

# Traits

Learn to Code with Rust

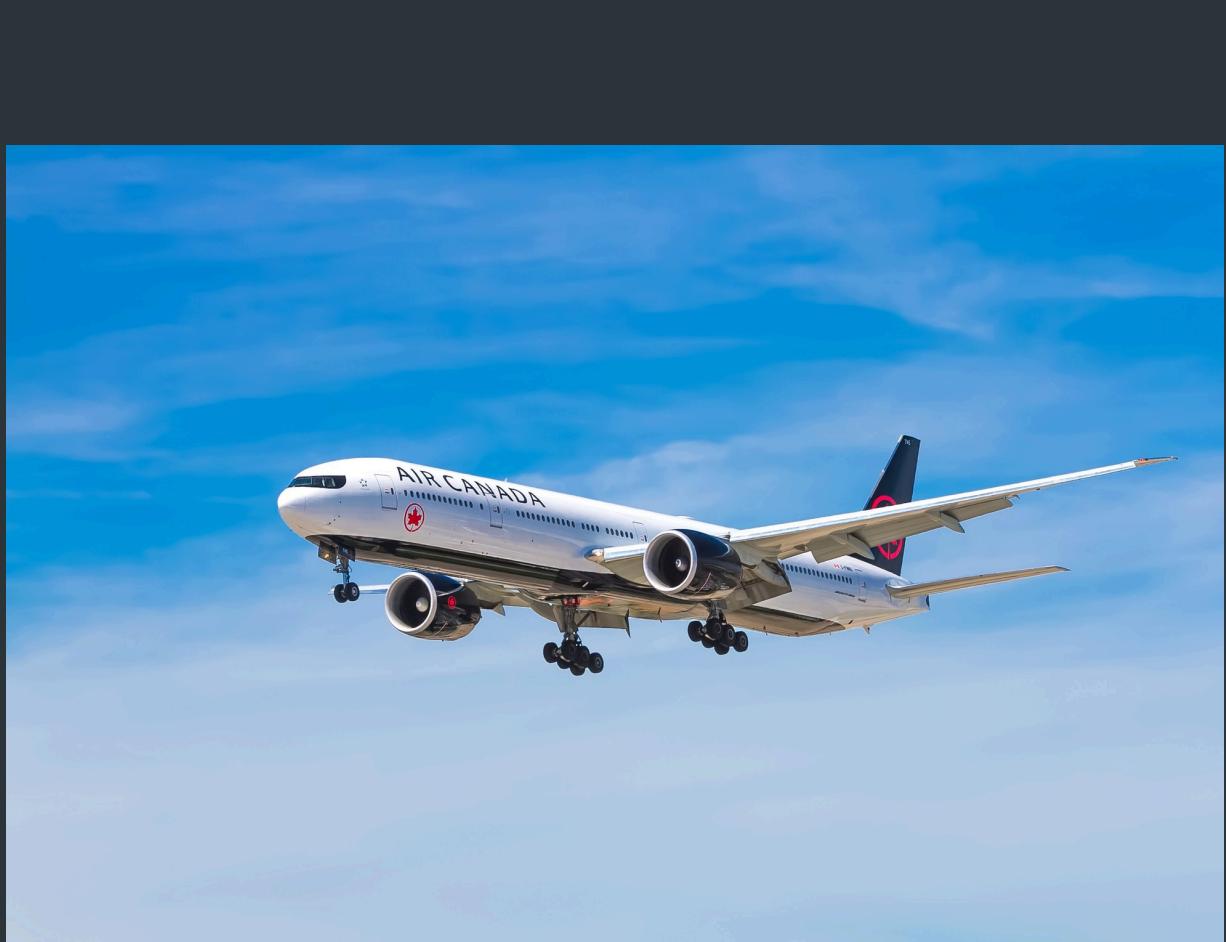
# Intro to Traits

---

- A **trait** is a "a distinguishing quality or characteristic".
- Consider a trait like **flight**. Flight describes the quality of being able to fly.

# Trait: Flight

---



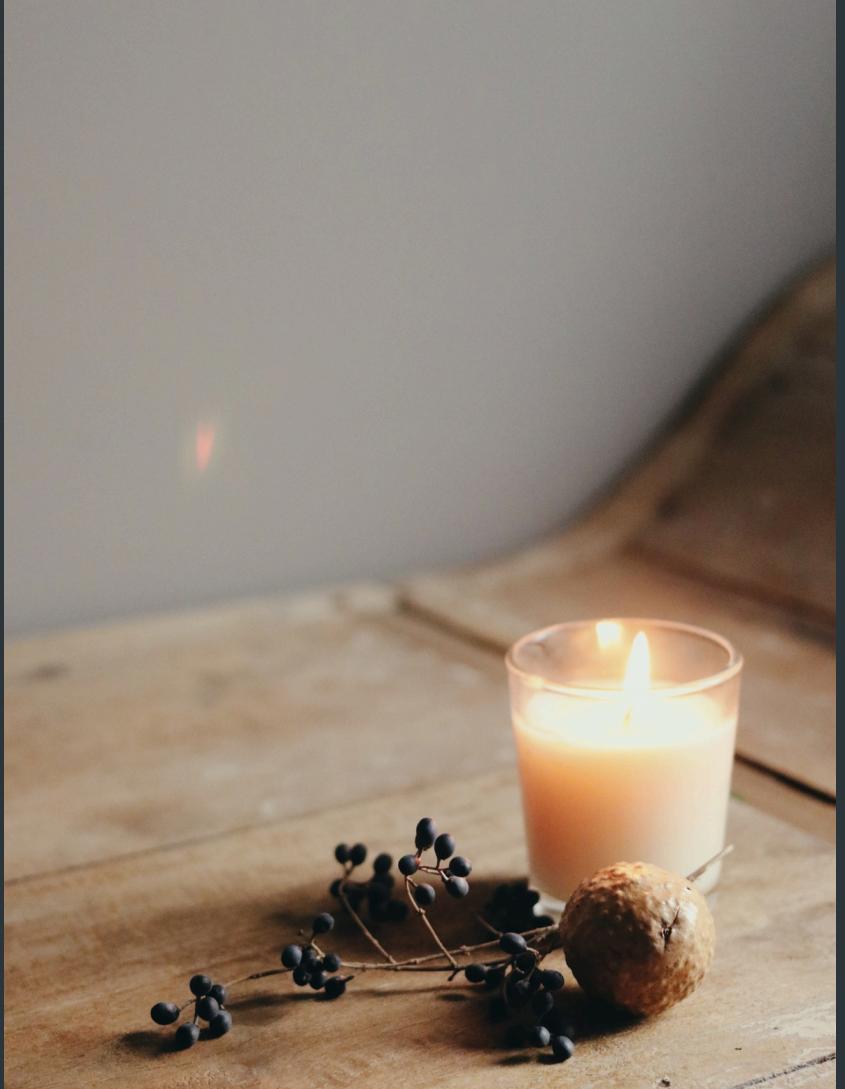
# Trait: Storage

---



# Trait: Illumination

---



# Traits in Rust

---

- A **trait** is a contract that describes functionality that a type should have.
- We use the word **implement** to describe when a type honors a trait's requirements.
  - For the previous example, we can say that a candle, a computer screen, and the sun **implement** the "illumination" trait.
- A **trait** definition declares the method(s) that a type implementing that trait must have.
  - Method name
  - Parameters with types
  - Return value type

# Traits We've Seen Before

---

- The **Display** and **Debug** traits require a type to define methods for presenting itself as a string.
- The **Clone** trait requires a type to define a **clone** method for creating a duplicate of itself.

# Implementations

---

- Once we have defined a trait, we can implement it on structs and enums. The type promises to honor the trait's contract.
  - A type implementing the **Flight** trait promises it can fly.
- Multiple types can implement the same trait.
  - A Bird and a Plane type both implement the **Flight** trait.
- A type can implement multiple traits.
  - A plane can implement both the **Flight** and **Storage** traits.