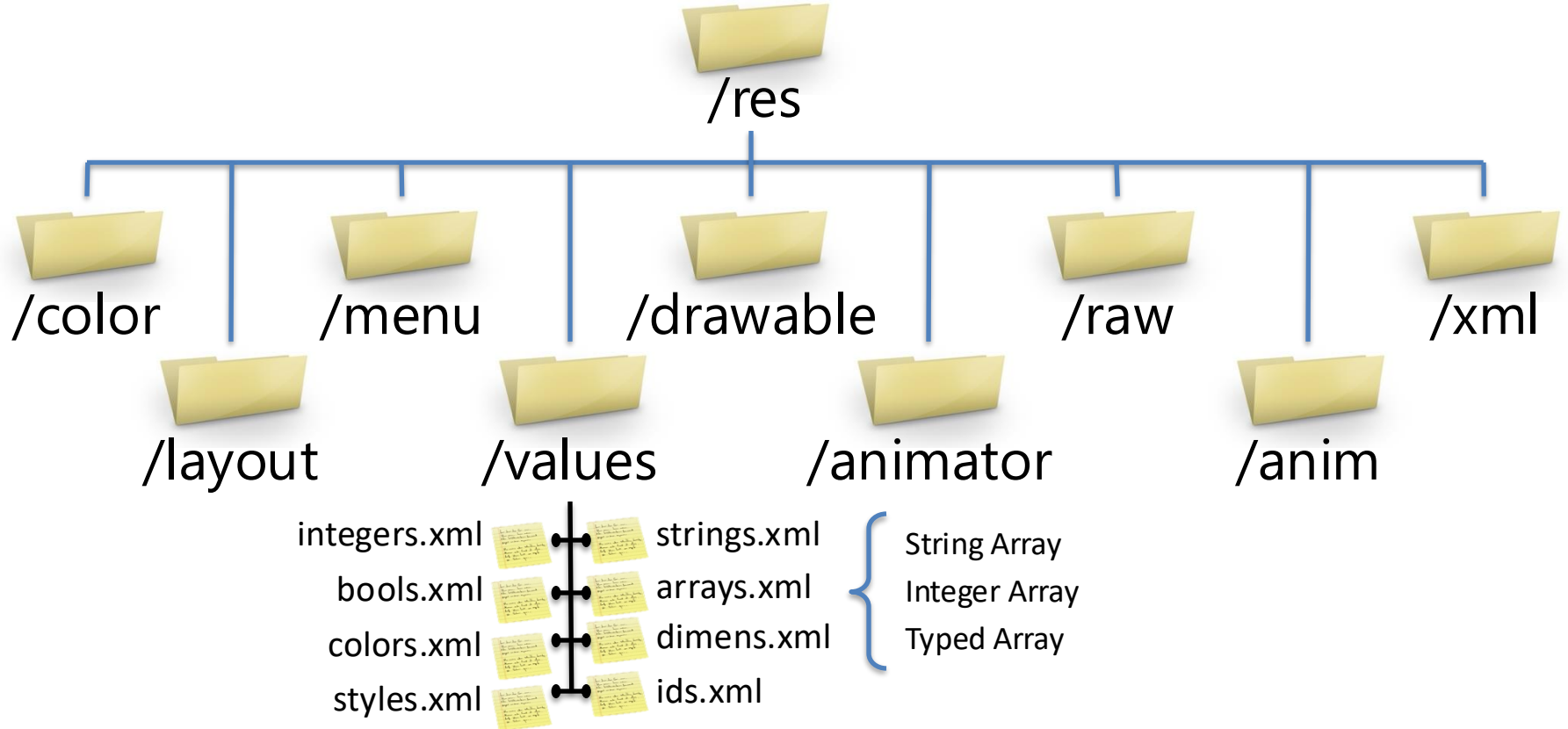


Resources



resources

Strings

Strings

/res/values/strings.xml

Definición:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    :::
    <string name="identificador">valor</string>
    :::
</resources>
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <string name="saludo">Buenos días</string>
    <string name="despedida">Adiós</string>
</resources>
```

Strings

/res/values/strings.xml

Definición:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    ...
    <string name="identificador"><![CDATA[Cadena con caracteres no permitidos en XML]]></string>
    ...
</resources>
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <string name="saludo">Buenos días</string>
    <string name="despedida"><![CDATA[Uno->Dos & Uno<-Dos]]></string>
</resources>
```

Strings

/res/values/strings.xml

Referencia desde otro recurso:

@string/identificador

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <string name="saludo">Buenos días</string>
  <string name="despedida">Adiós</string>
</resources>
```

```
<LinearLayout android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
  <TextView android:id="@+id/tv_saludo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/saludo" />
</LinearLayout>
```

Strings

/res/values/strings.xml

Referencia desde código (Views):

`R.string.identificador`

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <string name="saludo">Buenos días</string>
  <string name="despedida">Adiós</string>
</resources>
```

```
override fun onCreate(savedInstanceState : Bundle?)
{
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    val tv : TextView =
        findViewById<TextView>(R.id.tv_saludo)
    tv.setText( R.string.saludo )
}
```

Strings

/res/values/strings.xml

Cómo obtener los valores desde código (Views):

```
String Context.getString(R.string.identificador)
```

String

Ejemplo:

```
Context.getResources().getString(R.string.identificador)
```

```
override fun onCreate(savedInstanceState : Bundle?)
{
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    val tv : TextView = findViewById<TextView>(R.id.tv_saludo)
    tv.text = getString(R.string.saludo)
}
```

Strings

/res/values/strings.xml

Cómo obtener los valores desde código (Compose):

```
String stringResource(R.string.identificador)
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <string name="saludo">Buenos días</string>
    <string name="despedida">Adiós</string>
</resources>
```

```
@Composable
fun SaludoText()
{
    val saludoText = stringResource(R.string.saludo)
    Text(text = saludoText)
}
```


Strings

/res/values/strings.xml

Con formato:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  :::
  <string name="identificador">literal1 %1$s literal2
  %2$d</string>
  :::
</resources>
```

%index\$s	String	%index\$c	char
%index\$d	int	%index\$b	boolean
%index%f	float	%index\$h	hexadecimal
%index\$g	double		\n new line
		...	

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <string name="saludo_personal">Buenos días %1$s</string>
  <string name="temperatura">Hoy hace %1$g grados
  centígrados</string>
</resources>
```

Strings

/res/values/strings.xml

Cómo obtener los valores desde código de strings con formato (Views):

```
String Context.getString(R.string.identificador, Object... parameters)
```

```
String Context.getResources().getString(R.string.identificador, Object... parameters)
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <string name="saludo_personal">Buenos días %1$s</string>
  <string name="temperatura">Hoy hace %1$g grados
centígrados</string>
</resources>
```

```
{
    val tvSaludo : TextView = findViewById<TextView>(R.id.tv_saludo)
    val tvTemperatura : TextView =
        findViewById<TextView>(R.id.tv_temperatura)

    tvSaludo.text = getString(R.string.saludo_personal, "Carlos")
    tvTemperatura.TEXT = getString(R.string.temperatura, 33.14)
```

Strings

/res/values/strings.xml

Cómo obtener los valores desde código de strings con formato (Compose):

```
String stringResource(R.string.identificador, Object... parameters)
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <string name="saludo_personal">Buenos días %1$s</string>
  <string name="temperatura">Hoy hace %1$g grados
```

centígrados<

</resources>

@Composable

```
fun SaludoText()
```

```
{
```

```
  val saludoText = stringResource(R.string.saludo_personal, "Carlos")
```

```
  val temperaturaText = stringResource(R.string.temperatura, 36.5)
```

```
  Text(text = saludoText)
```

```
}
```

Strings

/res/values/strings.xml

Con estilo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    :::
    <string name="identificador"><b>negrita</b> <i>cursiva</i>
<u>subrayado</u></string>
    :::
</resources>
```

	negrita
<i>	<i>cursiva</i>
<u>	<u>subrayado</u>
<sup>	superíndice
<sub>	subíndice
<big>	25% más grande
<small>	20% más pequeño
	viñetas
 	salto línea
<span style="	color
	background_color
	text-decoration">

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <string name="frio">Hoy hace <b>frío</b></string>
    <string name="cita">Me dijo: <i>No</i></string>
</resources>
```

Strings

/res/values/strings.xml

¿Cómo obtener los valores de cadenas con estilo:

```
CharSequence Context.getText(R.string.identificador)
```

Ejemplo:

```
<resources>
  <string name="frio">Hoy hace <b>frío</b></string>
</resources>
```

```
{
    TextView tvSaludo = (TextView) findViewById(R.id.tv_saludo);
    TextView tvTemperatura = (TextView) findViewById(R.id.tv_temperatura);

    CharSequence frio = getText(R.string.frio)
    tvTemperatura.setText( frio );
}
```

Strings

/res/values/strings.xml

Cómo obtener los valores de cadenas con estilo (Compose):

Ejemplo:

```
import androidx.compose.ui.text.AnnotatedString
import androidx.compose.ui.text.SpanStyle
import androidx.compose.ui.text.buildAnnotatedString
import androidx.compose.ui.text.font.FontStyle
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextDecoration
```

```
fun getStyledAnnotatedString(): AnnotatedString {
    return buildAnnotatedString {
        append("Esto es ")
        withStyle(style = SpanStyle(fontStyle = FontStyle.Italic)) { append("cursiva") }
        append(", ")
        withStyle(style = SpanStyle(fontWeight = FontWeight.Bold)) { append("negrita") }
        append(", ")
        withStyle(style = SpanStyle(textDecoration = TextDecoration.LineThrough)) { append("tachado") }
        append(", ")
        withStyle(style = SpanStyle(textDecoration = TextDecoration.Underline)) { append("subrayado") }
        append(", y ")
        withStyle(style = SpanStyle(fontWeight = FontWeight.Bold,
            fontStyle = FontStyle.Italic )) { append("boldItalic") }
        append(".")
    }
}
```

Esto es *cursiva*, **negrita**, tachado, subrayado, y ***boldItalic***.

Strings

/res/values/strings.xml

Cómo obtener los valores de cadenas con formato y estilo (Views):

String Context.getString(R.string.identificador, Object... parameters)

Spanned Html.fromHtml(String text)

Ejemplo:

```
<resources>
  <string name="cita">Dijo literalmente \"
  <string name="cita_styled">Dijo literalmente \"
  <string name="cita_html">Dijo literalmente \"
</resources>
```

< <
> >

```
{
    val tvCita : TextView = findViewById<TextView>(R.id.tv_cita)

    val cita : String = getString(R.string.cita, "Hoy será un gran día")
    val citaStyled : Spanned = Html.fromHtml(cita,
    Html.FROM_HTML_MODE_LEGACY)
    tvCita.text = citaStyled
}
```

resources

String array

String array

/res/values/**arrays.xml**

Definición:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  ...
  <string-array name="identificador">
    <item>valor1</item>
    ...
    <item>valorn</item>
  </string-array>
  ...
</resources>
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <string-array name="fin_de_semana">
    <item>sábado</string>
    <item>domingo</string>
  </string-array>
</resources>
```

String array

/res/values/**arrays.xml**

Referencia desde otro recurso:

R.array.**identificador**

Ejemplo:

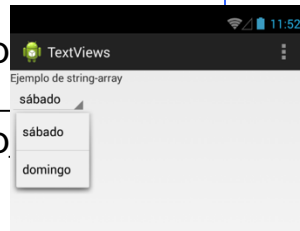
```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <string-array name="fin_de_semana">
        <item>sábado</item>
        <item>domingo</item>
    </string-array>
</resources>
```

```
<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView android:id="@+id/tv"
        android:text="@string/ejemp"
        android:layout_width="wrap"
        android:layout_height="wrap"
    />

    <Spinner android:id="@+id/sp"
        android:entries="@array/fin_de_semana"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

</LinearLayout>
```



String array

/res/values/**arrays.xml**

Cómo obtener los valores desde código (Views):

```
String[] Context.resources.getStringArray(R.array.identificador)
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <string-array name="laborables">
    <item>lunes</string>
    <item>martes</string>
    <item>miércoles</string>
    <item>jueves</string>
    <item>viernes</string>
  </string-array>
</resources>
```

```
{
  val dias : Array<String> =
    resources.getStringArray(R.array.laborables);
}
```

String array

/res/values/**arrays.xml**

Cómo obtener los valores desde código (Compose):

```
LocalContext.current.resources.getStringArray(R.array.identificador)
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <string-array name="laborables">
    <item>lunes</string>
    <item>martes</string>
    <item>miércoles</string>
    <item>jueves</string>
    <item>viernes</string>
  </string-array>
</resources>
```

```
{
  val dias : Array<String> = LocalContext.current.resources.getStringArray(R.array.laborables)
}
```

resources

Quantity string

Quantity string /res/values/plurals.xml

Definición:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  ...
  <plurals name="identificador">
    <item quantity="zero" | "one" | "two" | "few" | "many" | "other">valor1</item>
    ...
    <item quantity="zero" | "one" | "two" | "few" | "many" | "other">valorn</item>
  </plurals>
  ...
</resources>
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <plurals name="mensajes">
    <item quantity="one">Tienes un mensaje nuevo</item>
    <item quantity="other">Tienes %d mensajes nuevos</item>
  </plurals>
</resources>
```

Quantity string /res/values/plurals.xml

Cómo obtener los valores desde código (Views):

```
String Context.getResources().getQuantityString(R.plurals.identificador)
```

```
String Context.getResources().getQuantityString(R.plurals.identificador, int quantity)
```

```
String Context.getResources().getQuantityString(R.plurals.identificador, int quantity, Object... params)
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <plurals name="new_messages">
    <item quantity="one">tienes un mensaje nuevo</string>
    <item quantity="other">tienes %d mensajes nuevos</string>
  </plurals>
</resources>
```

```
resources.getQuantityString(R.plurals.new_messages, 0, 0) → "tienes 0 mensajes nuevos"
resources.getQuantityString(R.plurals.new_messages, 1, 1) → "tienes un mensaje nuevo"
resources.getQuantityString(R.plurals.new_messages, 8, 8) → "tienes 8 mensajes nuevos"
```

Quantity string /res/values/plurals.xml

Cómo obtener los valores desde código (Compose):

```
String pluralStringResource(R.plurals.identificador)
```

```
String pluralStringResource(R.plurals.identificador, int quantity)
```

```
String pluralStringResource(R.plurals.identificador, int quantity, Object... params)
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <plurals name="new_messages">
    <item quantity="one">tienes un mensaje nuevo</string>
    <item quantity="other">tienes %d mensajes nuevos</string>
  </plurals>
</resources>
```

```
pluralStringResource(R.plurals.new_messages, 0, 0) → "tienes 0 mensajes nuevos"
pluralStringResource(R.plurals.new_messages, 1, 1) → "tienes un mensaje nuevo"
pluralStringResource(R.plurals.new_messages, 8, 8) → "tienes 8 mensajes nuevos"
```