Raw Pointers

Learn to Code with Rust

Pointers

- A variable is a name for a piece of data.
- The Rust book defines a **pointer** as "a variable that contains an address in memory".
- By storing an address, a pointer variable indirectly *points* to a piece of data.
- A pointer prevents the duplication of the original piece of data. Instead, we store the data's address.

References

- A **reference** is a type/category of pointer that points to valid data.
- Create a reference with the borrow operator (**&**).
- Rust's borrow checker guarantees that *references* point to valid/allocated data. *Pointers* in other languages do not make the same promise.

Raw Pointers

- A **raw pointer** is a variable that stores a memory address without any safety checks.
- A **raw pointer** *may* point to valid data but it can also point to deallocated memory.
- **Raw pointers** are common in languages like C and C++. Developers still call them "references" even though they lack the safety guarantees of Rust's references.
- Rust supports raw pointers too but references are *safer* and *recommended*.