

TP : Les styles

INTRODUCTION

Qu'est-ce qu'un style ?

Un style est un ensemble de critères esthétiques qui vont spécifier le design d'une vue ou d'un élément. Ces critères sont des propriétés qui peuvent être : Couleur d'un texte, taille du texte, margin, padding etc... Un style se définit dans un fichier XML (généralement dans res/values/styles.xml).

Utilité des styles

Les styles sont utilisés dans le but d'éviter de se répéter. Par exemple, considérons que nous ayons deux boutons et que nous voulons définir leur couleur à « Blanc ». Sans les styles, il faudrait définir pour chaque bouton sa couleur, tandis qu'avec un style, il suffira de définir la couleur blanche dans le style et d'appliquer le style aux deux boutons.

OBJECTIF DU TP

Dans ce TP, nous allons apprendre à personnaliser des éléments d'un projet Android à l'aide de styles. L'objectif est de définir un style avec des propriétés (couleur et dimension) pour l'appliquer à un élément. Nous allons également voir comment définir un thème pour notre application.

Documents

Le code initial est le code de base lors de la création d'un projet, dans ce TP, vous allez travailler sur un TextView (le « hello world » de base).

Code final : <https://drive.google.com/file/d/0Bxul46mcWBOtZfViMno1NDdBU0U/view?usp=sharing>

DEROULEMENT DU TP

Code initial : Création d'un projet -> Minimum SDK API 14 : Android 4.0 -> Empty Activity

A cette étape, vous devez disposer d'un fichier « activity_main.xml » correspondant à la figure ci-dessous.

```
<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context="com.example.nath.tpstyle.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
</RelativeLayout>
```

Définition d'un style

Pour définir un style, vous allez ouvrir le fichier styles.xml (situé dans res/values). Vous pouvez effacer son contenu, je vais vous expliquer comment créer votre propre style.

Vous devez respecter la syntaxe suivante :

```
<style name="nom_du_style">
    <item name="propriete_1">valeur_de_la_propriete_1</item>
    <item name="propriete_2">valeur_de_la_propriete_2</item>
    ...
    <item name="propriete_n">valeur_de_la_propriete_n</item>
</style>
```

- ➔ L'attribut « name » correspond au nom de votre style, c'est ce nom qu'il faudra appeler pour appliquer le style à un élément.
- ➔ Pour ajouter une propriété à un style, il suffit d'ajouter une balise « item », l'attribut « name » correspond à un attribut destiné aux vues : par exemple on utilisera « android:textSize » pour changer la taille du texte.
- ➔ La valeur de la propriété peut être une ressource ou alors vous pouvez écrire directement la valeur (il n'est pas conseillé de le faire).

Appliquer un style à un élément

Maintenant que votre style est défini, il faut l'appliquer à notre élément « TextView ». Pour cela, il suffit d'utiliser l'attribut « style » dans votre TextView et de lui donner le nom du style précédemment créé.

Voici la syntaxe à utiliser : style="@style/nom_du_style "

Je vais maintenant vous demander de créer un style pour définir la couleur et la taille du texte de notre TextView ainsi que pour lui attribuer une marge intérieur en haut. Vous êtes libre de choisir vos valeurs.

Aide :

- *N'oubliez pas de définir votre ressource couleur dans res/values/colors.xml*
- *Pour la marge, vous pouvez utiliser les dimensions déjà définies dans le projet (res/values/dimens.xml) :*
- *Pour la taille du texte, vous pouvez définir une nouvelle ressource dans res/values/dimens.xml*

Si vous avez réussi, vous pouvez voir les changements dans la vue graphique de votre layout et passez à la suite.

Maintenant que nous avons vu comment appliquer un style à un élément, il faut que vous sachiez qu'il est également possible d'appliquer un style à toute notre application.

On appelle cela un thème mais il s'agit en réalité d'un style, nous allons voir comment cela fonctionne.

Les thèmes

Les thèmes sont des styles que vous pouvez appliquer à l'ensemble des balises de votre application.

Pour définir un thème, vous pouvez créer un fichier `res/values/themes.xml`, mais vous n'y êtes pas obligé, vous pouvez très bien utiliser un style défini dans le fichier `styles.xml`.

La syntaxe est ensuite exactement la même que pour les styles.

Un exemple :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme">
        <item name="android:textSize">50dp</item>
        <item name="android:textColor">@color/bleu</item>
    </style>

</resources>
```

Dans votre fichier `themes.xml`, vous pouvez définir un style

Au final, un thème est défini comme un style, la seule différence va être la manière de l'appliquer.

Pour appliquer ce thème à l'ensemble de l'application, il faut l'ajouter à la balise `<application />` de notre `AndroidManifest.xml`.

La syntaxe est la suivante : `android:theme="@style/nom_du_theme"`.

Ce qui est important à retenir est que le thème sera appliqué pour tous les éléments de votre application.

A vous maintenant de créer un thème dans lequel vous allez définir des propriétés de couleurs et de taille de texte différentes de celles que vous avez créé avant et appliquer ce thème à votre application.

Faites un test en créant un autre « `TextView` » dans votre layout sans lui définir de style. Vous pouvez alors remarquer qu'il possède les propriétés que vous venez de définir.

Ce qui est important de voir également, c'est que votre `TextView` « Hello World » garde les propriétés du premier style que vous avez défini. Un style définit sur un élément sera donc prioritaire sur votre thème.

Voici ce TP se termine ici, j'espère qu'il vous aura aidé à comprendre comment fonctionne les styles et les thèmes et comment personnaliser votre application.