



**IMT Lille Douai**  
École Mines-Télécom  
IMT-Université de Lille



# Apprendre par la pratique la programmation mobile

Sauvegarder des données dans iOS...

Anthony Fleury ([anthony.fleury@imt-lille-douai.fr](mailto:anthony.fleury@imt-lille-douai.fr))  
IMT Lille Douai - Année 2020 - 2021



# Les données dans iOS



# Deux choix à votre disposition...

- iOS vous offre deux manières de sauvegarder des données dans vos applications de manière simple (en plus de la création de vos propre fichiers) :
  - Les NSUserDefaults, un fichier conservé dans votre répertoire d'application et stockant des données sous forme Clé - Valeur.
  - Core Data, une interface graphique pour créer une base de données sans vous ennuyer avec le SQL, en ayant des objets attachés à vos différentes tables.



# NSUserDefaults



# Charger les NSUserDefaults...

- Les NSUserDefaults vont être associer à votre application en cas de besoin, et le fait de vouloir y accéder va créer le fichier en question. Pour y accéder :

```
let maConfig = UserDefaults.standard
```

- Ensuite, vous pouvez ajouter par exemple une chaine de caractères nommée appliName :

```
maConfig.set("Mon Appli 2048", forKey: "appliName")
```



# Ecrire et relire...

- Une fois écrite, votre valeur peut ensuite être relue de la même manière :

```
let maConfig = UserDefaults.standard
if let nomAppli = maConfig.string(forKey: "appliName")
{
    print(nomAppli)
}
```



# Les différentes méthodes : pour écrire

- Pour écrire les différents types vous avez la fonction suivante :

```
func set(value: Type, forKey: String)
```

- Avec Type qui peut être de différents type (Bool, Int, Double, Float ou alors type Any et donc n'importe quel objet)



# Les différentes méthodes : pour lire

- De la même manière, pour lire, vous avez :

```
func bool(forKey: String) -> Bool
```

```
func integer(forKey: String) -> Int
```

```
func float(forKey: String) -> Float
```

```
func double(forKey: String) -> Double
```

```
func object(forKey: String) -> Any?
```

```
func dataForKey(defaultName: String) -> Data?
```

```
func string(forKey: String) -> String?
```

```
func stringArray(forKey: String) -> [String]?
```

```
func array(forKey: String) -> [Any]?
```

```
func dictionary(forKey: String) -> [String : Any]?
```

- Vous allez principalement utiliser les premières méthodes ainsi que `string`.



# Jouer avec des chaînes ?

- Vous pouvez, pour sauvegarder l'état d'une partie (par exemple pour reprendre entre deux utilisations ou pour sauvegarder les parties avec les meilleurs scores), créer une chaîne de caractères contenant les valeurs des différentes cases :
  - Exemple (pour un 2x2) : let grille = "2,4,8,16"
- Mais comment récupérer ces valeurs ?



# Jouer avec des chaînes ?

- Séparation des chaînes :

```
let grilleSep = grille.components(separatedBy: ",")  
var valeur1: String = grilleSep[0]  
var valeur2: String? = grilleSep[1]
```

- La première chaîne contient 2 et la seconde 4.  
grilleSep.count vous donne le nombre de valeurs et vous pouvez réintégrer cela dans votre tableau.
- Le tableau grilleSep sera : ["2", "4", "8", "16"]



**A vous de jouer...**

**Finissez maintenant votre application...**

**Sauvegardez le nom du joueur ? Son meilleur score ? Encore mieux, en plus des 5 meilleurs scores les grilles qui vous ont fait arriver à ces scores et les réafficher sous forme de grille ?**



**CoreData**

**Pour CoreData, voir le projet Hammer**