Collection Types

Collection Types

Containers that can store multiple values

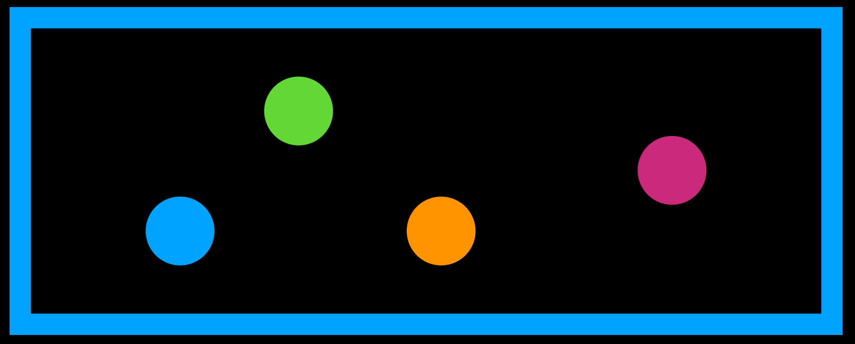
Like a “group of variables“

Structures for ordered and unordered groups of values

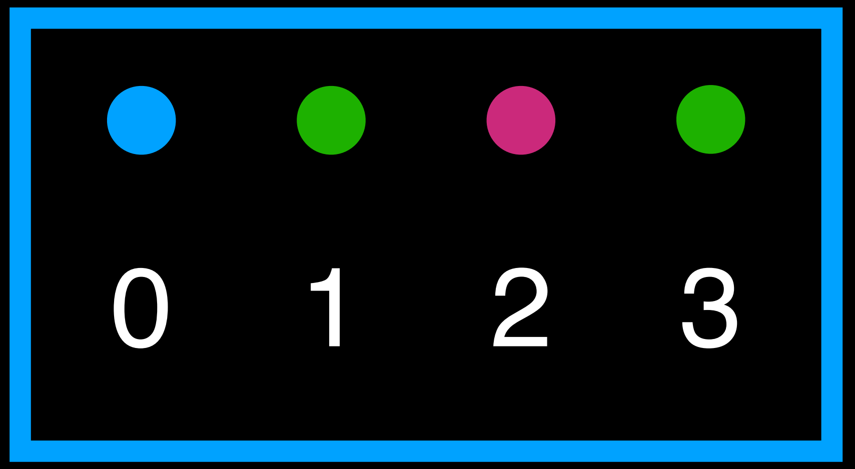
Set, **Array** and **Dictionary**

Collection Types

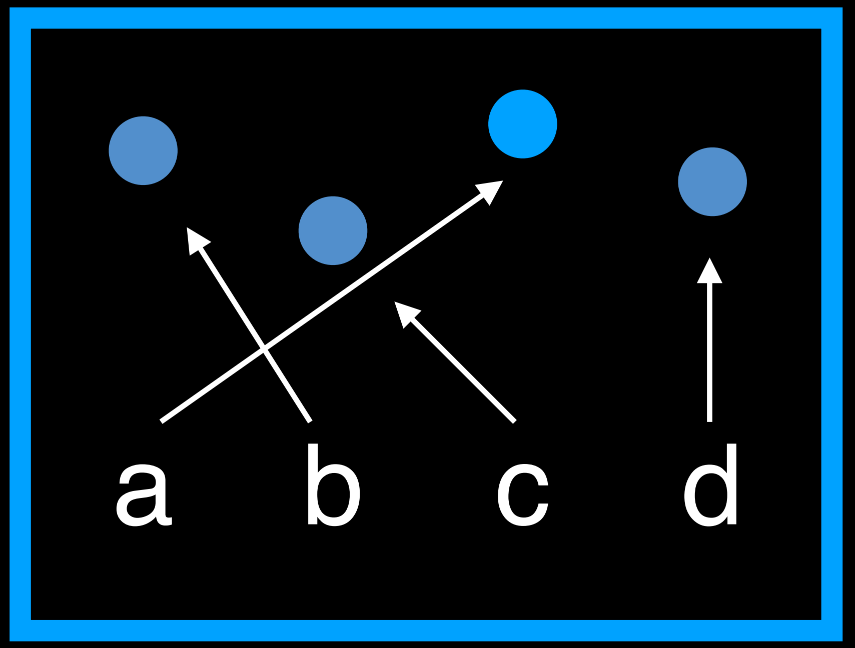
Set is an unordered collection of unique values



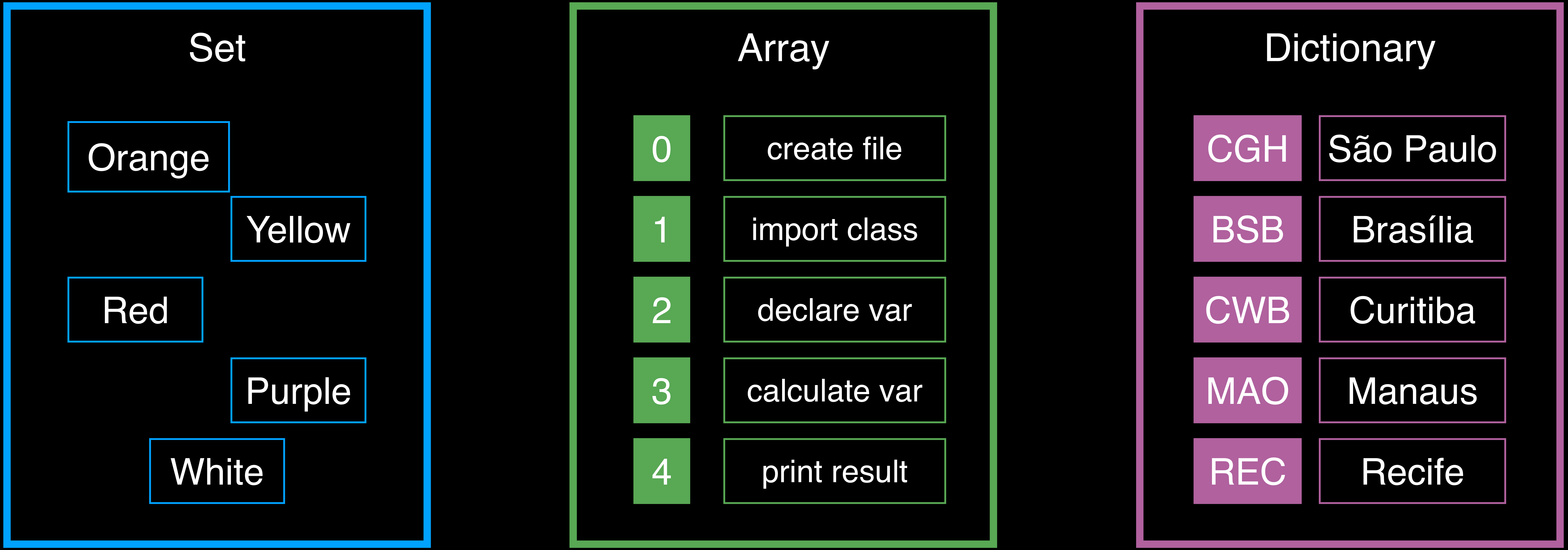
Array is an ordered collection of values accessible by index



Dictionary is an unordered collection of key-value associations



Collection Types



Set

Orange

Yellow

Red

Purple

White

Array

0	create file
1	import class
2	declare var
3	calculate var
4	print result

Dictionary

CGH	São Paulo
BSB	Brasília
CWB	Curitiba
MAO	Manaus
REC	Recife

Set

```
import Foundation

// Declaring an unordered collection of unique elements
var cakeIngredients: Set = ["Eggs", "Butter", "Flour", "Milk"]

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"

// Inserting a new element
cakeIngredients.insert("Milk")

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"
```

```swift
import Foundation

// Declaring an unordered collection of unique elements
var cakeIngredients: Set = ["Eggs", "Butter", "Flour", "Milk"]

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"

// Inserting a new element
cakeIngredients.insert("Milk")

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"
```

```swift
import Foundation

// Declaring an unordered collection of unique elements
var cakeIngredients: Set = ["Eggs", "Butter", "Flour", "Milk"]

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"

// Inserting a new element
cakeIngredients.insert("Milk")

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"
```

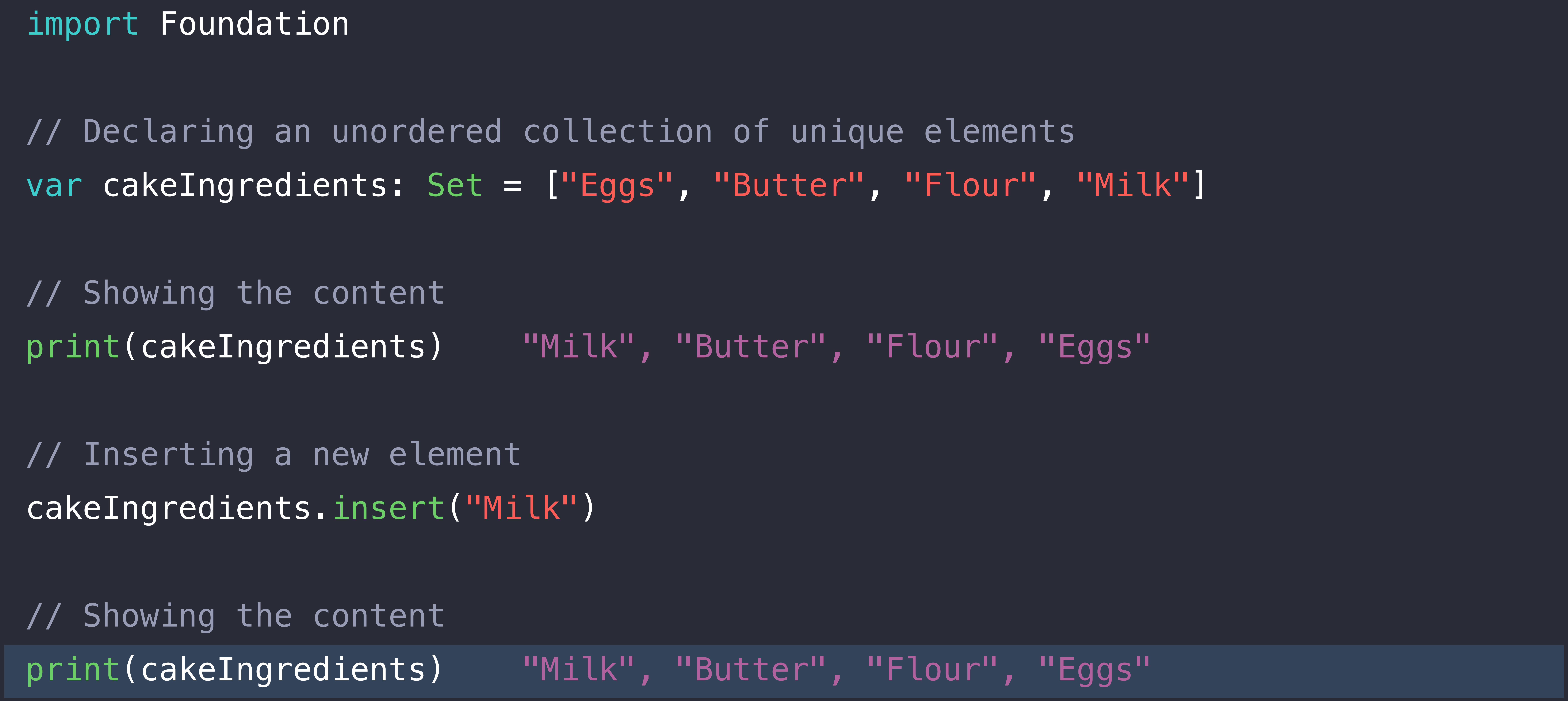
```swift
import Foundation

// Declaring an unordered collection of unique elements
var cakeIngredients: Set = ["Eggs", "Butter", "Flour", "Milk"]

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"

// Inserting a new element
cakeIngredients.insert("Milk")

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"
```



```swift
import Foundation

// Declaring an unordered collection of unique elements
var cakeIngredients: Set = ["Eggs", "Butter", "Flour", "Milk"]

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"

// Inserting a new element
cakeIngredients.insert("Milk")

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"
```

```
import Foundation

// Declaring an unordered collection of unique elements
var cakeIngredients: Set = ["Eggs", "Butter", "Flour", "Milk"]

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"

// Inserting a new element
cakeIngredients.insert("Milk")

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"
```

```
import Foundation

// Declaring an unordered collection of unique elements
var cakeIngredients: Set = ["Eggs", "Butter", "Flour", "Milk"]

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Flour", "Eggs"

// Inserting a new element
cakeIngredients.insert("Cocoa")

// Showing the content
print(cakeIngredients)    "Milk", "Butter", "Cocoa", "Flour", "Eggs"
```

Hands on

Array

```swift
// Ways to declare an array
var tracks: Array<String> = ["Interlagos", "Velocitta", "Auto Club Speedway”]

var tracks: [String] = ["Interlagos", "Velocitta", "Auto Club Speedway”]

var tracks = ["Interlagos", "Velocitta", "Auto Club Speedway"]
```

```swift
// Ways to declare an array
var tracks: Array<String> = ["Interlagos", "Velocitta", "Auto Club Speedway"]

var tracks: [String] = ["Interlagos", "Velocitta", "Auto Club Speedway"]

var tracks = ["Interlagos", "Velocitta", "Auto Club Speedway"]
```

```
// Ways to declare an array
var tracks: Array<String> = ["Interlagos", "Velocitta", "Auto Club Speedway"]

var tracks: [String] = ["Interlagos", "Velocitta", "Auto Club Speedway"]

var tracks = ["Interlagos", "Velocitta", "Auto Club Speedway"]
```

```swift
// Ways to declare an array
var tracks: Array<String> = ["Interlagos", "Velocitta", "Auto Club Speedway"]

var tracks: [String] = ["Interlagos", "Velocitta", "Auto Club Speedway"]

var tracks = ["Interlagos", "Velocitta", "Auto Club Speedway"]
```

```swift
// Ways to declare an array
var tracks: Array<String> = ["Interlagos", "Velocitta", "Auto Club Speedway"]

var tracks: [String] = ["Interlagos", "Velocitta", "Auto Club Speedway"]

var tracks = ["Interlagos", "Velocitta", "Auto Club Speedway"]
```

```
// Ways to declare an array
var tracks: Array<String> = ["Interlagos", "Velocitta", "Auto Club Speedway"]

var tracks: [String] = ["Interlagos", "Velocitta", "Auto Club Speedway"]

var tracks = ["Interlagos", "Velocitta", "Auto Club Speedway"]
```

Value	Interlagos	Velocitta	Auto Club Speedway
Index	0	1	2

```swift
// Declaring an ordered collection
var visitedContries: Array<String> = ["Brazil", "USA", "Sweden", "Colombia"]

// Accessing an element using its index
print(visitedContries[0])         "Brazil"

// Appending an element to the collection
visitedContries.append("Italy")

// Inserting an element at a specific index
visitedContries.insert("Indonesia", at: 5)

// Getting the number of elements
visitedContries.count             6
```

```swift
// Declaring an ordered collection
var visitedContries: Array<String> = ["Brazil", "USA", "Sweden", "Colombia"]

// Accessing an element using its index
print(visitedContries[0])         "Brazil"

// Appending an element to the collection
visitedContries.append("Italy")

// Inserting an element at a specific index
visitedContries.insert("Indonesia", at: 5)

// Getting the number of elements
visitedContries.count             6
```

```swift
// Declaring an ordered collection
var visitedContries: Array<String> = ["Brazil", "USA", "Sweden", "Colombia"]

// Accessing an element using its index
print(visitedContries[0])         "Brazil"

// Appending an element to the collection
visitedContries.append("Italy")

// Inserting an element at a specific index
visitedContries.insert("Indonesia", at: 5)

// Getting the number of elements
visitedContries.count             6
```

```swift
// Declaring an ordered collection
var visitedContries: Array<String> = ["Brazil", "USA", "Sweden", "Colombia"]

// Accessing an element using its index
print(visitedContries[0])         "Brazil"

// Appending an element to the collection
visitedContries.append("Italy")

// Inserting an element at a specific index
visitedContries.insert("Indonesia", at: 5)

// Getting the number of elements
visitedContries.count             6
```

```swift
// Declaring an ordered collection
var visitedContries: Array<String> = ["Brazil", "USA", "Sweden", "Colombia"]

// Accessing an element using its index
print(visitedContries[0])         "Brazil"

// Appending an element to the collection
visitedContries.append("Italy")

// Inserting an element at a specific index
visitedContries.insert("Indonesia", at: 5)

// Getting the number of elements
visitedContries.count             6
```

```
// Declaring an ordered collection
var visitedContries: Array<String> = ["Brazil", "USA", "Sweden", "Colombia"]

// Accessing an element using its index
print(visitedContries[0])          "Brazil"

// Appending an element to the collection
visitedContries.append("Italy")

// Inserting an element at a specific index
visitedContries.insert("Indonesia", at: 5)

// Getting the number of elements
visitedContries.count              6
```

Hands on

Dictionary

```swift
// Ways to declare a dictionary
var airports: Dictionary<String,String> = ["CGH":"São Paulo", "CWB":"Curitiba"]

var airports: [String:String] = ["CGH":"São Paulo", "CWB":"Curitiba"]

var airports = ["CGH":"São Paulo", "CWB":"Curitiba"]
```

```
// Ways to declare a dictionary
var airports: Dictionary<String,String> = ["CGH":"São Paulo", "CWB":"Curitiba"]

var airports: [String:String] = ["CGH":"São Paulo", "CWB":"Curitiba"]

var airports = ["CGH":"São Paulo", "CWB":"Curitiba"]
```

```swift
// Ways to declare a dictionary
var airports: Dictionary<String,String> = ["CGH":"São Paulo", "CWB":"Curitiba"]

var airports: [String:String] = ["CGH":"São Paulo", "CWB":"Curitiba"]

var airports = ["CGH":"São Paulo", "CWB":"Curitiba"]
```

```swift
// Ways to declare a dictionary
var airports: Dictionary<String,String> = ["CGH":"São Paulo", "CWB":"Curitiba"]

var airports: [String:String] = ["CGH":"São Paulo", "CWB":"Curitiba"]

var airports = ["CGH":"São Paulo", "CWB":"Curitiba"]
```

```swift
// Declaring an unordered collection
var countryCode: [String:Int] = ["BR":55, "US":1, "IT":39, "IN":62]

// Accessing an element using its key
countryCode["IT"]            39

// Adding an element to the collection
countryCode["UK"] = 44

// Printing the collection
print(countryCode)          ["IT": 39, "BR": 55, "US": 1, "UK": 44, "IN": 62]
```

```swift
// Declaring an unordered collection
var countryCode: [String:Int] = ["BR":55, "US":1, "IT":39, "IN":62]

// Accessing an element using its key
countryCode["IT"]            39

// Adding an element to the collection
countryCode["UK"] = 44

// Printing the collection
print(countryCode)          ["IT": 39, "BR": 55, "US": 1, "UK": 44, "IN": 62]
```

```
// Declaring an unordered collection
var countryCode: [String:Int] = ["BR":55, "US":1, "IT":39, "IN":62]

// Accessing an element using its key
countryCode["IT"]            39

// Adding an element to the collection
countryCode["UK"] = 44

// Printing the collection
print(countryCode)           ["IT": 39, "BR": 55, "US": 1, "UK": 44, "IN": 62]
```

```swift
// Declaring an unordered collection
var countryCode: [String:Int] = ["BR":55, "US":1, "IT":39, "IN":62]

// Accessing an element using its key
countryCode["IT"]            39

// Adding an element to the collection
countryCode["UK"] = 44

// Printing the collection
print(countryCode)          ["IT": 39, "BR": 55, "US": 1, "UK": 44, "IN": 62]
```

```
// Declaring an unordered collection
var countryCode: [String:Int] = ["BR":55, "US":1, "IT":39, "IN":62]

// Accessing an element using its key
countryCode["IT"]            39

// Adding an element to the collection
countryCode["UK"] = 44

// Printing the collection
print(countryCode)          ["IT": 39, "BR": 55, "US": 1, "UK": 44, "IN": 62]
```

Hands on

Stock register

name = Apple

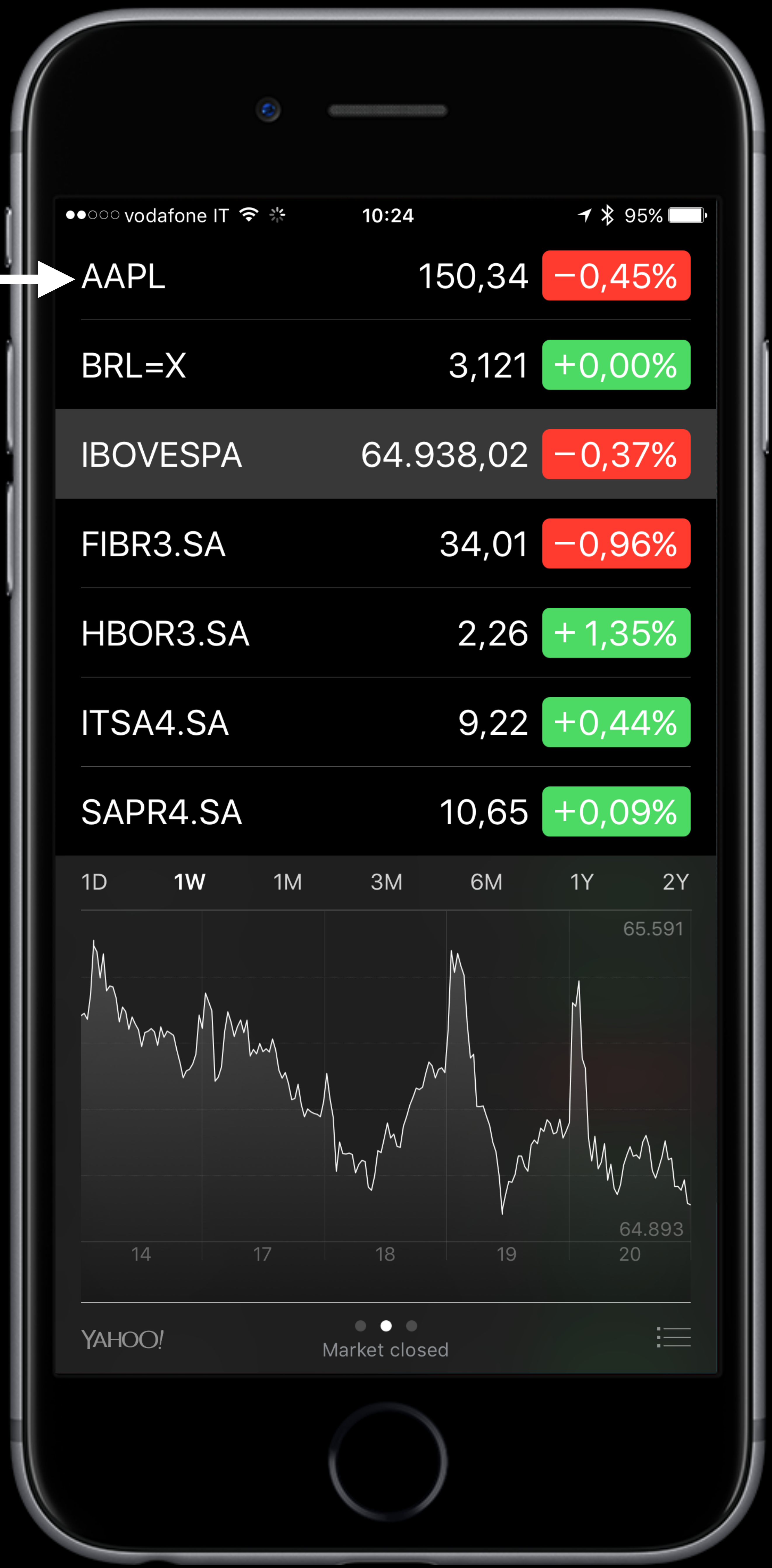
code = AAPL

value = 150,34

variation = 0,45%

volume = US$ 4B

...



AAPL	150,34	−0,45%
BRL=X	3,121	+0,00%
IBOVESPA	64.938,02	−0,37%
FIBR3.SA	34,01	−0,96%
HBOR3.SA	2,26	+1,35%
ITSA4.SA	9,22	+0,44%
SAPR4.SA	10,65	+0,09%

Code version

```
let aaplStock = [
"code" : "AAPL",
"value" : "150,34",
"variation" : "0,45%",
"volume" : "US$ 4B"
]
```