



What are data-types?

The data type of a value (or variable in some contexts) is an attribute that tells what kind of data that value can have.

In simple words it is the value of a variable

Data types in Rust are divided into different groups:

- Scalar Type
- Compound Type (we will discuss later in this course)

Scalar Type

A scalar type represents a single value. Rust has four primary scalar type.

- Integer
- Floating-point
- Boolean
- Character

Integer

An integer is a number without a fractional component. Integers can be further classified as Signed and Unsigned. Signed integers can store both negative and positive values. Unsigned integers can only store positive values. Integer is i32 by default

| size | Signed | Unsigned |
|---------|--------|----------|
| 8 bit | i8 | u8 |
| 16 bit | i16 | u16 |
| 32 bit | i32 | u32 |
| 64 bit | i64 | u64 |
| 128 bit | i128 | u128 |
| Arch | Isize | usize |

Each signed variant can store numbers from $-(2^n - 1)$ to $2^n - 1 - 1$ inclusive, where n is the number of bits that variant uses. So an i8 can store numbers from $-(2^7 - 1)$ to $2^7 - 1 - 1$, which equals -128 to 127. Unsigned variants can store numbers from 0 to $2^n - 1$, so a u8 can store numbers from 0 to $2^8 - 1$, which equals 0 to 255.

Additionally, the isize and usize types depend on the architecture of the computer your program is running on, which is denoted in the table as “arch”: 64 bits if you’re on a 64-bit architecture and 32 bits if you’re on a 32-bit architecture.

Floating-point Types

Rust also has two primitive types for floating-point numbers, which are numbers with decimal points. Rust’s floating-point types are f32 and f64, which are 32 bits and 64

bits in size, respectively. The default type is f64 because on modern CPUs it's roughly the same speed as f32 but is capable of more precision. All floating-point types are signed.

The f32 type is a single-precision float, and f64 has double precision.

Automatic Type casting

Automatic type casting is not allowed in Rust.

Number Separator

For easy readability of large numbers, we can use a visual separator _ underscore to separate digits. That is 50,000 can be written as 50_000 .

Boolean

Boolean types have two possible values – true or false. Use the bool keyword to declare a boolean variable .

Character

The character data type in Rust supports numbers, alphabets, Unicode and special characters. Use the char keyword to declare a variable of character data type.

Note that we specify char literals with single quotes, as opposed to string literals, which use double quotes. Rust's char type is four bytes in size and represents a Unicode Scalar Value, which means it can represent a lot more than just ASCII. Accented letters; Chinese, Japanese, and Korean characters; emoji; and zero-width spaces are all valid char values in Rust.