

# Data Types

Learn to Code with Rust

# Data Types

---

- Every Rust value has a **data type**.
- Rust is a **statically typed** language, which means the compiler must know the types of all variables at compile time.
- The compiler can *infer* the types of variables based on their initial assignments.

# Scalar Types

---

- A **scalar type** is a type that holds a single value.
- Rust has 4 scalar types: integers, floating-point numbers, Booleans, and characters.
- An integer is a whole number.
- A floating-point number is a decimal number.

# Signed and Unsigned Integers

---

- **Signed integer types** support positive and negative values.
- Signed integers like **i32** start with an **i** (integer).
- **Unsigned integer types** only support zero and positive values. They can store a larger max value in the positive direction.
- Unsigned integers like **u32** start with an **u** (unsigned).

# Bits

---

- The number after the **i** or **u** refers to the amount of bits the integer needs in your computer's memory.
- A **bit** is the smallest unit of computer memory. It represents a value of either 0 or 1.
- 8 bits is equal to 1 byte.
  - 1024 bytes in a kilobyte
  - 1024 kilobytes in a megabyte
  - 1024 megabytes in a gigabyte
- An **i32** requires 32 bits (4 bytes) of memory. An **f64** requires 64 bits (8 bytes).

# Integer Lower and Upper Bounds

Type	Smallest Value	Largest Value
i8	-128	127
u8	0	255
i16	-32,768	32,767
u16	0	65,535
i32	-2,147,483,648	2,147,483,647
u32	0	4,294,967,295
i64	-9,223,372,036,854,775,808	9,223,372,036,854,775,808
u64	0	18,446,744,073,709,551,616
i128	-170,141,183,460,469,231,731,687,303,715,884,105,728	170,141,183,460,469,231,731,687,303,715,884,105,727
u128	0	340,282,366,920,938,463,463,374,607,431,768,211,455

# Float Lower and Upper Bounds

---

Type	Precision
f32	6-9 digits of precision
f64	15-17 digits of precision