10-quotient.lean

```
import Mathlib
```

Quotients

After building up the theory of groups, homorphisms, and subgroups, we are now ready to define quotient groups.

In fact, quotients are so fundamental that Lean makes them a primitive way of constructing new types.

We shall first illustrate the general quotient construction in Lean, and then specialize it to quotient groups.

At the end of the journey, we show the first isomorphism theorem for groups as promised.

1 Quotient Sets

We still build up the theory from sets. First, we need to define equivalence relations on a type `α`. As you can guess, a binary relation `r` is just a `α → α → Prop`.

```
section

variable {α : Type*} {r : α → α → Prop} (a b c : α)
```

`Equivalence r` is a bundled structure that packages the three properties:

- reflexivity of `r`
- symmetry of `r`
- transitivity of `r`

```
variable (r_equiv : Equivalence r)

example : r a a := by exact r_equiv.refl _
example : r a b → r b a := by exact r_equiv.symm
example : r a b → r b c → r a c := by exact r_equiv.trans
```

In mathmatics we often denote equivalence relations by `~`. In Lean, we can also use `≈` as notation for equivalence relations, once we register the `Setoid` instance for `α`.

The `Setoid` typeclass is a bundle of

- an equivalence relation `r` on `α`
- the proof that `r` is an equivalence relation.

```
instance r_setoid : Setoid α where
  r := r
  iseqv := r_equiv

end
```