String.xml

- A string resource provides text strings for your application with optional text styling and formatting.
- A single string that can be referenced from the application or from other resource files (such as an XML layout).
- A string is a simple resource that is referenced using the value provided in the name attribute
- You can combine string resources with other simple resources in the one XML file, under one <resources> element.

XML file saved at res/values/strings.xml

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

<resources>
    <string name="hello">Hello!</string>
</resources>
```

This layout XML applies a string to a View:

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello" />
```

This application code retrieves a string:

```
String string = getString(R.string.hello);
```

XML file saved at res/values/strings.xml

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

<resources>
    <string-array name="planets_array">
        <item>Mercury</item>
        <item>Venus</item>
        <item>Earth</item>
        <item>Mars</item>
    </string-array>
</resources>
```

This application code retrieves a string array:

```
Resources res = getResources();
String[] planets = res.getStringArray(R.array.planets_array);
```

Advantages

- It centralizes the strings used by the application in a single location that is easily managed (by the developer or a non-developer).
- Strings can be defined as a resource once, and used throughout the code. Therefore, it will have consistent spelling, case and punctuation.
- Strings can be internationalized easily, allowing your application to support multiple languages with a single application package file (APK).
- Strings don’t clutter up your application code, leaving it clear and easy to maintain.