

# 1.3 Text and scrolling views

# Contents

- `TextView`
- `ScrollView`

# TextView

# TextView for text

- TextView is View subclass for single and multi-line text
- EditText is TextView subclass with editable text
- Controlled with layout attributes
- Set text:
  - Statically from string resource in XML
  - Dynamically from Java code and any source

# Formatting text in string resource

- Use `<b>` and `<i>` HTML tags for bold and italics
- All other HTML tags are ignored
- String resources: one unbroken line = one paragraph
- `\n` starts a new a line or paragraph
- Escape apostrophes and quotes with backslash (`\"`, `\'`)
- Escape any non-ASCII characters with backslash (`\`)

# Creating TextView in XML

```
<TextView android:id="@+id/textview"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="@string/my_story"/>
```

# Common TextView attributes

android:text—text to display

android:textColor—color of text

android:textAppearance—predefined style or theme

android:textSize—text size in sp

android:textStyle—normal, bold, italic, or bold|italic

android:typeface—normal, sans, serif, or monospace

android:lineSpacingExtra—extra space between lines in sp

# Formatting active web links

```
<string name="article_text">...www.rockument.com...</string>
```

```
<TextView  
    android:id="@+id/article"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:autoLink="web"  
    android:text="@string/article_text"/>
```

[autoLink](#) values: "web", "email", "phone", "map", "all"

Don't use HTML  
for a web link in  
free-form text



# Creating TextView in Java code

```
TextView myTextview = new TextView(this);  
myTextView.setWidth(LayoutParams.MATCH_PARENT);  
myTextView.setHeight(LayoutParams.WRAP_CONTENT);  
myTextView.setMinLines(3);  
myTextView.setText(R.string.my_story);  
myTextView.append(userComment);
```

# ScrollView

# What about large amounts of text?

- News stories, articles, etc...
- To scroll a TextView, embed it in a [ScrollView](#)
- Only *one* View element (usually TextView) allowed in a ScrollView
- To scroll multiple elements, use one ViewGroup (such as LinearLayout) within the ScrollView

# ScrollView for scrolling content

- [ScrollView](#) is a subclass of [FrameLayout](#)
- Holds all content in memory
- Not good for long texts, complex layouts
- Do not nest multiple scrolling views
- Use [HorizontalScrollView](#) for horizontal scrolling
- Use a [RecyclerView](#) for lists

# ScrollView layout with one TextView

```
<ScrollView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_below="@id/article_subhead
```

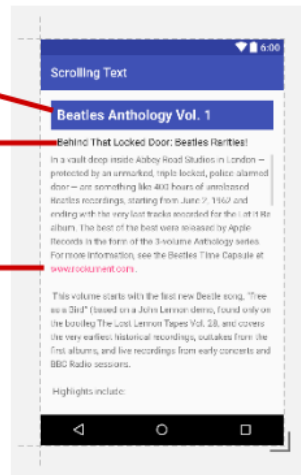
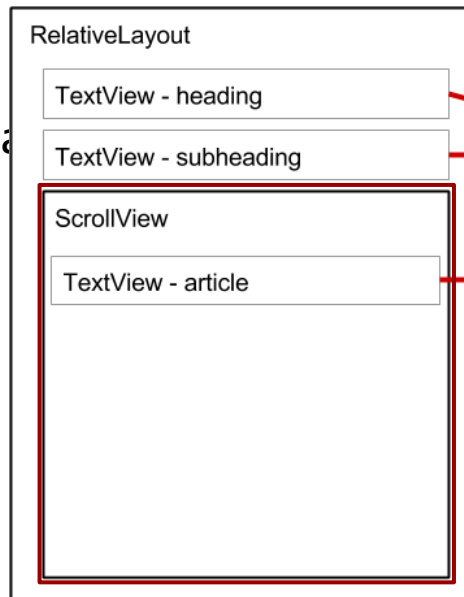
```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

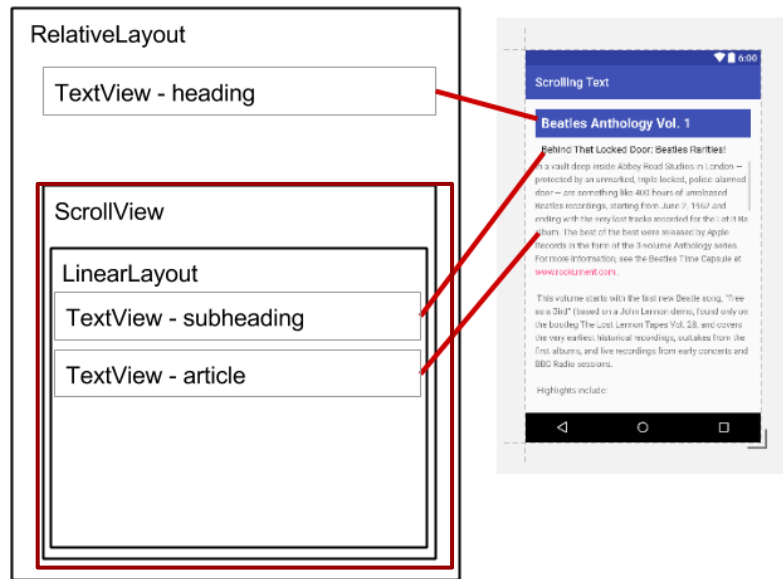
```
.../>
```

```
</ScrollView>
```



# ScrollView layout with a view group

```
<ScrollView ...  
  <LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="vertical">  
  
    <TextView  
      android:id="@+id/article_subheading"  
      .../>  
  
    <TextView  
      android:id="@+id/article" ... />  
  </LinearLayout>  
</ScrollView>
```



# ScrollView with image and button

```
<ScrollView...>
```

```
  <LinearLayout...>
```

```
    <ImageView.../>
```

```
    <Button.../>
```

```
    <TextView.../>
```

```
  </LinearLayout>
```

```
</ScrollView>
```

← One child of ScrollView  
which can be a layout

← Children of the layout

# Learn more

## Developer Documentation:

- [TextView](#)
- [ScrollView](#) and [HorizontalScrollView](#)
- [String Resources](#)

## Other:

- Android Developers Blog: [Linkify your Text!](#)
- Codepath: [Working with a TextView](#)



# What's Next?

- Concept Chapter: [1.3 Text and scrolling views](#)
- Practical: [1.3 Text and scrolling views](#)