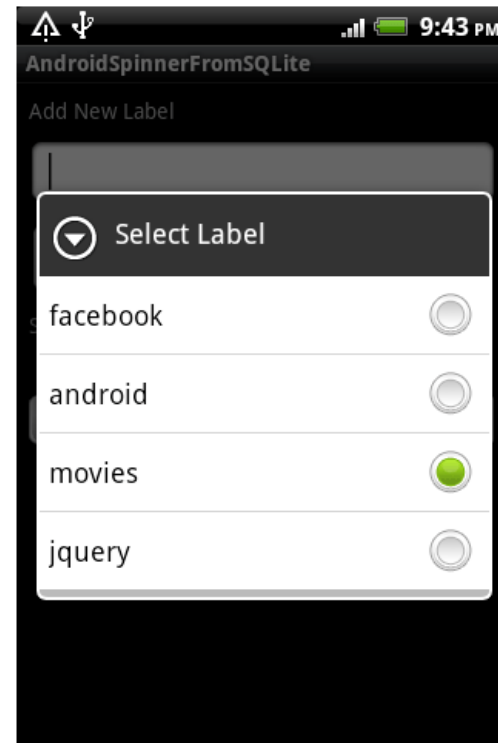


AFFICHER DES LISTES

- ❑ Spinner
- ❑ ListView
- ❑ Adapter

Android Spinner dropdown from SQLite

AndroidHive

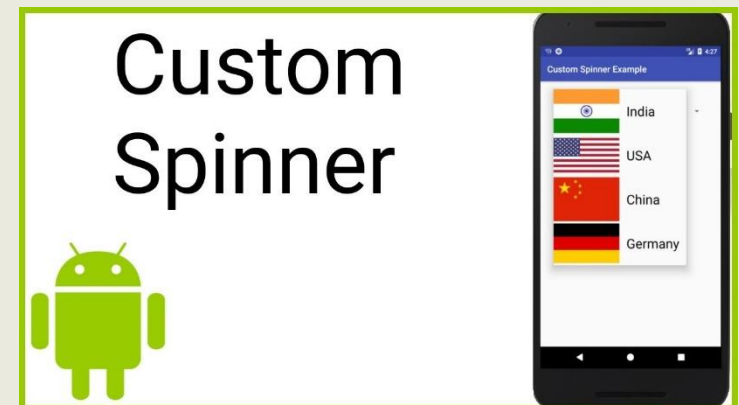
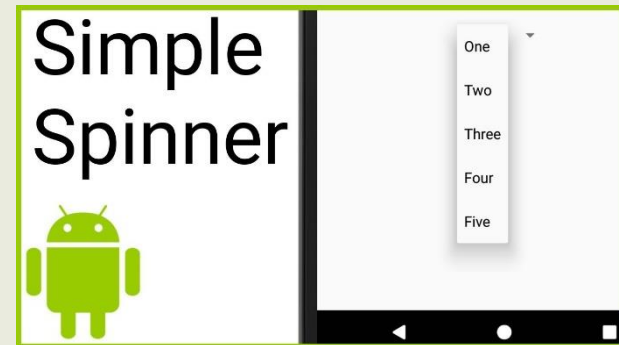
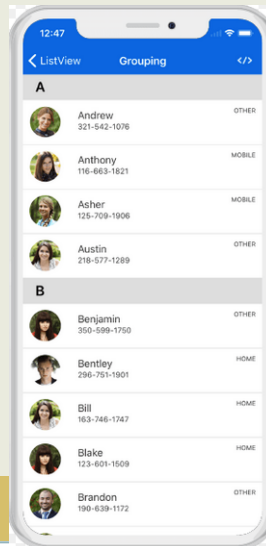
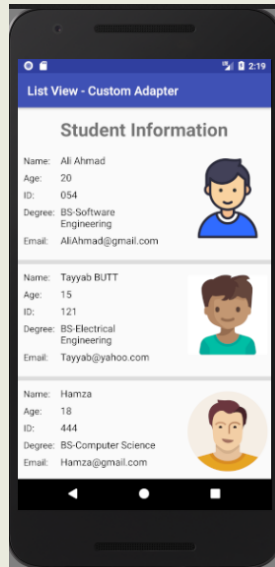
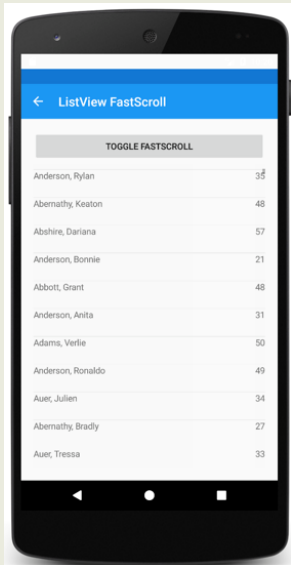


CONTEXTE D'UTILISATION

Dans un Spinner

Dans un ListView

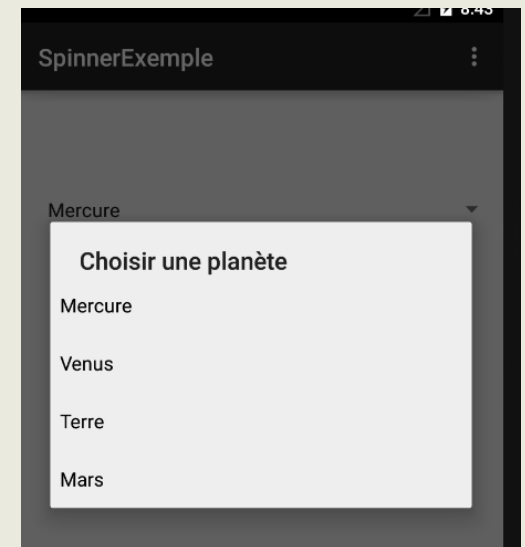
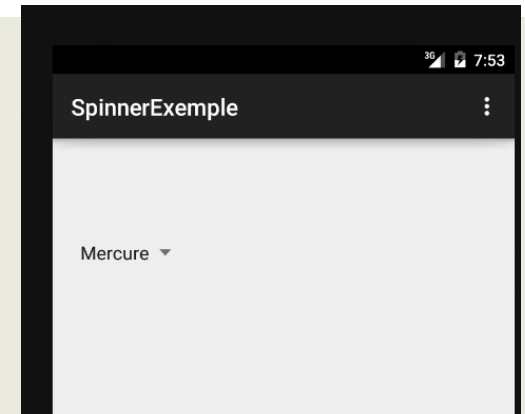
(Dans un RecyclerView → non étudié), ...



SPINNER – ÉLÉMENTS STATIQUES

- ◆ Définir le titre de la fenêtre qui s'ouvre lorsque l'on fait un choix
 - ✦ `android:prompt="texte"`
 - ✦ `android:spinnerMode="dialog"`
- ◆ Définir le contenu de la liste à partir d'une ressource (statique) `android:entries="@array/maliste"`
 - tableau de chaînes de caractères dans un fichier xml placé dans `res/values/` sous la forme :

```
<resources>
    <string-array name="maliste">
        <item>Mercure</item>
        <item>Venus</item>
        <item>Terre</item>
        <item>Mars</item>
    </string-array>
</resources>
```



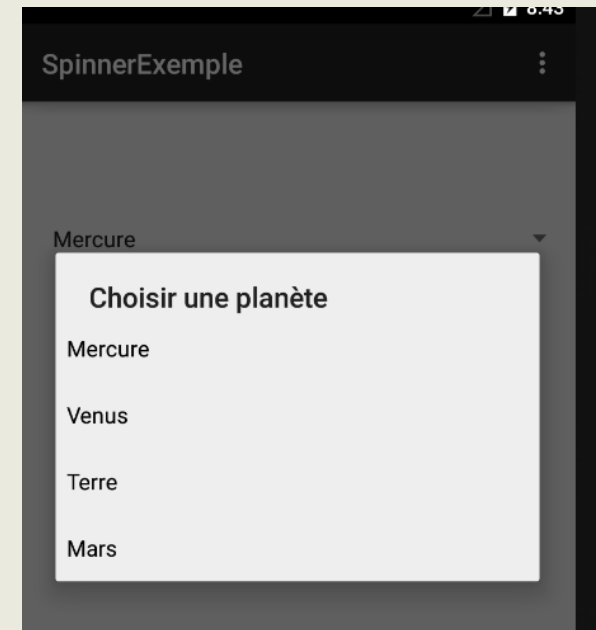
SPINNER

<Spinner

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:id="@+id/spinner"  
    android:entries="@array/maliste"  
    android:prompt="@string/planete"  
    android:layout_marginTop="71dp"  
    android:spinnerMode="dialog" />
```

Dans string.xml

```
<string name="planete">Choisir une planète</string>
```



SPINNER

❑ Evènement

- ◆ **setOnItemSelectedListener(AdapterView.OnItemSelectedListener)**
associe un écouteur d'événements à la sélection d'un choix.
- ✦ La méthode **onItemSelected (AdapterView<?>, View, int, long)** de l'interface AdapterView.OnItemSelectedListener est surchargée pour traiter l'événement.

Le 3^{ème} paramètre indique **le rang de l'élément** et le dernier son identifiant.

SPINNER

```
final Spinner spinner = findViewById(R.id.spinner);
spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        Toast.makeText(MainActivity.this, String.valueOf(sp.getSelectedItem()), Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
        Toast.makeText(MainActivity.this, "Vous n'avez rien sélectionné", Toast.LENGTH_SHORT).show();
    }
});
```

SPINNER – ÉLÉMENTS DYNAMIQUES

❑ Ajout des items par programmation

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    final Spinner spinner = findViewById(R.id.spinner);  
    List<String> items = new ArrayList<String>();  
    items.add("France");  
    items.add("Italie");  
    items.add("Portugal");  
    items.add("Espagne");  
    ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(  
        getApplicationContext(),  
        android.R.layout.simple_spinner_item,  
        items);  
    spinner.setAdapter(dataAdapter);  
}
```

```
ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(this,  
    android.R.layout.simple_list_item_single_choice, list);
```

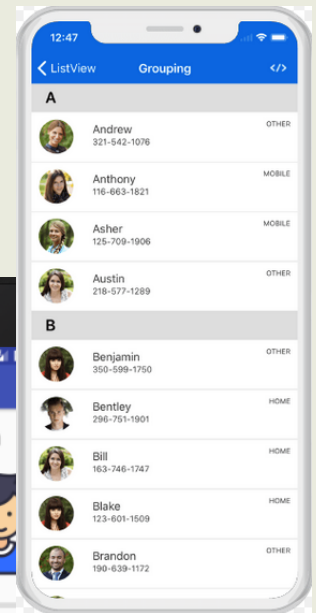
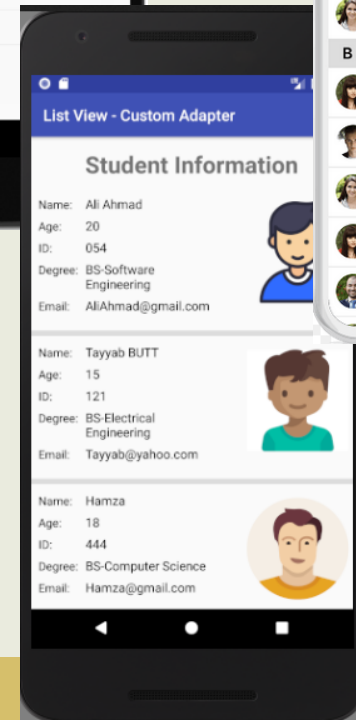
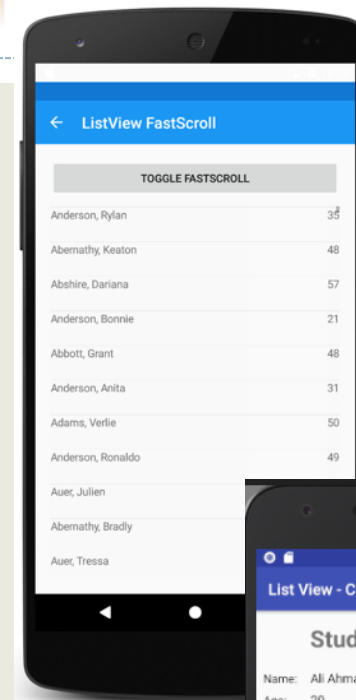
SPINNER

❑ **Vider une liste**

```
public void supTout(View v) {  
    final Spinner spinner = findViewById(R.id.spinner);  
    ArrayAdapter<String> dataAdapter = sp2.getAdapter();  
    dataAdapter.clear();  
}
```


LISTVIEW

- ❑ Une ListView permet de placer des éléments en liste avec un ascenseur vertical si nécessaire.
- ❑ ListView est normalement utilisé pour afficher des éléments textuels éventuellement accompagnés d'une case à cocher lorsqu'il s'agit d'une liste à choix multiples.
- ❑ Une ListView permet d'afficher des éléments plus complexes en utilisant un gestionnaire de contenu.



LISTVIEW – ÉLÉMENTS STATIQUES

❑ Contenu de type texte

android:entries="@array/maliste" définit le contenu de la liste à partir d'une ressource de type tableau de chaînes de caractères définie dans un fichier xml placé dans **res/values/**

```
<string-array name="maliste">  
    <item>premier élément</item>  
    <item>deuxième élément</item>  
    ...  
    <item>dernier élément</item>  
</string-array>
```

LISTVIEW

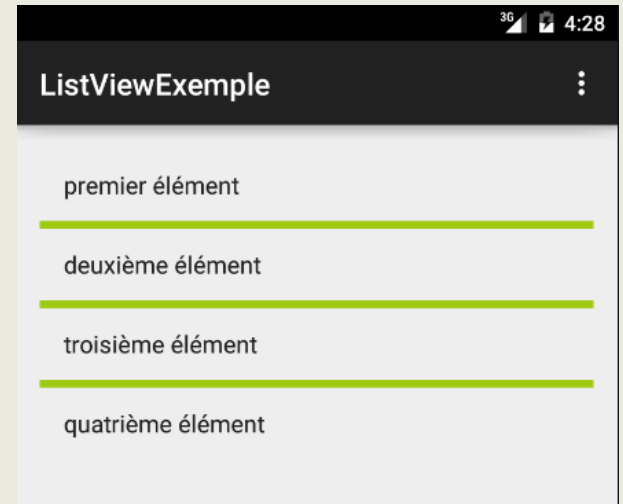
❑ Séparateurs

- ◆ **android:divider**="couleur" définit la couleur des séparateurs
- ◆ **android:divider**="@drawable/monimage" pour utiliser une image.
L'image est placée dans res/drawable/ et s'appelle monimage.x (où x est png, jpg, ou gif).
- ◆ **android:dividerHeight**="unité" définit la hauteur des séparateurs (si c'est une image elle sera déformée).
- ◆ **android:choiceMode**="c"
c peut prendre les valeurs : none, singlechoice, multipleChoice

LISTVIEW

<ListView

```
    android:layout_width="wrap_content"  
    android:layout_height="200dp"  
    android:id="@+id/listView"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true"  
    android:entries="@array/maliste"  
    android:dividerHeight="5dp"  
    android:divider="#ffa0ca12" />
```



LISTVIEW – ÉLÉMENTS DYNAMIQUES

❑ Ajout des items par programmation

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    final Spinner spinner = findViewById(R.id.spinner);  
    List<String> items = new ArrayList<String>();  
    items.add("France");  
    items.add("Italie");  
    items.add("Portugal");  
    items.add("Espagne");  
    ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(  
        getApplicationContext(),  
        android.R.layout.simple_list_item_1,  
        items);  
    spinner.setAdapter(dataAdapter);  
}
```

LISTVIEW

❑ Evènements

setOnClickListener(AdapterView.OnClickListener)

associe un écouteur d'événements au clic sur un élément.

La méthode **onItemClick(AdapterView<?>, View, int, long)**

de l'interface AdapterView.OnItemClickListener est surchargée pour traiter l'événement.

Le 3ème paramètre indique le rang de l'élément et le dernier son identifiant.

LISTVIEW

setOnItemSelectedListener(AdapterView.OnItemSelectedListener)

associe un écouteur d'événements à la sélection d'un élément

La méthode **onItemSelected (AdapterView<?>, View, int, long)** est surchargée pour traiter l'événement de sélection,

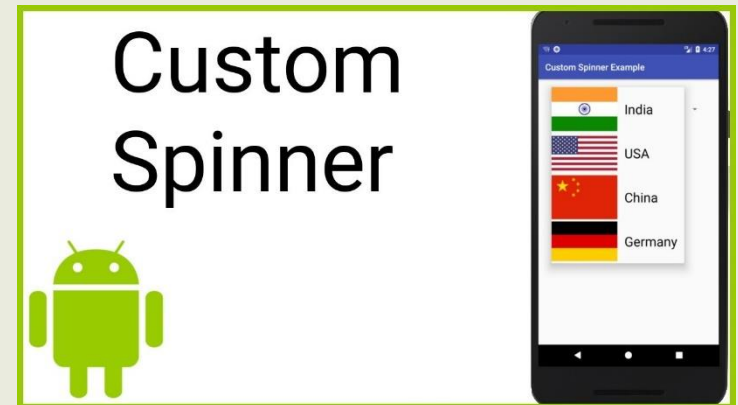
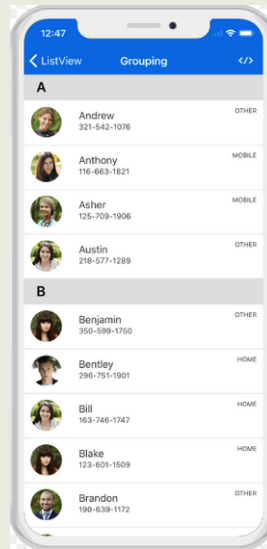
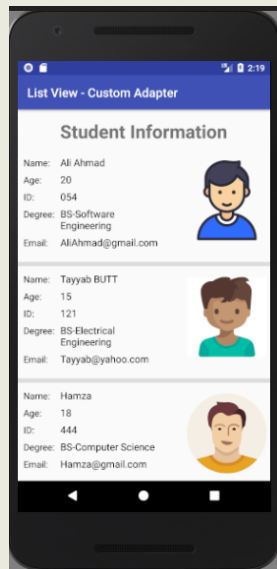
La méthode **onNothingSelected(AdapterView<?>)** est surchargée pour traiter l'événement de non sélection

PERSONNALISER UN ADAPTER

Dans un Spinner

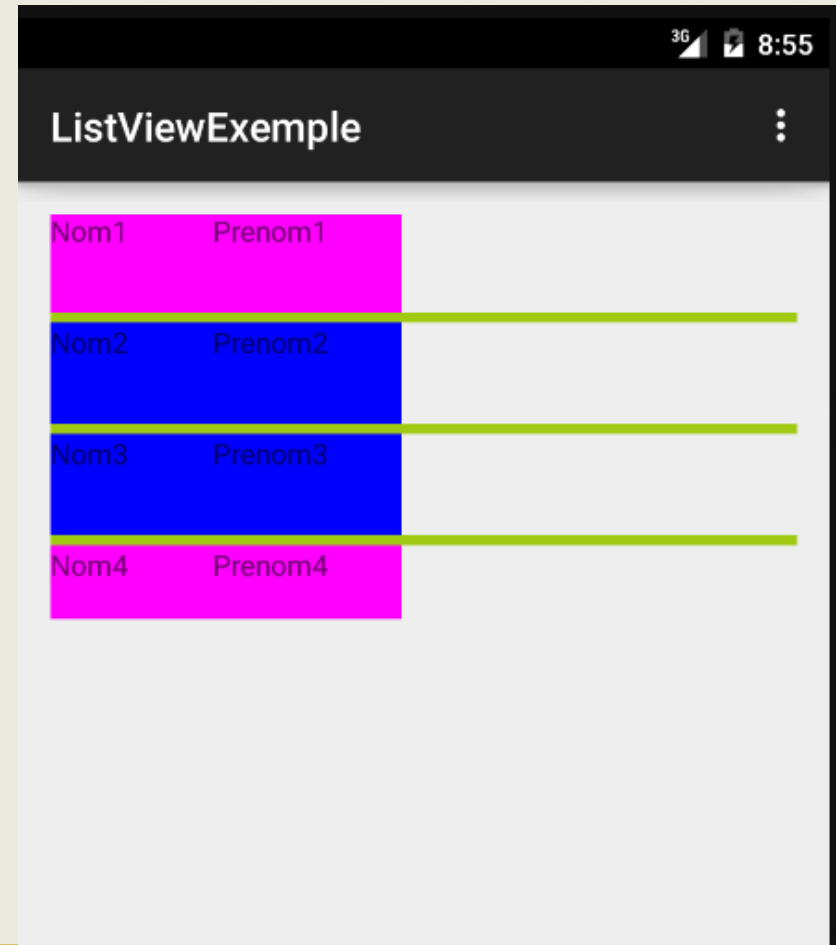
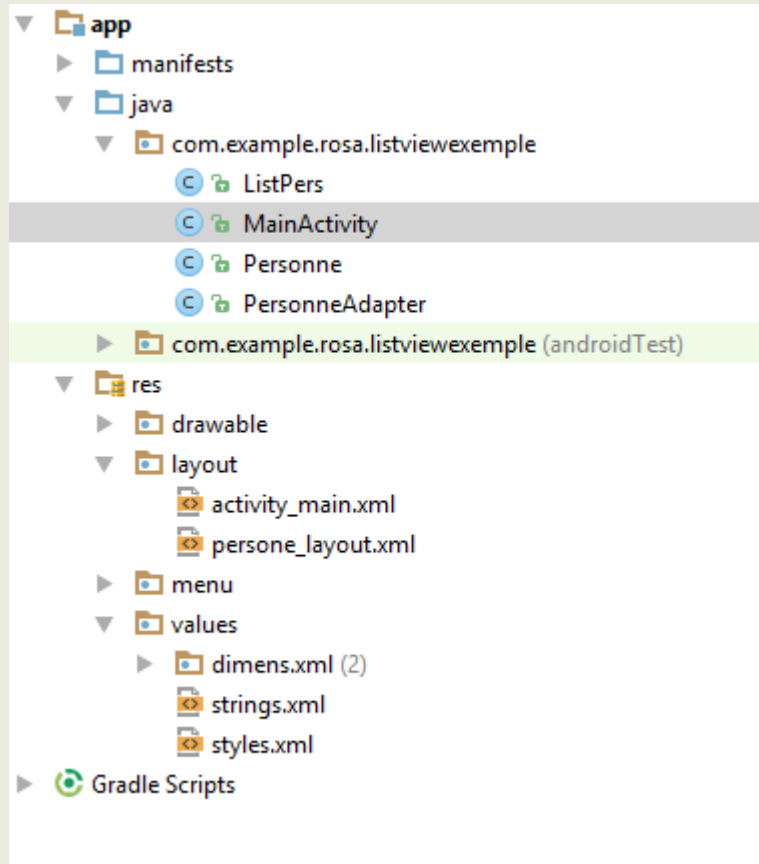
Dans un ListView

(Dans un RecyclerView → non étudié), ...



LISTVIEW

❑ Affichage d'éléments complexes



LISTVIEW

```
public class Personne {  
    private String nom;  
    private String prenom;  
    private int genre;
```

//ACCESSEURS

```
    public Personne(String nom, String prenom, int genre) {  
        this.nom = nom;  
        this.prenom = prenom;  
        this.genre = genre;  
    }  
}
```

LISTVIEW

```
public class ListePersonnes extends ArrayList<Personne> {  
  
    public void construireListe(){  
        add(new Personne("Nom1", "Prenom1", 2));  
        .....  
    }  
}
```

LISTVIEW

❑ Layout principal (fichier activity_main)

```
<ListView  
    android:layout_width="wrap_content"  
    android:layout_height="200dp"  
    android:id="@+id/listView"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true"  
    android:dividerHeight="5dp"  
    android:choiceMode="singleChoice"  
    android:divider="#ffa0ca12" />
```

LISTVIEW

❑ Layout pour visualiser une personne (fichier personne_layout)

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:id="@+id/LL_Fond">
```

```
    <TextView android:text="Nom"
        android:id="@+id/TV_Nom"
        android:layout_width="80dp"
        android:layout_height="50dp">
```

```
</TextView>
```

```
    <TextView android:text="Prénom"
        android:id="@+id/TV_Prenom"
        android:layout_width="93dp"
        android:layout_height="50dp">
```

```
</TextView>
```

```
</LinearLayout>
```

REMARQUE: peut contenir une image

ADAPTER

❑ Classe PersonneAdapter

ClickableActivity est
une interface !

```
public class PersonneAdapter extends BaseAdapter {  
    private Persons items;  
    private LayoutInflater mInflater; //Un mécanisme pour gérer l'affichage graphique depuis un layout XML  
    private ClickableActivity activity;  
  
    public PersonneAdapter(ClickableActivity activity, Persons items) {  
        this.activity = activity;  
        this.items = items;  
        mInflater = LayoutInflater.from(activity.getContext());  
    }  
  
    public int getCount() {  
        return items.size();  
    }  
  
    public Object getItem(int position) {  
        return items.get(position);  
    }  
  
    public long getItemId(int position) {  
        return position;  
    }  
}
```

LISTVIEW & ADAPTER

```
public View getView(int position, View convertView, ViewGroup parent) {
    View layoutItem;

    //(1) : Réutilisation des layouts
    layoutItem = convertView == null ? mInflater.inflate(R.layout.personne_layout, parent, false) : convertView;

    //(2) : Récupération des TextView de notre layout
    TextView displayFirstname = layoutItem.findViewById(R.id.firstname);
    TextView displayLastname = layoutItem.findViewById(R.id.lastname);

    //(3) : Renseignement des valeurs
    displayFirstname.setText(items.get(position).getFirstName());
    displayLastname.setText(items.get(position).getLastName());

    //(4) Changement de la couleur du fond de notre item
    layoutItem.setBackgroundColor( (items.get(position).getSex() == 1)? Color.BLUE : Color.MAGENTA );

    displayFirstname.setTag(position);
    layoutItem.setOnClickListener(v -> activity.onClickNom(items.get(position)) );

    return layoutItem; //On retourne l'item créé.
}
```

LISTVIEW

```
public interface ClickableActivity {  
    void onClickNom(Persons.Person item);  
    Context getContext();  
}
```


LISTVIEW

```
public class MainActivity extends AppCompatActivity implements ClickableActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Recuperation de la liste des personnes
        Persons persons = new Persons();

        //Creation et initialisation de l'Adapter pour les personnes
        PersonneAdapter adapter = new PersonneAdapter(this, persons);

        //Recuperation du composant ListView
        ListView display = findViewById(R.id.listView1);

        //Initialisation de la liste avec les donnees
        display.setAdapter(adapter);
    }

    public void onClickNom(Persons.Person item) {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Personne");
        builder.setMessage("Vous avez clique sur : " + item.getFirstName());
        builder.setNeutralButton("Ok", null);
        builder.show();
    }

    @Override
    public Context getContext() {
        return getApplicationContext();
    }
}
```

