

TP

Gestionnaires de positionnement Android (Layout)

LinearLayout, RelativeLayout & ConstraintLayout

LinearLayout

Exercice 1 — Colonne simple de 3 boutons (exo1_ly.xml)

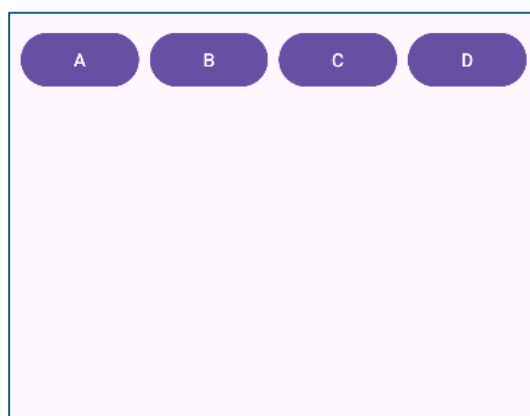
Crée une activité avec un LinearLayout vertical contenant **3 boutons** : Ajouter, Modifier, Supprimer.

Les boutons doivent occuper **toute la largeur** et s'afficher les uns sous les autres.



Exercice 2 — Ligne de 4 boutons avec largeur égale(exo2_ly.xml)

Crée une activité avec un LinearLayout **horizontal** contenant **4 boutons** (A, B, C, D) ayant **la même largeur** et un **petit espace** entre eux.



Annexe LinearLayout

Balises utilisées

- <LinearLayout>
Conteneur qui organise ses enfants en **ligne verticale ou horizontale**.
Attribut clé : android:orientation="vertical|horizontal" → disposition des vues.
- <Button>

Attributs communs

- xmlns:android="http://schemas.android.com/apk/res/android"
Déclare l'espace de noms Android (obligatoire dans chaque fichier XML).
- android:orientation="vertical|horizontal"
Définit la direction des enfants dans le LinearLayout.
- android:padding="16dp"
Ajoute un espace **interne** autour du contenu du layout.
- android:layout_width="match_parent"
La vue occupe **toute la largeur** du parent.
- android:layout_height="match_parent" (pour le parent)
Le conteneur occupe **toute la hauteur** disponible.
- android:layout_height="wrap_content" (pour les boutons)
La hauteur s'adapte au contenu (texte du bouton).
- android:id="@+id/btnAjouter"
Identifiant unique pour référencer la vue dans le code Java.
- android:text="Ajouter"
Texte affiché sur le bouton.
- android:layout_width="0dp"
Largeur fixée à 0 pour activer le **poids** (dans un LinearLayout horizontal).
- android:layout_weight="1"
Définit la proportion d'espace attribuée à la vue (ici, parts égales).
- android:layout_marginEnd="8dp"
Ajoute un espace à droite du bouton (utile pour séparer les boutons).
- android:gravity="center" (sur le parent)
Centre les enfants dans le conteneur.

RelativeLayout — 2 exercices

Exercice 3 — Trois boutons alignés par rapport au parent et entre eux (exo3_rl.xml)

Crée une activité avec un RelativeLayout et 3 boutons :

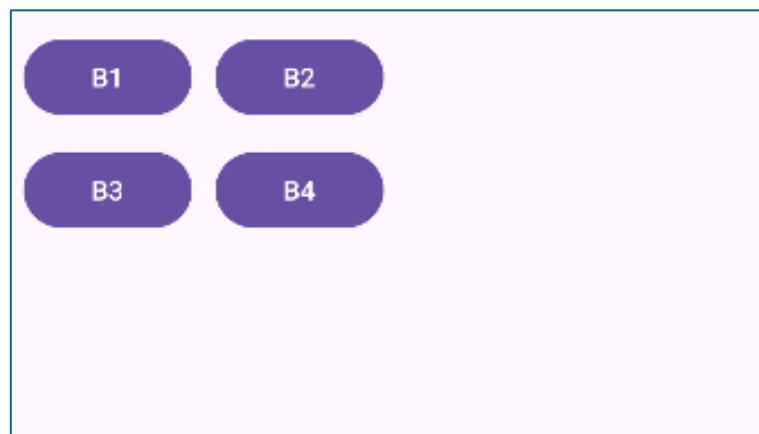
- Bouton 1 : centré **horizontalement** en haut du parent.
- Bouton 2 : placé **en dessous** du Bouton 1, **aligné à gauche**.
- Bouton 3 : placé **en dessous** du Bouton 1, **aligné à droite**.



Exercice 4 — Quatre boutons positionnés les uns par rapport aux autres(exo4_rl.xml)

Crée une activité avec un RelativeLayout et **4 boutons** formant un carré logique :

- B1 en **haut à gauche** du parent.
- B2 à **droite** de B1, aligné en haut.
- B3 **en dessous** de B1, aligné à gauche.
- B4 **en dessous** de B2, aligné à droite.



Annexe RelativeLayout

Balises utilisées

<RelativeLayout> Conteneur qui positionne ses enfants relativement au parent ou aux autres vues.

Attributs communs

Attributs communs

- xmlns:android="http://schemas.android.com/apk/res/android"
Déclare l'espace de noms Android (obligatoire dans chaque fichier XML).
- android:layout_width="wrap_content|match_parent"
Largeur de la vue (adaptée au contenu ou occupant tout l'espace).
- android:layout_height="wrap_content|match_parent"
Hauteur de la vue.
- android:id="@+id/btnX"
Identifiant unique pour référencer la vue dans le code ou dans les relations.
- android:text="..."
Texte affiché sur le bouton.
- android:layout_marginTop="12dp"
Ajoute un espace au-dessus du bouton.

Spécifiques à RelativeLayout

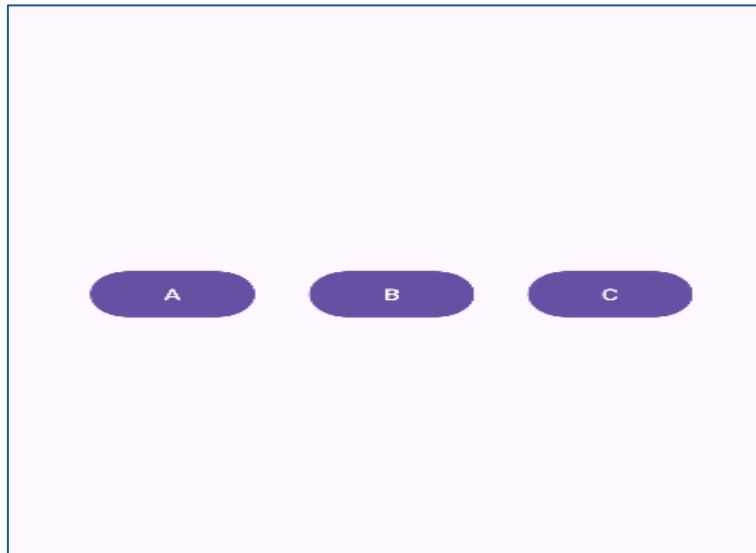
- android:layout_alignParentTop="true"
Aligner la vue en haut du parent.
- android:layout_alignParentBottom="true"
Aligner la vue en bas du parent.
- android:layout_alignParentStart="true"
Aligner la vue au début du parent.
- android:layout_alignParentEnd="true"
Aligner la vue à la fin du parent.
- android:layout_centerHorizontal="true"
Centrer la vue horizontalement dans le parent.
- android:layout_centerVertical="true"
Centrer la vue verticalement dans le parent.
- android:layout_below="@id/..."
Placer la vue **en dessous** d'une autre vue.
- android:layout_above="@id/..."
Placer la vue **au-dessus** d'une autre vue.

- `android:layout_toStartOf="@id/..."`
Place la vue à **gauche** d'une autre vue.
- `android:layout_toEndOf="@id/..."`
Place la vue à **droite** d'une autre vue.
- `android:layout_alignStart="@id/..."`
Aligne le bord gauche avec celui d'une autre vue.
- `android:layout_alignEnd="@id/..."`
Aligne le bord droit avec celui d'une autre vue.
- `android:layout_alignTop="@id/..."`
Aligne le haut avec celui d'une autre vue.
- `android:layout_alignBottom="@id/..."`
Aligne le bas avec celui d'une autre vue.

ConstraintLayout — 2 exercices

Exercice 5 — Chaîne horizontale de 3 boutons centrée(exo5_cl.xml)

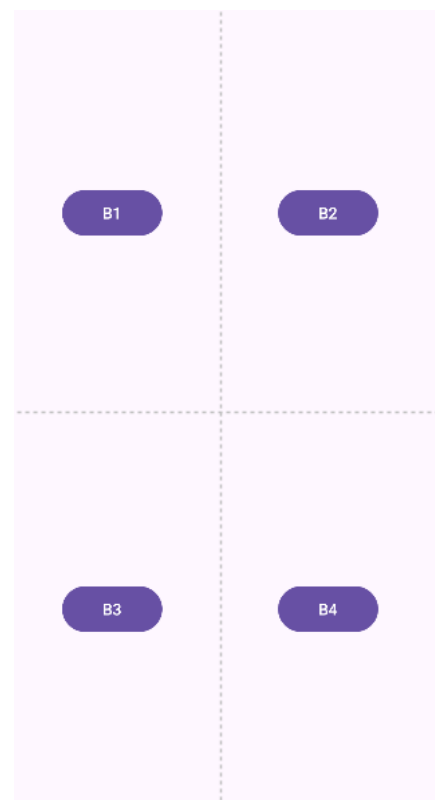
Crée une activité avec un ConstraintLayout contenant **3 boutons** (A, B, C) disposés **horizontalement au centre de l'écran**, avec un espacement régulier entre eux. Les boutons doivent être **alignés verticalement au centre**.



Exercice 6 — 4 boutons en grille 2×2 avec Guideline(exo6_cl.xml)

Crée une activité avec un ConstraintLayout, ajoute **deux guidelines** (verticale et horizontale à 50%), puis pose **4 boutons** formant une **grille 2×2** :

- B1 en haut-gauche du premier quadrant,
- B2 en haut-droite,
- B3 en bas-gauche,
- B4 en bas-droite.



Annexe ConstraintLayout

Balises utilisées

- `<androidx.constraintlayout.widget.ConstraintLayout>` : conteneur flexible basé sur des contraintes.
- `<Button>` : élément interactif pour déclencher une action.
- `<androidx.constraintlayout.widget.Guideline>` : repère invisible pour positionner les vues.

Attributs communs

- `xmlns:android="http://schemas.android.com/apk/res/android` : espace de noms Android (obligatoire).
- `xmlns:app="http://schemas.android.com/apk/res-auto` : espace de noms pour les attributs spécifiques à ConstraintLayout.
- `android:layout_width="match_parent|wrap_content` : largeur de la vue.
- `android:layout_height="match_parent|wrap_content` : hauteur de la vue.
- `android:id="@+id/btnX` : identifiant unique pour la vue.
- `android:text="..."` : texte affiché sur le bouton.

Spécifiques à ConstraintLayout

- `app:layout_constraintStart_toStartOf="parent|@id/..."` : contrainte horizontale début.
- `app:layout_constraintEnd_toEndOf="parent|@id/..."` : contrainte horizontale fin.
- `app:layout_constraintStart_toEndOf="@id/..."` : contrainte pour placer une vue à droite d'une autre.
- `app:layout_constraintEnd_toStartOf="@id/..."` : contrainte pour placer une vue à gauche d'une autre.
- `app:layout_constraintTop_toTopOf="parent|@id/..."` : contrainte verticale haut.
- `app:layout_constraintBottom_toBottomOf="parent|@id/..."` : contrainte verticale bas.
- `app:layout_constraintTop_toBottomOf="@id/..."` : contrainte pour placer une vue en dessous d'une autre.
- `app:layout_constraintBottom_toTopOf="@id/..."` : contrainte pour placer une vue au-dessus d'une autre.
- `app:layout_constraintGuide_percent="0..1` : position d'une Guideline en pourcentage.
- `android:orientation="vertical|horizontal` (pour Guideline) : orientation du repère.

- `app:layout_constraintHorizontal_bias="0..1` : ajustement horizontal (0 = gauche, 1 = droite).
- `app:layout_constraintVertical_bias="0..1` : ajustement vertical (0 = haut, 1 = bas).