# Hack `postgres` Source Code: Vol I

## Chapter 1 : C & Rust

### 1.26 Rust - Traits

So, what is this "traits" in rust?

Umm.. you can think of them as "interfaces" in other languages. In short, traits allow "types" in rust to implement some custom methods.

Show me an example... It would help me understand more.

```rust
struct Car {
    _type: String,
    brand: String,
}

struct Bike {
    _type: String,
    brand: String,
}

// Create a trait called summary
trait Summary {
    fn print_summary(&self);
}

// Make the Car type to implement trait — Summary's function — `print_summary`
impl Summary for Car {
    fn print_summary(&self) {
        println!("type:{} and brand: {}", self._type, self.brand);
        }
}

impl Summary for Bike {

    fn print_summary(&self) {
        println!("type:{} and brand: {}", self._type, self.brand);
        }
}


  fn main() {

    // Create instances of Car
    let bmw_car: Car = Car { _type: String::from("car"),
```

```rust
        brand: String::from("BMW"),
    };

    // Bike instances
    let bmw_bike: Bike = Bike { _type: String::from("bike"),
    brand: String::from("BMW"),
    };

    // Calling methods on instances
    bmw_car.print_summary();
    bmw_bike.print_summary();


}
```