

Les Layouts

Formateur : Mohammed LAMNAOUR



Présentation Generale

- Un Layout définit la manière dont les Vues contenues sont disposées les unes par rapport aux autres.
- Les Layouts que nous allons traiter dans ce cours sont :
 - LinearLayout
 - RelativeLayout
 - TableLayout



LinearLayout

Développement Mobile

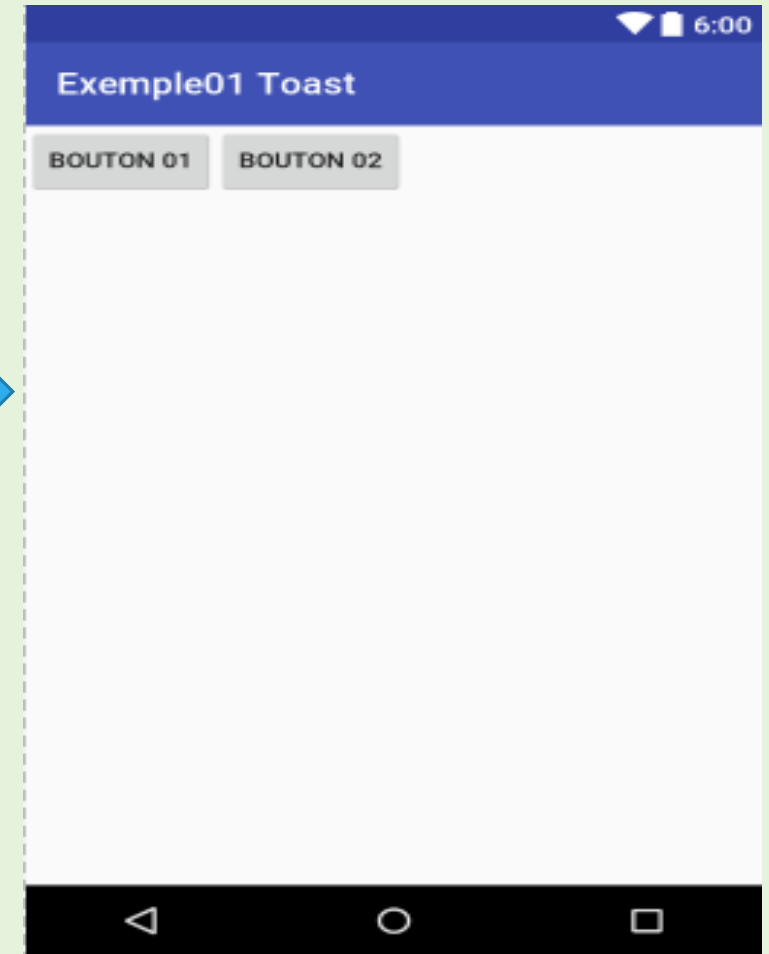
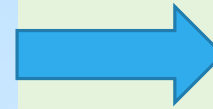
Présentation

- **LinearLayout** : disposition des éléments en 1 ligne ou 1 colonne dans l'ordre où ils sont définis dans le XML
- La propriété **android:orientation** permet de définir le sens du layout (**horizontal** ou **vertical**).



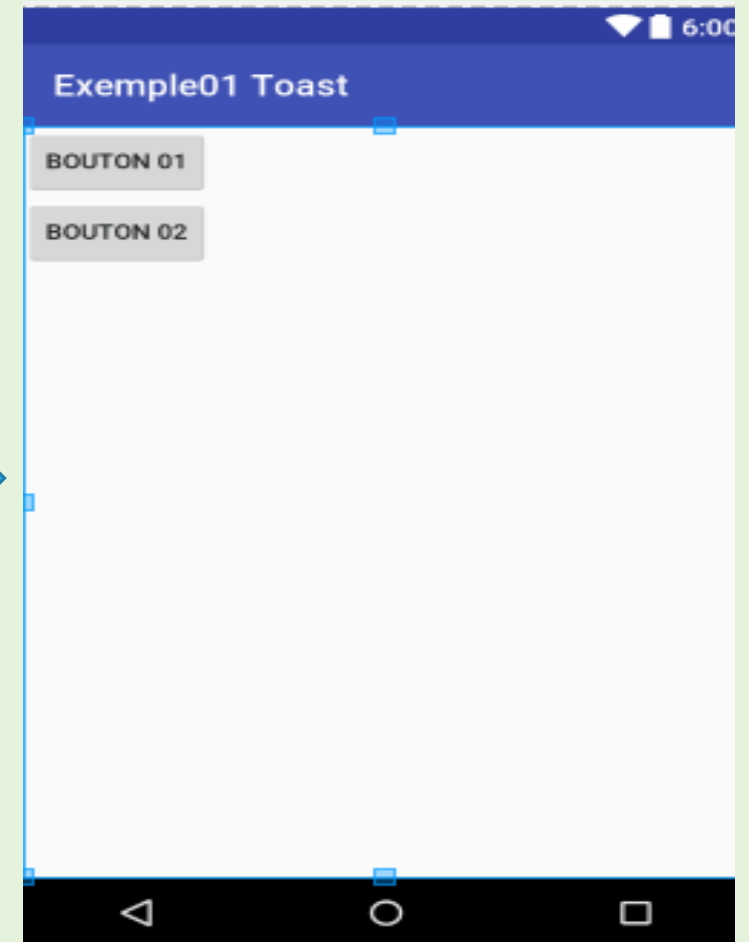
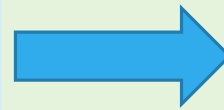
LinearLayout – Example 1

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn01"
        android:text="Bouton 01"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn02"
        android:text="Bouton 02"/>
</LinearLayout>
```



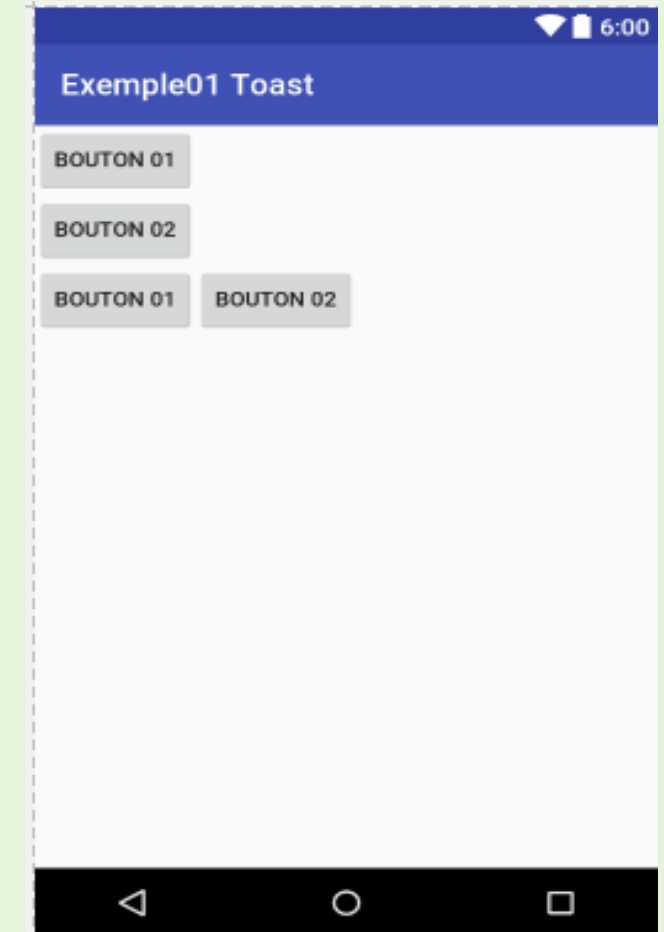
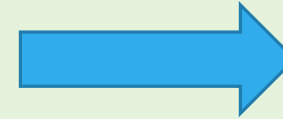
LinearLayout – Exemple 2

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn01"
        android:text="Bouton 01"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn02"
        android:text="Bouton 02"/>
</LinearLayout>
```



LinearLayout Imbriques

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn01"
        android:text="Bouton 01"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn02"
        android:text="Bouton 02"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 03"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 04"/>
    </LinearLayout>
</LinearLayout>
```



LinearLayout – Poids des éléments

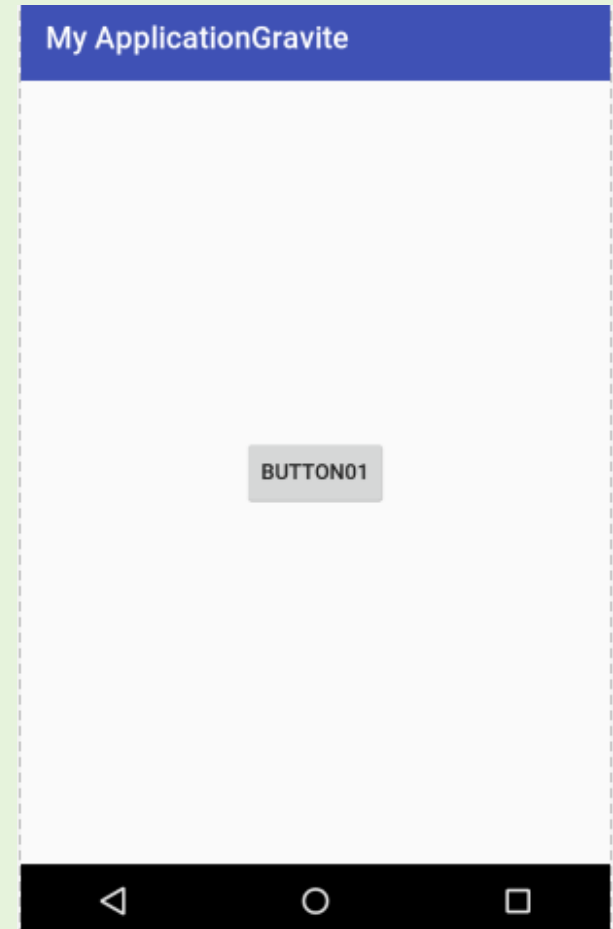
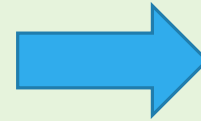
- imaginez que vous avez 2 boutons en colonne et que vous voulez que chacun occupe tout l'espace disponible :
 - Si on leur affecte la même valeur au **poids** (**android:layout_weight**), l'espace libre sera partagé équitablement entre les deux.
 - Si l'on met une valeur 1 au premier et 2 au second, le deuxième prendra deux fois plus d'espace libre que le premier (Par défaut la valeur du poids est de 0).

LinearLayout – Gravite des éléments

- Dans un **LinearLayout** les éléments sont alignés de gauche à droite et de haut en bas.
- La question qui se pose est comment faire si l'on veut placer notre élément tout en bas ou à droite. ➔ La réponse se trouve dans la **gravité** (**android:gravity** ou **android:layout_gravity**).
- Les gravités les plus couramment utilisées sont **left**, **center_horizontal**, **top**, **bottom**, **right**, pour respectivement aligner les éléments à gauche, au centre, en haut, en bas et à droite. On peut aussi utiliser **center_vertical** pour centrer verticalement l'élément.

LinearLayout – Gravite des éléments – Exemple 1

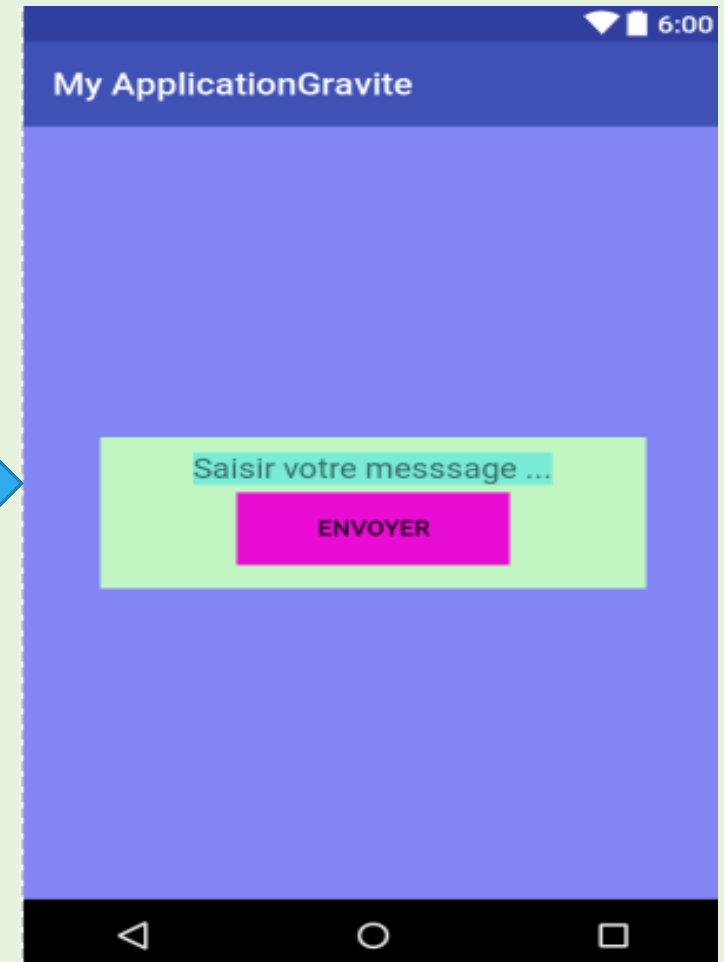
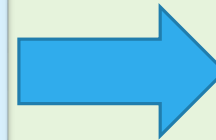
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button01"
        android:id="@+id/button" />
</LinearLayout>
```



LinearLayout – Gravite des éléments – Exemple 2

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:background="#8385f7">

    <LinearLayout
        android:layout_width="300dp"
        android:layout_height="100dp"
        android:orientation="vertical"
        android:background="#c1f5c1"
        android:gravity="center"
        >
        <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:hint="Saisir votre message ..."
            android:background="#78ecd7"/>
        <Button
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:text="Envoyer"
            android:layout_gravity="center"
            android:id="@+id/button"
            android:layout_margin="5dp"
            android:background="#e90dd3"/>
    </LinearLayout>
</LinearLayout>
```



Exercices d'application

- Créer les interfaces suivantes en se basant sur le LinearLayout :

Calculatrice V2

Nombre 1 :

Nombre 2 :

Resultat :

Linear Layout Cas 1

Nom :

Prénom :

☒ Homme ☐ Femme

Date Naissance :

Liste des diplôme :

☐ Bac + 2 ☐ License

☐ Master ☐ Doctorat

Linear Layout Cas 1

To

From

Message

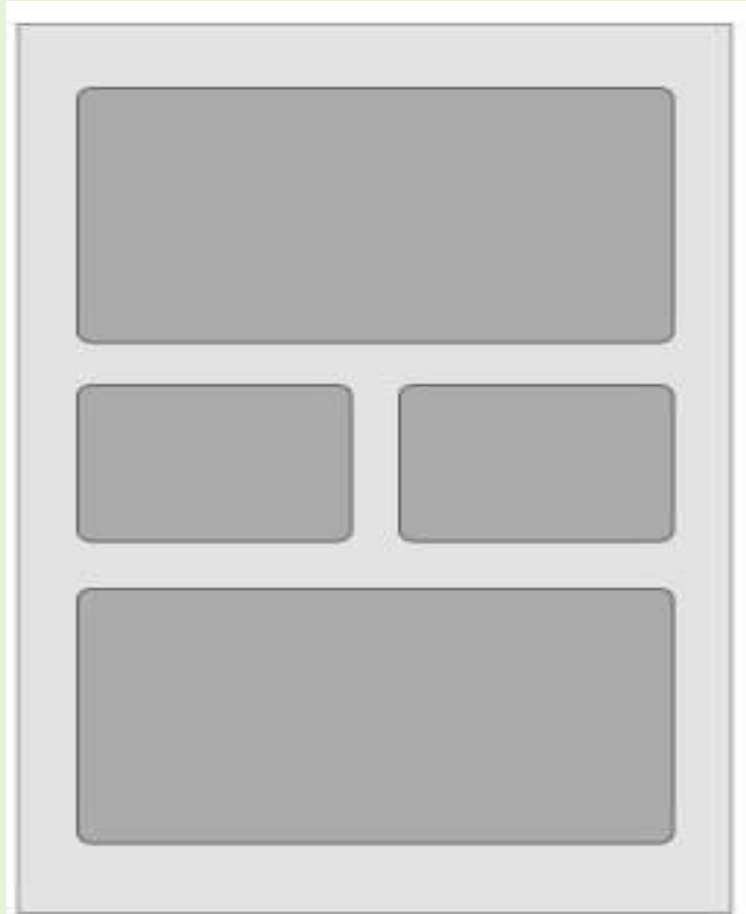


RelativeLayout

Développement Mobile

Présentation

- **RelativeLayout** : chaque Vue définit son déplacement par rapport à une autre Vue.



Positionnement relatif au conteneur

- **android:layout_alignParentTop** (true / false) : Cette option permet de préciser si le haut de l'élément doit être aligné avec celui de son conteneur. Même principe pour : **android:layout_alignParentBottom**, **android:layout_alignParentLeft** et **android:layout_alignParentRight**.
- **android:layout_centerHorizontal** : Indique si l'élément doit être **centré horizontalement** dans son conteneur. Même principe pour : **android:layout_centerVertical**.
- **android:layout_centerInParent** : Vous permet d'indiquer que l'élément doit être **centré horizontalement et verticalement** dans le conteneur.

Position relative aux autres éléments

- **android:layout_above** : Indique que l'élément sera placé au-dessus de celui indiqué par son id.
- **android:layout_below** : Indique que l'élément sera placé en dessous de celui indiqué par son id.
- **android:layout_toLeftOf** : Indique que l'élément sera placé à gauche de celui indiqué par son id.
- **android:layout_toRightOf** : Indique que l'élément sera placé à droite de celui indiqué par son id.
- **android:layout_alignTop** : Indique que le haut de notre élément est aligné avec le haut de l'élément indiqué.
- **android:layout_alignBottom** : Indique que le bas de notre élément est aligné avec le bas de l'élément indiqué.
- **android:layout_alignLeft** : Indique que le côté gauche de notre élément est aligné avec le côté gauche de l'élément indiqué.
- **android:layout_alignRight** : Indique que le côté droit de notre élément est aligné avec le côté droit de l'élément indiqué.

Exercices d'application

- Exercice 1:

Créer les interfaces suivantes en se basant sur RelativeLayout

Forme 1

BOUTTON1 BOUTTON2

BOUTTON6 BOUTTON5 BOUTTON7

BOUTTON3 BOUTTON4

Forme 2

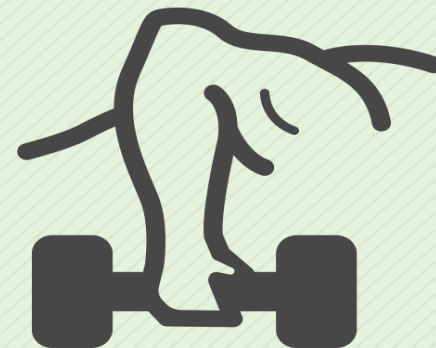
LOGIN

Username :

Password :

Comments

DELETE SAVE





TableLayout

Développement Mobile

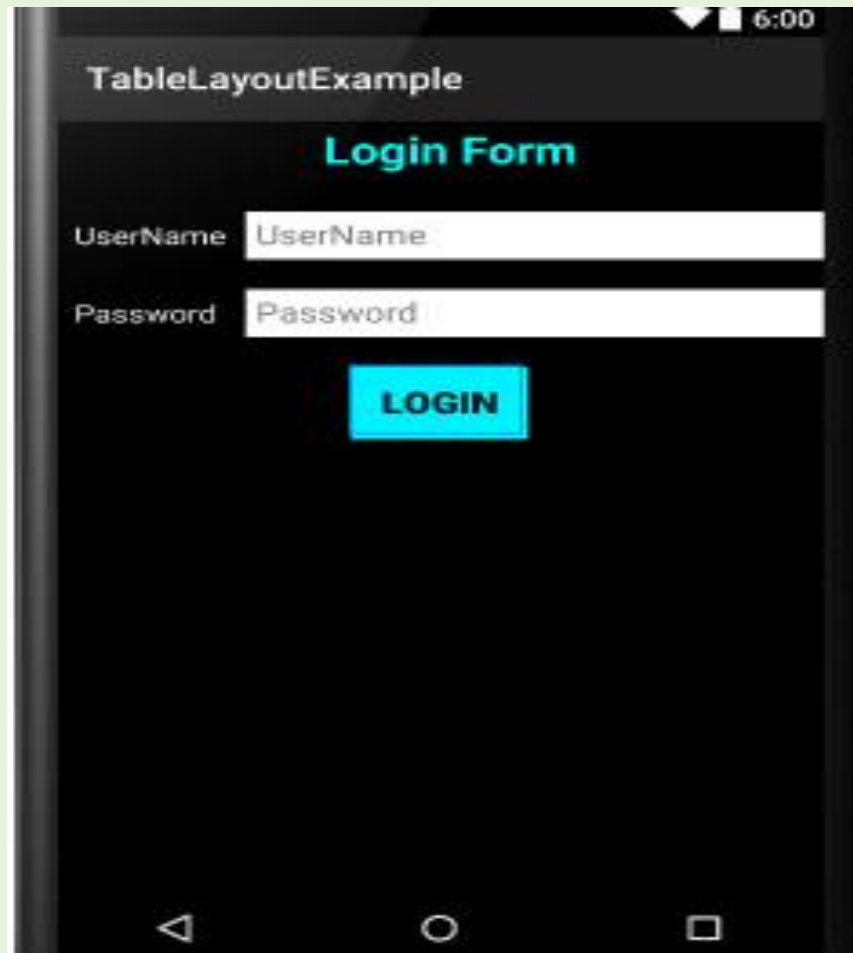
Présentation

- **TableLayout** : Les vues sont positionnées sous forme d'un tableau

Row 1 Column 1	Row 1 Column 2	Row 1 Column 3
Row 2 Column 1		Row 2 Column 2
Row 3 Column 1		

Exercice d'application

- Créer l'interface suivant en se basant sur TableLayout :



The screenshot displays an Android application interface. At the top, the title bar reads 'TableLayoutExample'. Below it, the text 'Login Form' is centered in a green font. The form consists of two rows: the first row has the label 'UserName' followed by a text input field containing 'UserName'; the second row has the label 'Password' followed by a text input field containing 'Password'. Below these fields is a red rectangular button with the text 'LOGIN' in white. The status bar at the top right shows the time '6:00'. The bottom of the screen features the standard Android navigation bar with back, home, and recent apps icons.