Android Récapitulatif

Table des matières

**Aucune entrée de table des matières n'a été trouvée.**

# Manifest

Déclare tout ce qu’Android doit savoir sur l’appli’. Elle définit les permissions.

On définit dans le manifest les activités :

<**activity android:name=".NextActivity"**/>

# Implémenter une activité

Quand une classe est appelé, c’est l’évènement onCreate qui est déclenché.

* On définit la vue avec setContentLayout.
* Pour récupérer un élément de la vue on la récupère avec l’ID : finViewById().
* On peut définit un listener au click avec setOnClickListener().

On peutredéfinir un texte avec setText().

# Unités, Layout, Configurations

## Unités :

Taille des éléments : DP

Taille des textes : SP

## Layout :

Définissent la vue, doivent définir au moins :

* Layout\_height
* Layout\_width

Qui peuvent prendre comme valeur wrap\_content (plus petit possible),match\_parent(plus grand possible), une valeur en DP.

Les plus utilisés :

* FrameLayout : n’arrange pas les éléments entre eux, peuvernt se chevaucher.
* LinearLayout permet d’aligner les éléments
* RelativeLayout : positionne où on veut.

## Configurations :

Informations définissant comment fonctionne le périphérique. Ils sont du type : Matériel, Logiciel, Situationnel. Comme taille d’image différente en fonction de la taille d’écran, changer langue, interface différente en fonction de l’orientation de l’écran.

### Image :

En mode project : app/src/main/res 🡺 new android repository. Drawable avec contrainte densité. Les plus utilisés : de XX à Medium. Les images doivent avoir EXACTEMENT le même nom.

On peut redéfinir les marges pour grand écran dans le fichier w820dp, le fichier de base redéfinit les marges à 60dp normalement.

### Langages :

Pour récupérer ce String :

<TextView

Android :text= »@string/hello\_world »

On fait getString(R.string.hello\_world) ;

On doit créer un nouveau string.xml avec la contrainte française. Les nouveaux textes doivent avoir EXACTEMENT le même nom. LKa langue changera en fonction du pays dans lequel elle est téléchargée.

# Rétro-compatible : thèmes et styles

Deux couleurs à choisir :

* Primary : couleur primaire de l’appli’.
* Accent : couleur qui attire l’œil (bouton flottant ?)

### Styles :

Récupères informations des layout pour les mettre dans res/style.

### Thèmes :

Ensemble de style appliqué à une vue.

# Action bar et menu :

Overfow : 3 petits points à droite du menu s’il n’y a pas assez de palce dans le bar, ou placés volontairement.

## Le titre :

setTitle(String) ;

## Ajouter un menu :

Res 🡺 new resource repository : menu sans contrainte.

Dans menu on crée un nouveau fichier main\_menu :

*<?***xml version="1.0" encoding="utf-8"***?>*<**menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"**>  
  
 <**item  
 android:id="@+id/action\_edit"  
 android:title="Edit"  
 android:icon="@android:drawable/ic\_menu\_edit"  
 app:showAsAction="always"** />  
  
 <**item  
 android:id="@+id/action\_add"  
 android:title="Edit"  
 android:icon="@android:drawable/ic\_menu\_add"  
 app:showAsAction="ifRoom"**/>  
  
 <**item  
 android:id="@+id/action\_delete"  
 android:title="Delete"  
 android:icon="@android:drawable/ic\_menu\_delete"  
 app:showAsAction="never"**/>  
</**menu**>

Pour créer un menu on Override onCreateOptionsMenu

@Override  
**public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***main\_menu***, menu);  
 **return true**;  
}