Haute École Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

2021 – 2022
MOBG5
JLC - SDR

**MOBG5 – Développement mobile**

# Affichage d'une liste d'éléments

*RecyclerView*

**Consignes**

Tous les exercices proposés dans ce TD sont disponibles sur `https://codelabs.developers.google.com/android-kotlin-fundamentals/`.
Nous ne ferons pas tous les exercices de ce tutoriel mais seulement une sélection d'exercices essentiels pour développer une première application.
Lisez les explications présentes dans ce TD et réalisez les exercices en ligne.
Les challenges et devoirs (Homeworks) proposés en ligne sont optionnels.

## 1 Application `TrackMySleepQuality`

### Afficher une liste d'éléments : `RecyclerView`

Le `layout` `ListView` affiche le contenu d'une collection d'éléments dans une liste verticale et qui peut défiler. Avec ce `layout` **tous** les éléments de la collection sont **chargés en mémoire**, même ceux qui ne sont pas visibles à l'écran.

Pour afficher une liste d'éléments tout en optimisant la gestion de la mémoire et en variant les présentations (verticale, horizontale, sous forme de grille), on utilise plutôt un `RecyclerView`.

La particularité de ce `layout` est d'instancier uniquement les éléments de la vue dont il a besoin. Pour implémenter ce `layout` vous allez avoir besoin des classes :

- ▷ `ViewHolder` : décrit la vue d'un élément de la liste ;
- ▷ `Adapter` : crée les `ViewHolder` à afficher et les relient aux données à afficher ;
- ▷ `RecyclerView` : affiche la liste d'éléments.

**Lien vers l'exercice**

07.1 : RecyclerView fundamentals [a]

a. https://codelabs.developers.google.com/codelabs/kotlin-android-training-recyclerview-fundamentals

### Utilisation du Data Binding

Afin de rendre le code plus lisible, le `Data Binding` va être utilisé. Les méthodes `findViewById()` et les déclarations des propriétés utilisées dans le vue ne seront plus nécessaire dans le `ViewHolder`.



Version du document
8 septembre 2021

**Lien vers l'exercice**

07.2 : DiffUtil and data binding with RecyclerView [a]

a. https://codelabs.developers.google.com/codelabs/kotlin-android-training-diffutil-databinding

## Affichage sous forme de grille : le `GridLayout`

Le `RecylcerView` est lié à un `LayoutManager` qui peut gérer la présentation des éléments. Dans cet exercice vous allez créer un `LayoutManager` qui organise les éléments en une grille. Par exemple pour demander au `LayoutManager` d'afficher les éléments en 3 colonnes, vous écrirez :

```
val manager = GridLayoutManager(activity, 3)
```

**Lien vers l'exercice**

07.3 : GridLayout with RecyclerView [a]

a. https://codelabs.developers.google.com/codelabs/kotlin-android-training-grid-layout

## Gérer les événements associés aux éléments

Finalement vous allez ajouter un gestionnaire d'événement `clickListener` afin d'associer une action aux clics effectués sur un élément de la liste.

**Lien vers l'exercice**

07.4 : Interacting with RecyclerView items [a]

a. https://codelabs.developers.google.com/codelabs/kotlin-android-training-interacting-with-items