



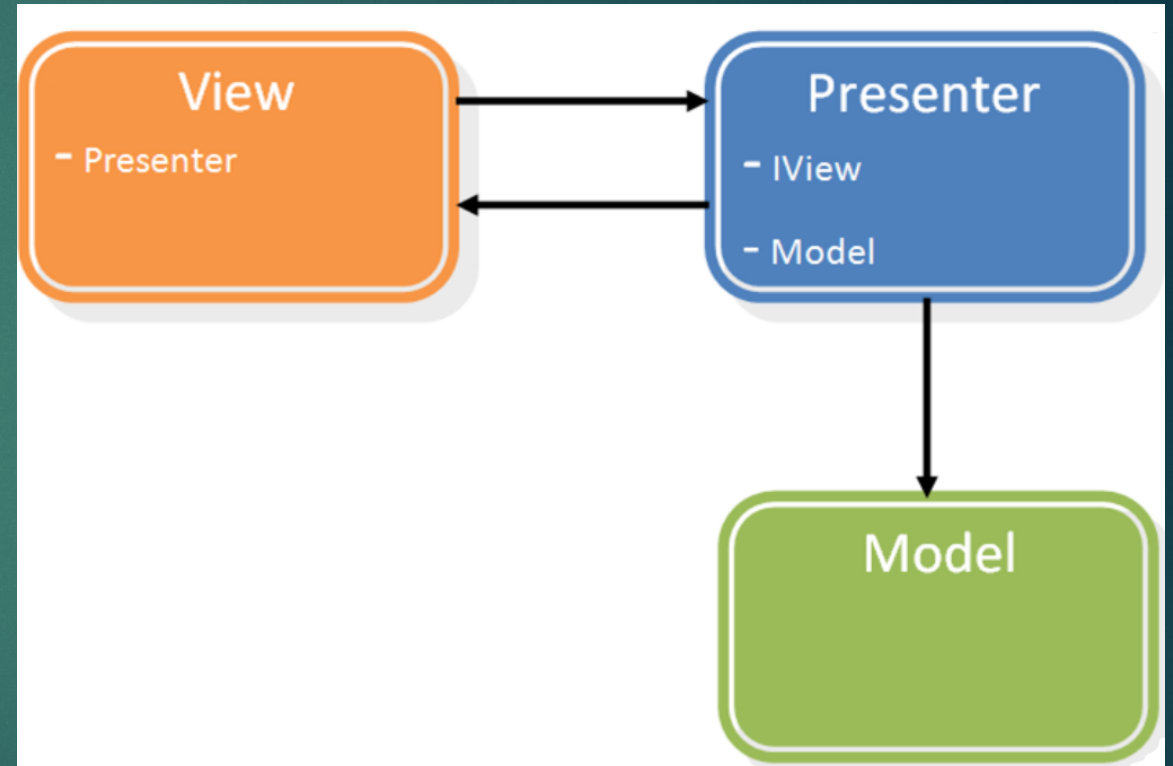
# Développement mobile

CRÉATION D'UNE INTERFACE



# Model View Presenter

- ▶ Presenter : Répartitions des données dans la vue, récupération des évènements de la vue
- ▶ View : Interface graphique passive
- ▶ Model : Données





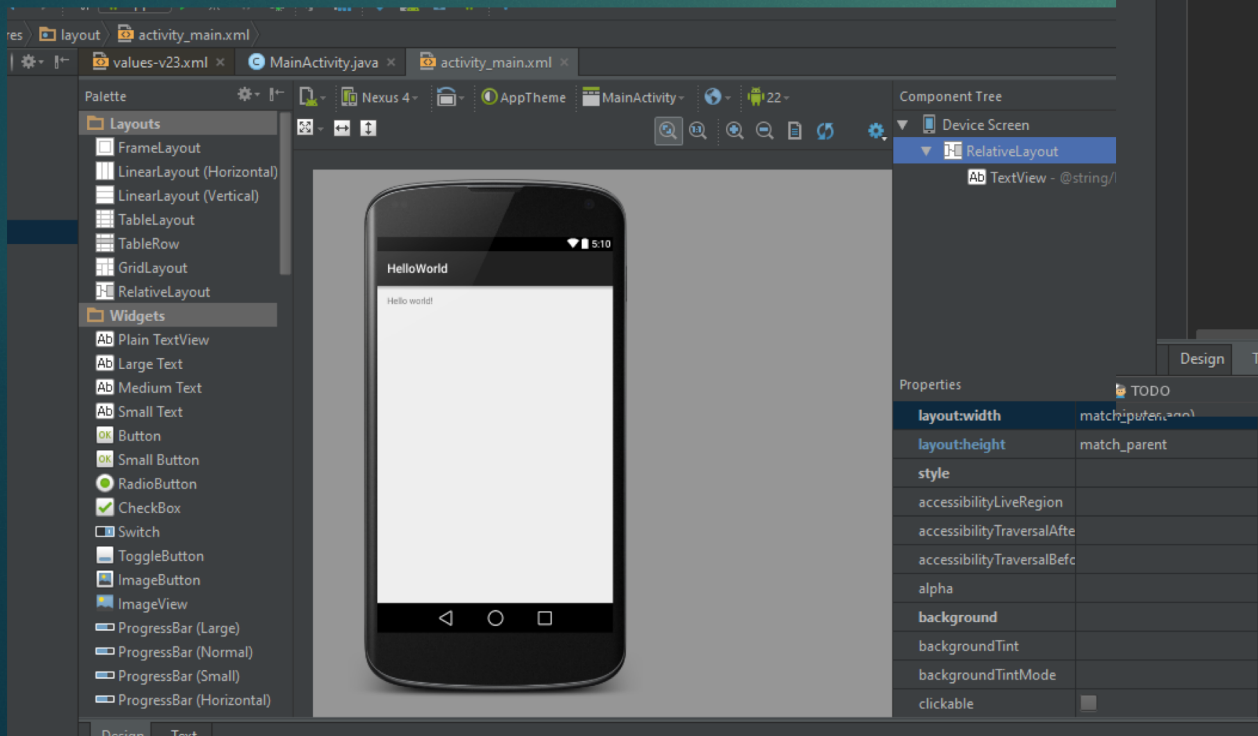
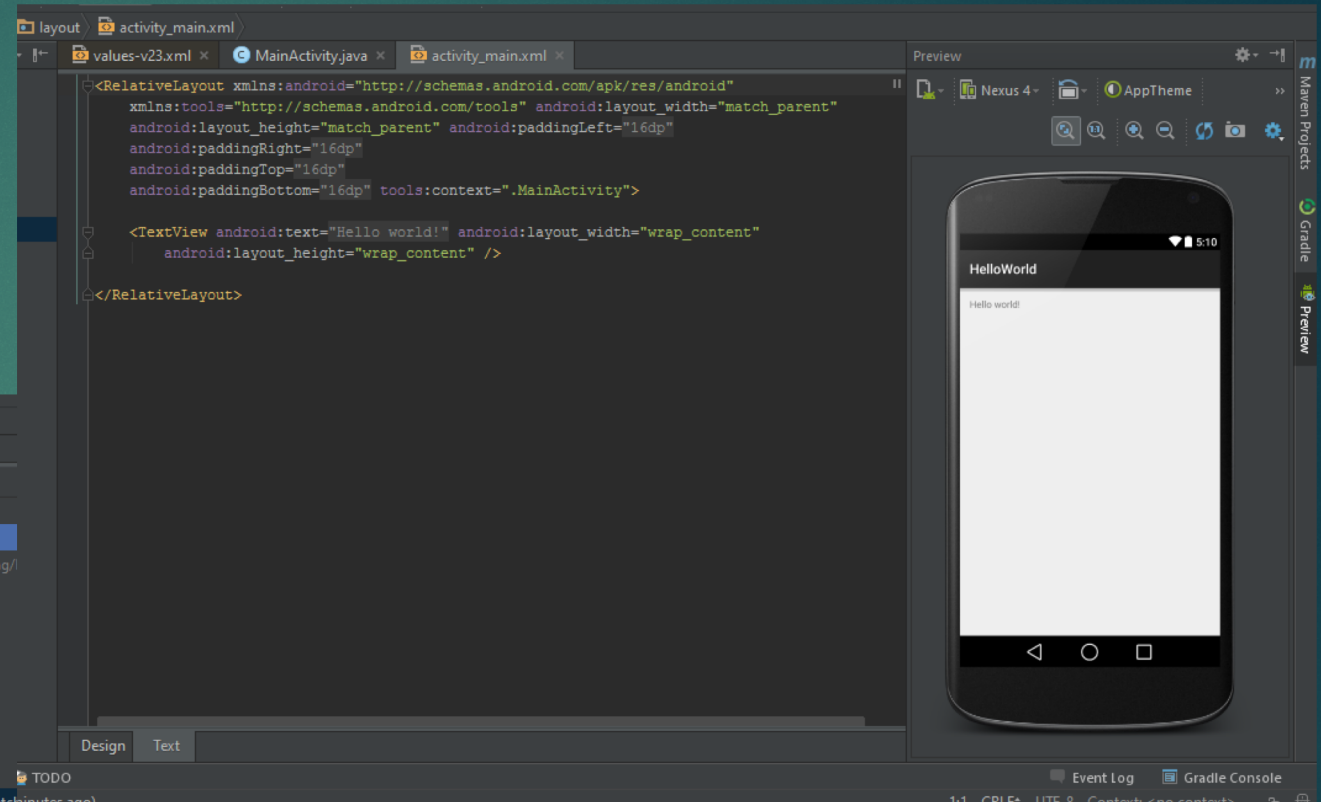
# Constitution d'un écran

- ▶ Activity
  - ▶ Comportement d'un écran
  - ▶ Recupération, injection de données
  - ▶ Réaction aux évènements
- ▶ Layout
  - ▶ Design de l'écran
  - ▶ Gestion des résolutions
  - ▶ Configuration des briques visuelles élémentaires



# Construction d'un layout

- ▶ Sous eclipse:
  - ▶ Editeur visuel
  - ▶ Editeur textuel <= à préférer





# Les layout

- ▶ Fichier xml
  - ▶ Jeux de balises
  - ▶ Un seul élément racine
- ▶ Briques de bases nombreuses
  - ▶ Layout (Relative, Linear)
  - ▶ TextView
  - ▶ Button
  - ▶ Checkbox
  - ▶ etc...

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.helloworld.MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```



# Quelques vues basique

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center_horizontal" >

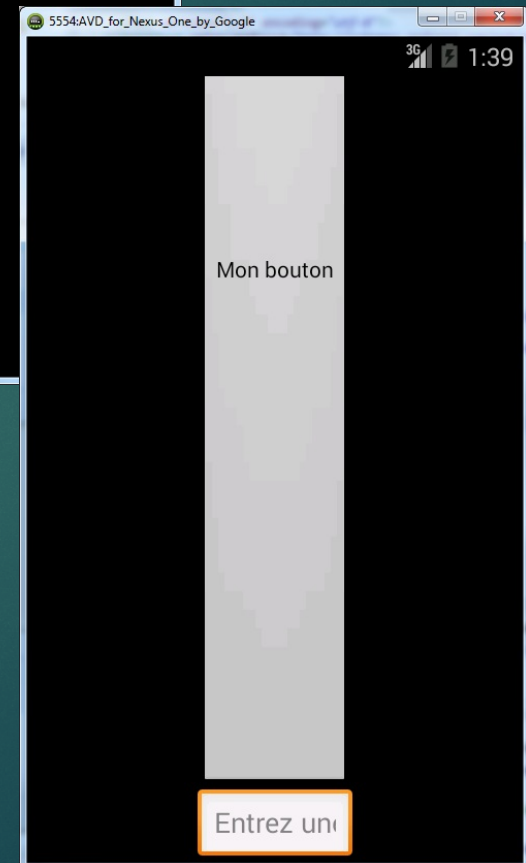
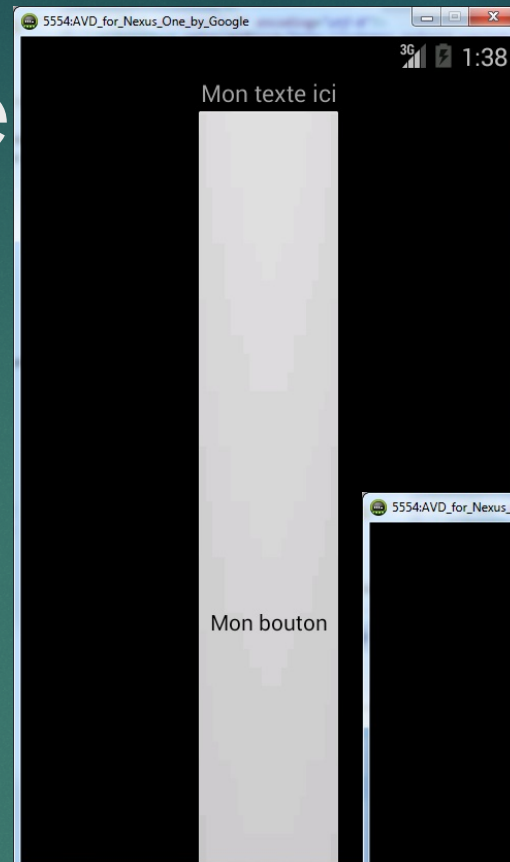
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Mon texte ici"
            android:textColor="#a3a3a3"
            android:textSize="15sp"/>

        <Button
            android:id="@+id/main_b_mon_bouton"
            android:layout_width="wrap_content"
            android:layout_height="1000px"
            android:text="Mon bouton" />

        <EditText android:id="@+id/main_et_mon_input"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:hint="Entrez une valeur"
            android:inputType="text"
            android:maxLines="1"/>

    </LinearLayout>

</ScrollView>
```





# Gestion des tailles

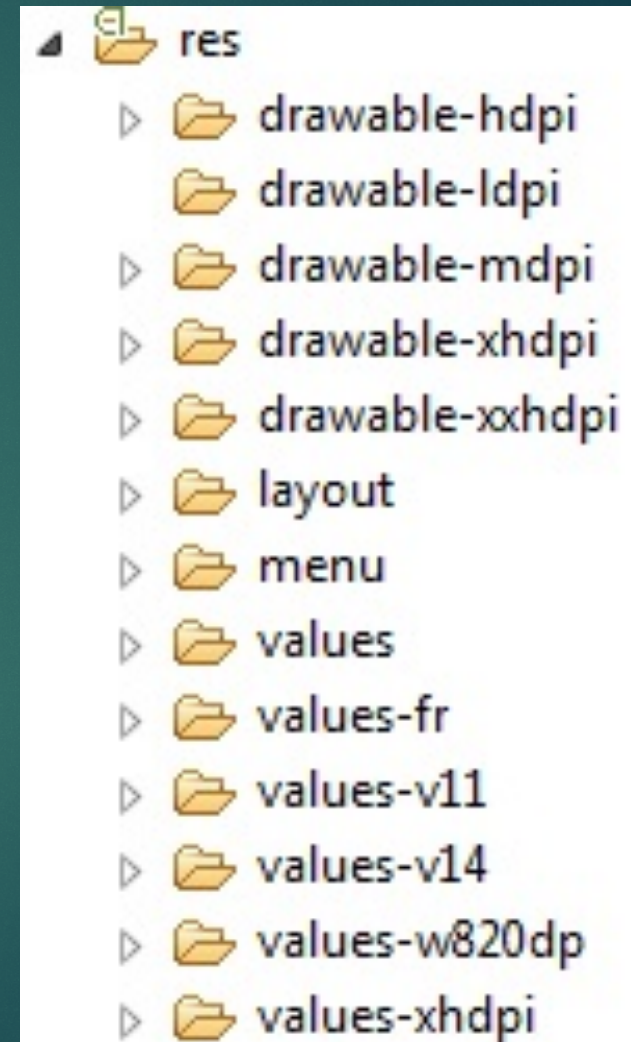
- ▶ Le fichier `dimens.xml`
  - ▶ Valeur utilisable dans les layouts
  - ▶ Multivalué

```
<resources>

    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16px</dimen>
    <dimen name="activity_default_font_size">15sp</dimen>

</resources>
```

- ▶ Les formats possibles :
  - ▶ Pixel : px
  - ▶ Scale-independant pixel : sp
  - ▶ Density-independant pixel : dp (ou dip)





# Lien entre layout et activity

- Récupérer un élément d'un layout dans une activity

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.draft.MainActivity" >

    <TextView
        android:id="@+id/main_tv_helloworld"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

```
public class MainActivity extends ActionBarActivity {

    private static final String HELLO_WORLD_TEXT = "Nouveau hello world !";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

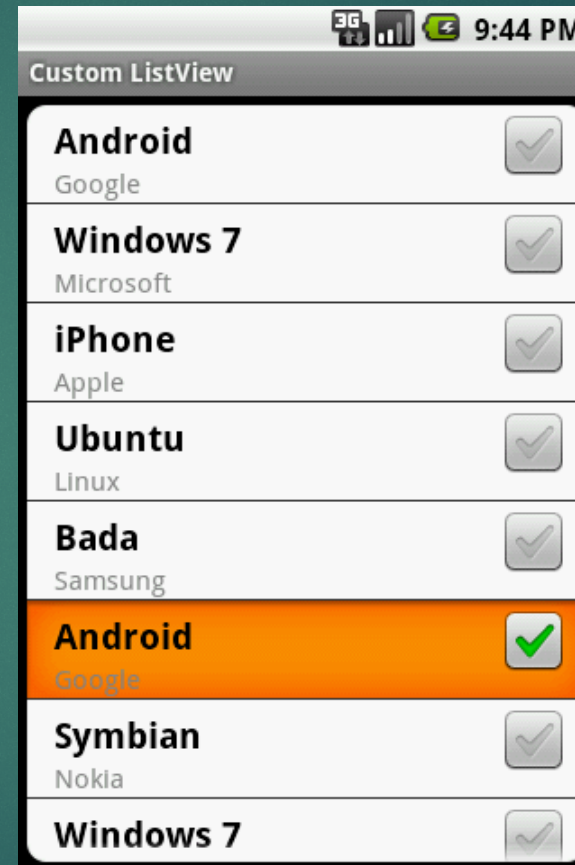
        updateHelloWorldText();
    }

    private void updateHelloWorldText() {
        TextView helloWorld = (TextView) findViewById(R.id.main_tv_helloworld);
        helloWorld.setText(HELLO_WORLD_TEXT);
    }
}
```



# Les ListView

- ▶ But
  - ▶ Affichage de nombreux éléments
  - ▶ Economies de mémoire
  - ▶ Economies de calculs





# Les ListView

- Implémentation de la liste

- Côté xml

```
<ListView
    android:id="@+id/main_lv_my_listview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

- Côté activity

```
private String[] dataToDisplay;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    dataToDisplay = initializeDataToDisplay();
    initializeListView(dataToDisplay);
}

private void initializeListView(String[] dataToDisplay) {
    ListView myListView = (ListView) findViewById(R.id.my_list_view);
    myListView.setAdapter(new CustomAdapter(this, dataToDisplay));
}
```



# Les ListView

- ▶ Les adapters
  - ▶ Rendent les éléments de la liste

```
public class CustomAdapter extends ArrayAdapter<String> {
    private static int ITEM_LAYOUT = R.layout.listview_item;
    private Context ctx;
    private String[] items;

    public CustomAdapter(Context context, String[] myItems) {
        super(context, ITEM_LAYOUT);
        ctx = context;
        items = myItems;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

        if (convertView == null) {
            LayoutInflater inflater = ((Activity) ctx).getLayoutInflater();
            convertView = inflater.inflate(ITEM_LAYOUT, parent, false);
        }

        String currentItem = items[position];

        TextView itemLabel = (TextView) convertView.findViewById(R.id.listview_item_tv_item_label);
        itemLabel.setText(currentItem);

        return convertView;
    }
}
```



# TP 02 : Interfaces

- ▶ Disponible ici :
  - ▶ <http://www.labri.fr/~zemmari/pam>