ANDROID USER INTERFACE

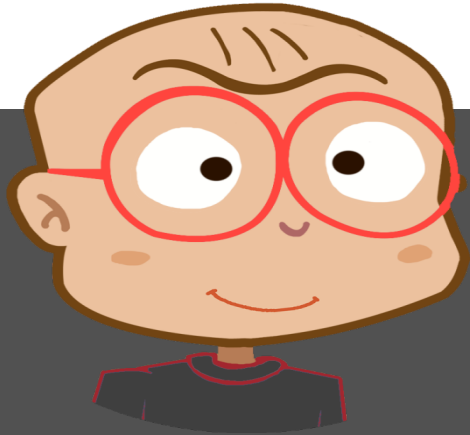**START** »

Table of contents

Android UI Layout

Android UI Controls

Android Event Handling

Android Style and Themes
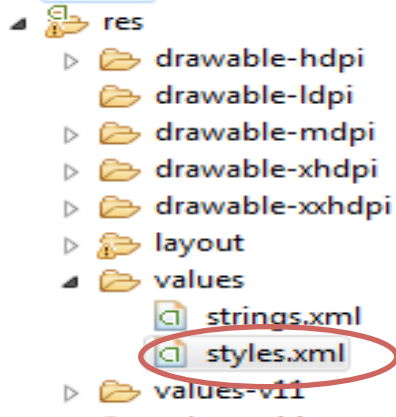
Android Custom Components

Android Styles & Themes

Android Style & Theme

```xml
<?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:orientation="vertical" >

    <TextView
        android:id="@+id/text_id"
        style="@style/CustomFontStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onButtonClicked"
        android:text="TestStyle" />
</LinearLayout>
```

Android Style & Theme

```
res
  drawable-hdpi
  drawable-ldpi
  drawable-mdpi
  drawable-xhdpi
  drawable-xxhdpi
  layout
  values
    strings.xml
    styles.xml
  values-v11
```

```xml
<resources>
    <style name="CustomFontStyle">
        <item name="android:capitalize">characters</item>
        <item name="android:typeface">monospace</item>
        <item name="android:textSize">12sp</item>
        <item name="android:textColor">#00FF00</item>
    </style>
    <style name="CustomFontStyle.LargeFont">
        <item name="android:textSize">22sp</item>
    </style>
    <style name="CustomFontStyle.LargeFont.Red">
        <item name="android:textColor">#FF0000</item>
    </style>
</resources>
```
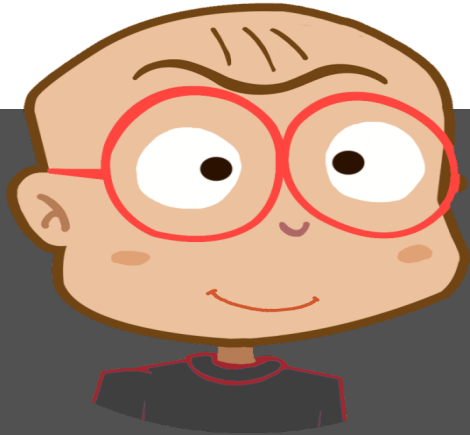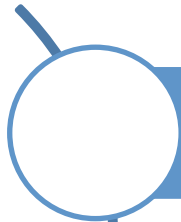
Android Style & Theme

```xml
<activity
        android:name="com.tecdkids.ZZZActivity"
        android:theme="@style/CustomFontStyle" >
</activity>
```

Android Custom Components
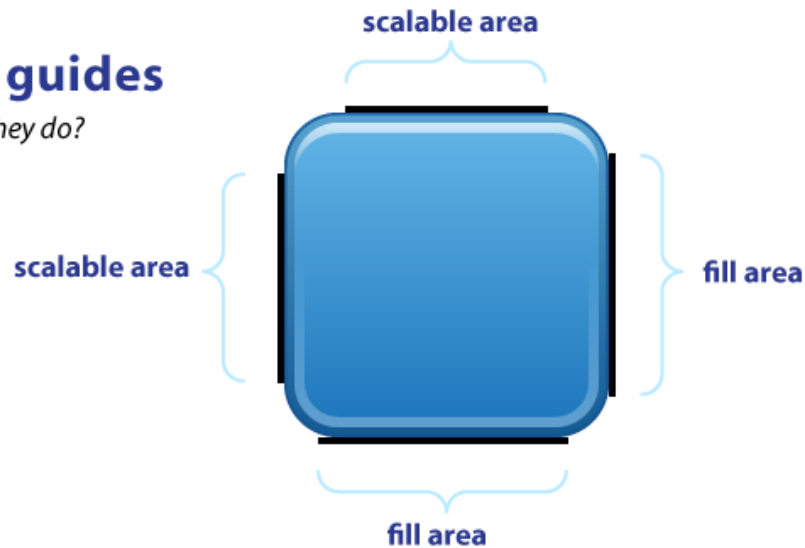
Android Custom Components

Draw 9-patch

Custom Component with Custom Attributes

Android Custom Components

9-patch guides
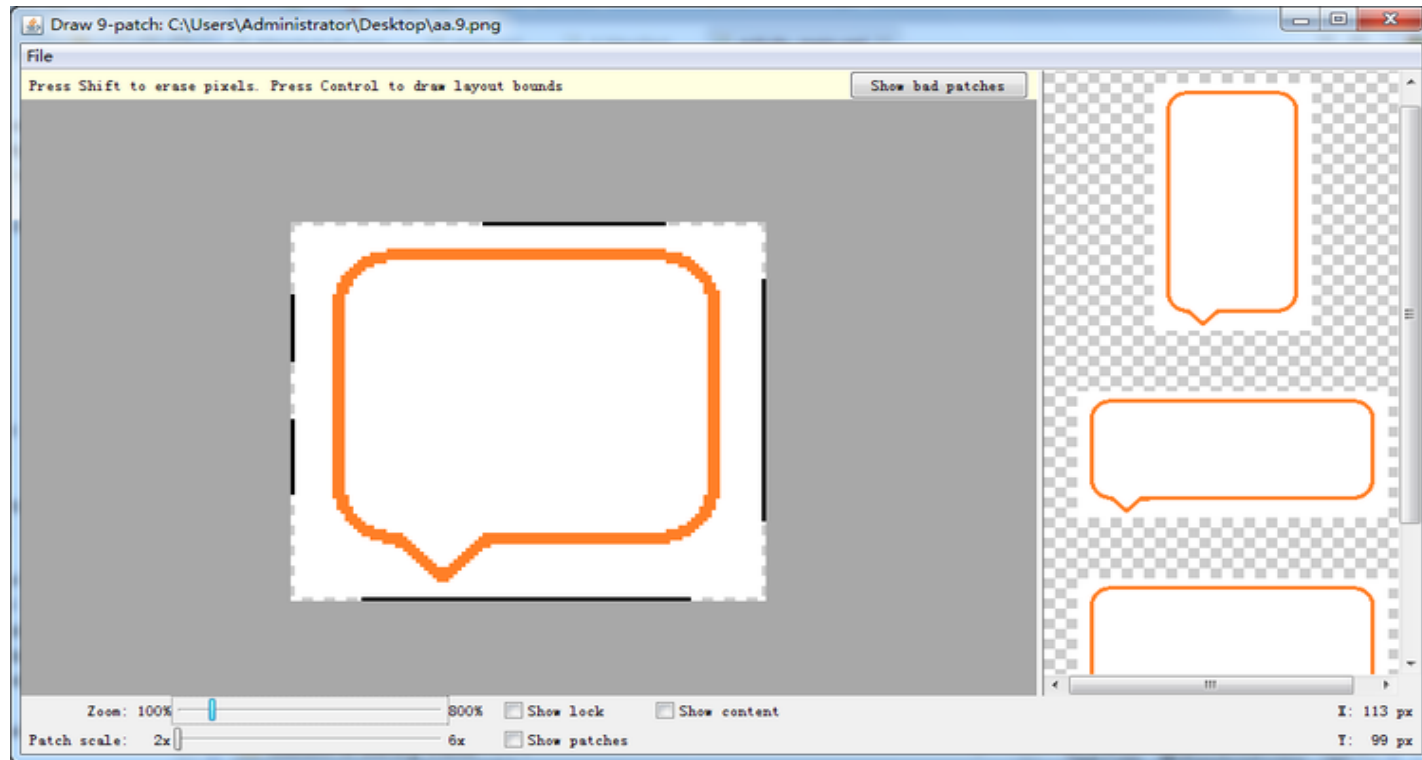what do they do?

scalable area

scalable area

fill area

fill area

The **Draw 9-patch** tool is a WYSIWYG editor that allows you to create bitmap images that automatically resize to accommodate the contents of the view and the size of the screen. Selected parts of the image are scaled horizontally or vertically based indicators drawn within the image.
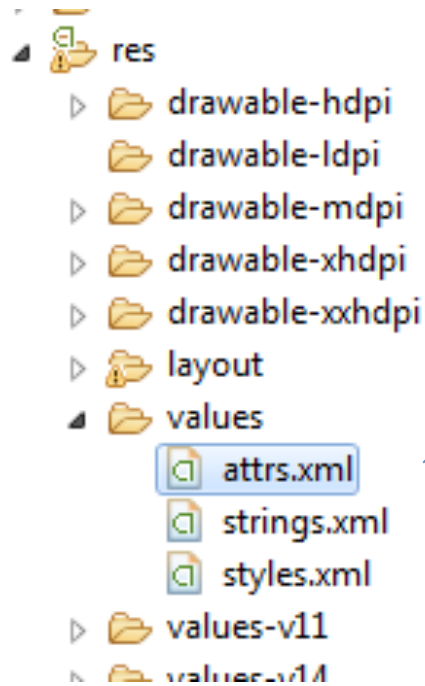
Android Custom Components

Custom Component with Custom Attribute

<?xml version="1.0" encoding="utf-8"?>

```
res
  drawable-hdpi
  drawable-ldpi
  drawable-mdpi
  drawable-xhdpi
  drawable-xxhdpi
  layout
  values
    attrs.xml
    strings.xml
    styles.xml
  values-v11
  values-v14
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="CustomEditText">
        <attr name="needNumber" format="boolean" />
    </declare-styleable>
</resources>
```

Using

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    .......
<com.techkids.androidui.widget.CustomEditText
        android:id="@+id/customEdittext"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TestStyle"
        app:needNumber="true" />
</LinearLayout>
```

Custom Component with Custom Attribute

Get Attribute

```java
public class CustomEditText extends EditText
{
        ...

TypedArray typedArray = this.context.obtainStyledAttributes(attrs,R.styleable.CustomEditText);
        final int sizeOfType = typedArray.getIndexCount();
        for (int i = 0; i < sizeOfType; ++i) {
                int attr = typedArray.getIndex(i);
                switch (attr) {
                        case R.styleable.CustomEditText_needNumber:
                        isNeedNumber = typedArray.getBoolean(attr, false);
                        break;
                }
        }
        typedArray.recycle();

        ...
}
```
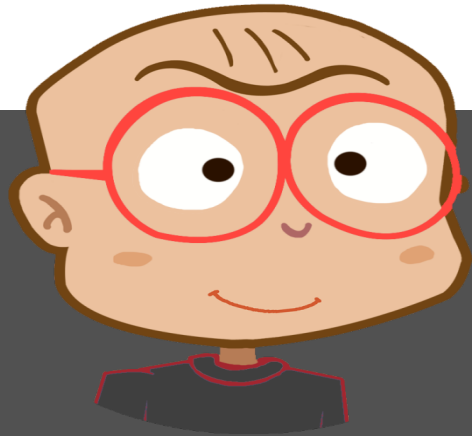
Custom Component with Custom Attribute

Using attribute

```java
public void validate() {
        this.setError(null);
        if (isNeedNumber) {
                try {
                        String text = this.getText().toString();
                        text = text.replaceAll("\\D", "");
                        if (text.length() == 0) {
                        this.setError("Please check data input!!!");
                }
                } catch (Exception e) {
                }
        }
}
```

Shape & Selector

Shape

Shape Drawable
This is a generic shape defined in XML.

- FILE LOCATION:res/drawable/*filename*.xml

- RESOURCE REFERENCE:In Java: R.drawable.*filename*

- In XML: @[*package*:]drawable/*filename*

Shape

```xml
<?xml version="1.0" encoding="UTF-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android" >
    <solid android:color="#1e1e1e" />
    <stroke
        android:width="1dp"
        android:color="#55ffffff" />
    <corners
        android:bottomLeftRadius="8dp"
        android:bottomRightRadius="8dp"
        android:topLeftRadius="8dp"
        android:topRightRadius="8dp" />
    <gradient
        android:angle="0"
        android:endColor="#ef5429"
        android:startColor="#ad3210" />
</shape>
```

Selector

- FILE LOCATION:res/drawable/*filename*.xml

- RESOURCE REFERENCE:In Java: R.drawable.*filename*

- In XML: @[*package*:]drawable/*filename*

Shape

```xml
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:drawable="@drawable/bg_blue_btn1_p"
android:state_pressed="true"/>
    <item android:drawable="@drawable/bg_blue_btn1_p"
android:state_focused="true"/>
    <item android:drawable="@drawable/bg_blue_btn1"/>

</selector>
```

Exit Course

THANK YOU