Boolean

This lesson discusses the boolean data type.

We'll cover the following

Boolean

The boolean variable can take a value either **true or false**.

☑ **True**

□ **False**

Example

The following code explains how to define a boolean variable in three different ways:

Explicit Definition

The following code explicitly defines the variable using the `bool` keyword:

```
fn main() {
    //explicitly define a bool
    let is_bool:bool = true;
    println!("explicitly_defined: {}", is_bool);
}
```

Implicit Definition

The following code implicitly defines the boolean type of a variable by assigning the value `true` or `false` to the variable.

```
fn main() {
    // assign a boolean value
    let a = true;
    let b = false;
    println!("a: {}", a);
    println!("b: {}", b);
}
```

Result of an Expression

The result of an expression that evaluates to either true or false (for example a comparison of two values) can be assigned to an implicit boolean variable.

```
fn main() {
    // get a value from an expression
    let c = 10 > 2;
    println!("c: {}", c);
}
```

Quiz

Test your understanding of boolean data type in Rust!

Quick Quiz on Boolean!

Q

What is the output of the following code?

```
let value = 13 > 20;
println!("{}", value);
```

Retake Quiz

Now that you have learned about boolean data type, let’s learn about character and string data type in Rust in the next lesson.