



Head First XML Layout on Android

Yuki ANZAI
@yanzm

BootCamp 27 Sep 2010

自己紹介



あんざいゆき
@yanzm

日本Androidの会 女子部副部長
デ部とかデザイン部とか。。。

Android アプリ開発者「Libraroid」
「BooXpress」 「Wolfraroid」 など

Question

Android 経験値を教えてください！

1. Android開発、まったくやったことないです
2. HelloWorldはやりました
3. バリバリ開発してます



Layout

View の配置



画面のレイアウト方法

コードで生成

動的に変更したい場合

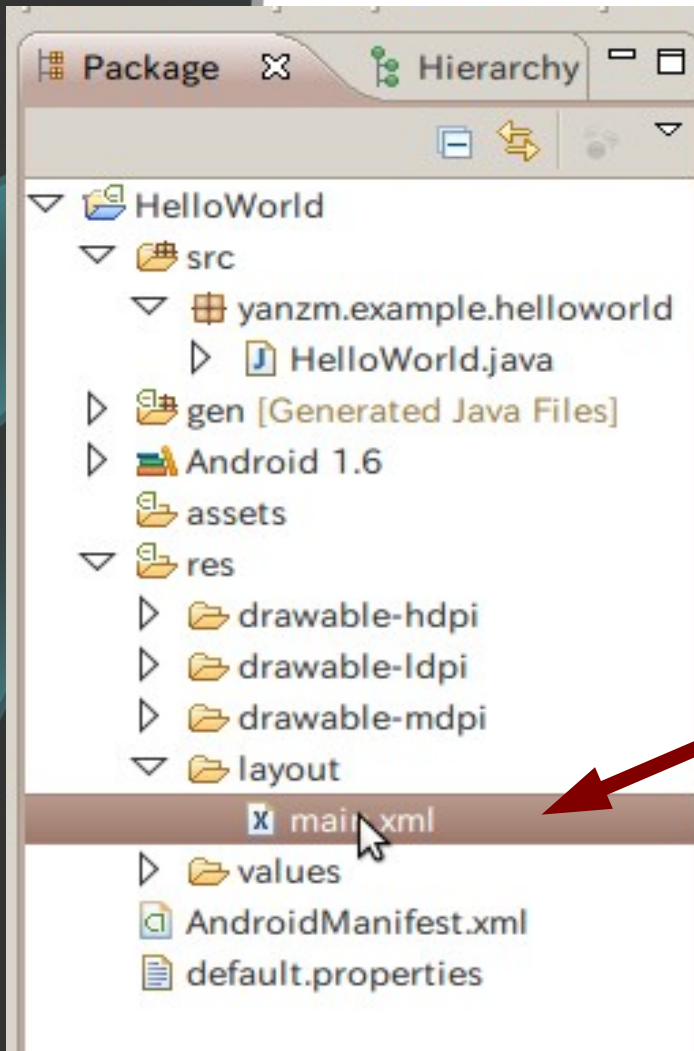
XML で定義

静的なレイアウトを定義する場合

XMLで定義する利点

- デザインとアクションの分離
- コードの見通しがよくなる
- メンテナンスが楽 = 変更しやすい
- 複雑なレイアウトをコードで生成するのはバグの元
- 縦・横画面用のレイアウトが別々に定義できる
- 再利用しやすい

レイアウト定義用XMLの場所



res/layout/ の下

res/layout/ *filename.xml*

Eclipseで Android プロジェクトを作成すると、
main.xml が作られる

The background features a dark teal gradient with several overlapping, curved, semi-transparent teal shapes that create a sense of depth and movement. The text is centered in a white, serif font.

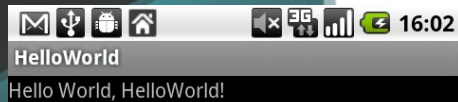
HelloWorld Demo

HelloWorld main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/ap
k/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
    />
</LinearLayout>
```

View

ViewGroup



HelloWorld
Hello World, HelloWorld!

View の配置を定義するのが ViewGroup

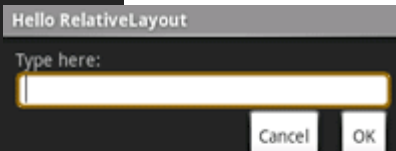
A dark teal background with a large, curved, lighter teal abstract shape on the left side, resembling a stylized wave or a curved arrow pointing towards the right.

LayoutGroup

LayoutGroup



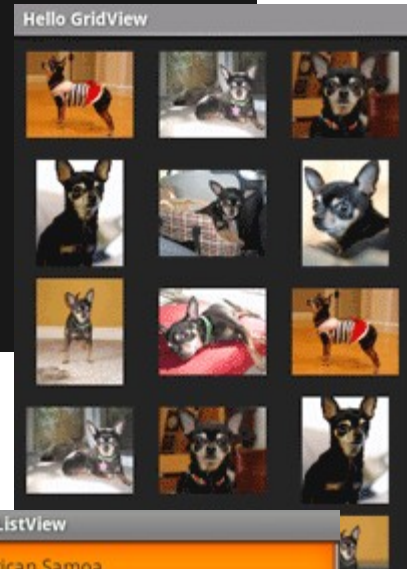
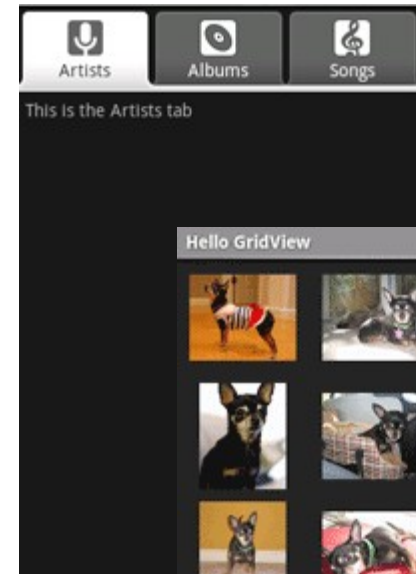
row one
row two
row three
row four



LinearLayout
RelativeLayout
TableLayout
FrameLayout
(AbsoluteLayout)

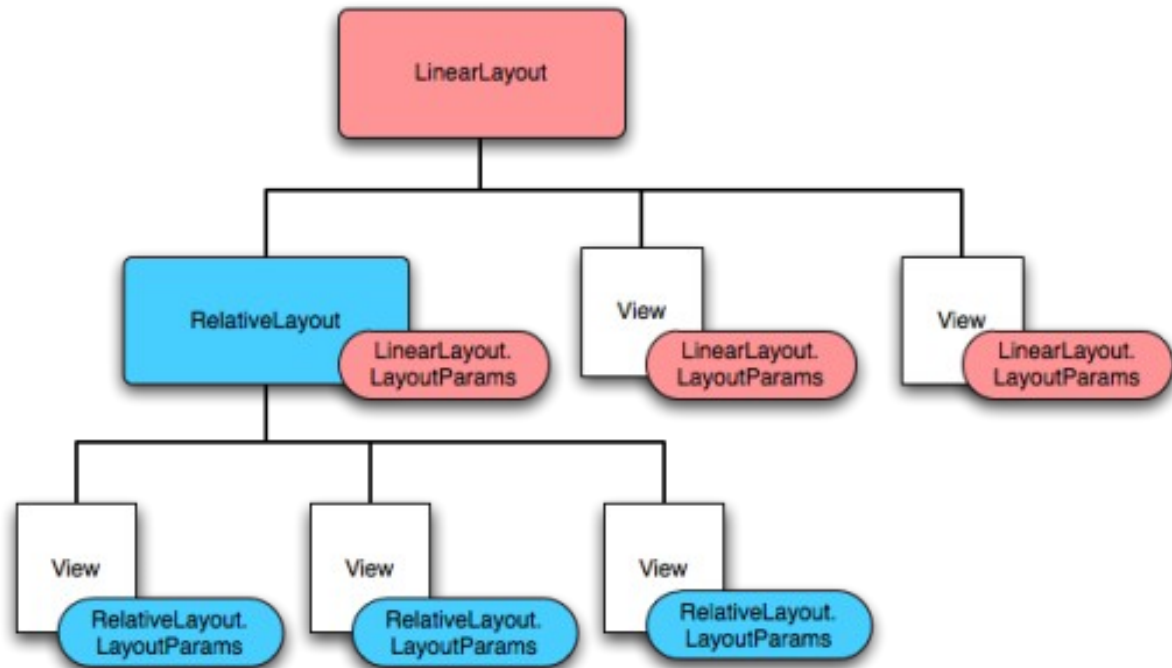
ListView
GridView
TabHost

...



LayoutGroup の注意点

- layout_width と layout_height は必須
- LayoutGroup は入れ子にできる
- View には親のレイアウトパラメータを指定



LinearLayout

LinearLayout



LinearLayout



<LinearLayout>

<LinearLayout>

<EditText />

<Button />

<Button />

<LinearLayout>

<LinearLayout>

<TextView />

<ImageView />

<LinearLayout>

<Button />

<Button />

<ScrollView>

<TextView />

<LinearLayout>

<ScrollView>

<TextView />

<Button />

<LinearLayout>

<Button />

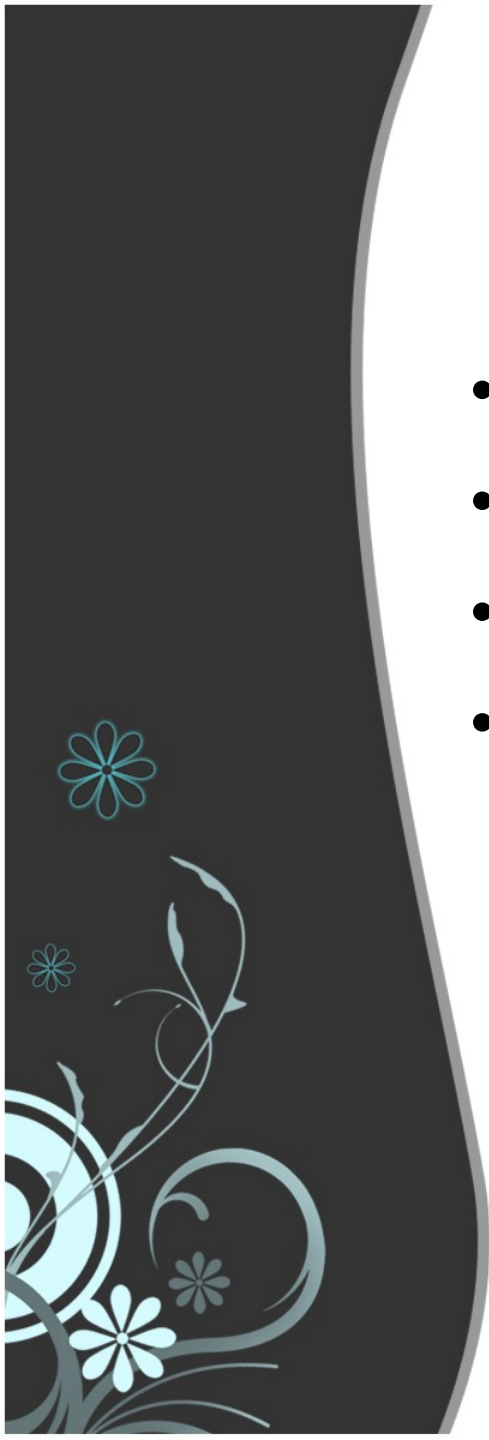
<Button />

<Button />

<Button />

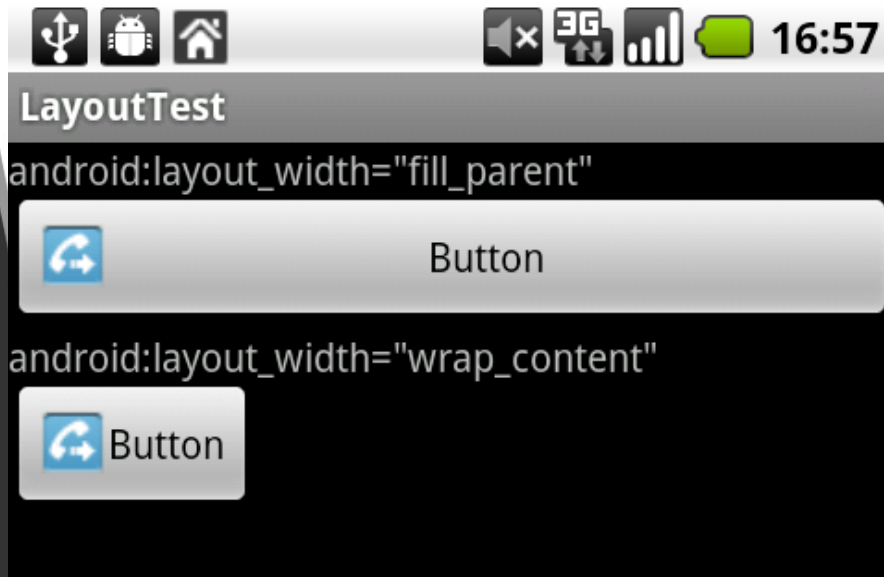
LinearLayout

- fill_parent と wrap_content
- android:weight
- 画面を区切って配置する方法に最適
- 複雑なレイアウトには不向き



fill_parent ・ wrap_content

- android:height, android:width に指定
- fill_parent : 親の幅一杯を指定
- wrap_content : View 中のコンテンツがちょうど入る大きさ

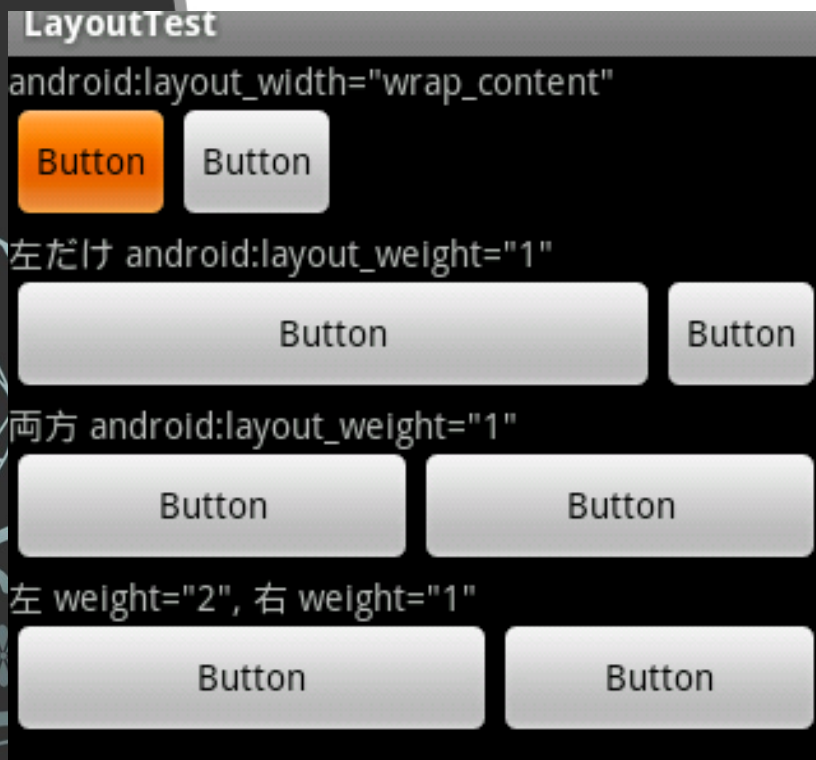


fill_parent

wrap_content

android:layout_weight

- LinearLayout で余った領域を割り当てる時の重みを指定
- 高さ、幅の % 指定のようなもの



wrap_content

左だけ layout_weight="1"

両方 layout_weight="1"

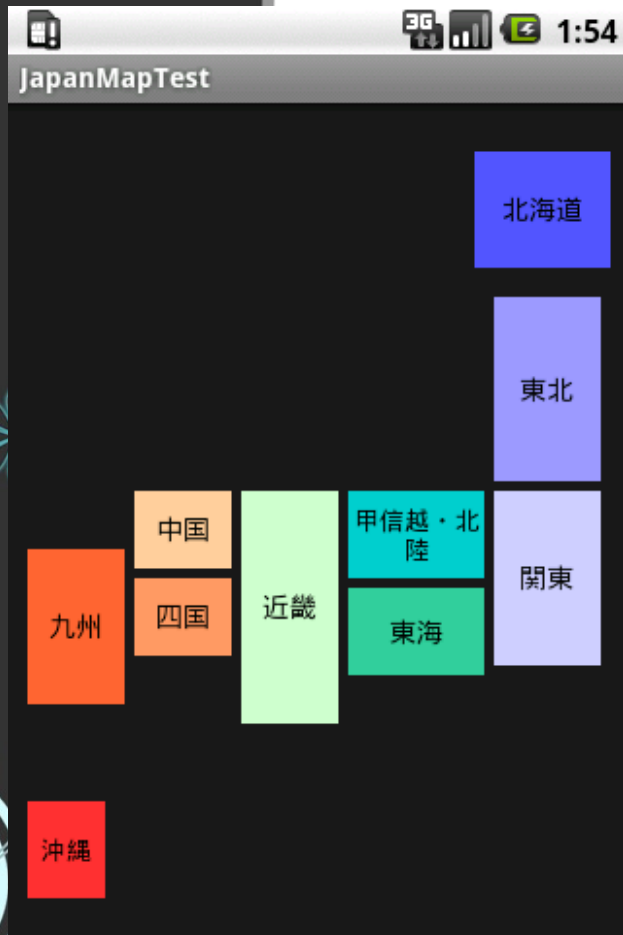
左 layout_weight="2"
右 layout_weight="1"

A dark teal background with a large, curved, lighter teal abstract shape on the left side, resembling a stylized wave or a curved arrow pointing towards the right.

RelativeLayout

RelativeLayout

LinearLayout では
難しい



RelativeLayout

<RelativeLayout>

<Button /> : 中心

```
id="@+id/center"  
centerInParent="true"
```

親（この場合画面）の中心

<Button /> : 近畿

```
id="@+id/kinki"
```

```
layout_below="@id/center"
```

中心の下

<Button /> : 中国

```
id="@+id/chugoku"
```

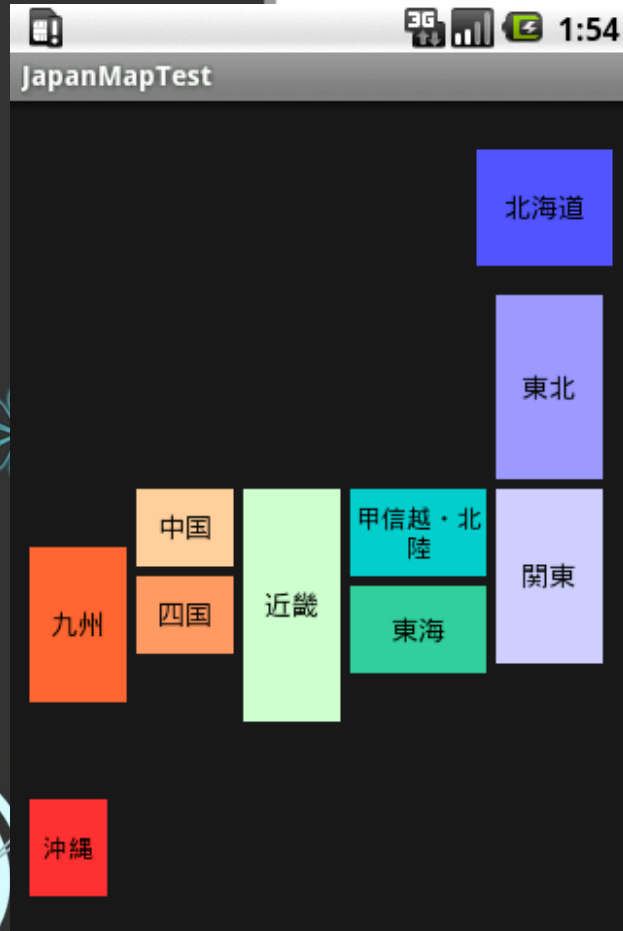
```
layout_toLeftOf="@id/kinki"
```

```
layout_alignTop="@id/kinki"
```

近畿の左

<Button />

...

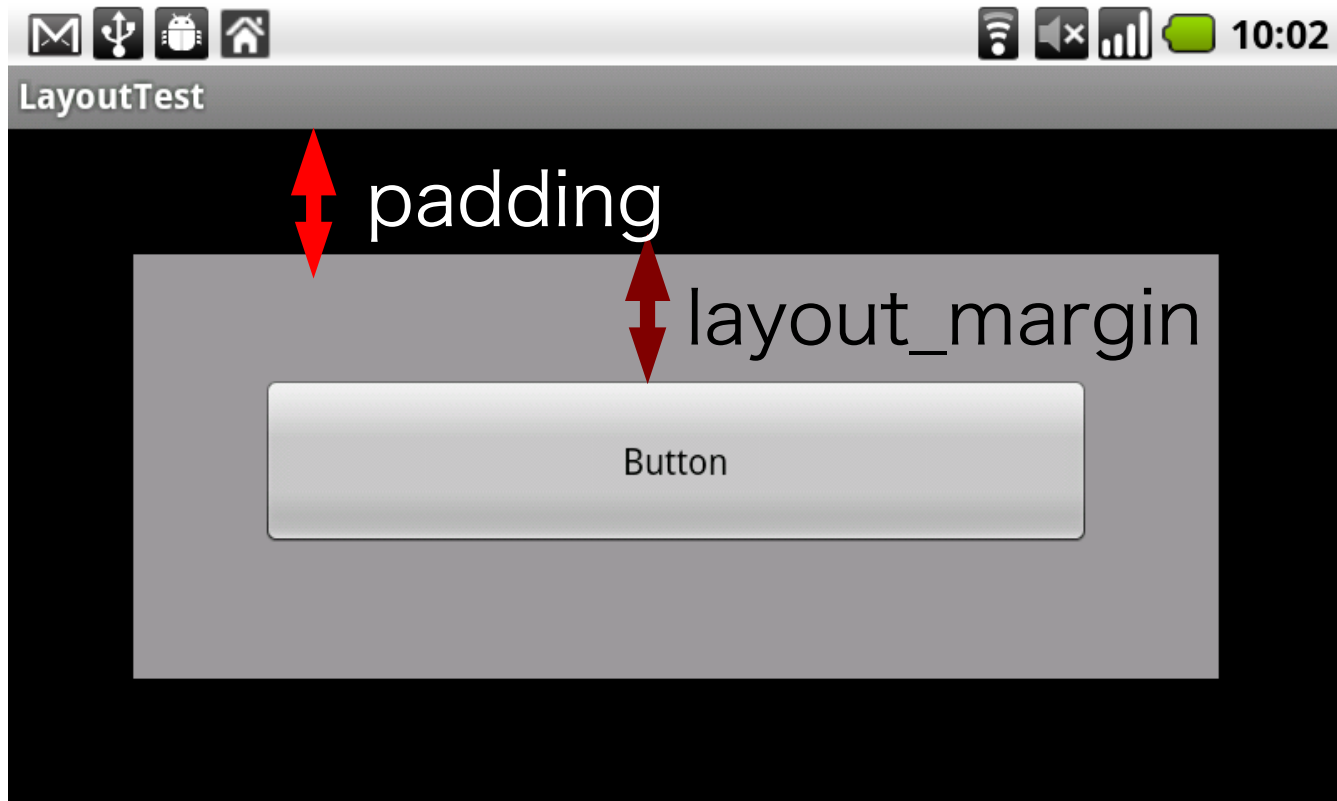


よりよいレイアウトへ

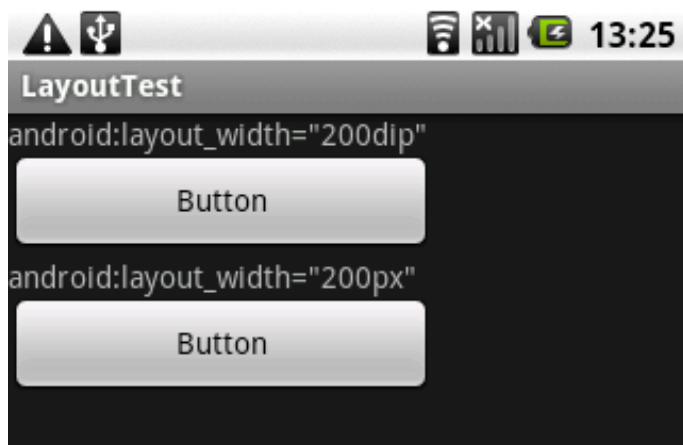
ここちよいレイアウト

- マージン、パディングを適切に設定
 - `android:layout_margin`
 - `android:padding`
 - padding は ViewGroup のみ
View でも定義はできるがサポートされていない
- dipを使う！
 - px を使ったピクセル指定はだめ

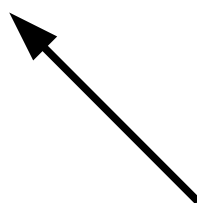
margin と padding



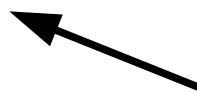
dip vs px



- 上のボタン: 200 dip
- 下のボタン: 200 px
- dip なら、画面に対する割合が同じ



Nexus One



HT-03A

フォーカスハンドリング

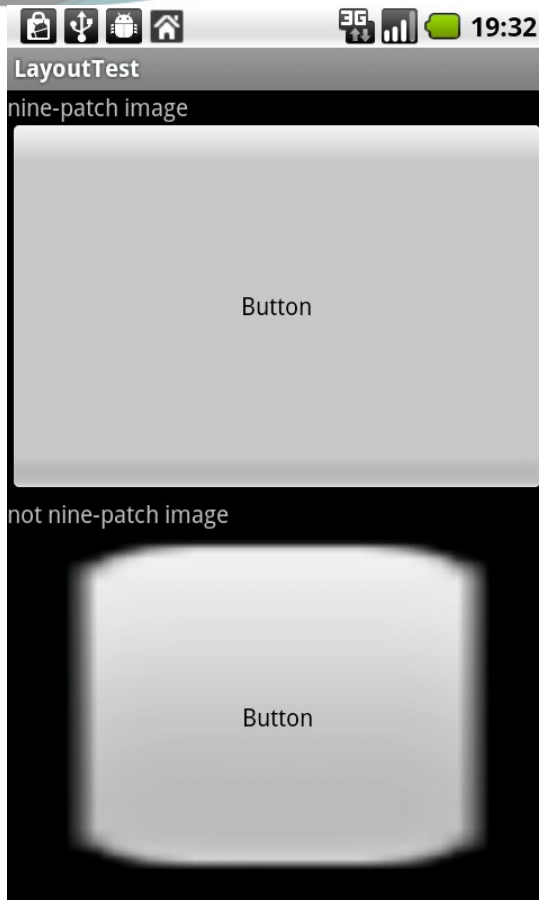
- キーボード、トラックボール操作を意識する
- 今どこにフォーカスがある？
- スクロールバーにフォーカス
 - `focusable="true"`
 - `focusableInTouchMode="true"`
- フォーカスのリクエスト
 - `<View>` タグ内に `<requestFocus />` を入れる
 - 1画面に1つだけ
 - コードからリクエスト

マルチデバイス対応

- `nine-patch (img.9.png)`
- `maxHeight`, `maxWidth`, `minHeight`, `minWidth`
- `<dimen>`
 - リソースとしてサイズを指定
 - `<dimen name="listHeight">64dip</dimen>`
 - コードから `setWidth(width)` など
- Not use absolute layout

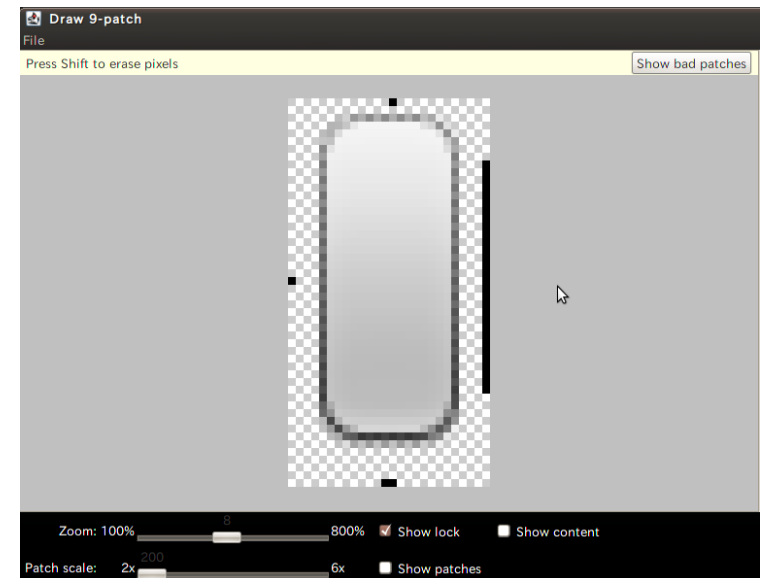
nine-patch

- 拡大する領域を 1px で指定
- `sdk-dir/tools/draw9patch`



nine-patch

not
nine-patch

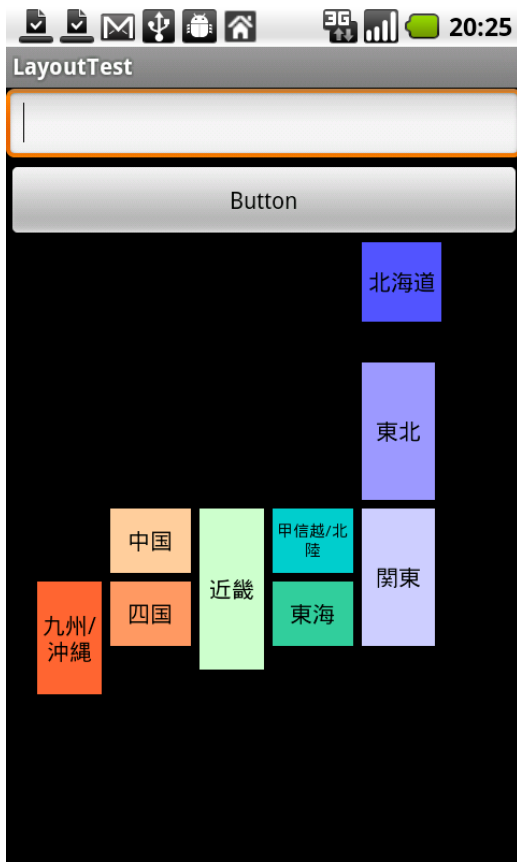


より開発をスムーズに

レイアウトの再利用

- `<include>` タグ
 - `<include layout="@layout/layout_resource"/>`
- `<merge>` タグ
 - custom view を作るときに有用
 - 最適化のため、view-tree のレベルを削減
- `ViewStub`を使う

<include> tag



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent" >
  <EditText
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
  <Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Button" />
  <include
    layout="@layout/japanmap_layout" />
</LinearLayout>
```

ViewStub

- たまにしか使われない View に使う
- dumb and lightweight view
- dimension を持たない、何も描画しない
- layout file の中では ViewStub で定義
- 必要なときにコードから inflate する

ViewStub

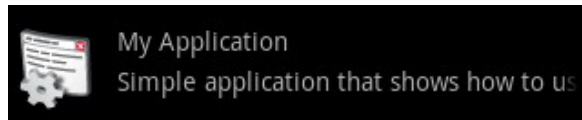
```
<ViewStub
    // inflate するための id
    android:id="@+id/stub_import"
    // inflate したあと override する id
    android:inflatedId="@+id/panel_import"
    // 置き換える layout file
    android:layout="@layout/progress_overlay"
/>
```

inflate() or visibility をセットする(とinflateされる)

```
((ViewStub) findViewById(R.id.stub_import))
    .setVisibility(View.VISIBLE);
// or
View importPanel = ((ViewStub) findViewById(R.id.stub_import))
    .inflate();
```

効果的なレイアウト

- ListViewの子要素のXML定義の注意点
 - 何度も inflate されるので簡素化
 - LinearLayout の入れ子よりも RelativeLayout 1個で実現すべし



```
<LinearLayout>  
  <ImageView />  
  <LinearLayout>  
    <TextView />  
    <TextView />
```



```
<RelativeLayout>  
  <ImageView />  
  <TextView />  
  <TextView />
```

Summary

- XML でレイアウトを定義すると
 - メンテナンスが楽
 - 再利用しやすい
 - 省エネにする手法がある
- ユーザーを意識したレイアウト
 - マージン、パディング
 - フォーカス
 - マルチデバイス対応