Methods

**Exercise 1**

Declare a struct that represents a baseball player. Include name, atBats and hits. Declare a method that calculates a players batting average. The formula is Hits / AtBats. Declare a slice of this type and initialize the slice with several players. Iterate over the slice displaying the players name and batting average.

Interfaces

**Exercise 1**

**Part A** Declare an interface named speaker with a method named speak. Declare a struct named english that represents a person who speaks english and declare a struct named chinese for someone who speaks chinese. Implement the speaker interface for each struct using a value receiver and these literal strings "Hello World" and "你好世界". Declare a variable of type speaker and assign the address of a value of type english and call the method. Do it again for a value of type chinese.

**Part B** Add a new function named sayHello that accepts a value of type speaker. Implement that function to call the speak method on the interface value. Then create new values of each type and use the function.

Embedding

**Exercise 1**

Copy the code from the template. Add a new type CachingFeed which embeds Feed and overrides the Fetch method.

Exporting

Exporting

**Exercise 1**

**Part A** Create a package named toy with a single exported struct type named Toy. Add the exported fields Name and Weight. Then add two unexported fields named onHand and sold. Declare a factory function called New to create values of type toy and accept parameters for the exported fields. Then declare methods that return and update values for the unexported fields.

**Part B** Create a program that imports the toy package. Use the New function to create a value of type toy. Then use the methods to set the counts and display the field values of that toy value.