

Série de TP 2: Application de démonstration**Convertisseur de Température****Enoncé :**

Dans cet exercice, nous allons apprendre à créer et utiliser des ressources Android et réviser la création d'une application interactive.

Cette application est disponible sur Google Play avec l'URL suivante : **Android Temperatureconverter**

1. **Créer un nouveau projet** : créer un nouveau projet Android avec les informations suivantes :

Propriété	Valeur
Application Name	Temperature Converter
Package name	com.toshiba.temperatureconverter
API (Minimum, Target, Compile with)	Latest
Template	Empty Activity
Activity	MainActivity
Layout	activity_main

2. **Créer les attributs** : Android vous permet de créer des ressources statiques pour définir des attributs, par exemple des chaînes de caractères ou bien des couleurs. Ces attributs peuvent être utilisés dans les fichiers XML ou par le code source Java.

- Sélectionnez le fichier res/values/string.xml pour ouvrir l'éditeur de ce fichier.
- Ajouter une définition de couleur à ce fichier. Pour cela, Appuyez sur le bouton « + »
- Entrez le nom **myColor** et **#F5F5F5** comme valeur.

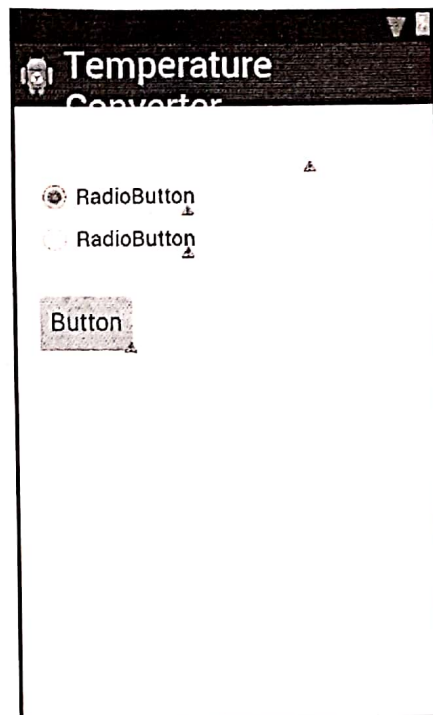
3. **Ajoutez d'autres attributs** : cette fois-ci des chaînes de caractères (String)

Nom	Valeur
celsius	to Celsius
fahrenheit	to Fahrenheit
cal	Calculate

- Basculez en mode « représentation XML » et validez que les valeurs sont correctes :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">Temperature Converter</string>
  <string name="action_settings">Settings</string>
  <string name="hello_world">Hello world!</string>
  <color name="myColor">#F5F5F5</color>
  <string name="celsius">to Celsius</string>
  <string name="fahrenheit">to Fahrenheit</string>
  <string name="calc">Calculate</string>
</resources>
```

4. **Ajouter des objets à votre mise en page** : Le résultat de votre application doit ressembler à ceci.



5. Éditer les propriétés des objets :

- Affectez la valeur `@string/Celsius` à la propriété `android:text` du premier bouton de la radio.
- Affectez la valeur `@string/Fahrenheit` à la propriété `android:text` du deuxième bouton radio
- Assurez-vous que la propriété « Checked » a la valeur « true » pour le premier RadioButton.
- Attribuez `@string/cal` à la propriété `texte` de votre bouton et affectez la valeur `onClick` à la propriété `onClick`.
- Définissez la propriété « `inputType` » à « `numberSigned` » et « `numberDecimal` » pour le champ `EditText`.

Votre page XML doit ressembler à ceci :

```
<?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"
    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.toshiba.temperatureconverter.MainActivity"
    android:background="@string/myColor"
>
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:inputType="numberSigned|numberDecimal"
    android:text="lolololo"
/>
```

```

<RadioGroup
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:layout_below="@+id/editText"
  android:layout_alignParentLeft="true"
  android:layout_alignParentBottom="true"

  >
  <RadioButton
    android:id="@+id/radioButton0"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:checked="true"
    android:text="@string/celsius"
  />

  <RadioButton
    android:id="@+id/radioButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:text="@string/fahrenheit"
  />

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/cal"
    android:id="@+id/button"
    android:onClick="onClick"/>
</RadioGroup>
</RelativeLayout>

```

6. Créer la classe utilitaire : convertir des degrés Celsius en Fahrenheit et vice versa.

```

package com.toshiba.temperatureconverter;
public class ConverterUtil {
  // converts to celsius
  public static float convertFahrenheitToCelsius(float fahrenheit) {
    return ((fahrenheit - 32) * 5 / 9);
  }
  // converts to fahrenheit
  public static float convertCelsiusToFahrenheit(float celsius) {
    return ((celsius * 9) / 5) + 32;
  }
}

```

7. Modifier le code de l'activité

```

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.Toast;

```

```
publicclass MainActivityextends Activity {
privateEditText text;

publicvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
text= (EditText) findViewById(R.id.editText);
}
// this method is called at button click because we assigned the name to the
// "OnClick property" of the button
publicvoid onClick(View view) {
switch(view.getId()) {
caseR.id.button:
RadioButtoncelsiusButton= (RadioButton) findViewById(R.id.radioButton0);
RadioButtone FahrenheitButton= (RadioButton) findViewById(R.id. radioButton1);
if(text.getText().length() == 0) {
Toast.makeText(this, "Please enter a valid number",
Toast.LENGTH_LONG).show();
return;
}
floatinputValue= Float.parseFloat(text.getText().toString());
if(celsiusButton.isChecked()) {
text.setText(String
.valueOf(ConverterUtil.convertFahrenheitToCelsius(inputValue)));
celsiusButton.setChecked(false);
fahrenheitButton.setChecked(true);
} else {
text.setText(String
.valueOf(ConverterUtil.convertCelsiusToFahrenheit(inputValue)));
fahrenheitButton.setChecked(false);
celsiusButton.setChecked(true);
}
break;
}
}
}
```