

Résumé pour TP 4 : Le style, les chaînes de caractères et les menus

TP4 avec 2 parties (IHM+ Programmation)

Objectif :

1. Savoir créer un menu
2. Savoir créer, modifier et appliquer les chaînes de caractères (strings),
3. Savoir créer les tableaux de chaînes de caractères, et les afficher dans une **liste** ou **Spinner**.

Travail demandé

1. Démarrer eclipse
2. Créer un nouveau projet et l'appeler par « TP4»

I). L'interface utilisateur

1. Ouvrir le fichier MainActivity.xml du projet.
2. Créer le formulaire qui représente les données suivantes (travail du TP1) :



3. contenant les éléments suivants :

- Trois menus principaux (Fichier, Edition, Aide)
- Le menu Fichier comporte les items : Nouveau, Ouvrir, Fermer.
- Le menu Edition comporte les items : Copier, Coller, Supprimer.
- Le menu ? comporte les items : Aide, A propos.

4. Créer les chaînes de caractères et les appliquer aux : Textview (propriété Hint) (nom, prénom, groupe), Button (Ajouter, Initialiser).
5. Créer un tableau de caractères « string-array »
6. `<string-array name="groupes">`
7. `<item >G1</item>`
8. `<item >G2</item>`
9. `<item >G3</item>`
10. `</string-array>`
11. Ajouter la propriété android : `android:scrollbarStyle="insideOverlay"`
12. Créer le style défini dans l'exemple dans le résumé et l'appliquer aux « liste des étudiants ».

Le Style, Les chaînes de caractères, et le menu

Chaîne de caractère :

Les **strings** (chaînes de caractères) définies par la syntaxe suivante :

```
<string name="name">le texte</string>
```

Les chaînes de caractères doivent être placées dans le fichier string.xml qui se trouve dans « res/layout/string.xml » pour permettre les utiliser dans l'application développée.

Exemple :

```
<resources>

...
<string name="aff">Afficher</string>
<string name="init">Initialiser</string>
<string name="matric">Matricule</string>
<string name="nomprenom">Nom et Prénom</string>

</resources>
```

Modifier les balises des boutons : `editText1`, `editText2`, `button1`, `button2`, en ajoutant les attributs suivants :

```
android :hint= "@string/matric" et android :hint= "@string/nomprenom"
android :text= "@string/aff" et android :text= "@string/init"
respectivement.
```

Le style

Pour définir un style personnalisé, il est nécessaire d'ouvrir le fichier styles.xml qui se trouve dans la racine « res/values/styles.xml »

Exemple : Soit le style suivant :

```
<style name="StylePersonnalise" parent="AppTheme">
<item name="android:textColor">#ffff0000</item>
<item name="android:textSize">15dp</item>
<item name="android:textStyle">bold|italic</item>
<item name="android:width">90dp</item>
<item name="android:height">20dp</item>
</style>
```

Pour appliquer ce style sur « Button1 » il faut ajoute l'attribut : **style= « @style/StylePersonnalise »** entre les balises du button1.

Le Menu

Pour définir un nouveau menu, il est nécessaire d'ouvrir le fichier main.xml qui se trouve dans la racine « res/menu/main.xml » et ajouter les attributs suivants :

```
<item android:id="@+id/item1" android:title="Fichier">
  <menu>
    <item android:id="@+id/item2" android:title="Nouveau"/>
    <item android:id="@+id/item3" android:title="Ouvrir"/>
    <item android:id="@+id/item4" android:title="Fermer"/>
  </menu>
</item>
<item android:id="@+id/item5" android:title="Edition">
  <menu>
    <item android:id="@+id/item6" android:title="Copier"/>
    <item android:id="@+id/item7" android:title="Coller"/>
    <item android:id="@+id/item8" android:title="Supprimer"/>
  </menu>
</item>
<item android:id="@+id/item9" android:title="Aide">
  <menu>
    <item android:id="@+id/item10" android:title="Aide"/>
    <item android:id="@+id/item11" android:title="A propos"/>
  </menu>
</item> </menu>
```

Les images

Pour charger les images dans un widget image, il est nécessaire que ces images soient dans les ressources images (dans les dossiers : src\drawable-ldpi, src\drawable-mdpi, src\drawable-hdpi, src\drawable-xhdpi, src\drawable-xxhdpi).

Ces dossiers sont de différentes résolutions 3, 4, 6, 8 scaling ratio, par exemple les images de dossier drawable-ldpi sont de résolution de 36X36 pixels, drawable-mdpi contient les images de 48x48, etc.

- L'attribut : android:background="@drawable/background", pour mettre l'image nommée « background » qui se trouve dans le dossier « drawable » comme un arrière-plan.

- L'attribut : android:src="@drawable/image1", pour charger l'image « image1 » dans le widget d'image, comme (ImageButton, ImageView,...), tel que l'image « Image1 » doit placer dans le dossier « drawable ».

Partie Programmation

L'évènement Selection d'item (ListView, Spinner, GridView)

```
sp.setOnItemClickListener(new OnItemSelectedListener() {  
    @Override  
    public void onItemClick(AdapterView<?> arg0, View arg1, int  
arg2, long arg3) {  
        // TODO Auto-generated method stub  
        value = arg0.getItemAtPosition(arg2);  
        switch (arg2) {  
            case 0:  
                position = 1; Toast.makeText(getApplicationContext(),  
value.toString(), 2000).show();  
                sv.setBackgroundColor(0xffffffff);  
                break;  
  
            case 1:  
                position = 2; Toast.makeText(getApplicationContext(),  
value.toString(), 2000).show();  
                sv.setBackgroundColor(0xffff0000);  
                break;  
  
            case 2:  
                position = 3;  
                Toast.makeText(getApplicationContext(), value.toString(),  
2000).show();  
                sv.setBackgroundColor(0xff0000ff);  
                break;  
        }  
    }  
}
```

L'évènement Selection d'item (Menu)

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.item1:  
            Toast.makeText(getApplicationContext(), item.getTitle(), 1000).show();  
            break;  
        case R.id.item2:  
            Toast.makeText(getApplicationContext(), item.getTitle(), 1000).show();  
            break;  
  
        default:  
            return false;  
    }  
    return super.onOptionsItemSelected(item);  
}
```