CAI-IP02 SMB116

Les **Layouts**

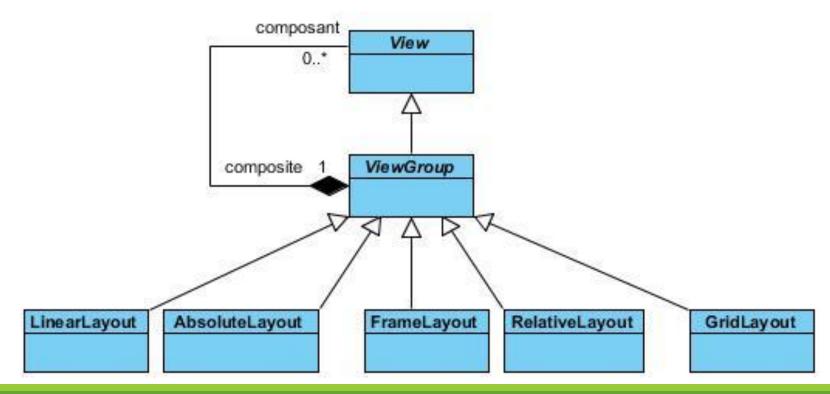






ViewGroup

- ► Container de vues enfants qui gère leur positionnement (Composite)
- ► Les **Layouts** sont des **ViewGroups**





Layouts

- Ce sont des Vues qui gèrent la disposition de vues filles (Composants)
- ▶ Un **Layout** impose aux vues filles un ensemble de contrainte de placement
- Chaque type de Layout implémente une stratégie de placement des vues composantes
- Les Layouts sont définis de manière statique dans des ressources XML (rappel)



Les Différents Layouts

- ▶ FrameLayout : permet de superposer des composants les uns au-dessus des autres de manière à ce qu'un seul ne soit visible à la fois
- ▶ LinearLayout : dispose les composants les uns à la suite des autres dans le sens vertical ou bien horizontal
- ▶ RelativeLayout : dispose les composants relativement les uns par rapport aux autres
- ▶ TableLayout : disposition des composants dans une Table comme en Html
- ▶ GridLayout : disposition des composants dans les cellules d'une grille



Le FrameLayout

► Superposition de composants les uns au-dessus des autres de manière à ce qu'un seul ne soit visible à la fois. (utilisation propriété **visibility**)

▶ Possibilité d'afficher plusieurs composants à la fois en les positionnant en utilisant la

propriété gravity

```
<FrameLayout</pre>
    android:layout width="match parent"
    android:layout height="match parent">
    < ImageView
        android:id="@+id/imageView"
        android:layout width="114dp"
        android:layout height="160dp"
        android:layout gravity="center horizontal"
        app:srcCompat="@android:drawable/btn star big on" />
    <TextView
        android:id="@+id/textView3"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:layout gravity="center"
        android:text="Une étoile"
        android:textSize="24sp" />
</FrameLayout>
```





Le LinearLayout

▶ Placement des composants les uns à la suite des autres dans le sens vertical ou bien horizontal

```
<LinearLayout</pre>
    android:layout width="match parent"
   android:layout height="match parent"
    android:orientation="vertical">
    <TextView
        android:id="@+id/titre"
        android:layout width="match parent"
        android:layout height="wrap content"
        android:text="Titre"
        android:textSize="30sp" />
    <TextView
        android:id="@+id/texte"
        android:layout width="match parent"
        android:layout height="0dp"
        android:layout gravity="top"
        android:layout weight="1"
        android:text="texte"
        android:textSize="18sp"
        android:textStyle="italic" />
    <Button
        android:id="@+id/button2"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:layout gravity="right"
        android:text="Button" />
</LinearLayout>
```



Contraintes applicables à chaque composant:

- layout_width et layout_height: décrit la manière de dimensionner sa case
 - match_parent: taille du parent
 - wrap_content: s'adapte à la taille du contenu
- weight: poids pour le redimensionnement valeur entière XXX
- gravity: placement à l'intérieur de la case
 (left, top, bottom, ..., center, ...)

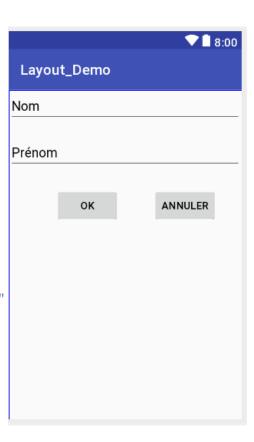


android:text="Prénom" />

Le RelativeLayout (1)

▶ Placement des composants les uns par rapport aux autres

```
<RelativeLayout
                                                   <Button
   android:layout width="match parent"
                                                           android:id="@+id/btnAnnuler"
   android:layout height="match parent"
                                                           android:layout width="wrap content"
   app:layout constraintHorizontal bias="0.0"
                                                           android:layout height="wrap content"
   app:layout constraintLeft toLeftOf="parent"
                                                           android:layout alignParentEnd="true"
   app:layout constraintRight toRightOf="parent";
                                                           android:layout below="@+id/txtPrenom"
                                                           android:layout marginEnd="34dp"
   <EditText
                                                           android:layout marginTop="26dp"
        android:id="@+id/txtName"
                                                           android:text="Annuler" />
        android:layout width="match parent"
        android:layout height="wrap content"
                                                       <Button
        android:layout alignParentStart="true"
                                                           android:id="@+id/btnOk"
        android:layout alignParentTop="true"
                                                           android:layout width="wrap content"
        android:ems="10"
                                                           android:layout height="wrap content"
        android:inputType="textPersonName"
                                                           android:layout alignTop="@+id/btnAnnuler"
        android:text="Nom" />
                                                           android:layout marginEnd="46dp"
    <EditText
                                                           android:layout toStartOf="@+id/btnAnnuler"
        android:id="@+id/txtPrenom"
                                                           android:text="Ok" />
        android:layout width="match parent"
        android:layout height="wrap content"
                                                   </RelativeLayout>
        android:layout alignParentStart="true"
        android:layout below="@+id/txtName"
        android:layout marginTop="19dp"
        android:ems="10"
        android:inputType="textPersonName"
```





Le RelativeLayout (2)

- Contraintes applicables à chaque composant:
 - ▶ layout_width et layout_height: décrit la manière de dimensionner sa case
 - ► match_parent: taille du parent
 - wrap_content: s'adapte à la taille du contenu
 - ▶ layout_below, layout_above : placement relatif à un autre composant
 - layout_alignLeft, layout_alignRight : alignement relatif entre composants
 - ▶ layout_marginTop, layout_marginLeft ... : marges autour du composant

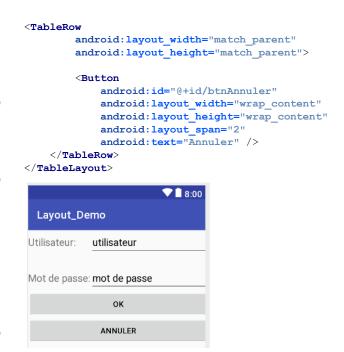


Le TableLayout (1)

- ▶ Positionnement des composants dans des lignes les TableRow
- ▶ le **TableLayout** est similaire à une Table en Html

```
<TableLayout
   android:layout width="match parent"
   android:layout height="match parent">
    <TableRow
        android:layout width="match parent"
       android:layout height="match parent">
        <TextView
           android:id="@+id/lblLogin"
           android:layout width="wrap content"
            android:layout height="wrap content"
            android:text="Utilisateur:"
            android:textSize="18sp" />
        <EditText
            android:id="@+id/txtLogin"
           android:layout width="wrap content"
            android:layout height="wrap content"
            android:ems="10"
            android:inputType="textPersonName"
            android:text="utilisateur" />
   </TableRow>
    <TableRow
        android:layout width="match parent"
       android:layout height="match parent"
        android:layout margin="10dp" />
```

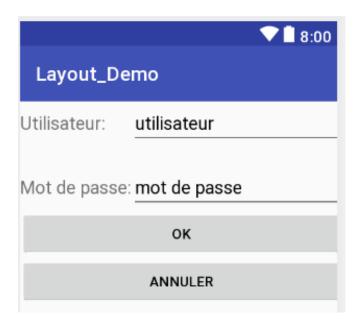
```
<TableRow
    android:layout width="match parent"
    android:layout height="match parent" >
    <TextView
        android:id="@+id/lblPassword"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:text="Mot de passe:"
        android:textSize="18sp" />
    <EditText
        android:id="@+id/editText3"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="mot de passe" />
</TableRow>
<TableRow
    android:layout width="match parent"
    android:layout height="match parent" >
    <Button
        android:id="@+id/btnOk"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:layout span="2"
        android:text="OK" />
</TableRow>
```





Le TableLayout (2)

- ► Contraintes applicables à chaque composant:
- ► Layout_column : numéro de colonne dans la ligne
- ► Layout_span : nombre de colonnes occupées

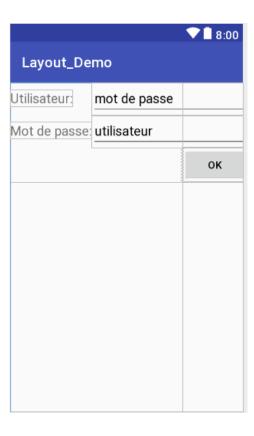




Le GridLayout (1)

Découpage de l'espace en Lignes et colonnes pour obtenir des cellules :

```
<GridLayout
                                                <EditText
   android:layout width="match parent"
                                                      android:id="@+id/txtLogin2"
   android:layout height="match parent"
                                                      android:layout width="wrap content"
    android:columnCount="2">
                                                      android:layout height="wrap content"
                                                      android:ems="10"
    <TextView
                                                      android:inputType="textPersonName"
        android:id="@+id/lblLogin2"
                                                      android:labelFor="@+id/lblLogin"
       android:layout width="wrap content"
                                                      android:text="utilisateur" />
       android:layout height="wrap content"
        android:text="Utilisateur:"
                                                  <Button
        android:textSize="18sp" />
                                                      android:id="@+id/btnOk2"
                                                      android:layout width="wrap content"
    <EditText
                                                      android:layout column="1"
        android:id="@+id/editText2"
                                                      android:layout gravity="right"
       android:layout width="wrap content"
                                                      android:freezesText="false"
       android:layout height="wrap content"
                                                      android:gravity="center"
       android:ems="10"
                                                      android:text="OK" />
        android:inputType="textPersonName"
        android:labelFor="@+id/lblPassword"
                                              </GridLayout>
        android:text="mot de passe" />
    <TextView
        android:id="@+id/lblPassword2"
        android:layout width="wrap content"
       android:layout height="wrap content"
       android:text="Mot de passe:"
        android:textSize="18sp" />
```





Le GridLayout (2)

- Découpage de l'espace en Lignes et colonnes pour obtenir des cellules :
- ▶ Propriété du GridLayout : columnCount : nombre de colonnes par ligne
- Propriétés pour les composants:
 - ► Layout_columnSpan : nombre de colonnes occupées
 - ▶ Layout_column : numéro de la première cellule à occuper