Types de variables
Structures de contrôle

OCInformatique

1

Types de variables

2

Deux «familles»

primitifs

int, float, BOOL, ... (dans le stack)

objets

NSString, id, NSArray, ... (dans le heap)

3

int

Nombre entier

- **Exemples: 12, 0, -525, 10995435**

float

Nombre à virgule

- **Exemples: 5.65, 0.457, -23.356, 432153252.65463**

4

BOOL

Opérateur booléen

- **VRAI ou FAUX, 1 ou 0**
- **en Objective-C: YES or NO**
- **utile pour effectuer des tests**

5

NSString

Chaîne de caractères fixe

- **non modifiable**

NSMutableString

Chaîne de caractère mutable (sous-classe de NSString)

- **idem NSString, mais modifiable**

6

NSArray

Tableau fixe

- **unidimensionnel**
- **non modifiable**
- **contient des objets**

NSMutableArray

Tableau mutable (sous-classe de NSArray)

- **idem NSArray, mais modifiable**

7

id

Type très important en Objective-C

- **pointeur vers n'importe quel objet, ou plutôt**
- **pointeur vers un objet dont on ne connaît pas forcément la classe**

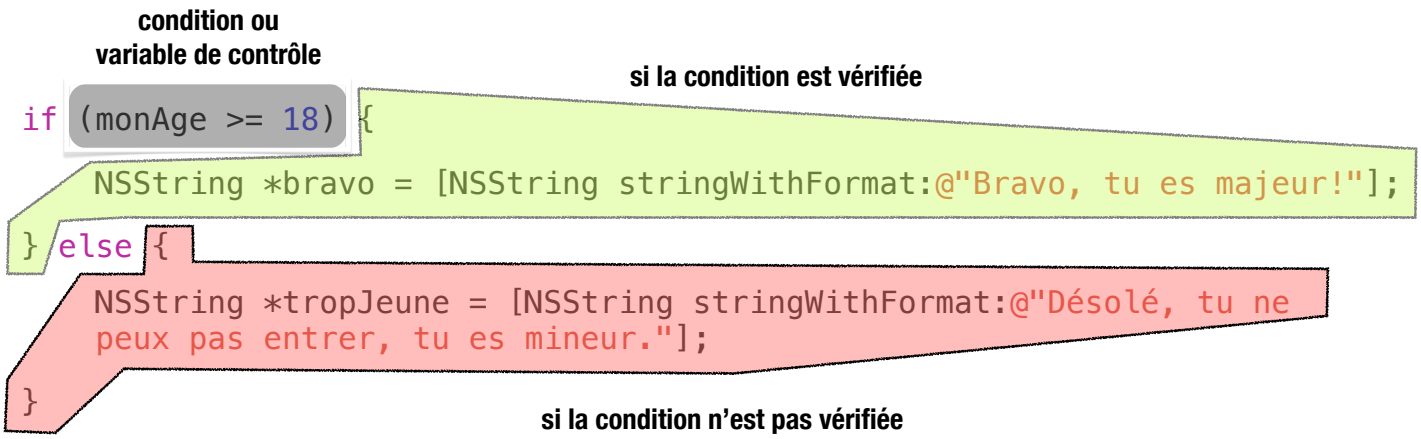
8

Structures de contrôle

if ( ) {...} else {...}

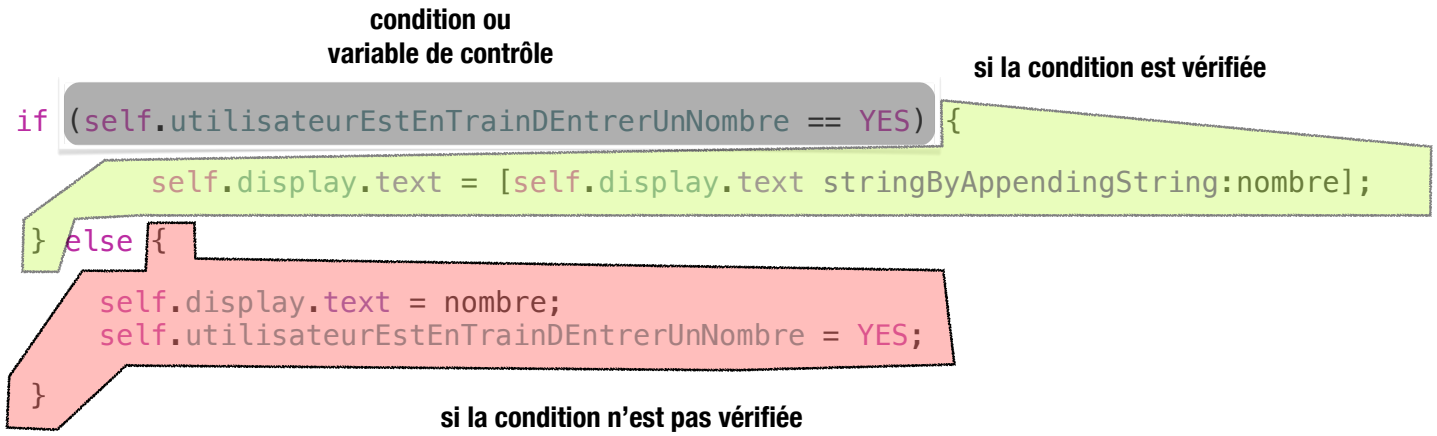
Exemple

de structure de contrôle en if else

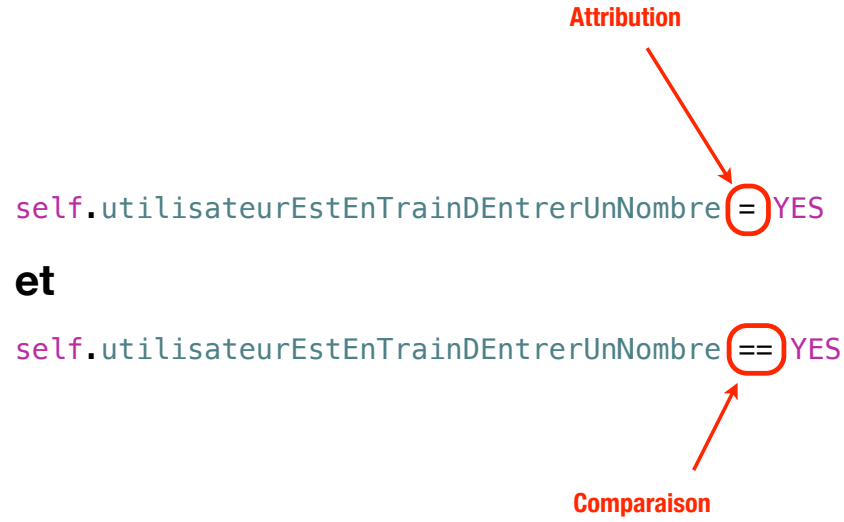


BOOL // Obj-C: YES, NO; NO = zéro, YES = non-zéro

```
@property (nonatomic) BOOL utilisateurEstEnTrainDEntrerUnNombre;
```



Attention de distinguer!



BOOL

```
@property (nonatomic) BOOL utilisateurEstEnTrainDEntrerUnNombre;
```

c'est la version «dot notation», mais en réalité on utilise le getter de la @property:

```
[self utilisateurEstEnTrainDEntrerUnNombre] == YES;
```

On vérifie que l'utilisateur est bien en train d'entrer un nombre

```
if (self.utilisateurEstEnTrainDEntrerUnNombre == YES) {
        self.display.text = [self.display.text stringByAppendingString:nombre];
 } else {
     self.display.text = nombre;
     self.utilisateurEstEnTrainDEntrerUnNombre = YES;
 }
```

On attribue la valeur YES à la propriété `utilisateurEstEnTrainDEntrerUnNombre`

c'est la version «dot notation», mais en réalité on utilise le setter de la @property:

```
[self setUtilisateurEstEnTrainDEntrerUnNombre:YES];
```


Au lieu de

```
if (self.utilisateurEstEnTrainDEntrerUnNombre == YES) {
                        […]
}
```

on va utiliser

```
if (self.utilisateurEstEnTrainDEntrerUnNombre) {
        […]
}
```

et dans la négative

```
if (!self.utilisateurEstEnTrainDEntrerUnNombre) {
        […]
}
```

! signifie «non ...»
utilisé avec un booléen, il donne le contraire de sa valeur (de YES à NO ou de NO à YES)