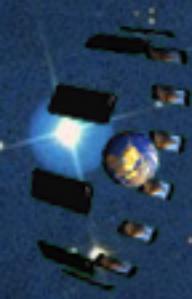


Guide minimum de survie en Objective-C



Fabrice.Kordon@lip6.fr



En guise d'introduction...

2

■ Historique

- 1983
- S'inspire de C et de smalltalk
- Premier utilisateur: NextStep (NeXT)

■ Nous aborderons...

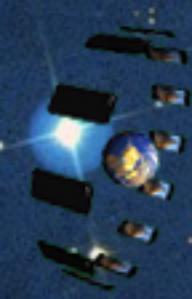
- Principes du langage
- Principaux éléments de syntaxe
- La classe NSString

► Vous en aurez rapidement besoin



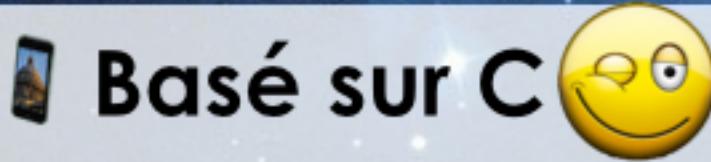
Objective-C, concepts de base

- ▶ Méthodes de classes et d'instances
- ▶ Communication entre classes



Principes d'Objective-C

4



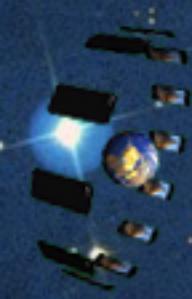
■ Basé sur C

■ Sur ensemble strict de C

- Nouvelle syntaxe, nouveaux types
 - ▶ «[c m]», @property, @interface, @implementation, @synthesize, ...
 - ▶ id, class, selector
- Héritage simple, «protocole» (délégation = framework)
- Typage (relativement) fort (par rapport à C), mécanismes d'introspection...

■ Classes et instances sont des objets

- Méthodes de classes
- Méthodes d'instances



Les méthodes

5

■ Les instances répondent à des méthodes d'instances

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

■ Les classes répondent à des méthodes de classe

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

*Usage régulier
dans les API d'iOS*



En objective-C, tout est message

6



Syntaxe

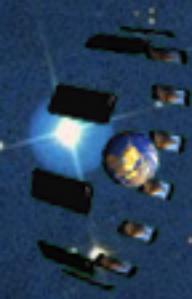
```
[dest message];  
[dest message:argument];  
[dest message:arg1 etAussi:arg2];
```



Exemple

```
meteo *logParis;
```

```
[logParis tempsPourDemain];  
[logParis ajoutTemperature:13.0];  
[logParis tempsMatin:@"beau" apresMidi:@"nuages"];
```



Morphologie d'un message

7

```
[logParis ajoutTemperature:13.0];
```

```
[logParis tempsMatin:@“Beau” apresMidi:@“Nuages”];
```

Morphologie d'un message

7

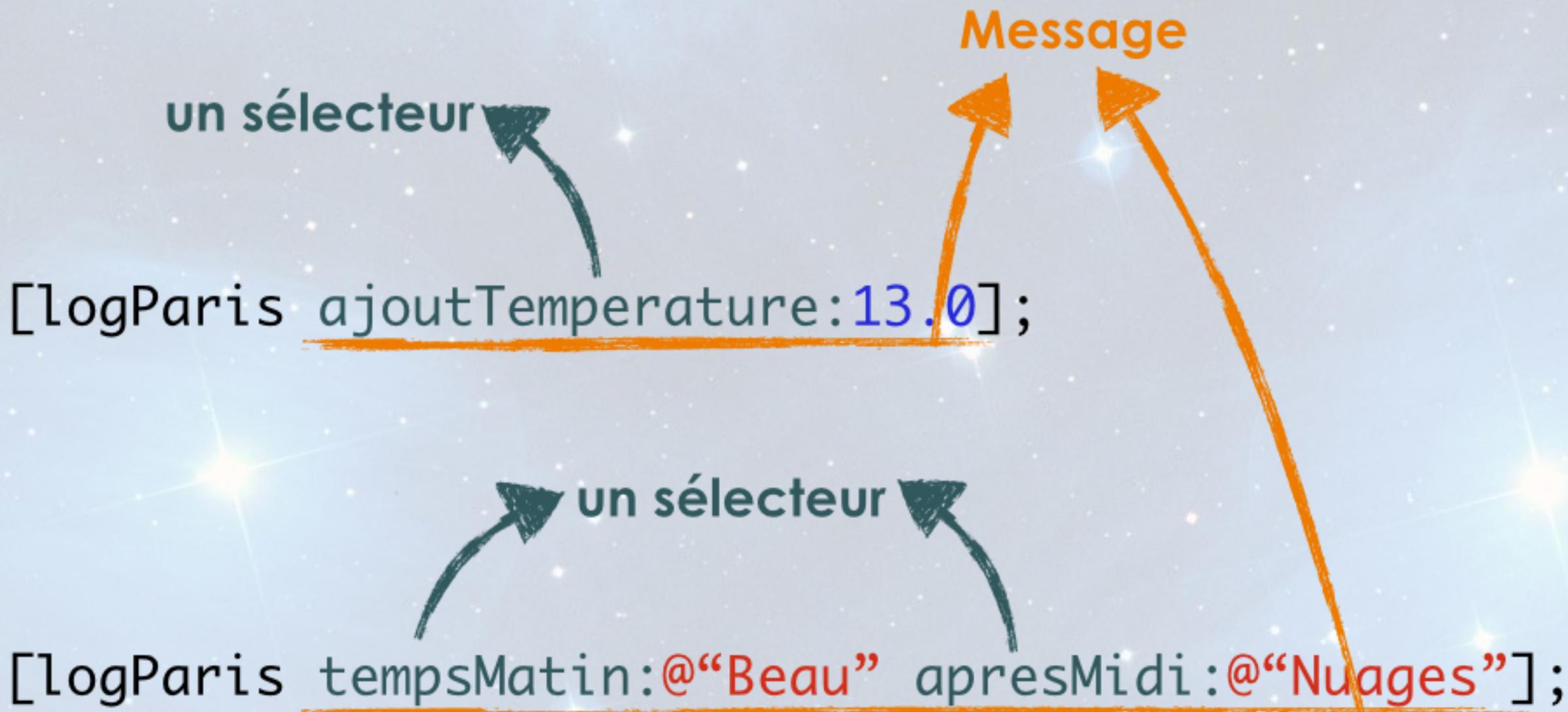
Message

[logParis ajoutTemperature:13.0];

[logParis tempMatin:@“Beau” apresMidi:@“Nuages”];

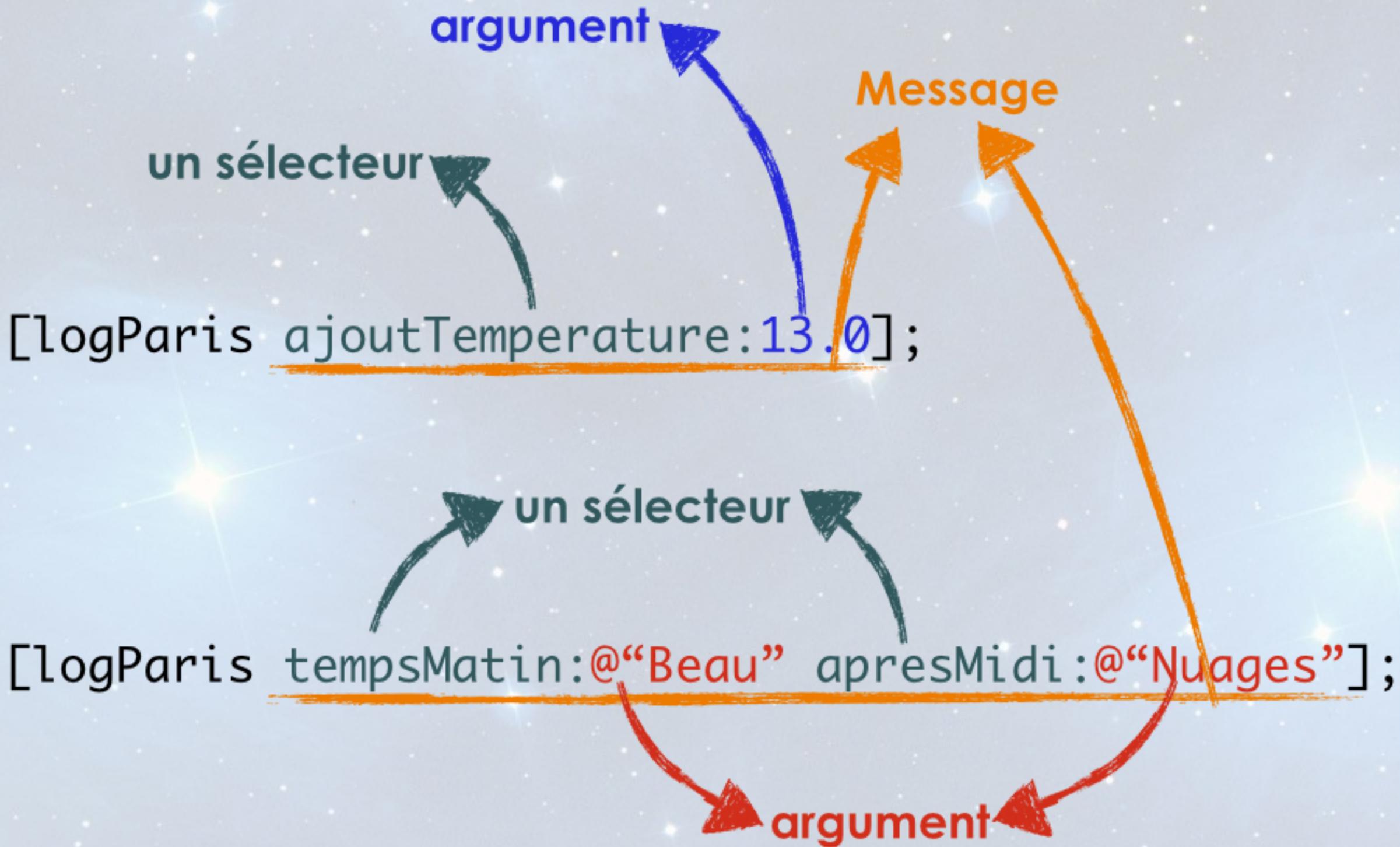
Morphologie d'un message

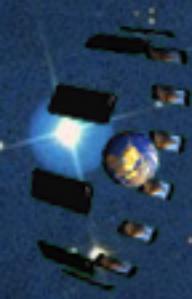
7



Morphologie d'un message

7





Le «type» sélecteur

8

■ Mécanisme de construction de sélecteurs

■ Extrêmement utile pour référencer des méthodes

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

■ Utile pour l'introspection aussi

- (BOOL) respondToSelector:(SEL)sel;

Premiers éléments utiles

- ▶ **NSLog**
- ▶ **NSString**

Générer une trace sur la console...

10

Instruction NSLog

- Prend une NSString en paramètre
- Concaténation/substitutions gérées par NSString (cf plus loin)

Exemple

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



Les chaînes de caractères

11



NSString plutôt que char*

- C'est une classe (support d'Unicode)
- Toujours utilisé...



Évolution de la syntaxe

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

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

Manipulation des NSString

Les NSString ne sont pas modifiables!!!

- Il existe les NSMutableString pour cela

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

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

```
NSString *msg = [NSString stringWithFormat:@"Il fait %@", meteo];
```

```
NSLog(@"Le vent souffle à %d km/h : %d sur l'échelle de Beaufort",
      [logParis vent],
      [meteo equivalentBeaufort:[logParis vent]]);
```

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

```
NSString *fullString;
```

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



Pour le reste...

En guise de conclusion...

■ Vous disposez (presque 😊) des bases pour construire votre première application en Objective-C



à propos de mon style d'écriture en Objective-C

14

■ Historique...

- Vision objet assez «pure» (méthodes = messages)
- Différentes notations
 - ▶ «À la objective C» : notation entre [et]
 - ▶ «À la C» : notation pointée
 - ▶ «Mixte»

■ iOS, c'est de l'Objective C mais aussi parfois du C

- Choix d'écriture (style)
- Respecter les conventions [] en Objective C
- Faire du C sinon

■ Mais vous faites ce que vous voulez...

- Et vous verrez que les gens aiment bien mélanger

One more more thing... Objective-C & Swift?

15

L'un va-t-il remplacer l'autre?

- Dans l'esprit d'Apple sans doute... mais c'est plus compliqué



Etat des lieux

- Un existant colossal (rappel, 900K applications!)
 - ▶ TIOBE Index (Objective-C): 2004 = 39^{ème} — 2009 = 30^{ème} — 2014 = 3^{ème} (~10%)
 - ▶ TIOBE Index (Swift) : introduction en 18^{ème} position (coté buzz?)
- Aaron Hillegass (Juin 2014)
 - ▶ «les développeurs iOS ont besoin de connaître Objective-C»
 - ▶ Aspects patrimoine (iOS contient encore des couches en C)
 - ▶ Facilité de lecture?
- Il y aura sans doute une longue cohabitation des deux langages

Désolé, vous devez connaître les deux



