

ToggleButton



Definición en Layout

```
<ToggleButton android:id="@+id/tb"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:checked="true"  
    android:textOff="@android:string/capital_off"  
    android:textOn="@android:string/capital_on" />
```

checked indica el estado checked del botón

· true · false

textOff texto que se mostrará cuando checked==false

textOn texto que se mostrará cuando checked == true

Código

```
import android.widget.ToggleButton;  
...  
val tb : ToggleButton = findViewById<ToggleButton>(R.id.tb)
```

Métodos

1 2

ToggleButton



setTextOn(textOn : CharSequence)

setTextOn(stringOnID : Int)

Establece el texto que se mostrará en el botón cuando éste se encuentre seleccionado. El texto se puede facilitar como una cadena de texto o a través del identificador de una definida como recurso (**R.string.xxx**)

setTextOff(textOff : CharSequence)

setTextOff(stringOffID: Int)

Establece el texto que se mostrará en el botón cuando éste no se encuentre seleccionado. El texto se puede facilitar como una cadena de texto o a través del identificador de una definida como recurso (**R.string.xxx**)

setChecked(checked : Boolean)

isChecked() : Boolean

Define y recupera, respectivamente, el estado del botón

toggle()

Invierte el estado del botón

ToggleButton



`setOnCheckedChangeListener(listener : CompoundButton.OnCheckedChangeListener)`

Establece el listener que se ejecutará cuando el usuario modifique el estado del ToggleButton

```
val tb : ToggleButton = findViewById<ToggleButton>(R.id.tb)
tb.setOnCheckedChangeListener { button, isChecked -> :: } }
```

// Recibe como parámetros el botón pulsado y su estado

Switch



Definición en Layout

```
<Switch android:id="@+id/sw"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:checked="true"  
    android:textOff="@android:string/capital_off"  
    android:textOn="@android:string/capital_on"  
    android:showText="false"  
    android:thumb=""  
    android:track="" />
```

Código

```
import android.widget.Switch;  
...  
val sw : Switch = findViewById<Switch>(R.id.sw)
```

checked indica el estado checked del botón

· true · false

textOff texto que se mostrará cuando checked==false

textOn texto que se mostrará cuando checked == true

showText indica si se muestran o no los textos textOff y textOn

· true · false

thumb imagen del botón (opcional)

track imagen del scroll (opcional)

Switch



setTextOn(textOn : CharSequence)

setTextOn(stringOnID : Int)

Establece el texto que se mostrará en el switch cuando éste se encuentre seleccionado. El texto se puede facilitar como una cadena de texto o a través del identificador de una definida como recurso (**R.string.xxx**)

setTextOff(textOff : CharSequence)

setTextOff(stringOffID: Int)

Establece el texto que se mostrará en el switch cuando éste no se encuentre seleccionado. El texto se puede facilitar como una cadena de texto o a través del identificador de una definida como recurso (**R.string.xxx**)

setChecked(checked : Boolean)

isChecked() : Boolean

Define y recupera, respectivamente, el estado del switch

toggle()

Invierte el estado del switch

Switch



showText(shwText : Boolean)

Indica si el switch muestra o no los textos textOn y TextOff

setOnCheckedChangeListener(listener : CompoundButton.OnCheckedChangeListener)

Establece el listener que se ejecutará cuando el usuario modifique el estado del Switch

```
val sw : Switch = findViewById<Switch>(R.id.sw)  
sw.setOnCheckedChangeListener { button, isChecked -> :: } }
```

// Recibe como parámetros el botón pulsado y su estado

CheckBox

- ☐ primavera
- ☐ verano
- ☐ otoño
- ☐ invierno

Definición en Layout

```
<CheckBox android:id="@+id/cb"  
    android:text="@string/spring_text"  
    android:checked="false"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

text representa el texto del botón. Puede ser un literal o la referencia a un recurso string (preferible)

checked indica el estado checked del botón
· true · false

Código

```
import android.widget.CheckBox  
:::  
val cb : CheckBox = findViewById<CheckBox>(R.id.cb)
```

Métodos

CheckBox

- ☐ primavera
- ☐ verano
- ☐ otoño
- ☐ invierno

setChecked(*checked* : Boolean)

isChecked() : Boolean

Definen y recuperan, respectivamente, el estado del checkbox

toggle()

Invierte el estado del checkbox

setOnCheckedChangeListener(*listener* : CompoundButton.OnCheckedChangeListener)

Establece el listener que se ejecutará cuando el usuario modifique el estado del CheckBox

```
val cb : CheckBox = findViewById<CheckBox>(R.id.cb)
cb.setOnCheckedChangeListener { button, isChecked -> :: }

```

// Recibe como parámetros el botón pulsado y su estado