Lesson 16 – Maps

⏱ **2h**

What students should know

- What is a map?
- Basic Operations with maps
- For Each command

A map is a collection that contains pairs of keys and values. For example, records in a phone book or a dictionary can be a map. Other times it is also referred to by the name dictionary.

Figure 1 Yellow Pages Map

Each key is unique so you can refer to it uniquely. This means that if you add a key/value pair (entry) and the collection already contains an entry with the same key, the previous entry will be removed from the map.

The value can be any type from a simple variable to an object.

Create a map

A map is declared in B4X as below:

```
Private EnglishGreek As  Map
EnglishGreek.Initialize

Private EnglishItalian As  Map
EnglishItalian.Initialize
```

Where **EnglishGreek** the name of the first map was created and EnglishItalian the second. In addition, a map to be used must be initialized.

Insert items into Map

Put method allow to add keys/values to a map. For example the two maps bellow use the put method.

English	**Greek**
Memory	Memory
Screen	Screen
Printer	Printer
Programming Language	Programming Language'

Map 1 **EgnlishGreek**

English	**Italian**
Memory	Memoria
Screen	Schermo
Printer	Stampante
Programming Language	Linguaggio di Programmazione

Map 2 **EnglishItalian**

The Put method is stated as follows:

<map name>.put(key, value)

The following example creates two Maps with key words from The English language and values in Greek and Italian.

```
Private EgnlishGreek  As  Map
EgnlishGreek. Initialize

Private EnglishItalian As  Map
EnglishItalian. Initialize

EgnlishGreek. Put("Memory","Memory")
EgnlishGreek. Put("Screen", "Screen  ")
EgnlishGreek. Put("Printer ","Printer")
EgnlishGreek. Put("Programming Language", "ProgrammingLanguage"))

EnglishItalian. Put(“Memory”, "Memoria")
EnglishItalian. Put(“Screen”, "Schermo")
EnglishItalian. Put(“Printer”, "Stampante")
EnglishItalian. Put(“Programming Language”, "Linguaggio di  Pro-
grammazione")
```

Use a map value

To use a value from a map, you only need to use the Get command.

<map name>. Get(Key As Object)

```
GRWord = EnglishGreek.Get("Screen")
Log(GRWord)     ' Shows  Screen

ITWord = EnglishItalian.Get("Screen")
Log(ITWord)  'Shows  Schermo
```

Returns Value with Key “Screen” in Variable GRWord From map English-Greek, and the Value which Corresponds In Key “Screen” From map EnglishItalian In Variable ITWord.

> **Remember**
>
> The type of variable that the value accepts from the map must be the same type As the value. Key types must always be string or number.

If the key does not exist on the map, then Null is returned

```
ITWord = EnglishItalian.Get("Keyboard")
Log(ITWord)  'Shows null
```

Indexes in Maps

You can use indexes as in lists to access either the key or key value.

<map name>. GetKeyAt(Index As Int)

```
Key = EnglishItalian. GetKeyAt(2)
Log(key)  ` Shows Printer
```

<map name>. GetValueAt(Index As Int)

```
Value = EnglishItalian.GetValueAt(2)
Log(Value)    ` Shows  Stampante
```

Also, the GetKeyAt and GetValueAt commands can be used in a iterative process to get all map values:

```
  For i = 0 To EnglishGreek.Size - 1
    Log(EnglishGreek.GetValueAt(i))
  Next
```

The above iteration shows the values of map EnglishGreek.

The command “for each”

One iteration command that has not been discussed so far is the command **for each**. This command creates an iteration that accepts items from a list of values such as a map for example.

```
For Each word As  String In  EnglishItalian.Values
   Log(word)
Next
```

The above iteration defines a variable which (the word in the example) will accept each time the value of the current position. You do not need a counter or step like for loop. Next steps give next map values and stops when map keys end.

```
For Each key  As  String In  EnglishItalian.Keys
   Log(key)
Next
```

Displays the keys stored in the map.

To get both keys and values at the same time you can use the command for each as below:

```
For Each key  As  String In  EnglishItalian.Keys
  Log(key  &  EnglishItalian.GetValueAt(key))
Next
```

Check key existence

If you are looking for a key, you can scan the table to find out if it exists or not, but it's easier and faster to use the **ContainsKey command**

<map name>. ContainsKey(Key As Object)

```
If EnglishGreek.ContainsKey("Keyboard")  Then
  Log("There is already an entry with this key !", "")
Else
  Log("There is  not such a key!", "")
End If
```

Delete Key and Empty Map

The Remove command deletes a key (and of course its value from a Map)

<map name>. Remove(Key As Object)

```
EnglishGreek. Remove("Memory")
```

Deleting all items from a Map is done with the Clear command.

```
EnglishGreek. Clear
```

Exercises

1. Create a map with Country names as keys and their capitals as values.

CUBA	HAVANA
CYPRUS	NICOSIA
CZECHIA	PRAGUE
EGYPT	CAIRO
KENYA	NAIROBI
MEXICO	MEXICO CITY
PERU	LIMA
VIETNAM	HANOI
PORTUGAL	LISBON

Source: https://www.boldtuesday.com/pages/alphabetical-list-of-all-countries-and-capitals-shown-on-list-of-countries-poster

2. Add 3 countries with their capitals.
3. Display the names of countries and their capitals by using the **for each** command
4. Create a new map that contains the names of the capitals as keys and values country's names.
5. In an appropriate text field, enter the name of a city and then display the country that owns the city.