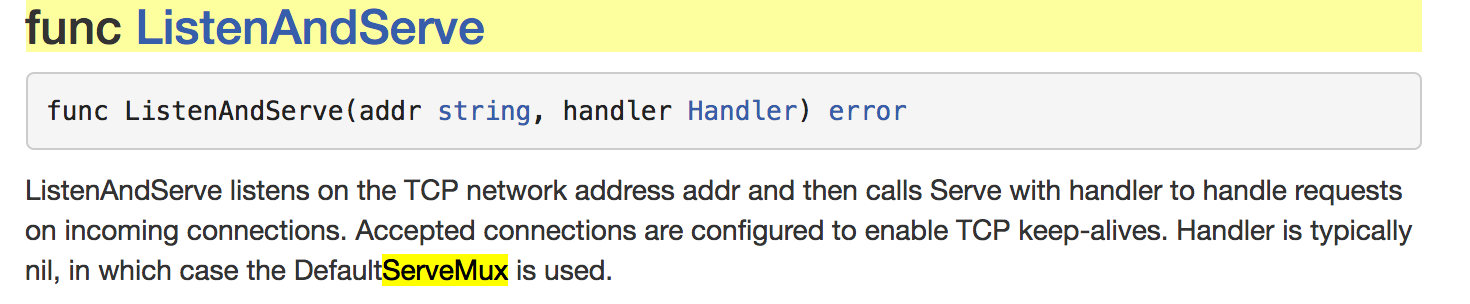
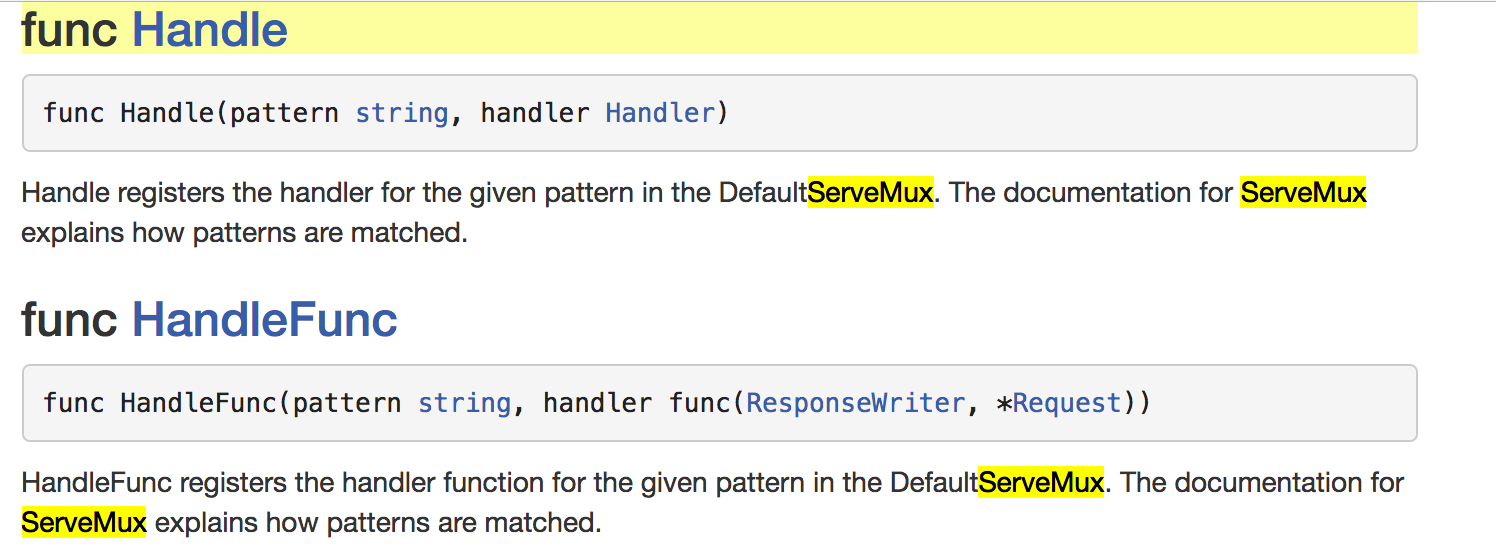
It takes some getting used to.

You use the default servemux when you pass nil into ListenAndServe: http.ListenAndServe(":8080", nil)​

​

http.Handle & http.HandleFunc are used when you are using the default servemux.​

​

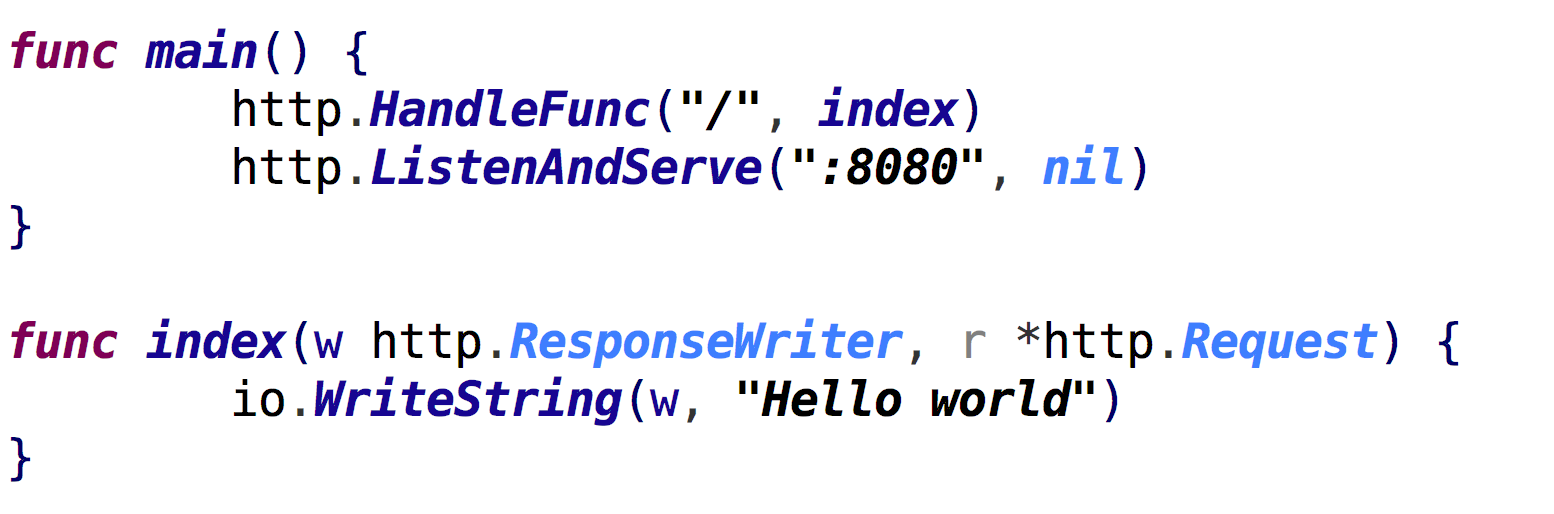
​http.Handle asks for a pattern and a HANDLER

http.HandleFunc asks for a pattern and a FUNC(REPONSEWRITER, \*REQUEST)

For http.Handle you need to pass an argument into it which is a VALUE of type HANDLER.

For http.HandleFunc​ you need to pass an argument into it which is a VALUE of type FUNC(REPONSEWRITER, \*REQUEST)​.​

Here is how you would use http.HandleFunc

​

Notice how the variable with the identifier "index" is storing a value of type FUNC(REPONSEWRITER, \*REQUEST)​. That is just what http.HandleFunc wants.

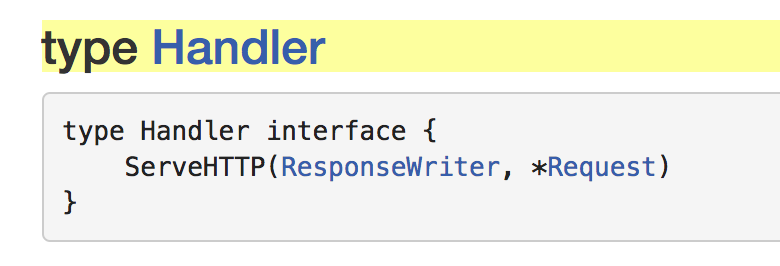
Here is one of the ways you could use http.Handle:

​

Notice how the variable with the identifier "foo" is of type "hotdog" and how type "hotdog" has the method "ServeHTTP(ResponsWriter, \*Request)" attached to it? This is what makes something type HANDLER (which is what http.Handle wants). Type HOTDOG is ALSO of type HANDLER. Any TYPE that has the methods listed in an INTERFACE implicitly fulfills that interface and IS ALSO OF THAT TYPE.

So "FOO" is of type HOTDOG and ALSO of type HANDLER.

Here is the HANDLER interface

​

You see how the HANDLER INTERFACE says you must have this method to also be my type?

My joke: An interface says, "Hey baby, if you have these methods, then you're my type."

VALUES CAN BE OF MORE THAN ONE TYPE.