

Views

Widgets in Android



- eXtensible Markup Language
- A programming language
- Used in Android to set up the screen
- Tags
 - Have a start: `<TextView`
 - Many attributes
 - An end: `/>`

Android uses widgets, called views, to set up the screen.



VIEWS



TextView

Morning Pizza

I'll be in your neighborhood doing errands this weekend if you want to meet up for brunch. Lemme know.

Show notifications

ImageView



Button

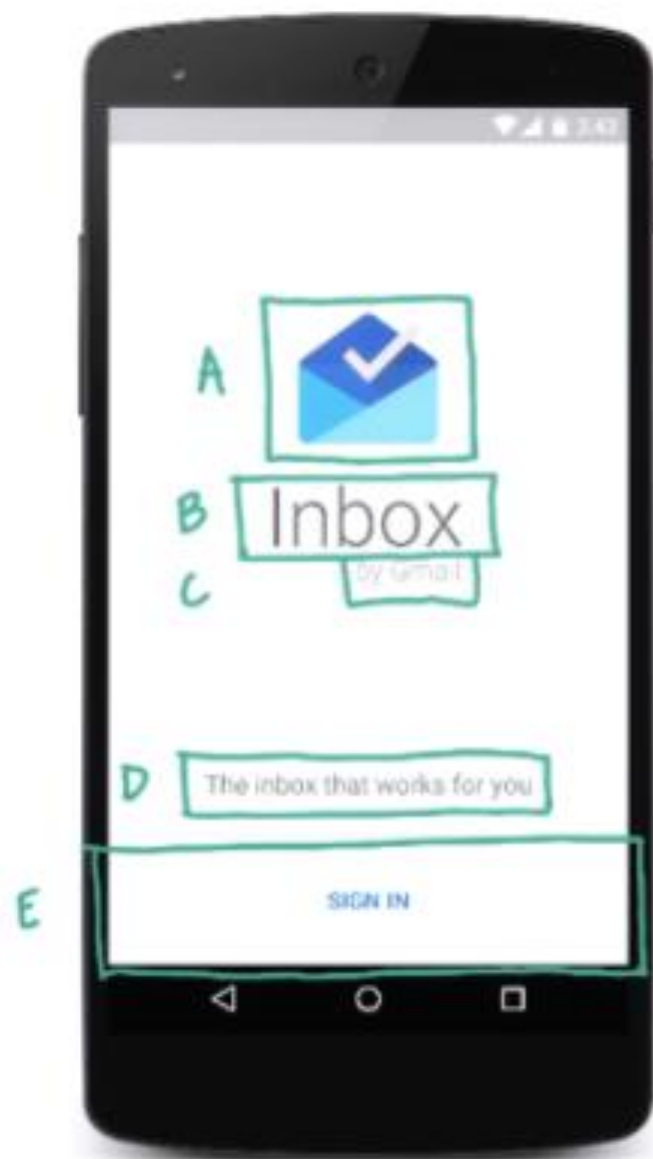


BUTTON

BUTTON

VIEWS

Identify the type of View



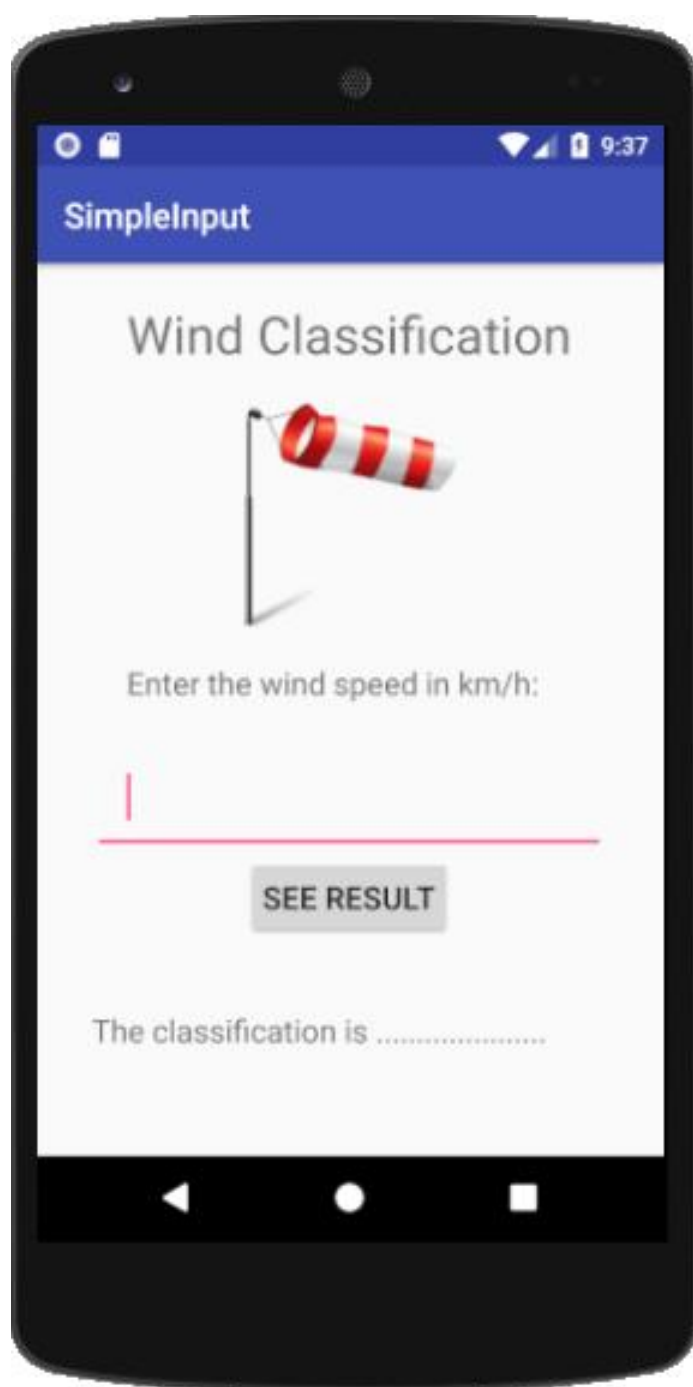
- | | | | |
|---|--------------------------------|---------------------------------|------------------------------|
| A | ☐ TextView | ☐ ImageView | ☐ Button |
| B | ☐ TextView | ☐ ImageView | ☐ Button |
| C | ☐ TextView | ☐ ImageView | ☐ Button |
| D | ☐ TextView | ☐ ImageView | ☐ Button |
| E | ☐ TextView | ☐ ImageView | ☐ Button |



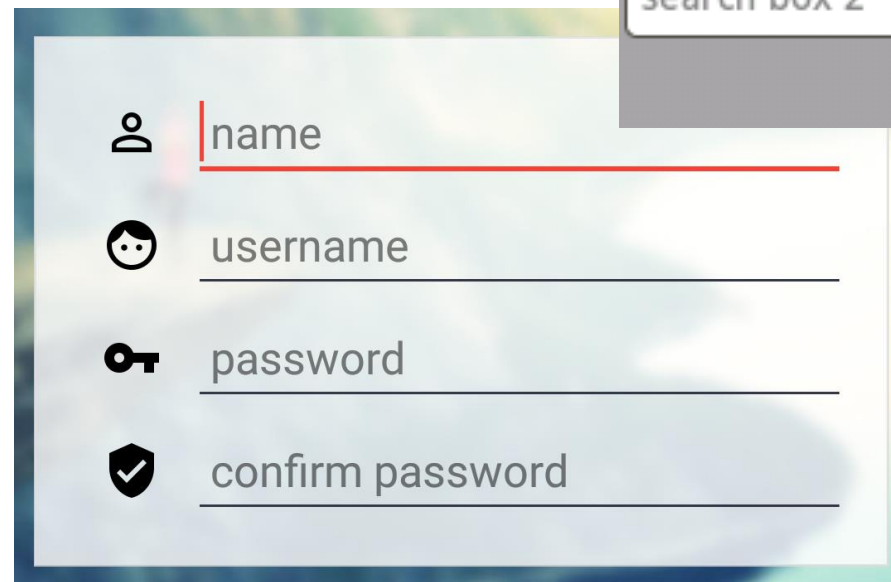
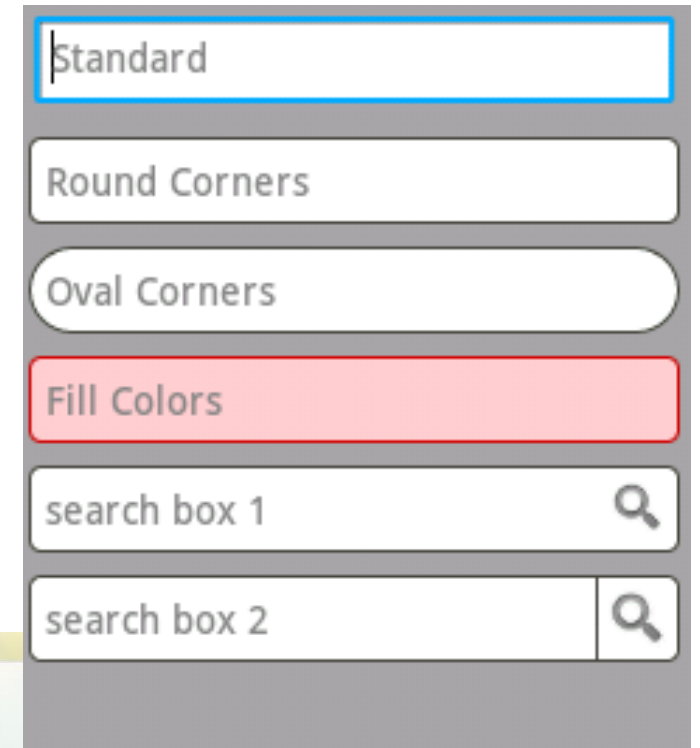
VIEWS

Identify the type of View

- | | | | |
|---|--------------------------------|---------------------------------|------------------------------|
| A | ☐ TextView | ☐ ImageView | ☐ Button |
| B | ☐ TextView | ☐ ImageView | ☐ Button |
| C | ☐ TextView | ☐ ImageView | ☐ Button |
| D | ☐ TextView | ☐ ImageView | ☐ Button |
| E | ☐ TextView | ☐ ImageView | ☐ Button |



There is also
a view called
an EditText.





You place your Views into a Layout and ta-dah!

You have an Android Screen.

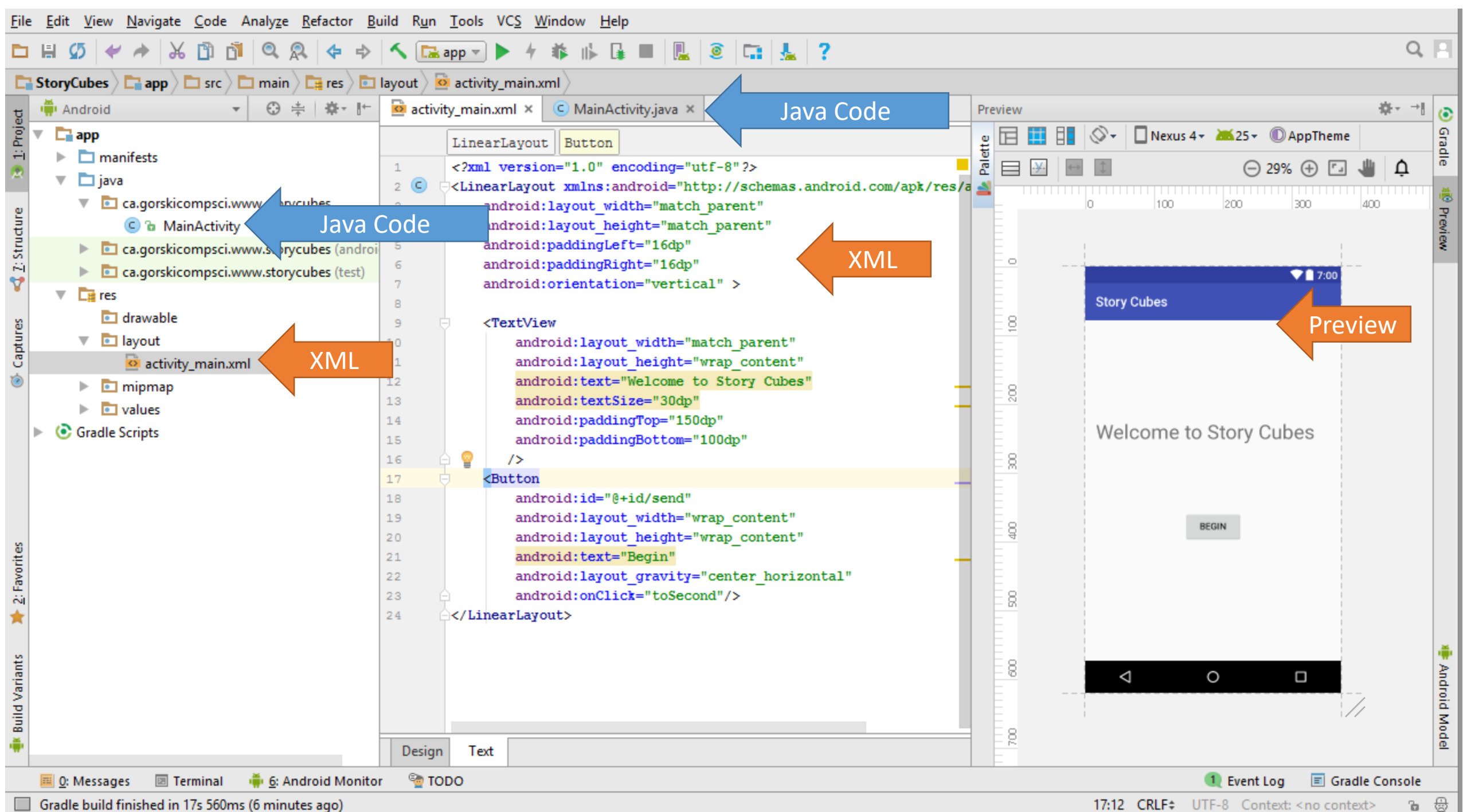
Android Studio is a real developers program.
The training wheels are gone.

If you make a foolish mistake,
that's your problem.



XML is
instructions for
how to draw the
image on the
screen.






```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Welcome to Story Cubes"
    android:textSize="30dp"
    android:paddingTop="150dp"
    android:paddingBottom="100dp" />
```

```
<Button
    android:id="@+id/send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Begin"
    android:layout_gravity="center_horizontal"
    android:onClick="toSecond" />
```

```
</LinearLayout>
```



Tags

- XML creates views using tags.
- They start with < and End with />
- Be careful of the start and end Brackets. If you lose them or misplace them, the code doesn't run.

Attributes

- When you create a new view, you tell XML how to format it.
- Colour, font, height, width are all attributes that can be specified

XML SYNTAX

<TextView

android:text="Happy Birthday!"
android:background="@android:color/darker_gray"
android:layout_width="150dp"
android:layout_height="75dp" />

↑
Attribute
Name

↑
Attribute
Value in
"Quotations"

XML SYNTAX

Line
numbers ↓

```
1 <TextView
2     android:text="Happy Birthday!"
3     android:textColor="@android:color/white"
4     android:background="@android:color/black"
5     android:layout_width="200dp"
6     android:layout_height="300dp" />
```

1. What's the name of the XML element?
2. List all attribute names (not attribute values) separated by commas
3. On what line number is the tag closed?

<TextView

```
1  android:text="Welcome!"
2  android:layout_width="match_parent"
3  android:layout_height="0dp"
4  android:layout_weight="1"
5  android:padding="20dp"

6  android:background="#FFFF00"
7  android:textColor="#FF0000"
8  android:layout_gravity="center"
9  android:fontFamily="sans-serif"
10 android:textSize="48sp" />
```



First
5!

GETTING PAST ERRORS

```
<Text View
  android:text="Hapy Birthday"
  android:layout_width="150dp"
  android:layout_height="150"
  android:background="@android:color/darker_groy"
>
```

Uh oh! Invalid
XML... help!

4 errors.

```
<TextView  
    android:text="I got you a free hug. Surprise!"  
    android:background="#9C27B0"  
    android:textColor="#DCE775"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textSize="45sp" />
```

