

---

# Petit Guide !

---

## I. Création d'une liste basique :

### String-array:

```
<string-array name="elements_liste">
    <item>Apple Pie</item>
    <item>Bananas Split</item>
    <item>Cupcake</item>
    <item>Donut</item>
    <item>Eclair</item>
    <item>Froyo</item>
    <item>Gingerbread</item>
    <item>Honeycomb</item>
    <item>JellyBean</item>
    <item>KitKat</item>
</string-array>
```

## II. Création d'une liste Personnalisée :

### Code remplissage liste :

```
elements = new ArrayList<Element>();
elements.add(new Element("Apple Pie", R.drawable.logo_iac, false));
elements.add(new Element("Bananas Split", R.drawable.logo_iac, false));
elements.add(new Element("Cupcake", R.drawable.logo_iac, false));
elements.add(new Element("Donut", R.drawable.logo_iac, false));
elements.add(new Element("Eclair", R.drawable.logo_iac, false));
elements.add(new Element("Froyo", R.drawable.logo_iac, false));
elements.add(new Element("Gingerbread", R.drawable.logo_iac, false));
elements.add(new Element("Honeycomb", R.drawable.logo_iac, false));
elements.add(new Element("Jellybean", R.drawable.logo_iac, false));
elements.add(new Element("KitKat", R.drawable.logo_iac, false));
```

### Code de la méthode getView(...) dans l'Adapter :

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    LayoutInflater inflater = (LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    if(convertView == null){
        //Convertir layout_element.xml en objet java
        convertView = inflater.inflate(R.layout.layout_element, parent, false);
    }

    //Récupérer les vues dont on va modifier le contenu depuis l'objet java
    TextView text = (TextView)convertView.findViewById(R.id.text_element);
    ImageView image = (ImageView)convertView.findViewById(R.id.image_element);
    CheckBox cb = (CheckBox)convertView.findViewById(R.id.cb_element);
```

```

//Modifier les valeurs des 3 vues
text.setText(elements.get(position).getTitle());
image.setImageResource(elements.get(position).getImage());
cb.setChecked(elements.get(position).isChecked());

//On voudrait pouvoir cliquer sur toutes les regions de l'élément donc on //ne veut aucun focus sur aucune vue particulière

text.setFocusable(false);
image.setFocusable(false);
cb.setFocusable(false);

//On greffe au checkbox de chaque élément, l'objet Element qui lui correspond pour pouvoir le récupérer dans la méthode onClick
cb.setTag(elements.get(position));

//On assigne un listener aux checkbox pour pouvoir mettre à jour l'état de chaque élément de la liste de l'adapter

cb.setOnClickListener(this);
//On retourne à la liste la vue de l'élément qui est en train d'être généré.
return convertView;
}

```

### III. Animation entre deux activités :

Translate-in.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:fromXDelta="-100%p"
        android:toXDelta="0"
        android:duration="2000"
    />
</set>

```

Translate-out.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">

<translate
    android:fromXDelta="0"
    android:toXDelta="100%p"
    android:duration="2000"
/>
</set>

```