Android: Menus et Messages

Assane SECK
Ingénieur-Informaticien

Sommaire

Menus et Messages

Afficher un toast
Ajouter un option menu
Ajouter un popup menu
Afficher un dialog
Spinner
Champ Autocomplete

AFFICHER UN Toast

AFFICHER UN Toast

- Un **Toast** est un message qui apparait et disparait
- Il est impossible de savoir si l'utilisateur l'a vu ou pas
- Configuration:
- Un **String** contenant le message
- Une durée
- **Toast.LENGTH_LONG** ou **Toast.LENGTH_SHORT**
- Il est aussi possible de donner une **View** de votre choix en paramètre

```
Toast.makeText(getApplicationContext(), "Ceci est un toast !",
Toast.LENGTH_LONG).show();
```

Sommaire

Menus et Messages

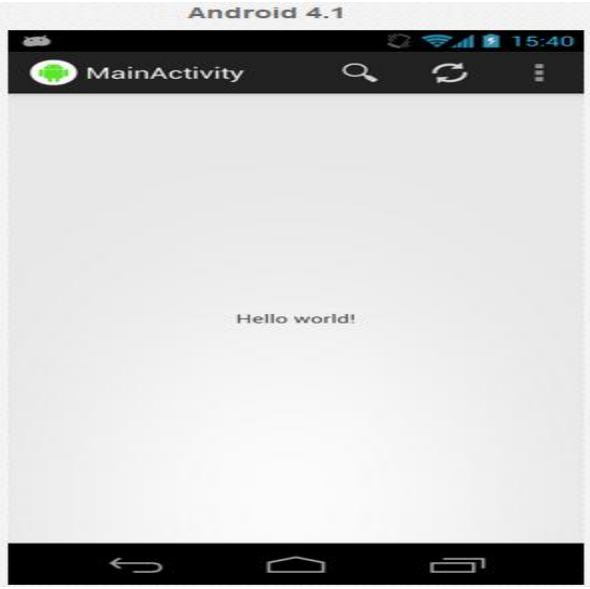
Afficher un toast
Ajouter un option menu
Ajouter un popup menu
Afficher un dialog
Spinner
Champ Autocomplete

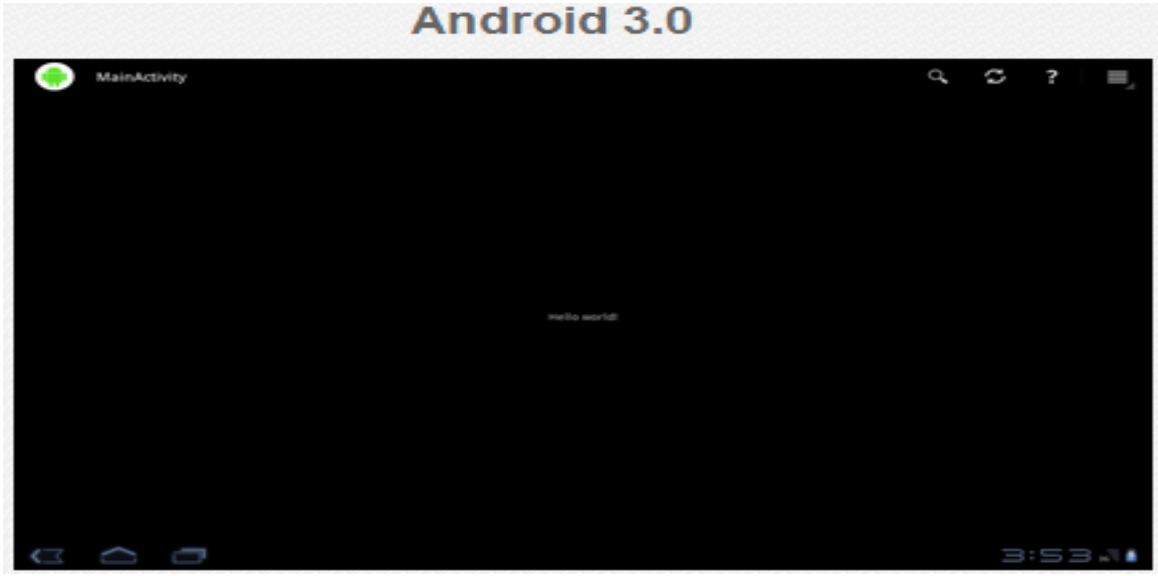
ActionBar|option menu : With Version API

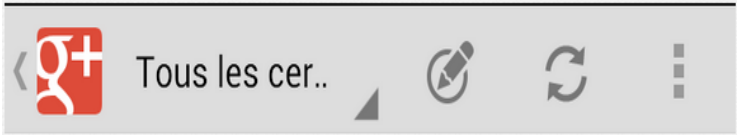
La version 3.0 d'Android (**ActionBar**).

Il remplace les menus disponibles dans les anciennes version d'Android

Permet à l'utilisateur d'accéder facilement aux fonctionnalités de l'application







AJOUTER UN OPTION MENU

- Lorsqu'un utilisateur appuie sur la touche menu, il lui est présenté un menu avec:

- maximum 6 options.

- Si vous avez besoin de plus, pensez à ajouter un submenu.

OPTION MENUS : COMMENT

- Overridez **onCreateOptionsMenu** dans votre Activity
- n'oubliez pas de faire appel à super.onCreateOptionsMenu()
- cette fonction reçoit une instance de Menu en paramètre
- appelez que lorsque la touche menu est appuyée pour la première fois
- Il est possible de modifier les éléments en faisant un override de onPrepareOptionsMenu

showAsAction

- L'attribut **showAsAction** permet de spécifier le comportement de l'élement dans une ActionBar, il peut posséder les valeurs suivantes :
  - **ifRoom** : L'élement sera ajouté aux actions principales de l'ActionBar si une place est disponible
  - **never** : Ne jamais rajouter l'action aux actions principales de l'ActionBar
  - **always** : Toujours rajouter l'action aux actions principales de l'ActionBar. Cette valeur n'est pas conseillé car peut entrainer une superposition d'élement (si nombre de place disponible < nombre d'élément ajouté à l'ActionBar), préférez la valeur ifRoom.
  - **withText** : Toujours afficher le texte représentant l'action

OPTIONS MENU : COMMENT

- On peut rajouter des éléments en utilisant la fonction **add()**
- La fonction **add() a différentes formes, mais nous utiliserons celle-ci:**
- Un identifiant de groupe (int), généralement 0, sauf si vous voulez grouper des choix de menu ensembles pour utiliser par exemple avec
- **setGroupCheckable()**
- Un identifiant de choix (int), pour identifier l'option sélectionnée dans
- **onOptionsItemSelected()**
- Un identifiant d'ordre qui indique où l'élément devrait être positionné. Il n'est cependant pas certain qu'Android en tiendra compte.
- Le texte de ce menu, peut être un string ou ressource id
- renvoie une instance de **MenuItem**
- Il est possible de mettre une icone sur un MenuItem en utilisant
- **setIcon()** avec une id d'image ou Drawable
- **addSubMenu()** fonctionne de la même manière.

Option Menu: activity_main.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/selection"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:drawSelectorOnTop="false" />
</LinearLayout>
```

Option Menu: menu.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/it1"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="Config"/>
    <item
        android:id="@+id/it2"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="A Propos"/>
    <item
        android:id="@+id/it3"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="Aide"
        android:icon="@drawable/ic_launcher"/>
</menu>
```

Option Menu: **onCreateOptionsMenu**

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {

new MenuInflater(getApplicationContext()).inflate(R.menu.menu,
menu);
return super.onCreateOptionsMenu(menu);

}
```

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.main, menu);

return true;

}
```

Option Menu: **onOptionsItemSelected**

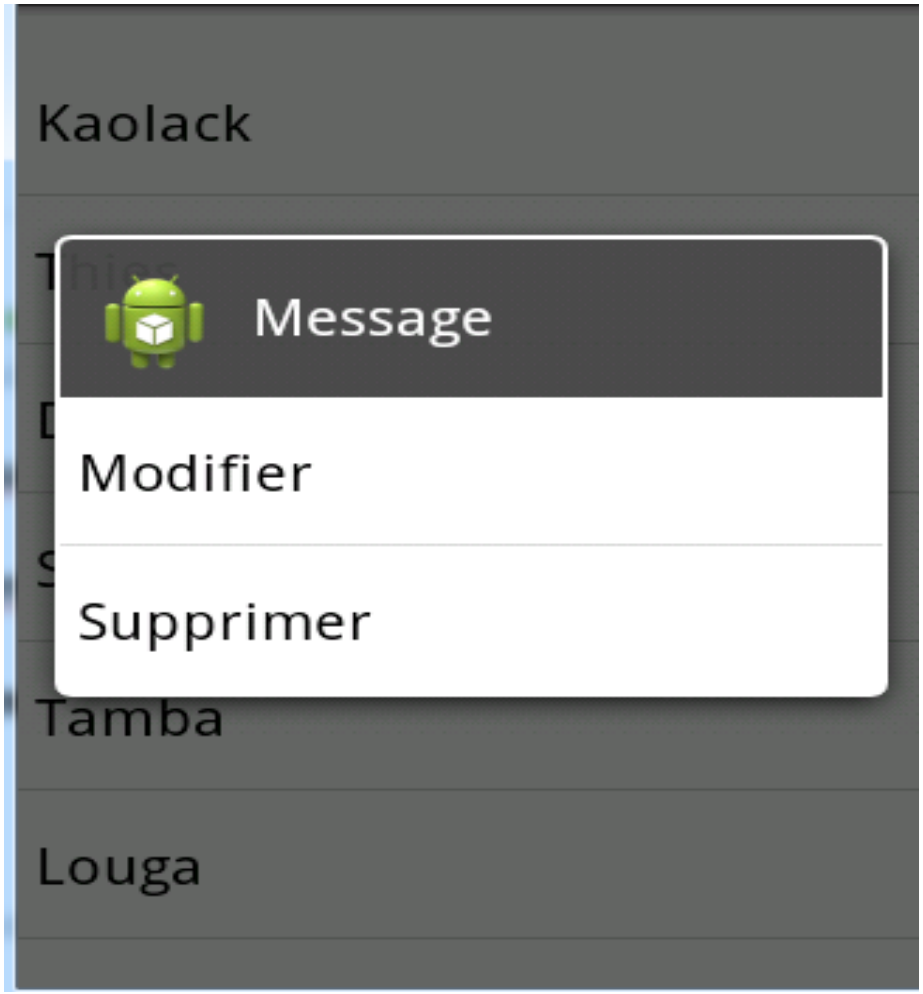
```
@Override
public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {
case R.id.it1:
Toast.makeText(getApplication(), item.getTitle(),
Toast.LENGTH_SHORT).show();
break;
case R.id.it2:
Toast.makeText(getApplication(), item.getTitle(),
Toast.LENGTH_SHORT).show();
break;
default:
break;
}
return super.onOptionsItemSelected(item);
}
```

Sommaire

Menus et Messages

Afficher un toast
Ajouter un option menu
Ajouter un popup menu
Afficher un dialog
Spinner
Champ Autocomplete

MENUS CONTEXTUELS



MENUS CONTEXTUELS

- On y fait souvent référence comme «popup menus»
- Fonctionne de la même manière que les option menus
- Il est possible ici d'ajouter une en-tête avec **setHeaderTitle(title)**
- Comment ?
- Enregistrez votre view pour affichage du menu contextuel sur votre activity
- **Activity#registerForContextMenu()**
- Implémentez **onCreateContextMenu()**
- Obtenez le **ContextMenu,View et ContextMenuInfo**
- Implémentez **onContextItemSelected()**
- Utilisez des identifiants uniques sur vos **MenuItem !**

MENU CONTEXTUEL: **onCreateContextMenu**

```
@Override
public void onCreateContextMenu(ContextMenu menu,
View v, ContextMenuInfo menuInfo) {

menu.setHeaderIcon(R.drawable.ic_launcher);
menu.setHeaderTitle("Message");
menu.add(0, 0, 0, "Modifier");
menu.add(0, 1, 0, "Supprimer");

super.onCreateContextMenu(menu, v, menuInfo);

}
```

MENU CONTEXTUEL: **onContextItemSelected**

```
@Override
public boolean onContextItemSelected(MenuItem item) {

switch (item.getItemId()) {

case 0:
Toast.makeText(getApplication(), item.getTitle(),
Toast.LENGTH_SHORT).show();
case 1:
Toast.makeText(getApplication(), item.getTitle(),
Toast.LENGTH_SHORT).show();
}
return super.onContextItemSelected(item);

}
```

MENUS CONTEXTUELS

. **registerForContextMenu(** widget);

. View est un widget

Un appui long sur le widget fera apparaitre le menu contextuel, appel de la méthode **onCreateContextMenu()**

Exemple Complet. activity_main.xml

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/selection"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:drawSelectorOnTop="false" />
</LinearLayout>
```

Exemple Complet. main.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/it1"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="Config"/>
    <item
        android:id="@+id/it2"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="A Propos"/>
    <item
        android:id="@+id/it3"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="Aide"
        android:icon="@drawable/ic_launcher"/>
</menu>
```

Exemple complet Menu Option: MainActivity.java

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.main, menu);
return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
switch (item.getItemId()) {
case R.id.it1:
    Toast.makeText(getApplication(), item.getTitle(),
    Toast.LENGTH_SHORT).show();
break;
case R.id.it2:
    Toast.makeText(getApplication(), item.getTitle(),
    Toast.LENGTH_SHORT).show();
break;
default:
break;
}
return super.onOptionsItemSelected(item);
}
```

Exemple complet menu Contextuel: MainActivity.java

```
public class MainActivity extends ListActivity {
TextView selection;
String[] items = { "Kaolack", "Thies", "Dakar", "Saint
Louis", "Tamba",
"Louga", "Fatick", "Touba", "Diourbel", "Kolda",
"Ziguinchor", };
@Override
public void onCreate(Bundle icicle) {
super.onCreate(icicle);
setContentView(R.layout.activity_main);
setListAdapter(new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, items));
selection = (TextView) findViewById(R.id.selection);
registerForContextMenu(getListView());
}
```

Exemple complet Menu Contextuel: MainActivity.java

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenuInfo menuInfo) {
menu.setHeaderIcon(R.drawable.ic_launcher);
menu.setHeaderTitle("Message");
menu.add(0, 0, 0, "Modifier");
menu.add(0, 1, 0, "Supprimer");
super.onCreateContextMenu(menu, v, menuInfo);
}
@Override
public boolean onContextItemSelected(MenuItem item) {
switch (item.getItemId()) {
case 0:
Toast.makeText(getApplication(), item.getTitle(),
Toast.LENGTH_SHORT).show();
case 1:
Toast.makeText(getApplication(), item.getTitle(),
Toast.LENGTH_SHORT).show();
}
return super.onContextItemSelected(item);
}
}
```

Sommaire

Menus et Messages

Afficher un toast
Ajouter un option menu
Ajouter un popup menu
Afficher un dialog
Spinner
Champ Autocomplete

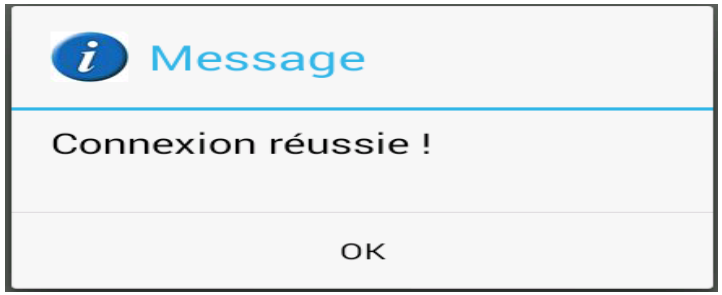
AFFICHER DES DIALOGUES

- Box dialogue classique : **AlertDialog ou AlertDialog.Builder**
- Moyen le plus facile de construire un alert dialog: en utilisant un **Builder**
- **AlertDialog** pour setButton uniquement
- **AlertDialog .Builder** pour setPositiveButton, setNegativeButton, etc…

- new **AlertDialog.Builder(this).create()** avec **AlertDialog**
- new **AlertDialog.Builder(this)** avec **AlertDialog.Builder**
- Fonctions de configuration:
- **setMessage()**
- **setTitle() and setIcon()**
- **setPositiveButton(), setNegativeButton()** and **setNeutralButton()**
- Finalement :
- Faites appel à show() afin de montrer l'**AlertDialog**
- Ou à create() pour seulement créer l'**AlertDialog**

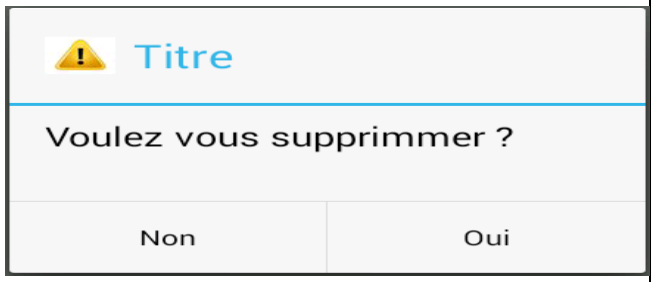
AlertDialog: **setButton**

```
AlertDialog dialog = new AlertDialog.Builder(MainActivity.this).create();
dialog.setTitle("Titre");
dialog.setMessage("Connexion réussie!");
dialog.setIcon(R.drawable.ic_launcher);
dialog.setButton(DialogInterface.BUTTON_NEUTRAL ,"OK", new
DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
Toast.makeText(getApplicationContext(), "OK",
Toast.LENGTH_LONG).show();
}
});
dialog.show();
```



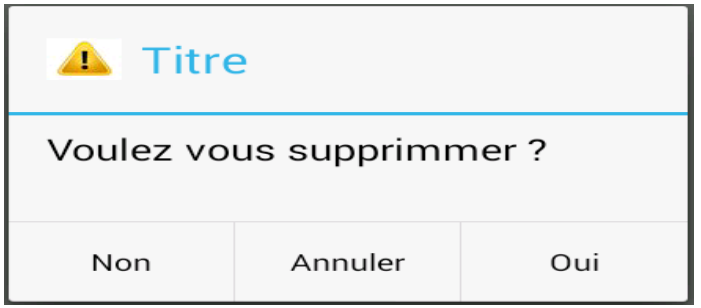
AlertDialog.Builder : setPositiveButton, setNegativeButton

```
AlertDialog.Builder dialog = new AlertDialog.Builder(
MainActivity.this);
dialog.setTitle("Titre");
dialog.setMessage("Voulez vous supprimmer ?");
dialog.setIcon(R.drawable.warning);
dialog.setPositiveButton("Oui",
new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog,
int which) {
}
});
dialog.setNegativeButton("Non",
new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog,
int which) {
}
});
dialog.show();
```

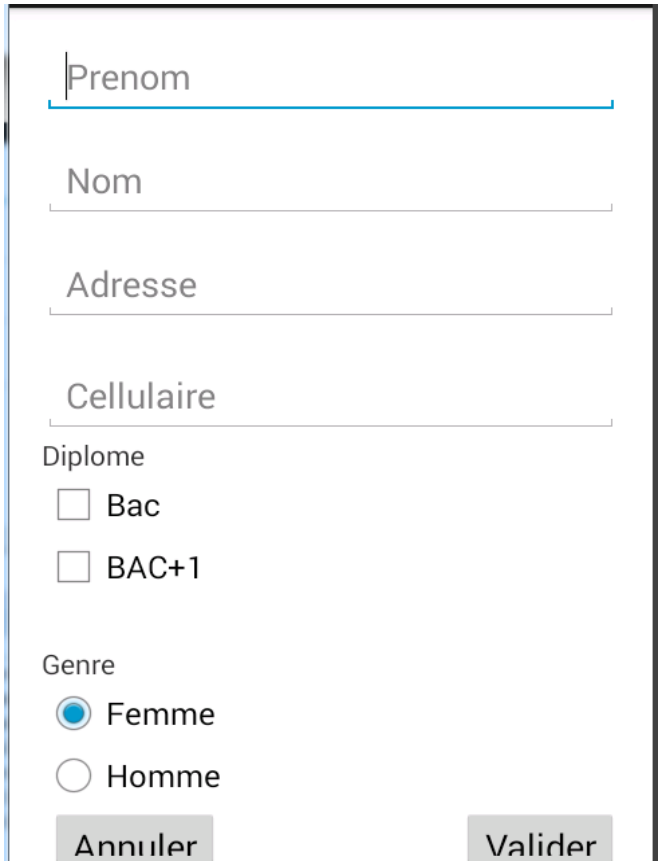


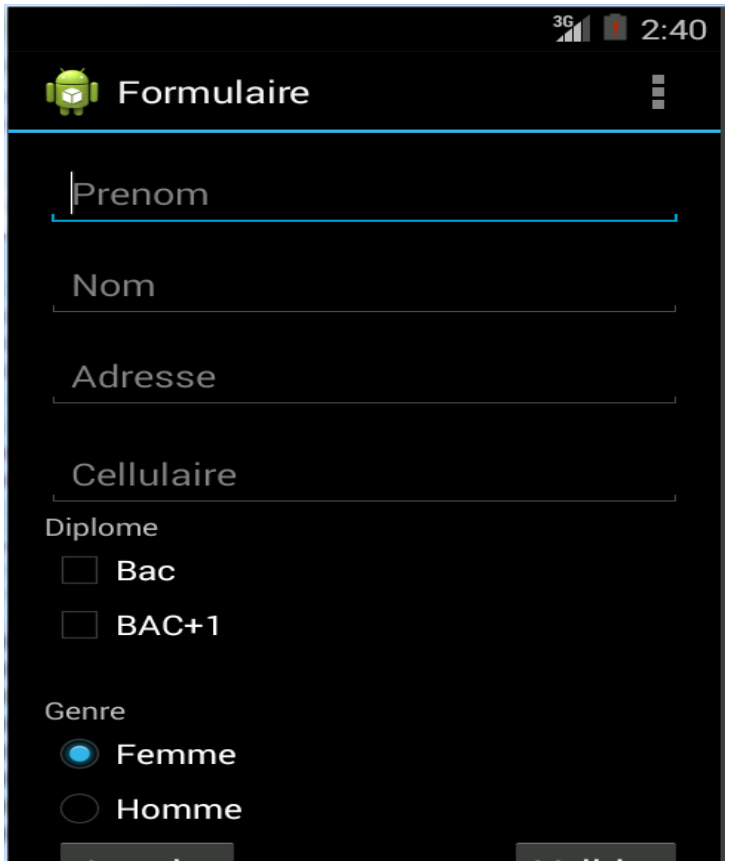
AlertDialog.Builder : setPositiveButton, setNegativeButton, setNeutralButton

```
AlertDialog.Builder dialog = new AlertDialog.Builder(
MainActivity.this);
dialog.setTitle("Titre");
dialog.setMessage("Voulez vous supprimer ?");
dialog.setIcon(R.drawable.warning);
dialog.setPositiveButton("Oui",
new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog,
int which) {
}
});
dialog.setNegativeButton("Non",
new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog,
int which) {
}
});
dialog.setNeutralButton("Annuler",
new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog,
int which) {
}
});
dialog.show();
```



Exercice (Prévoir le formulaire résumé)





FIN