Methods Versus Functions

This lesson explains differences between a function and a method in Golang.

Differences between a function and a method

A function has the variable as a parameter:

```
Function1(recv)
```

A method is called on the variable:

```
recv.Method1()
```

A method can change the values (or the state) of the receiver variable provided this is a pointer, just as is the case with functions (a function can also change the state of its parameter when this is passed as a pointer: call by reference).

> **Note**: Don't forget the ( ) after `Method1`, or you get the compiler error: `method recv.Method1 is not an expression, must be called`.

The receiver must have an explicit name, and this name must be used in the method. *receiver*-type is called the (receiver) base type; this type *must be declared within the same* package as all of its methods. In Go, the methods attached to a (receiver) type are not written inside of the structure, as is the case with classes; the coupling is much loose: the receiver establishes the association between method and type.

Methods are not mixed with the data definition (the structs). They are orthogonal to types; representation (data) and behavior (methods) are independent.

Now that you are familiar with the details of methods, in the next lesson, you have to write a program to solve a problem.