

Defined Types

- A **defined type** also called a **named type** is a **new type created by the programmer** from another existing type which is called the **underlying** or **source type**.
- A new defined type must have a new name and can have its new methods.
- The underlying type provides the representation, operations and size of the newly defined type.
- Even though the defined type and the source type share the same representation, operations and size **they are different types**. A new type it's not just an alias for an existing type, it's a completely new type.
- If we want to perform operations between source and defined types we must convert one type into the other type. A type can be converted to another type if they share the same underlying type.
- There is no type-hierarchy in Go.

Why define new types?

- We can **attach methods** to newly defined types.
- **Type safety.** We must convert one type into another to perform operations with them.
- **Readability.** When we defined a new type let's say `type usd float64` we know that new type represents the US Dollar, not only floats.