

# The empty interface

## The empty interface

The interface type that specifies zero methods is known as the *empty interface*:

```
interface{}
```

An empty interface may hold values of any type. (Every type implements at least zero methods.)

Empty interfaces are used by code that handles values of unknown type. For example, `fmt.Print` takes any number of arguments of type `interface{}`.

`interface{}` means you can put value of any type, including your own custom type. All types in Go satisfy an empty interface (`interface{}` is an empty interface).

What *is* an interface?

An interface is two things:

it is a set of methods,  
but it is also a type

The `interface{}` type, the **empty interface** is the interface that has no methods.

Since there is no `implements` keyword, all types implement at least zero methods, and satisfying an interface is done automatically, **all types satisfy the empty interface.**

That means that if you write a function that takes an `interface{}` value as a parameter, **you can supply that function with any value.**