

Observable Objects

L'équivalent de @State pour notre modèle de données

Les limites de @State

```
@State var book = Book()
```

Title

Author

Number of pages

Progression

update(progression: 80)



```
@State var book = Book()
```

Title

Author

Number of pages

Progression

Les limites de @State

```
@State var book = Book()
```

Title

Author

Number of pages

Progression

update(progression: 80)

```
@State var book = Book()
```

Title

Author

Number of pages

Progression

ObservableObject

```
class Book: ObservableObject {  
    let title: String  
    let author: String  
    var progress: Float  
  
    init(title: String,  
         author: String,  
         progress: Float) {  
        self.title = title  
        self.author = author  
        self.progress = progress  
    }  
}
```

“ Mon objet est observable :
observe tous les changements
et met à jour la vue
si nécessaire” 🤔

Le *property wrapper* @Published

```
class Book: ObservableObject {  
    let title: String  
    let author: String  
    @Published var progress: Float
```

```
    init(title: String,  
         author: String,  
         progress: Float) {  
        self.title = title  
        self.author = author  
        self.progress = progress  
    }  
}
```

“ Observe les changements
pour cette propriété seulement ”

Le property wrapper @ObservedObject

```
@??? var book = Book()
```

Title

Author

Number of pages

Progress

update(progression: 80)

```
@??? var book = Book()
```

Title

Author

Number of pages

Progress

Le *property wrapper* @StateObject

@??? var book = Book()

Title

Author

Number of pages

Progress

@StateObject var book = Book(...)

Création d'une source de vérité
de type personnalisé

Le *property wrapper* @ObservedObject

@StateObject **var** book = Book(...)  @ObservedObject **var** book: Book



Réception d'une source de vérité
de type personnalisé

Récapitulatif

- Créer une classe qui se conforme au protocole **ObservableObject**
- Publier certaines priorités avec **@Published**
- Créer une unique source de vérité avec **@StateObject**
- Lier des variables à cette source de vérité avec **@ObservedObject**