Existence of Key-Value Item

This lesson focuses on two major concepts: how to find a value associated with a key and how to delete a key-value pair from a map.

WE'LL COVER THE FOLLOWING

Testing the existence of a key

We saw in the previous lesson that `val1 = map1[key1]` returns the value `val1` associated with `key1`. If `key1` does not exist in the map, `val1` becomes the zero-value for the value's type, but this is ambiguous. Now we can't distinguish between this case or the case where `key1` does exist and its value is the *zero-value*! In order to test this, we can use the following **comma ok** form:

```
val1, isPresent = map1[key1]
```

The variable `isPresent` will contain a Boolean value. If `key1` exists in `map1`, `val1` will contain the value for `key1`, and `isPresent` will be true. If `key1` does not exist in `map1`, `val1` will contain the zero-value for its type, and `isPresent` will be false.

If you just want to check for the presence of a key and don't care about its value, you could write:

```
_, ok := map1[key1] // ok == true if key1 is present, false otherwise
```

Or combined with an `if`:

```
if _, ok := map1[key1]; ok {
```

```
  // ...
}
```

Deleting an element with a key

This is done with:

```
delete(map1, key1)
```

When `key1` does not exist, this statement doesn't produce an error.

Both the techniques are implemented in the following program.

```
package main
import "fmt"

func main() {
  var value int
  var isPresent bool
  map1 := make(map[string]int)
  map1["New Delhi"] = 55
  map1["Beijing"] = 20
  map1["Washington"] = 25
  value, isPresent = map1["Beijing"]  // checking existence of a key

  if isPresent {
    fmt.Printf("The value of \"Beijing\" in map1 is: %d\n", value)
  } else {
    fmt.Println("map1 does not contain Beijing")
  }
  value, isPresent = map1["Paris"]   // chekcing existence of a key
  fmt.Printf("Is \"Paris\" in map1 ?: %t\n", isPresent)
  fmt.Printf("Value is: %d\n", value)

  // delete an item:
  delete(map1, "Washington")
  value, isPresent = map1["Washington"] // checking existence of a key

  if isPresent {
    fmt.Printf("The value of \"Washington\" in map1 is: %d\n", value)
  } else {
    fmt.Println("map1 does not contain Washington")
  }
}
```

Finding and Deleting a Key from Map

In the above code, in `main` at **line** 7, we made a map `map1`. The declaration of `map1` shows that its keys will be of *string* type and values associated with its keys will be of *int* type. Now to check the presence of a key in this map, we

keys will be of *int* type. Now to check the presence of a key in this map, we made two variables: `value` of type *int* (at **line 5**) to get the value associated with that key and `isPresent` of type *bool* (at **line 6**) to check whether that key exists or not.

From **line 8** to **10**, we are making *key-value* pairs (each pair line by line) for `map1`. At **line 8**, we create a key `New Delhi` and give the value **55** to it. At **line 9**, we create a key `Beijing` and give the value **20** to it. At **line 10**, we create a key `Washington` and give the value **25** to it.

Now at **line 11**, we are checking the existence of the key `Beijing` in `map1` as: `value, isPresent = map1["Beijing"]`. We know that `Beijing` does exist in `map1` so `value` will get **20**, and `isPresent` will become `true`, causing the condition at **line 13** to become true. Consequently, **The value of "Beijing" in map1 is: 20** will be printed on the screen. Now at **line 18**, we are checking the existence of the key `Paris` in `map1` as: `value, isPresent = map1["Paris"]`. We know that `Paris` does not exist in `map1` so `value` will get **0**, and `isPresent` will become `false`, causing **line 19** to print **Is "Paris" in map1 ?: false**. **Line 20** will print **Value is: 0**.

At **line 23**, we are deleting a key `Washington` from `map1`. The execution of this line doesn't tell whether the key was deleted or not. So in the next line, we are verifying the existence of `Washington` in `map1` as: `value, isPresent = map1["Washington"]`. We know that `Washington` does not exist in `map1` anymore after deletion, so `value` will get **0**, and `isPresent` will become `false`, causing the condition at **line 26** to become false. The control will transfer to **line 28**, and **map1 does not contain Washington** will be printed on the screen.

That's it about the existence of a value in a map, in the next lesson, you'll see how to apply the for construct on maps.