QUIZ

Which of the following options is **NOT** true about updating values in a map?

To update a value of a key/value pair use the bracket notation

To update a value of a key/value pair use parentheses

Correct! This is a false statement. Using the bracket notation with its key allows us to update a key/value pair.

To update a value of a key/value pair use quotation marks

What is the main purpose of a Go map?

To access items using indexes.

To manage the data more efficiently.

Correct! It is also a good skill for programmers.

To have a collection of ordered key/value pairs.

Complete the following code to initialize a map that the map is declared as officeSupplies and key type as string and value type as int.

```
var officeSupplies = map[ string ] int { "notebooks":15 , }
```

You got it!

What can be done to fix the compiler error of the following code?

```
var origins = map[string]string{ "Mark": "New York" "John": "Los Angeles",
}

fmt.Println(origins)
```

Change it to make(map[string]string)

Add a comma between "New York" and "John"

Well done! Since the pairs are in the same line, a comma is required to separate pairs. If the pairs are in different lines, commas are not required.

Delete all the quotation marks

What does the following Go program output?

```
var guestList = map[string]string {
  "Table 1" : "Mr. Jack Robinson ",
}

fmt.Println(guestList["Table 1"])
```

Mr. Jack Robinson

Correct! Having the key-name in brackets allows us to find the value associated with the key "Table 1".

Which code snippet correctly adds a key/value pair to `vacationList` in the following Go program?

```
var vacationList = map[string]int{
  "towels":3,
  "plane tickets": 2,
  "meals" : 30,
}

// Add key/value pair here
```

`vacationList["car rides"] = 12`

👏 Correct! Using the bracket notation and the assignment operator allow us to add key/value pairs.

Which function is used to delete a key/value pair in a map?

`delete()`

👏 Well done! The `delete()` function is used to delete key/value pairs.

`null()`

`del()`

What would the following Go program output?

```
var bakingSupplies = map[string]int {
  "sugar": 4 ,
  "apples": 4,
  "flour": 4,
  "eggs":3,
}

fmt.Println(len(bakingSupplies))
```

6

2

4

Well Done! There are 4 key/value pairs in the map `bakingSupplies`.

Which of the following options is true about maps?

Indexing different items is possible in Go

Using the same key name in various key/value pairs is possible in Go

Adding key/value pairs of different keyTypes and valueTypes in the same map is possible in Go

Creating an empty map is accomplished by using the `make()` function or map literals.

Correct! Use either the `make()` function or map literals to create empty maps.