

# Objective-C, minimal survival guide

Fabrice.Kordon@lip6.fr

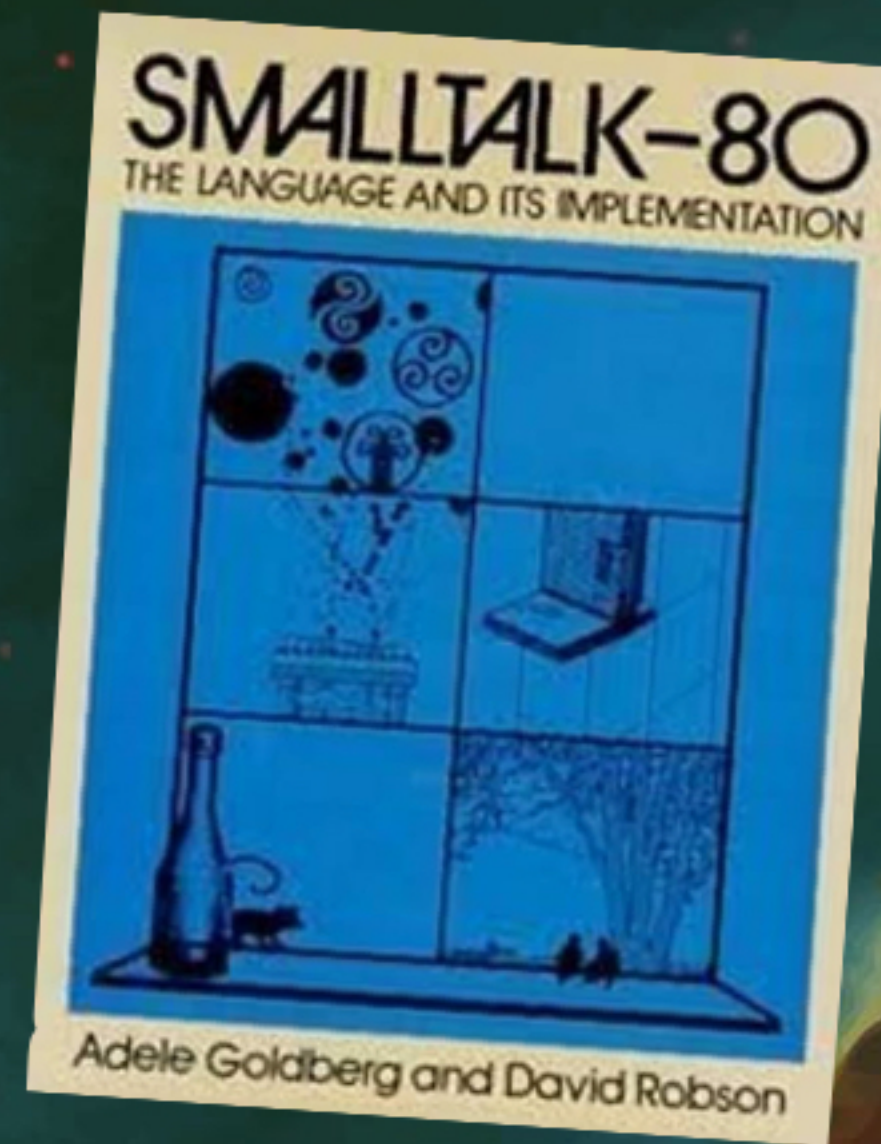




# As an introduction...

## History

- 1983
- Inspired from C and smalltalk
- First main use in NextStep (NeXT)



## We will see

- Principles
- Main syntactical constructions
- The NSString class
- ▶ You will need it rapidly





# Principles

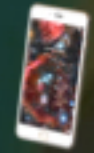


## Based on C



## Strict superset of C

- New syntax, new types
  - ▶ «[c m]», @property, @interface, @implementation, @synthesize...
  - ▶ id, class, selector...
- Simple inheritance, «protocols» (delegation = framework)
- Relatively strong typing (compared to C), introspection mechanisms...



## Two sorts of methods in classes

- Class methods
- Instance methods



# Methods



## Instances catch «instance methods»

- (`id`) `init`;
- (`float`) `temperature`;
- (`void`) `kekchose`;



## Classes catch «class methods»

- + (`id`) `alloc`;
- + (`UIDevice*`) `currentDevice`;



# Objective-C handles messages only

5



## Syntax

```
[dest message];  
[dest message:argument];  
[dest message:arg1 andAlso:arg2];
```



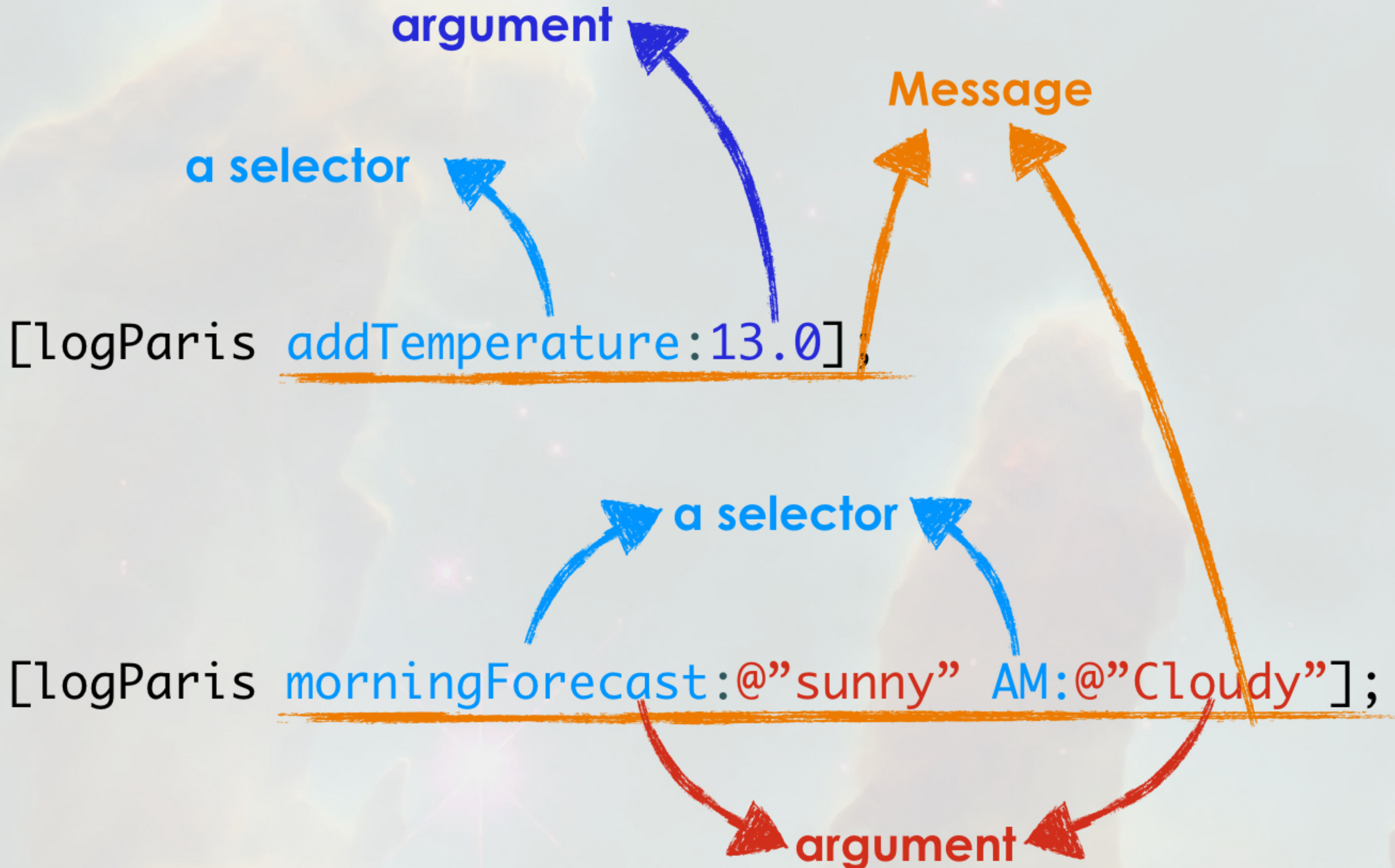
## Examples

```
Forecast *logParis;
```

```
[logParis tomorrowForecast];  
[logParis addTemperature:13.0];  
[logParis morningForecast:@"sunny" AM:@"cloudy"];
```



# Structure of an Objective-C message





# The selector «type»

 **∃ a way to create selectors**

 **Quite useful to point-out methods**

```
SEL sel1 = @selector(coucouLesGars);  
SEL sel1 = @selector(totoEstContent:);  
SEL sel2 = @selector(tempsMatin:ApresMidi:);
```

 **Useful for introspection too**

 Dynamic check for a selector

```
– (BOOL) respondsToSelector:(SEL)sel1;
```



# Produce a trace on the console



## NSLog

- Parameter: NSString
- Concatenations/substitutions handled par the NSString class




## Example

```
NSLog(@"Toto est content");
```





# NSString

## Strings, NSString rather than char\*

-  This is a class (Unicode support)

## From C to Objective-C

-  C: "Hello World"
-  Objective-C: @"Hello World"


```
NSString *myString = @"Hello World";
```



# NSString manipulation

10

## NSString cannot be modified

 When necessary, use NSMutableString

```
NSString *maChaine = @"Hello World";
```

```
NSString *meteo = @"sunny";
```

```
NSString *msg = [NSString stringWithFormat:@"it is %@", meteo];
```

```
NSLog(@"Wind goes %d km/h : %d on the Beaufort scale",  
      [logParis wind],  
      [meteo equivBeaufort:[logParis wind]]);
```

```
NSString *myString = @"Hello";
```

```
NSString *fullString;
```

```
fullString = [myString stringByAppendingString:@" world!"];
```



To get more...

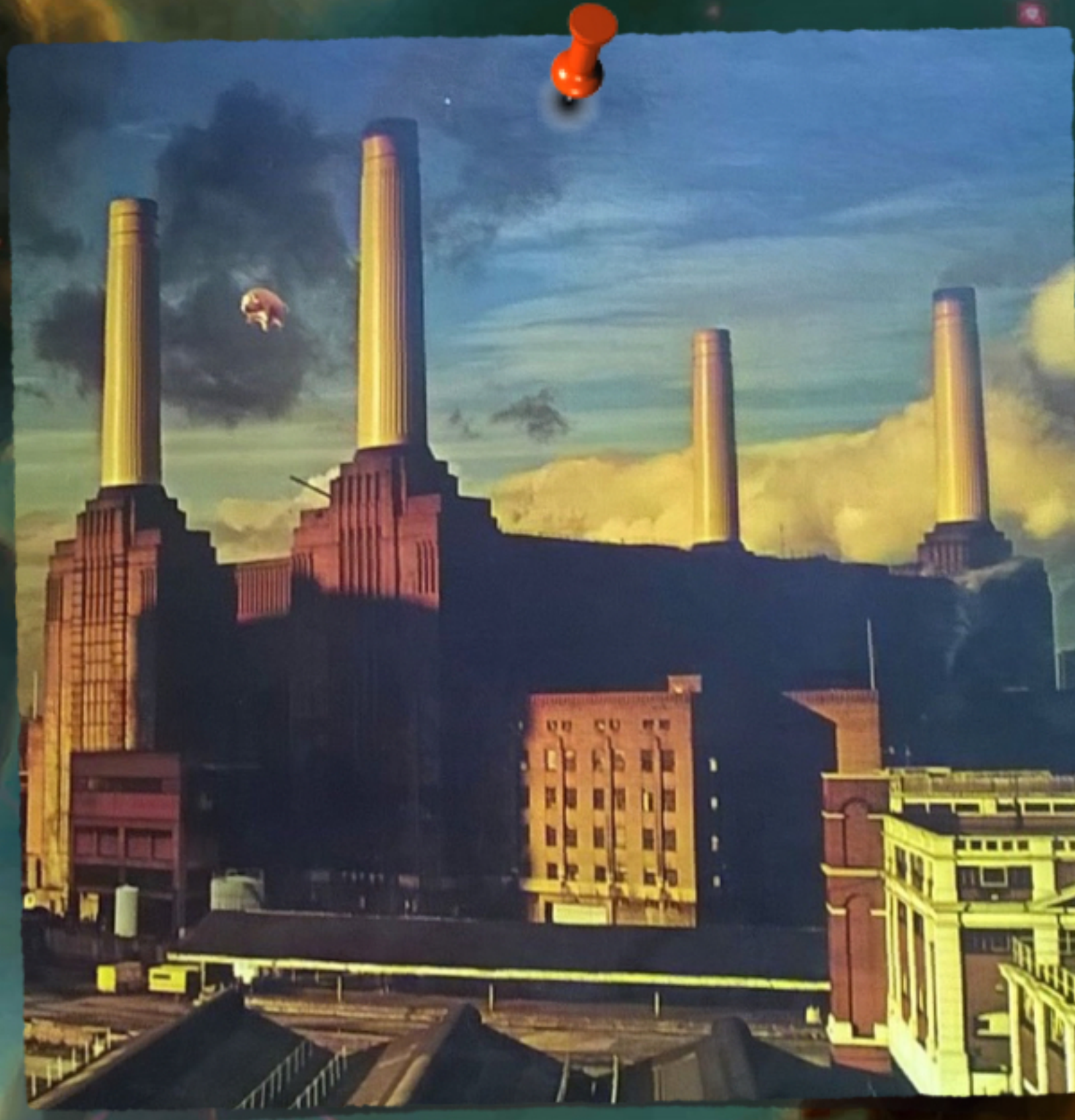




# As a conclusion...



You (almost 😊) have basics to build your first Objective-C application alone





# One more thing... about my Objective-C writing

12

## Historical vision

- Very «pure» vision of objects (method call = message)
- Several notations cohabit
  - ▶ À la «pure Objective-C» : [ and ] everywhere
  - ▶ À la «C» : dotted notation
- You may mix both writing

## in iOS, you have both C and Objective-C

- My choice of writing iOS programs
- Objective-C: Always respect conventions [ and ]
- C: write C

## You do what you want...

- People like to mix...



# One more more thing...

## What's next for Objective-C?

13

### 📱 Swift intended to replace Objective-C

- 👤 Apple engineers still used Objective-C until Swift 3

### 📱 But...

- 👤 Huge Legacy (remind the 2M applications)
- 👤 TIOBE index over the past 15 years

	2003	2008	2013	2018	last year
Objective-C	56	40	3	15	18 ↑ 10
Swift	-	-	-	15	13 ↓ 15

### 📱 Things are more complex

- 👤 Swift and Objective-C will cohabit a while



# One more more thing...

## What's next for Objective-C?

### Swift intended to replace Objective-C

- Apple engineers still used Objective-C until Swift 3



**Sorry,  
you must know both!**

### But...

- Huge Legacy (remember)
- TIOBE index over the past 15 years

	2003	2008	2013	2018	last year
Objective-C	56	40	3	15	18 ↑ 10
Swift	-	-	-	15	13 ↓ 15

### Things are more complex

- Swift and Objective-C will cohabit a while