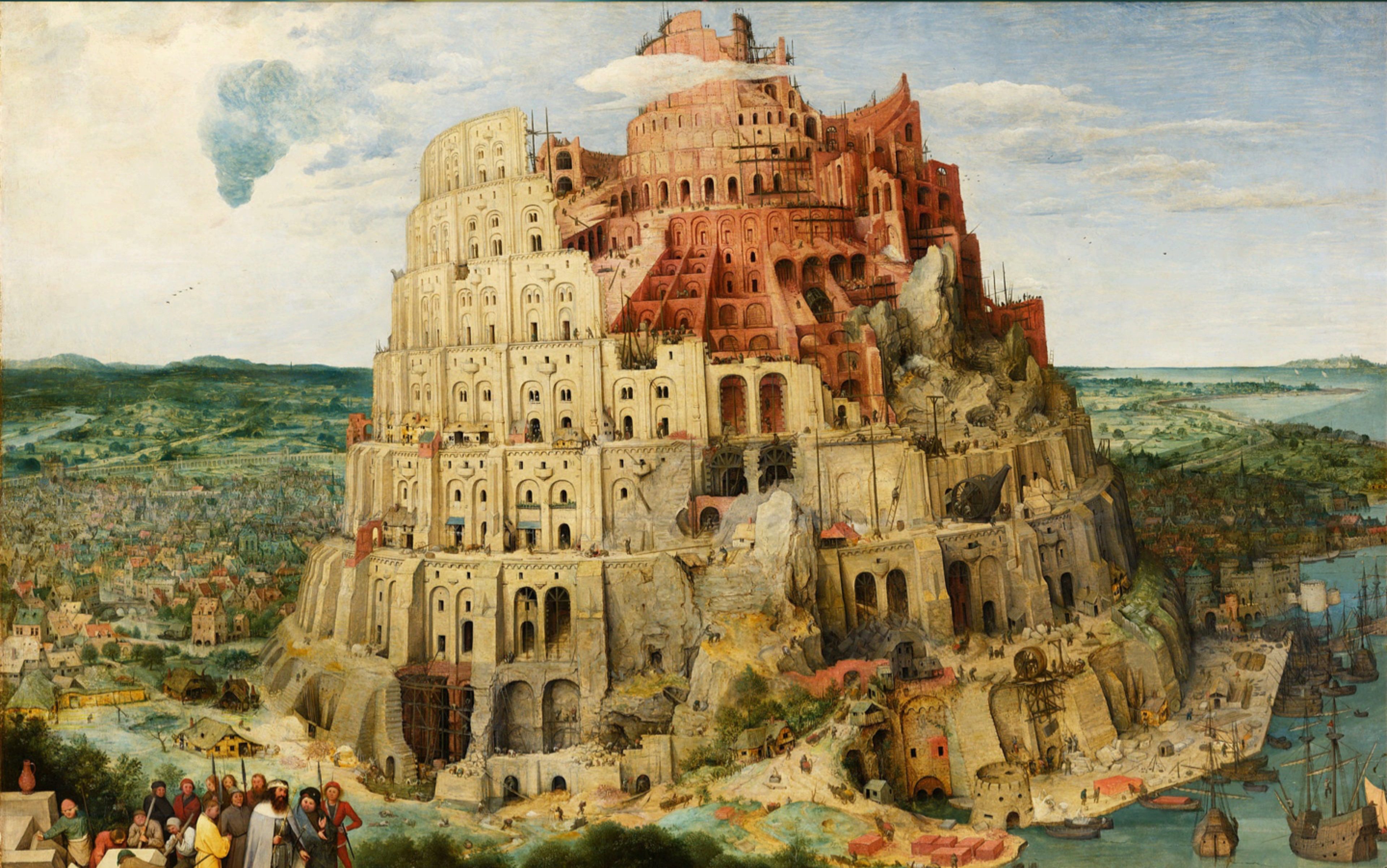


Multilingualism in your applications

Fabrice.Kordon@lip6.fr



Multilingualism, why?



Multilingualism, why?



Multilingualism, why?

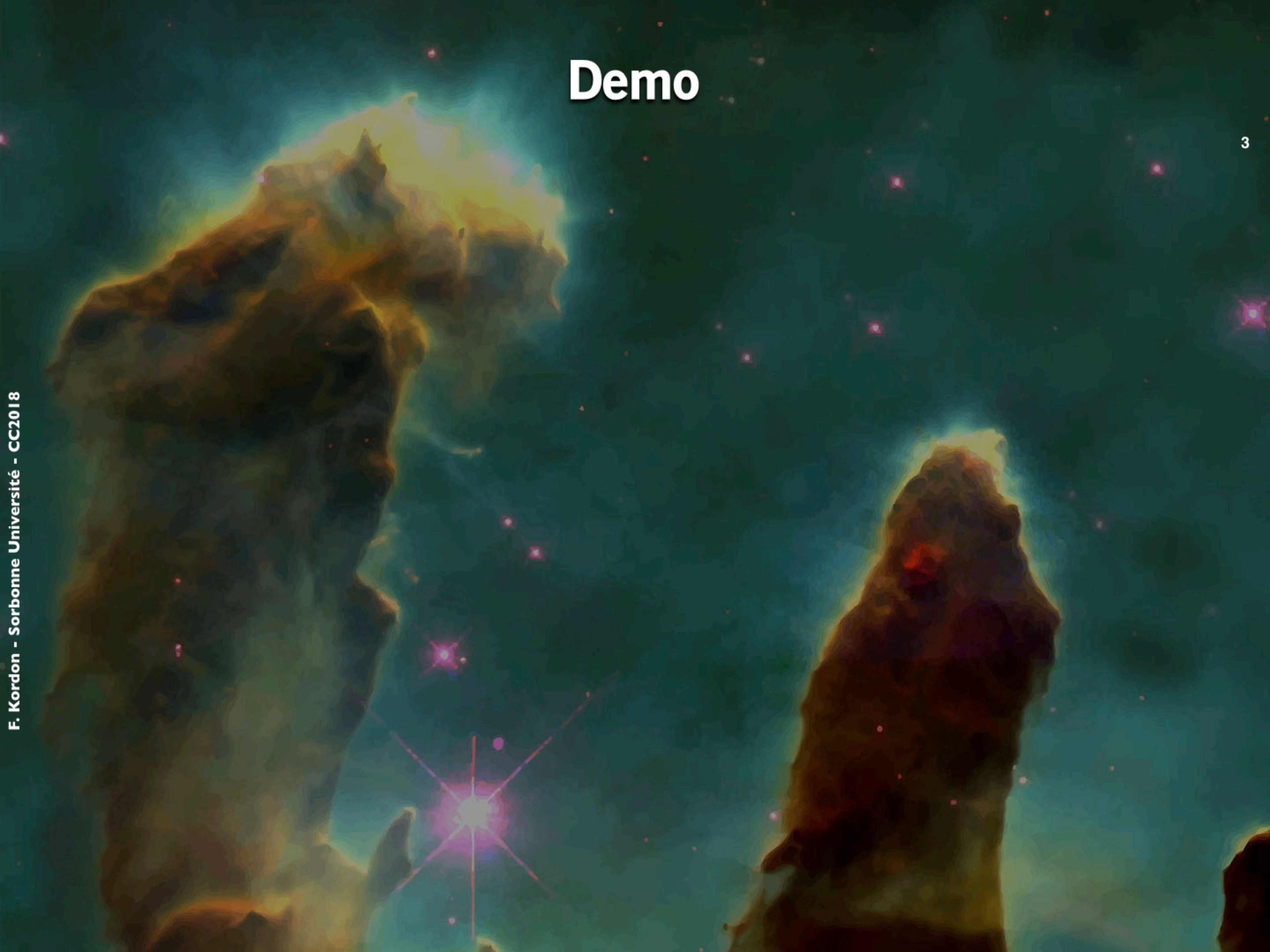


A good reason...

It is better to speak like your users




Demo



Steps


Step 1

-  Create a «localized» file of type string
 - ▶ **Alternative, create a «stringDict» file**

Step 2

-  Add languages in the project

Step 3

-  Translate...

Step 4

-  Implement

Steps

Step 1

- Create a «localized» file of type string
 - ▶ Alternative, create a «stringDict» file

Step 2

- Add language



About order

you may mix steps 2 and 3

Step 3

- Translate...



Caution...

Prepare your displays appropriately (size, consider reading direction, etc.)

Step 4

- Implement

Step 1 — creating the file



Step 1 — populate this file

6



Couple key/value

- 📍 The key to be referenced in a dictionary
- 📍 inserts possibles
 - ▶ **%@, %d, %x, etc.**



In out case...

```
/*  
  Message.strings  
  Babel  
  
  Created by Fabrice Kordon on 20/10/2018.  
  Copyright © 2018 Sorbonne Université. All rights reserved.  
*/  
"Hello!" = "Hello!";
```


Step 2 — add languages



Step 3 — Translate

```
/*  
  Message.strings  
  Babel  
  
  Created by Fabrice Kordon on 20/10/2018.  
  Copyright © 2018 Sorbonne Université. All rights reserved.  
*/  
"Hello!" = "Bonjour!";
```


Step 3 — Translate

```
/*  
  Message.strings  
  Babel  
  
  Created by Fabrice Kordon on 20/10/2018.  
  Copyright © 2018 Sorbonne Université. All rights reserved.  
*/  
"Hello!" = "¡Hola!";
```


Step 3 — Translate

```
/*  
  Message.strings  
  Babel  
  
  Created by Fabrice Kordon on 20/10/2018.  
  Copyright © 2018 Sorbonne Université. All rights reserved.  
*/  
"Hello!" = "卓悦";
```


Step 3 — Translate

```
/*  
  Message.strings  
  Babel  
  
  Created by Fabrice Kordon on 20/10/2018.  
  Copyright © 2018 Sorbonne Université. All rights reserved.  
*/  
"Hello!" = "صباح الخير";
```


Step 4 — ViewController



Sake of simplicity...

Code located in a
ViewController

Step 4 — ViewController

```
import UIKit

class ViewController: UIViewController {

    private let label = UILabel()
    private let number = UILabel()
    private let date = UILabel()

    override func viewDidLoad() {
        super.viewDidLoad()
        self.view = UIView()
        self.view.backgroundColor = UIColor.white
        self.view.frame = UIScreen.main.bounds
        // A localized text
        label.textAlignment = .center
        label.frame = CGRect(x: 10, y: 100,
                             width: UIScreen.main.bounds.width - 20,
                             height: 60)
        label.font = UIFont.boldSystemFont(ofSize: 22)
        label.text = Bundle.main.localizedString(forKey: "Hello!",
                                                  value: "",
                                                  table: "Message")

        label.textColor = UIColor.red
        self.view.addSubview(label)
    }
}
```


Step 4 — ViewController

```
// A localized number
number.textAlignment = .center
number.frame = CGRect(x: 10, y: 200,
                      width: UIScreen.main.bounds.width - 20,
                      height: 60)

number.font = UIFont.boldSystemFont(ofSize: 22)
let numFormat = NumberFormatter()
numFormat.numberStyle = .decimal
let rndNum = Double.random(in: 0.0 ... 100000.0)
print("rndNum= \(rndNum)")
number.text = numFormat.string(from: NSNumber(value: rndNum))
number.textColor = UIColor.orange
self.view.addSubview(number)
// A localized date
date.textAlignment = .center
date.frame = CGRect(x: 10, y: 300,
                    width: UIScreen.main.bounds.width - 20,
                    height: 60)

date.font = UIFont.boldSystemFont(ofSize: 22)
let dateFormat = DateFormatter()
dateFormat.dateStyle = .medium
date.text = dateFormat.string(from: Date())
date.textColor = UIColor.brown
self.view.addSubview(date)
```

}

}

As a conclusion...

Multilingualism is easy...



But you still need to translate

