

# TextView

*Muestra texto en pantalla sin posibilidad de editarlo*

## Definición en Layout

```
<TextView android:id="@+id/tv_1"  
    android:text="@string/texto_1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

**text** valor inicial del texto. Puede ser un literal o la referencia a un recurso string (preferible)

# TextView

## Definición en Layout — atributos visuales

```
<TextView android:id="@+id/tv_1"  
    android:text="@string/texto_1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:fontFamily="sans-serif-light"  
    android:textStyle="italic"  
    android:textSize="20sp"  
    android:allCaps="false"  
    android:textColor="@color/rojo"  
    android:background="@drawable/circulo"/>
```

**fontFamily** tipo de letra

- sans-serif
- sans-serif-light
- sans-serif-thin
- sans-serif-condensed
- sans-serif-medium

**textStyle** estilo del texto

- normal
- bold
- italic

**textSize** tamaño de la letra (*dimension*)

**allCaps** el texto siempre en mayúsculas

- true
- false

**textColor** color del texto (*color*)

**background** fondo del texto (*drawable* o *color*)

# TextView

## Definición en Layout — atributos visuales

```
<TextView android:id="@+id/tv_a"  
    android:text="Mínimo 2 líneas"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:minLines="2"/>
```

```
<TextView android:id="@+id/tv_b"  
    android:text="Exacto 1 línea"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:lines="1"/>
```

**minLines** debe tener, al menos, X líneas (*integer*)

**lines** debe tener, exactamente, X líneas (*integer*)

**maxLines** debe tener, como máximo, X líneas (*integer*)

**lineSpacingMultiplier** interlineado (*float*)

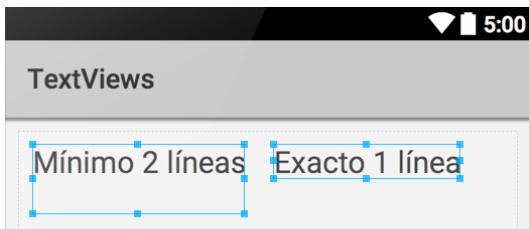
**scrollbars** *vertical* | *horizontal* | *none*

Java →

```
textView.setMovementMethod(ScrollingMovementMethod.getInstance());
```

Kotlin →

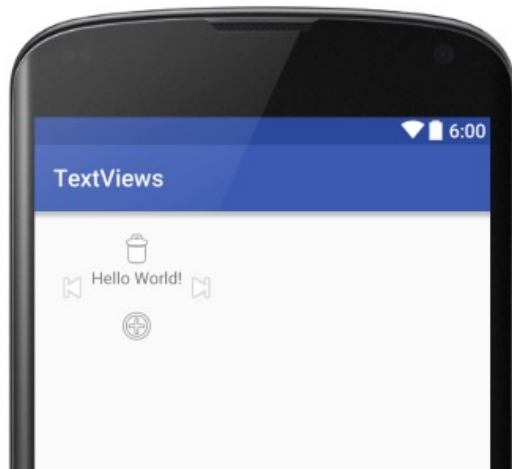
```
textView.movementMethod = ScrollingMovementMethod.getInstance()
```



# TextView

## Definición en Layout — atributos visuales

```
<TextView android:id="@+id/tv_a"  
    android:text="Hello World!"  
    android:drawableLeft="@android:drawable/ic_media_previous"  
    android:drawableTop="@android:drawable/ic_menu_delete"  
    android:drawableRight="@android:drawable/ic_media_next"  
    android:drawableBottom="@android:drawable/ic_menu_add"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```



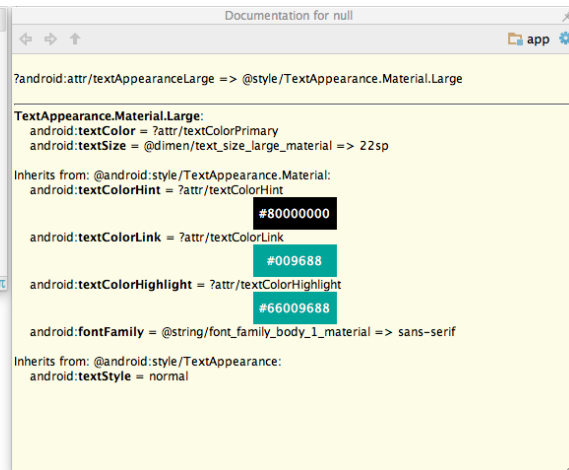
# TextView

## Definición en Layout — atributos visuales

```
<TextView android:id="@+id/tv_1"  
    android:text="@string/texto_1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance="?android:attr/textAppearanceLarge"/>
```

```
?android:textAppearanceLarge  
?android:text  
?android:textAppearance  
?android:textAppearanceSmall  
?android:absListViewStyle  
?android:accessibilityEventTypes  
?android:accessibilityFeedbackType  
?android:accessibilityFlags  
?android:accessibilityLiveRegion  
?android:accountPreferences  
?android:accountType
```

Dot, semicolon and some other keys will also close this lookup and be inserted into editor



# TextView

## Definición en Layout — links

```
<TextView android:id="@+id/tv_1"  
    android:text="http://www.upsa.es/mimo"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:linksClickable="true"  
    android:autoLink="web|email"  
    android:textColorLink="@color/verde"/>
```

**linksClickable** Si el texto de TextView contiene un enlace, se permite hacer click sobre él.

· true                      · false

**autoLink** representa un tipo de link (o varios separados por | )

· none    ninguno  
· web     <http://www.upsa.es/mimo>  
· email   <mailto:mimo@upsa.es>  
· phone   <tel:923277100>  
· map     [geo: 40.9624209,-5.6666993](geo:40.9624209,-5.6666993)  
· all      todos

**textColorLink** color del texto con el que resaltar el link (color)

## Alternativa:

```
String link = "<a href=\"geo:40.9624209,-5.6666993\">MIMO</a>";  
tv.setText(HtmlCompat.fromHtml(link,  
    HtmlCompat.FROM_HTML_MODE_LEGACY));  
tv.setMovementMethod(LinkMovementMethod.getInstance());
```

```
val link : String = "<a href=\"geo:40.9624209,-5.6666993\">MIMO</a>";  
tv.text = HtmlCompat.fromHtml(link,  
    HtmlCompat.FROM_HTML_MODE_LEGACY)  
tv.movementMethod = LinkMovementMethod.getInstance()
```

# TextView

## Métodos

`void setText(CharSequence text)`

`void setText(int stringID)`

Establece el texto que mostrará la vista. El texto puede indicarse a través de una cadena de caracteres o mediante un entero que represente la identidad de un recurso de tipo String (`R.String.xxx`)

`CharSequence getText()`

Permite obtener la cadena de caracteres se está mostrando la vista

`void setLinksClickable (boolean whether)`

`void setAutoLinkMask(int)`

Establece si el texto de la vista que cumple con un patrón puede ser seleccionado

Establece el tipo de patrón

`Linkify.ALL`

`Linkify.EMAIL_ADDRESSES`

`Linkify.MAP_ADDRESSES`

`Linkify.PHONE_NUMBERS`

`Linkify.WEB_URLS`