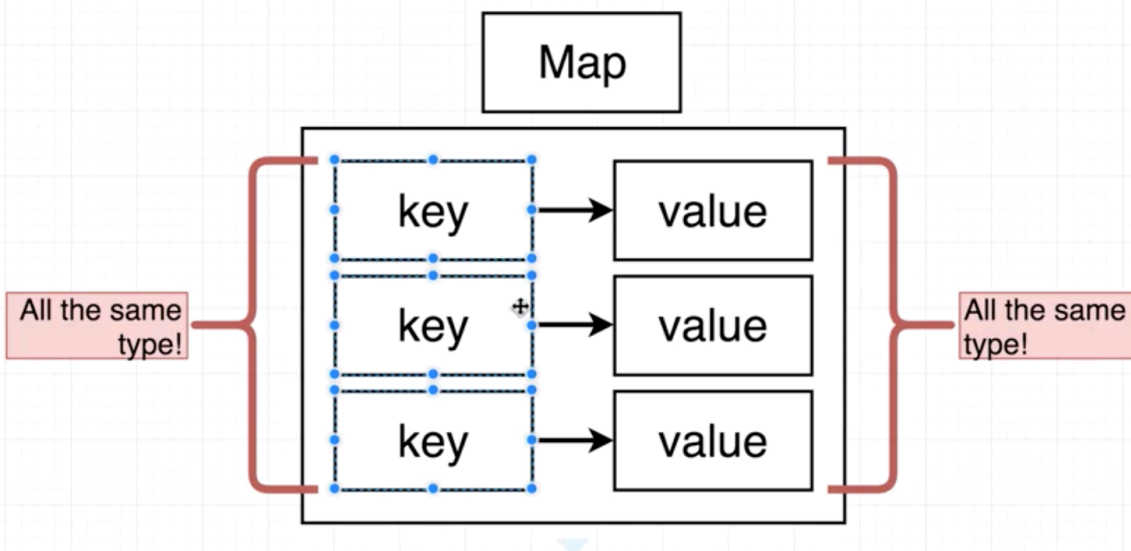
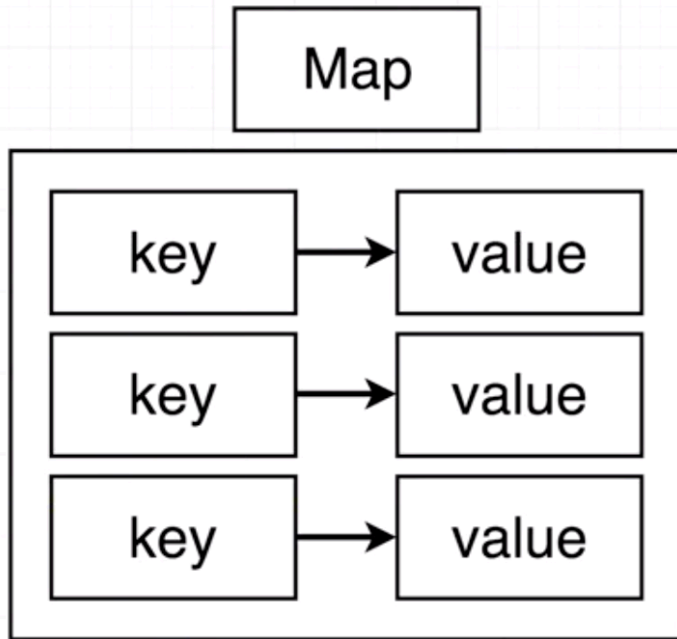


5 - Maps

What's a Map?



/src/map/main.go

```
1 func main() {  
2     colors := map[string]string{  
3         "red":    "#ff0000",  
4         "green": "#4bf745",  
    }
```

```

5     }
6     fmt.Println(colors)
7 }

```

Manipulating Maps

Another methods to create a map

```

1 var colors map[string]string

1 colors := make(map[string]string)

```

Assign value to the map

```

1 colors["white"] = "#ffffff"

```

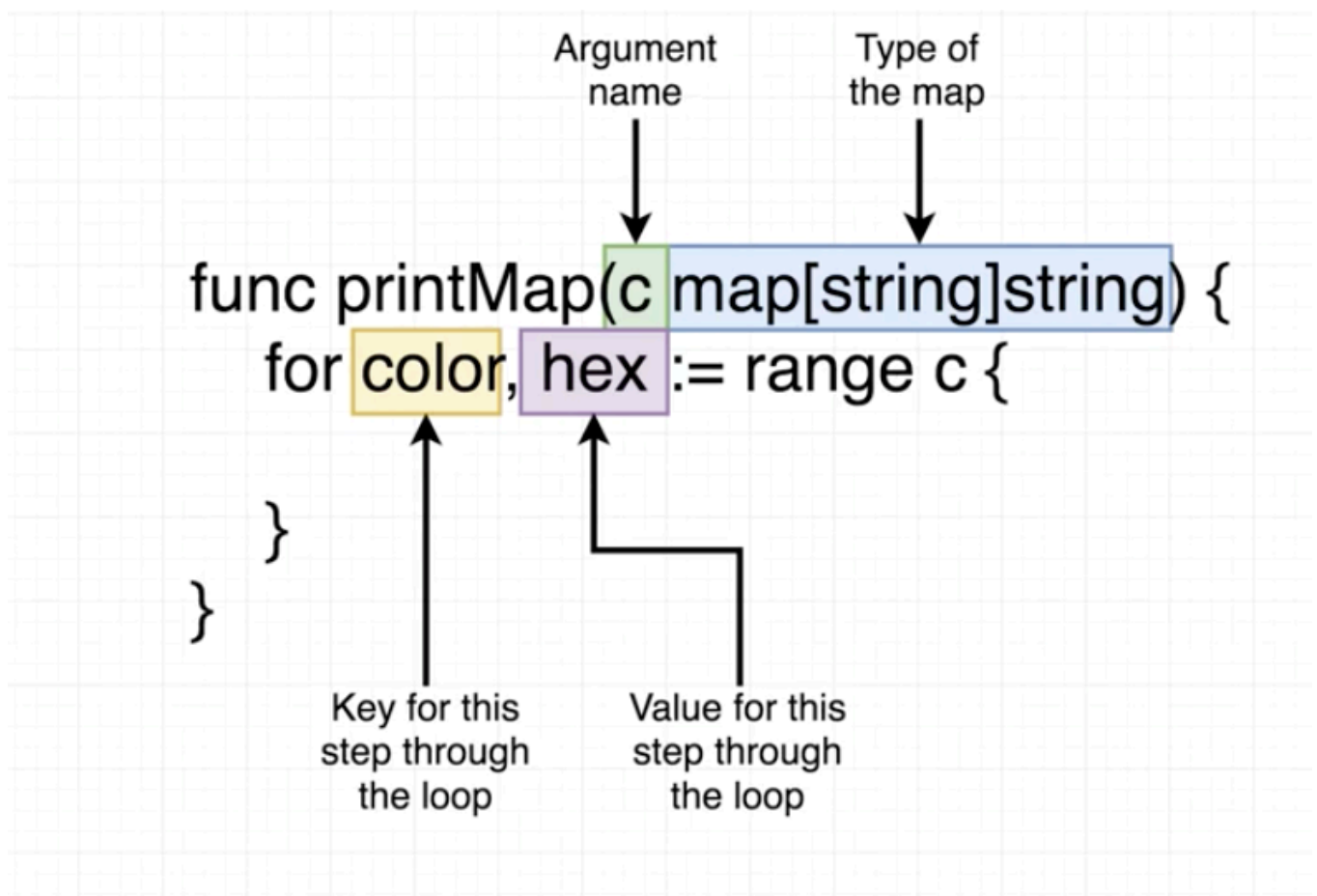
To delete key in map

```

1 delete(colors, "white")

```

Iterating Over Maps



main.go

```

1 func main() {
2     colors := map[string]string{
3         "red":    "#ff0000",
4         "green": "#00ff00",

```

```

5         "white": "#ffffff",
6     }
7
8     printMap(colors)
9 }
10
11 func printMap(c map[string]string) {
12     for color, hex := range c {
13         fmt.Println("Hex code for", color, "is", hex)
14     }
15 }

```

Differences Between Maps and Structs

Map		Struct	
All keys must be the same type	Use to represent a collection of related properties	Values can be of different type	You need to know all the different fields at compile time
All values must be the same type	Don't need to know all the keys at compile time	Keys don't support indexing	Use to represent a "thing" with a lot of different properties
Keys are indexed - we can iterate over them	Reference Type!	Value Type!	

