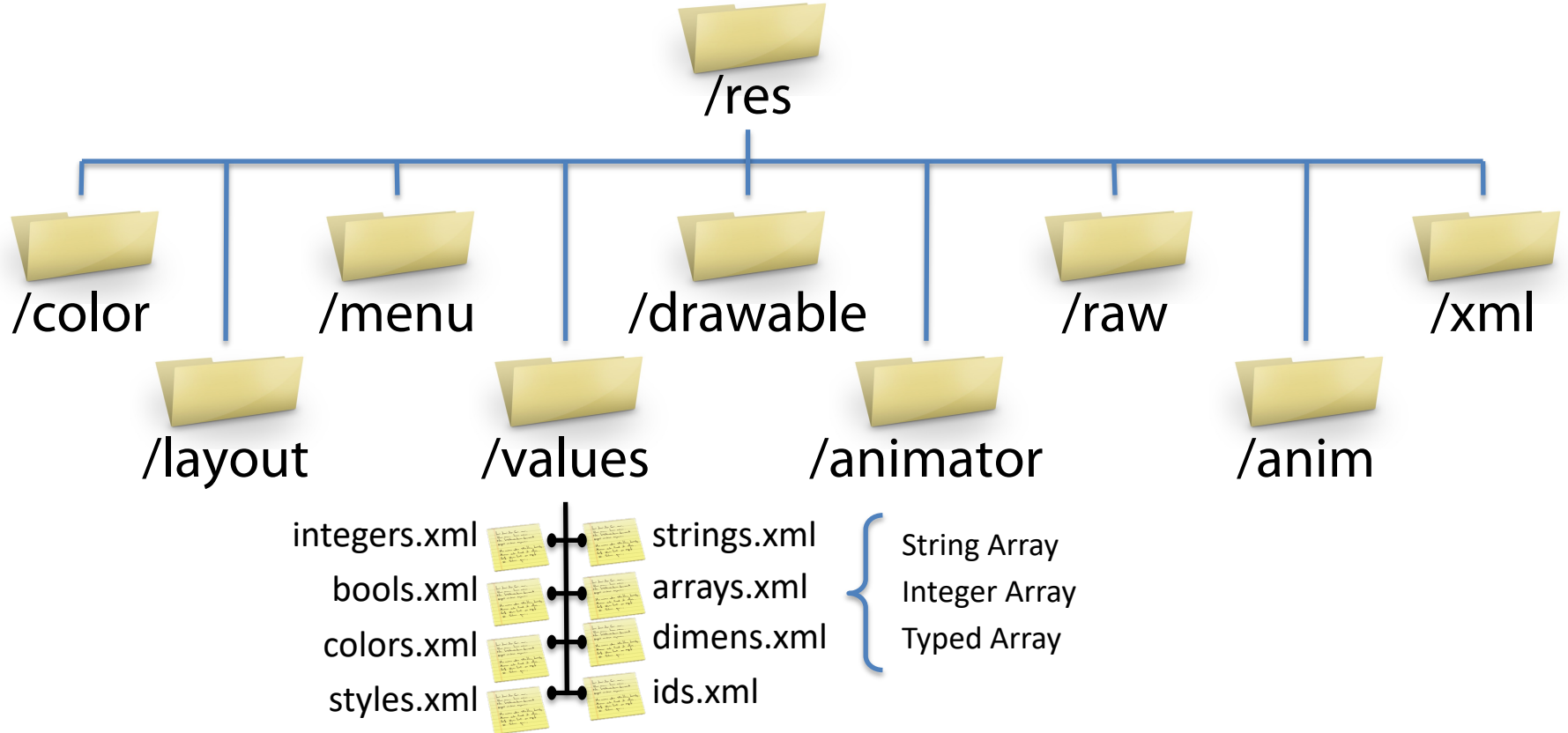


Resources



resources

Colors

Colors

/res/values/colors.xml

Definición:

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  ...
  <color name="identificador1">#RGB</color>
  <color name="identificador2">#ARGB</color>
  <color name="identificador3">#RRGGBB</color>
  <color name="identificador4">#AARRGGBB</color>
  ...
</resources>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <color name="resaltado">#FF0000</color>
  <color name="fondo">#88FF0000</color>
</resources>
```

Referencia desde otro recurso:
`@color/identificador`

Referencia desde código:
`R.color.identificador`

Obtención del valor desde código:
`int Context.getResources().getColor(R.color.identificador)`

resources

Color state list

Color state list

/res/color/**identifier.xml**

Definición:

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<selector>
  <item android:color="@color/identificador"
    [android:state_focused="true | false"]
    [android:state_selected="true | false"]
    [android:state_checkable="true | false"]
    [android:state_checked="true | false"]
    [android:state_enabled="true | false"]
    [android:state_window_focused="true | false"]
    [android:state_pressed="true | false"] />
  ...
</selector>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<selector>
  <item android:color="@color/red"
    android:state_pressed="true"/>
  <item android:color="@color/green"/>
</selector>
```

Referencia desde otro recurso:
@color/identifier

Referencia desde código:
R.color.identifier

Obtención del valor desde código:

`Context#resources.getColorStateList(R.color.identifier)` : `ColorStateList`

Color state list

/res/color/**identifier.xml**

Significado de state_xxx :

android:state_focused: Cuando la vista es la que tiene el foco (con la que interactúa el usuario)

android:state_selected: Indicaría si elemento está seleccionado (por ejemplo al hacer scroll en una lista)

android:state_checkable: Sólo es válido cuando una vista puede transitar entre estados checked y unchecked

android:state_checked: Si es true será válido si la vista es checkable y su estado es checked.

android:state_enabled: La vista es capaz de recibir clicks

android:state_window_focused: Cuando la vista está en primer plano (foreground)

android:state_pressed: Si es true determina por ejemplo que un botón ha sido pulsado