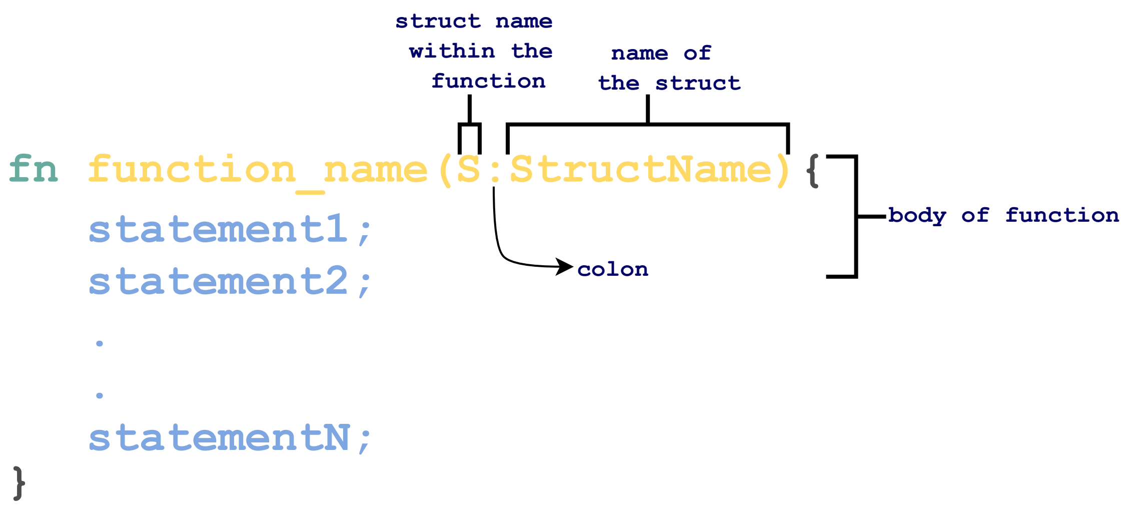
We'll cover the following

- Pass Structs to a Function
- Return Structs From a Function

Often, we need to pass a struct instance to a function. For example, in the previous lesson, every time we wanted to print a new struct instance we had to write a new print macro to print it. However, we can avoid multiple print statements by writing one print statement within a function and calling it when we need it.

Pass Structs to a Function

The structs can be passed to a function and the function can be invoked when required.

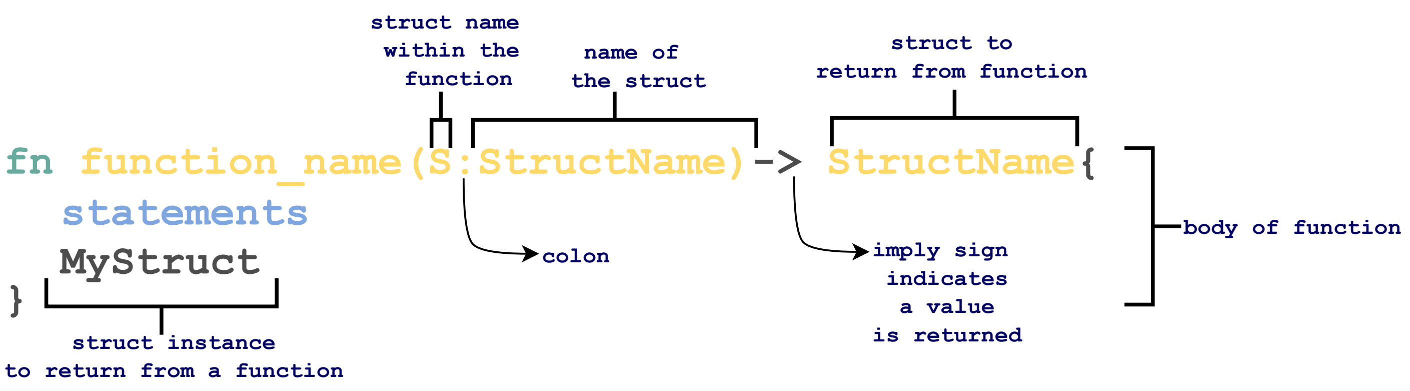


Pass struct to a function

```
//declare a struct
struct Course {
   code:i32,
   name:String,
   level:String,
}
fn display_mycourse_info(c:Course) {
println!("Name:{}, Level:{}, code: {}", c.name, c.level, c.code);
println!("Name:{}, Level:{} ,Code: {}", c.name, c.level, c.code);
}
fn main() {
   //initialize
   let course1 = Course  {
      name:String::from("Rust"),
      level:String::from("beginner"),
      code:130
   };
   display_mycourse_info(course1);
    let course2 = Course  {
      name:String::from("Java"),
      level:String::from("beginner"),
      code:130
   };
   display_mycourse_info(course2);
}
```

Return Structs From a Function

Structs can also be returned from the functions.



struct name within the function
name of the struct
struct to return from function
fn function_name(S:StructName)-> StructName{
statements
MyStruct
}
colon
imply sign indicates a value is returned
body of function
struct instance to return from a function

Return struct from a function

```
//declare a struct
struct Course {
   code:i32,
   name:String,
   level:String,
}
fn return_rust_course_info(c1:Course, c2:Course)-> Course{
   println!("I got into function and return values from there");
   if c1.name == "Rust" {
      return c1;
   }
   else{
      return c2;
   }
}

fn main() {
   //initialize
   let course1 = Course  {
      name:String::from("Rust"),
      level:String::from("beginner"),
      code:130

   };
    let course2 = Course  {
      name:String::from("Java"),
      level:String::from("beginner"),
      code:130
   };

  let choose_course = return_rust_course_info(course1, course2);
  println!("I choose to learn {} {} course with code:{}", choose_course.name, choose_course.level,
}
```

Now that you have learned about functions and structs, let’s learn about implementing methods in structs in the next lesson.