Université
Internationale
de Casablanca

Les composants graphiques

Cours 4

Programmation des systèmes embarqués Android :

Les layouts, les widgets, les listeners.

Wilfried Y. H. Adoni

Université Internationale de Casablanca

Programme de la séance

- Le Toast message
- Les widgets et listeners

Université Internationale de Casablanca

Les boites de dialogue: Toast

- Un **Toast** est un objet permettant d'afficher un message de type alerte à l'écran en tenant compte du contexte de l'activité associée.
- La définition de ce objet se fait dans la classe Activity.
- Syntaxe:

Toast.makeText(Context context , String message, int duration).show()

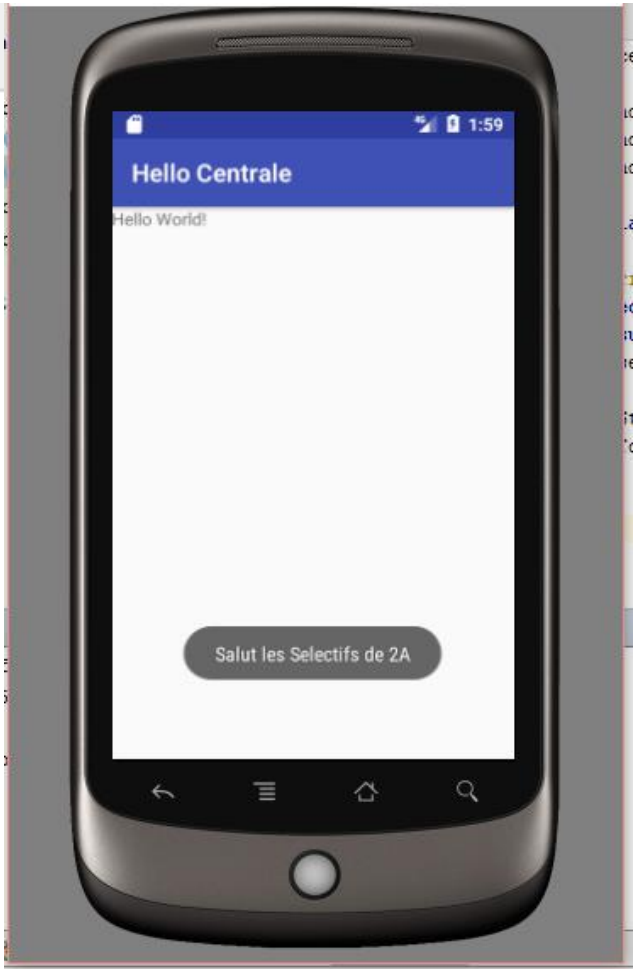
- **Context context** : définir la vue dans laquelle le message sera affiché.
- **String message** : le contenu du message à afficher.
- **Int duration** : la durée d'affiche du message.
- La méthode **show()** : permet d'afficher le toast message toast.

Université Internationale de Casablanca

Les boites de dialogue: Toast

- Exemple:

```
1  package centrale.ecole.ma.hellocentrale;
2
3  import android.support.v7.app.AppCompatActivity;
4  import android.os.Bundle;
5  import android.widget.Toast;
6
7  public class MainActivity extends AppCompatActivity {
8
9      @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_main);
13
14         String message = "Salut les Selectifs de 2A";
15         Toast.makeText(this.getApplicationContext(), message, Toast.LENGTH_LONG).show();
16     }
17 }
```



Université Internationale de Casablanca

Les widgets

- Les widgets ou composants graphiques sont des objets au sens POO du terme.
- Chaque type de widgets est une classe dérivée de la classe **View**.
- Ils sont placés à l'interieur des Layouts.
- Ils existent plusieurs types de widgets:
  - **TextView**
  - **EditView**
  - **Button**
  - **CheckBox**
  - **RadioButton**
  - **Spinner**

Université Internationale de Casablanca

TextView

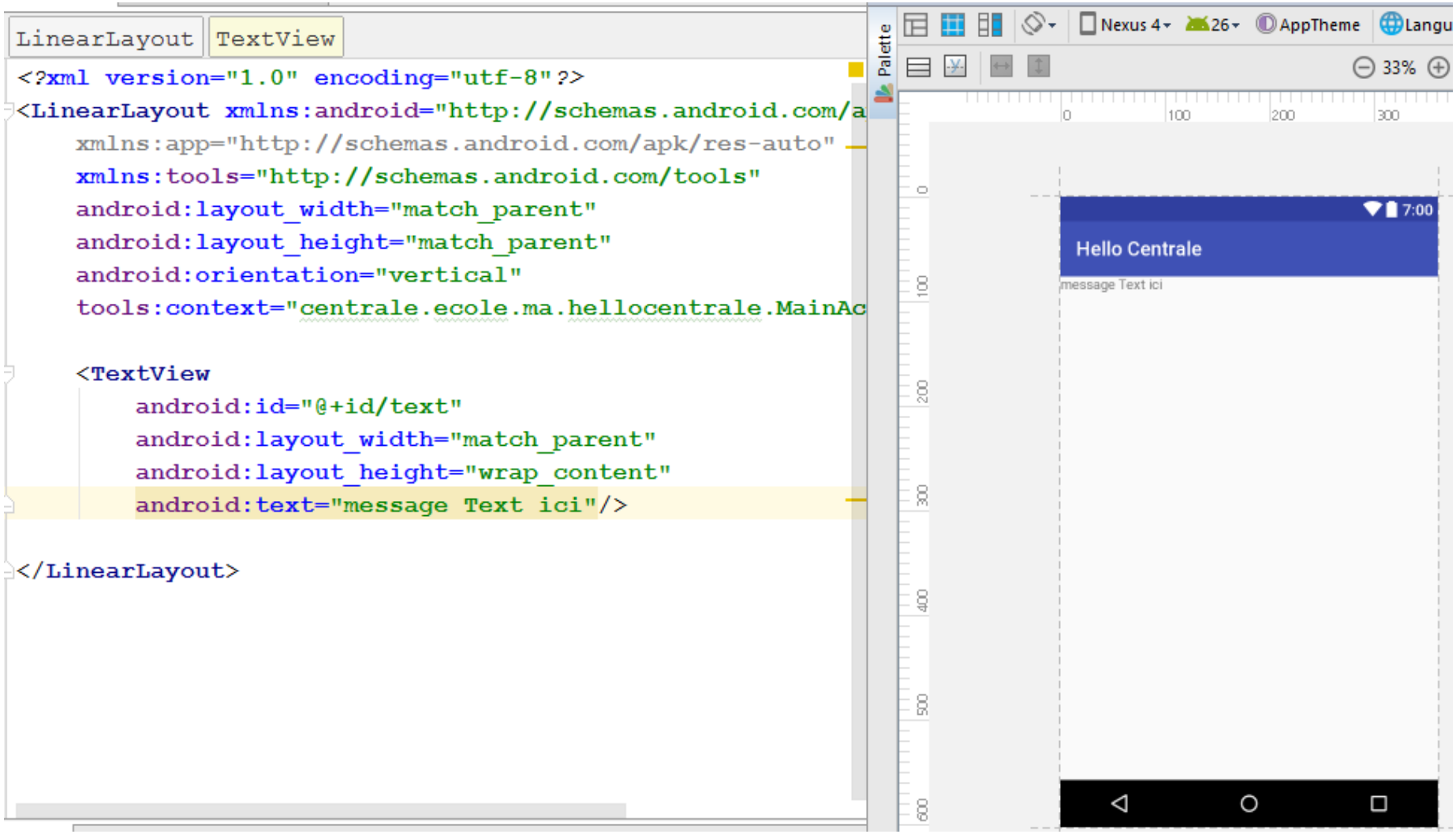
- Le widget **TextView** un objet permettant d'ajouter un texte sur l'écran du téléphone.
- La définition du TextView se fat dans le fichier layout et l'appel dans la classe Activity.
- Quelques méthodes associées à l'objet TextView:

méthode	rôle
`setText(String message)`	Modifie le texte à l'objet
`setTextColor(int color)`	Modifie la couleur du texte
`setBackgroundColor(int color)`	Modifie la couleur d'arrière du texte
`setVisible(int visibility);`	Modifie la visibilité du texte
`setTextSize(float size);`	Modifie la taille du texte

Université Internationale de Casablanca

Définition du TextView dans le fichier layout xml

- Le TextView est toujours à l'interieur d'un conteneur
- Ecriture basique d'un TextView



Université Internationale de Casablanca

Appel du TextView dans la classe d'activité

```
package centrale.ecole.ma.hellocentrale;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    // Déclaration d'une variable qui va contenir notre textView
    private TextView text;

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

        // recuperation de la vue du textView à l'aide de son ID
        text=(TextView)findViewById(R.id.text);

        //Modification du contenu du texte
        text.setText("Bonjour les élèves ingenieurs de 2A");
    }
}
```

Button

- Le widget **Button** est un objet permettant d'ajouter un bouton à l'écran du téléphone.
- La définition du Button se fait dans le fichier layout et l'appel dans la classe d'activité.
- Quelques méthodes associées à l'objet Button:

méthode	rôle
`setText(String message)`	Modifie le texte contenu dans le bouton
`setOnClickListener(OnClickListener l)`	Ajout une action d'écoute au bouton
`setClickable(boolean clickable)`	Modifie l'action du clique sur le bouton

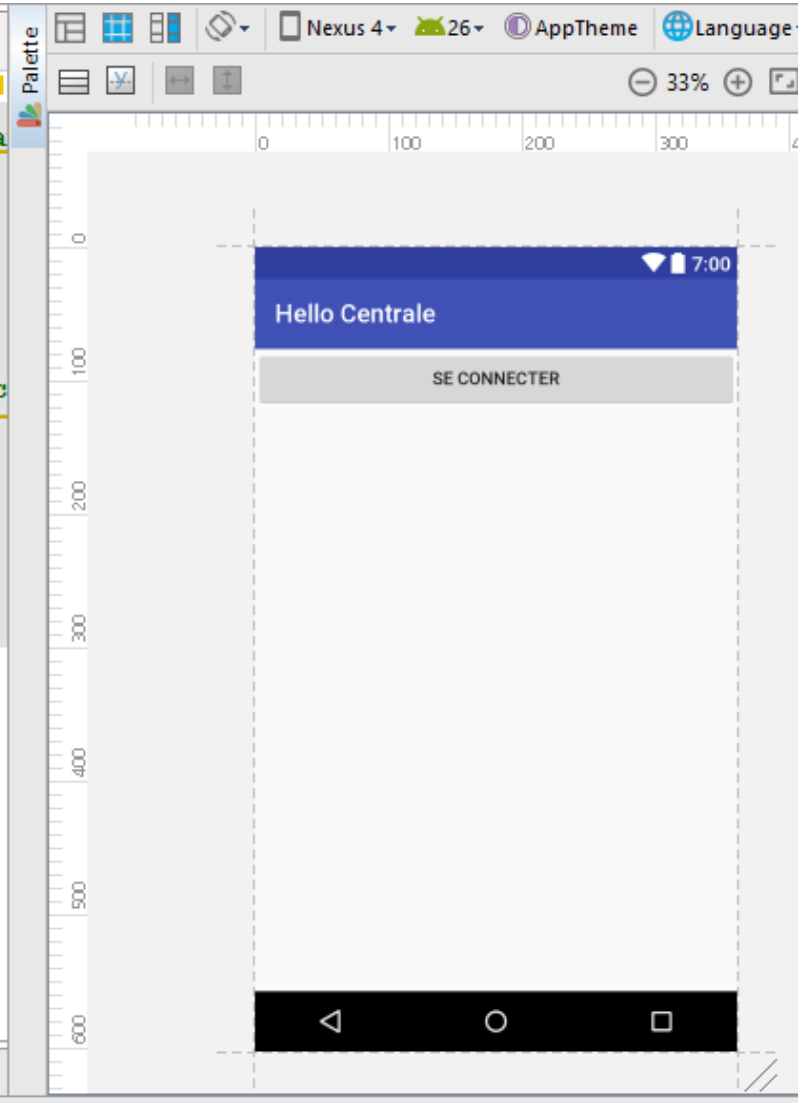
Université Internationale de Casablanca

Définition du Button dans le fichier layout xml

- Le Button est toujours à l'interieur d'un conteneur
- Ecriture basique d'un Button

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/a
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="centrale.ecole.ma.hellocentrale.MainAc

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Se Connecter"/>

</LinearLayout>
```

Text



Université Internationale de Casablanca

Appel du Button dans la classe d'activité

```
public class MainActivity extends AppCompatActivity {

    // Déclaration d'une variable qui va contenir notre bouton
    private Button btn;

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

        // recuperation de la vue du bouton à l'aide de son ID
        btn=(Button)findViewById(R.id.button);

        //Ajout d'une action d'ecoute à notre bouton
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String message = "Vous avez cliqué sur le bouton login!!";
                Toast.makeText(getApplicationContext(), message, Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

Université Internationale de Casablanca

EditText

- Le widget **EditText** est un objet permettant d'ajouter un champ de saisie à l'écran du téléphone.
- La définition du EditText se fait dans le fichier layout et l'appel dans la classe d'activité.
- Quelques méthodes associées à l'objet EditText:

méthode	rôle
`setText()`	Modifie le contenu du champ de saisie
`getText()`	Récupère la valeur saisie par l'utilisateur

Université Internationale de Casablanca

Définition du EditText dans le fichier layout xml

- Le EditText est toujours à l'interieur d'un conteneur

ty_main.xml × MainActivity.java ×

LinearLayout EditText

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context="centrale.ecole.ma.hellocentrale.MainActivity">


    <EditText
        android:id="@+id/edtEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Entrer votre email"
        android:inputType="textEmailAddress" />

    <EditText
        android:id="@+id/edtPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Entrer votre mot de passe"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/btnLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Login" />

</LinearLayout>
```

Preview

Palette

Nexus 4 26 AppTheme Language 50%

Université Internationale de Casablanca

Appel du EditText dans la classe d'activité

```
public class MainActivity extends AppCompatActivity {

    // Déclaration d'une variable qui va contenir notre bouton
    private Button login;
    private EditText email, password;

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

        // recuperation des widgets à l'aide de leur ID
        login=(Button)findViewById(R.id.btnLogin);
        email=(EditText)findViewById(R.id.edtEmail);
        password=(EditText)findViewById(R.id.edtPassword);

        //Ajout d'une action d'ecoute à notre bouton
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // recupere le contenu du champ email
                String inputEmail=email.getText().toString();
                //recupere le contenu du champ password
                String inputPassword=password.getText().toString();
                String message = "Bienvenue vos coordonnees sont : \n login :                    "+ in
                Toast.makeText(getApplicationContext(), message, Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

Université Internationale de Casablanca

CheckBox

- Le widget **CheckBox** est un objet jouant le rôle d'une case à cocher.
- Une case qui peut être dans deux états : cochée ou pas.
- Exemple en XML:
  - **android:checked="true"** signifie que la case est cochée par défaut.

```
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:checked="true"
    android:text="CheckBox"/>
```

Etat = **True** (Coché)

Etat = **False** (Non Coché)

- Quelques méthodes associées à l'objet CheckBox:

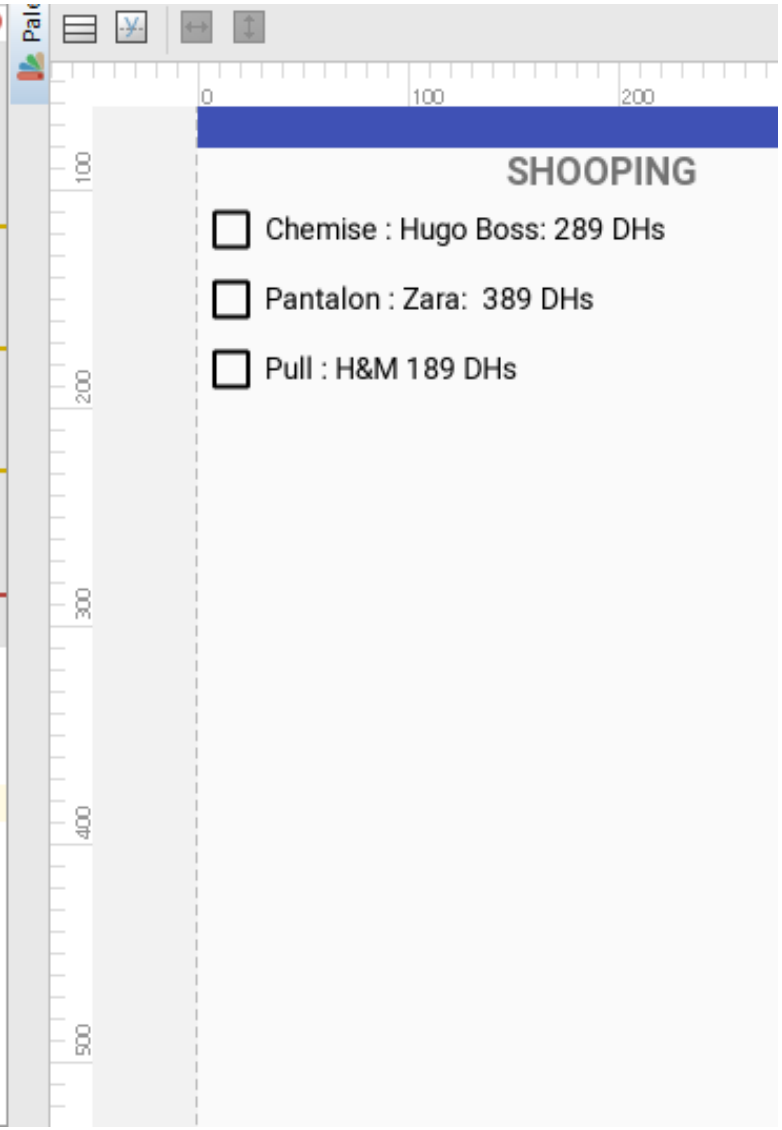
méthode	rôle
`setText()`	Modifie le contenu du champ de saisie
`setChecked()`	Récupère la valeur saisie par l'utilisateur
`isChecked()`	

Université Internationale de Casablanca

Définition du CheckBox dans le fichier layout xm

- Le CheckBox est toujours à l'interieur d'un conteneur

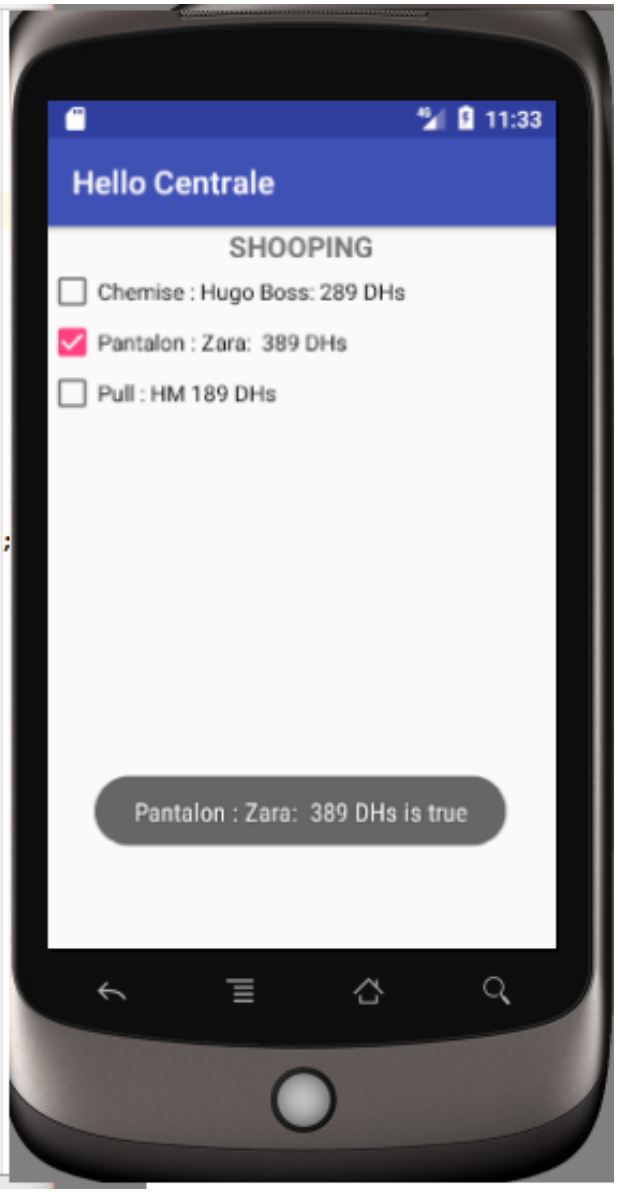
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="centrale.ecole.ma.hellocentrale.MainActivity">


    <TextView...>

    <CheckBox
        android:id="@+id/checkbox_chemise_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Chemise : Hugo Boss: 289 DHs"/>

    <CheckBox
        android:id="@+id/checkbox_pantalon_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Pantalon : Zara:  389 DHs"/>

    <CheckBox
        android:id="@+id/checkbox_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Pull : H&M 189 DHs"/>

</LinearLayout>
```



Université Internationale de Casablanca

Appel du CheckBox dans la classe d'activité

```
private CheckBox pantalon, chemise, pull;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    pantalon =(CheckBox)findViewById(R.id.checkbox_pantalon_id);
    chemise =(CheckBox)findViewById(R.id.checkbox_chemise_id);
    pull =(CheckBox)findViewById(R.id.checkbox_pull_id);

    pantalon.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(getApplicationContext(),
                    pantalon.getText()+" is "+pantalon.isChecked()+"", Toast.LENGTH_LONG).show();
        }
    });
    chemise.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(getApplicationContext(),
                    chemise.getText()+" is "+chemise.isChecked()+"", Toast.LENGTH_LONG).show();
        }
    });
    pull.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(getApplicationContext(),
                    pull.getText() + " is "+pull.isChecked()+"", Toast.LENGTH_LONG).show();
        }
    });
```

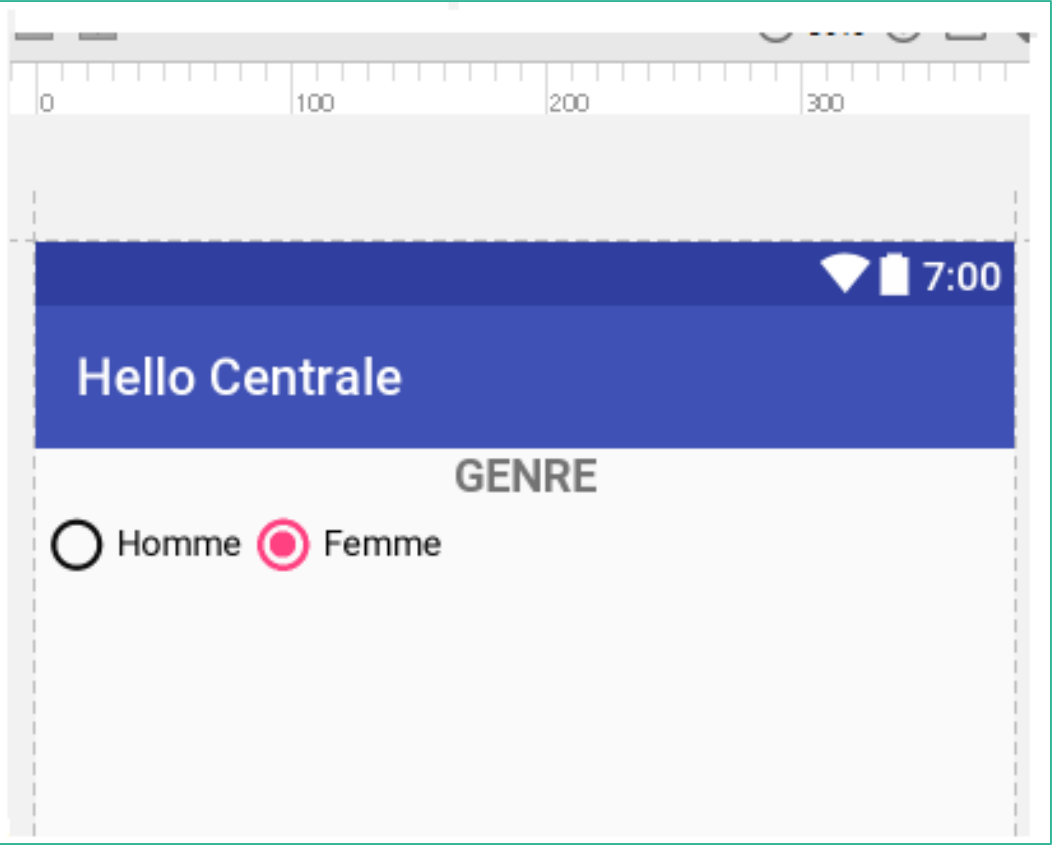


Université Internationale de Casablanca

RadioGroup et RadioButton

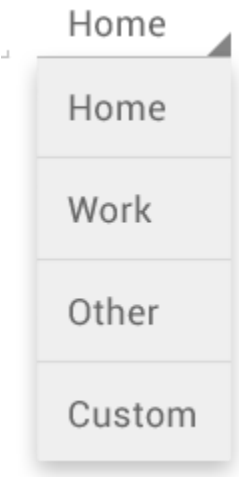
- Même principe que la CheckBox, à la différence que l'utilisateur ne peut cocher qu'une seule case.
- Il est plutôt recommandé de les regrouper dans un **RadioGroup**
- Exemple en XML:

```
    <TextView...>
<RadioGroup
    android:id="@+id/genre"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <RadioButton
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Homme"/>
    <RadioButton
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Femme"
        android:checked="true"/>

</RadioGroup>
```



Spinner

- **Spinner** est un widget similaire à une liste déroulante pour sélectionner des éléments.
- Toucher le **Spinner** affiche un menu déroulant avec toutes les autres valeurs disponibles, à partir de laquelle l'utilisateur peut en sélectionner un nouveau.
- La creation d'un widget de type **Spinner** se fait en 4 étapes :
  - ❑ Declaration du spinner dans le fichier xml
  - ❑ Chargé le Spinner à l'aide d'une liste de choix de type ArrayString
  - ❑ Ajouter un Adapter pour adapter le contenu
  - ❑ Associer une action lorsque l'utilisateur effectue un choix

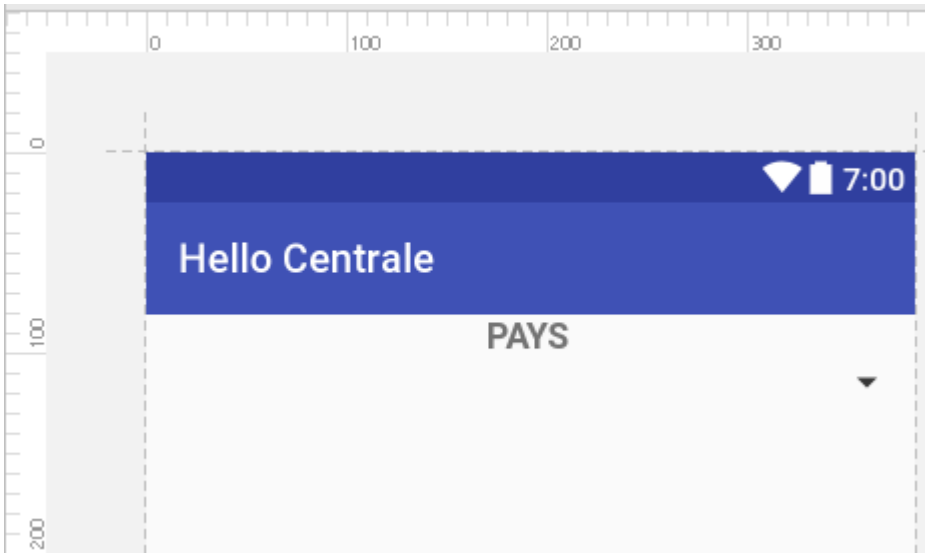


Université Internationale de Casablanca

Définition du Spinner dans le fichier layout xm

- Le Spinner est toujours à l'interieur d'un conteneur

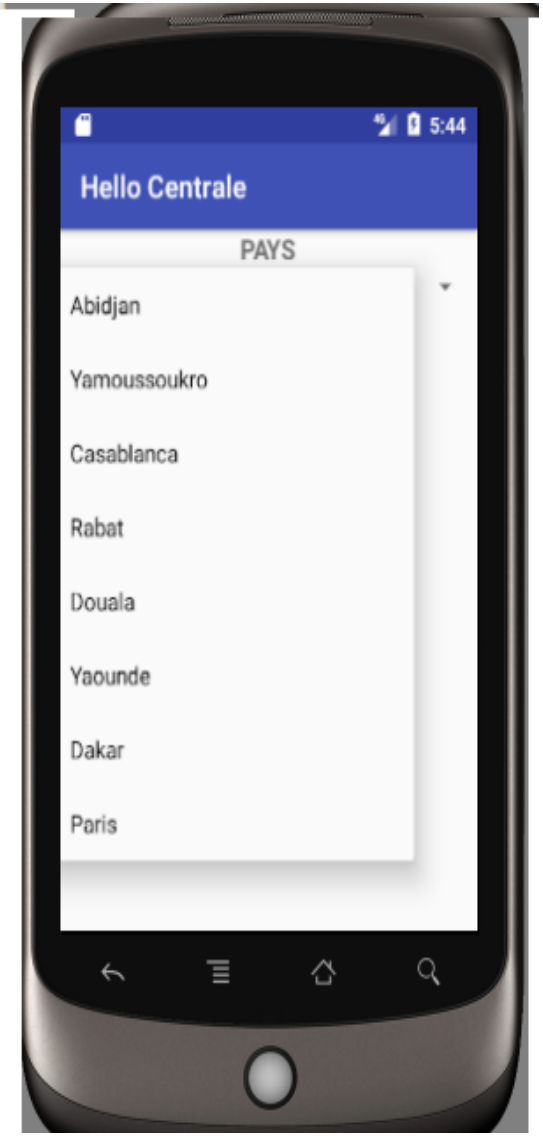
```
<Spinner
    android:id="@+id/spinner_pays"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```



Université Internationale de Casablanca

Appel du Spinner dans la classe d'activité

```
public class MainActivity extends AppCompatActivity {
    // Déclaration d'une variable qui va contenir notre spinner
    private Spinner spinner;

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

        // 1- recuperation de l'id de notre spinner
        spinner=(Spinner)findViewById(R.id.spinner_pays);
        // 2 - Creation des items du spinner via la une table de String
        final String capitale[] = {"Abidjan", "Yamoussoukro", "Casablanca", "Rabat", "Douala", "Yaounde", "Dakar", "Paris"};
        // 3- Creation d'un adapteur
        ArrayAdapter<String> adapter =new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item ,capitale);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        // 4- ajout de notre adapteur à notre spinner
        spinner.setAdapter(adapter);
        // 5 -  ajout d'une action au clique du spinner
        spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
                Toast.makeText(getApplicationContext(), capitale[position].toString(), Toast.LENGTH_LONG).show();
            }

            @Override
            public void onNothingSelected(AdapterView<?> parent) {
                Toast.makeText(getApplicationContext(), "Aucune selection", Toast.LENGTH_LONG).show();
            }
        });
    }
}
```



Université Internationale de Casablanca

Ressources

- Documentation officielle :
  - https://developer.android.com/index.html
- Tutoriel Android :
  - https://developer.android.com/studio/index.html