Methods of Structs

This lesson will get you acquainted with the methods in structs, i.e., what are they and how to create them in structs?

We'll cover the following

- What Are Methods?
- Declare a Method
- Call a Method
- Example
  - Explanation
- Quiz

What Are Methods?

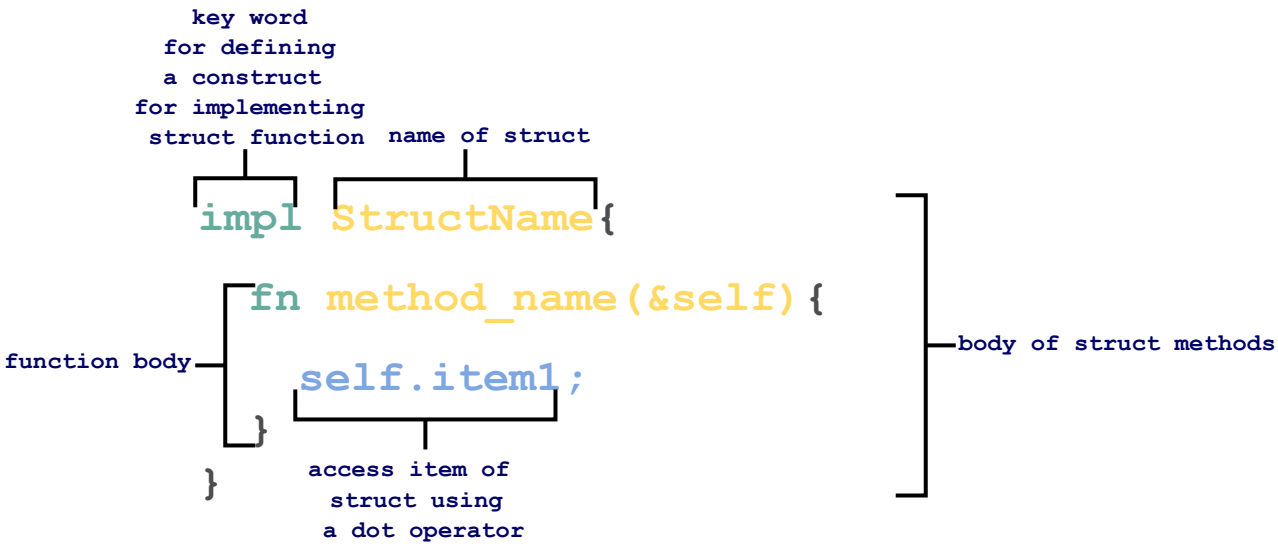
Methods are just like user-defined functions. They are like functions, but the only difference lies in the fact that methods are declared specifically within the struct context.

Declare a Method

The method is like a regular function except that the `&self` parameter is passed to it and the items within the function are accessed through it.

```
self.item
```

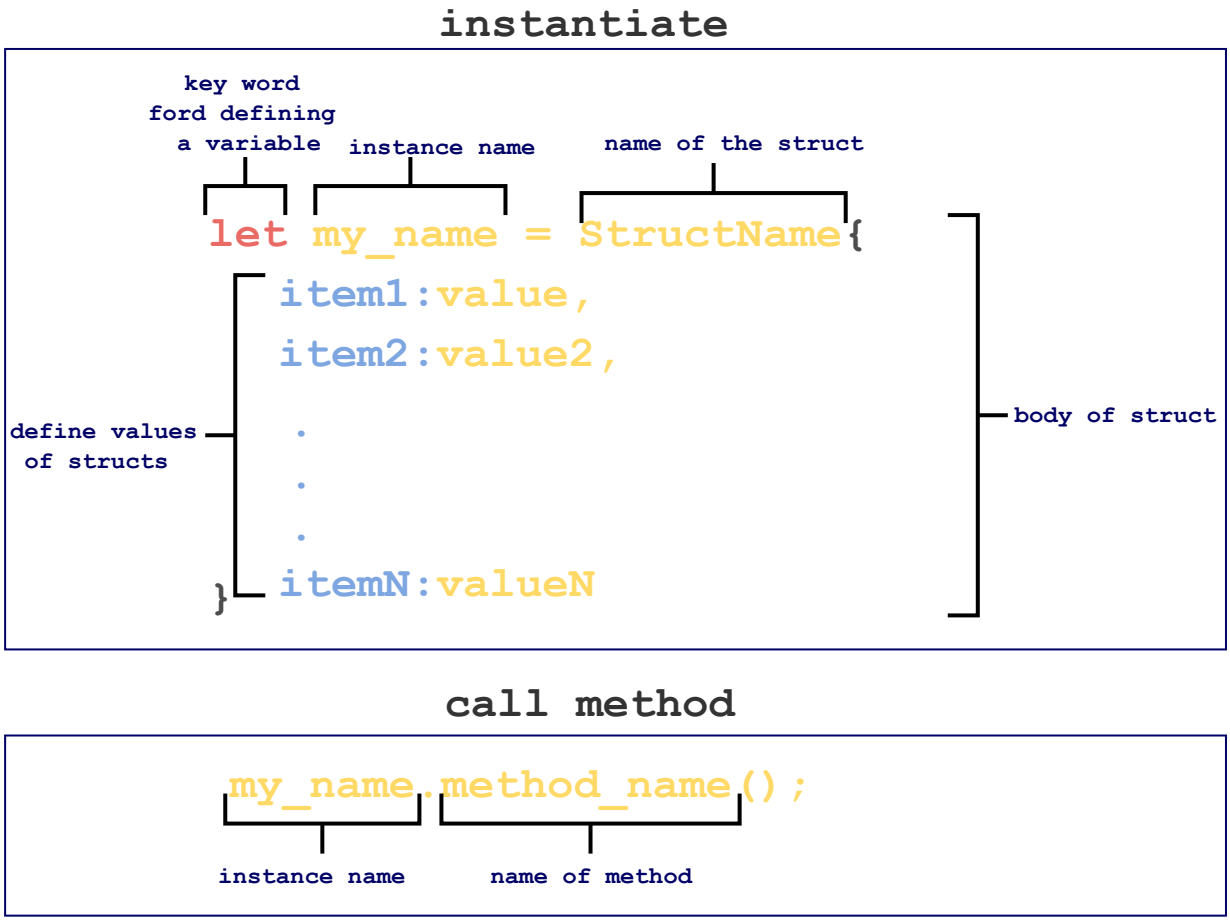
Here `self` is the calling instance, i.e., it is referencing to the struct.



impl construct for accessing value from struct

Call a Method

An instance of the struct has to be created to invoke it. This is similar to invoking a struct.



Initialize a struct

> **Why struct method**? The main advantage is that all the data related to the instance is put inside the `impl` block rather than putting it in different places.

Example

The example below shows declares a method `name_code` function within the `impl` construct:

```
//declare a struct
struct Course {
    name: String,
    level: String,
    code:i32
}
//impl construct to define struct methods
impl Course {
    fn name_code(&self) -> String {
        format!("{} {}", self.name, self.code)
    }
}

fn main() {
    let course_1 = Course {
        name: "Rust".to_string(),
        level:"beginner".to_string(),
        code:132
    };
    //call the non-static method
    println!("This is a {} course: {}", course_1.level, course_1.name_code());
}
```

Explanation

- **`main` function**

  From **line 14 to line 22**, the `main` function is defined.

  - **Lines 15 to 19**, creates an immutable instances `course1` of the struct `Course`.
  - On **line 21**,
    - prints the value of `level` item of struct instance using `course1.level`
    - invokes the method `name_code` within the `impl construct` and prints the value of name and code returned from the function.

- **`impl` `Course` construct**
  - The `impl` construct is defined from **line 8 to line 12**. Within the `impl`

construct has a function `name_code` that takes the parameter `self`, i.e., it

refers to the instance of the struct with which the function is invoked. It returns the name and code of the instance as `String` using the `format!` macro.

- `struct`

  On **line 2**, a `struct` `Course` is declared. Within the struct body, three items namely `code`, `name`, `level` are declared of type `i32`, `String` and `String` respectively.

Quiz

Test your understanding of methods of structs.

Quick Quiz on Methods of Structs!

Q

Which code block has the method enclosed in it?

Retake Quiz

In the next lesson, let's learn about "static" methods and contrast them with the ones you've seen before.