

# Maps in Go

---

- A **Map** is a collection type just like an array or a slice and stores **key:value** pairs.
- The main advantage of maps is that **add, get and delete operations take constant expected time**.
- All the keys and the values in a Map are statically typed and must have the same type.
- The keys in a map must be unique, but the values don't have to be unique.
- A Map allows us to **quickly access a value** using a unique key!
- We can use any **comparable type as a key map**. A comparable type is that type that supports the comparing operator which is the double equals sign.
- Even if it's possible, it's not recommended to use a float as a key. A float has some comparable issues.
- We can not compare a map to another map. We can only compare a map to **nil**.
- **Maps are unordered** data structures in Go.