

# イベント

【宜野湾】第2回 Android勉強会 全6回

- レイアウト構築について
- イベントについて
- リスト表示について

# レイアウト構築について

- テキスト : TextView

- 画像 : ImageView

- レイアウト :

  - LinearLayout/RelativeLayout

- View

- 様々なView

# レイアウト構築について

## ■ テキスト

XML

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:drawableStart="@mipmap/ic_launcher"
    android:gravity="center"
    android:text="@string/app_name"
    android:textColor="#f00"
    android:textSize="16sp"
    android:textStyle="bold"/>
```

レイアウト

My Application



My Application

### ■ textSize : サイズ指定単位 sp

setText(CharSequence, TextView.BufferType)

### ■ textColor : 色指定 #ffffffff ARGB

setTextColor(ColorStateList)

### ■ textStyle : Bold/Italic

setTypeface(Typeface)

### ■ gravity : 左寄せ、中央、右寄せ等

setGravity(int)

### ■ drawableStart : テキストと画像設定

setCompoundDrawablesWithIntrinsicBounds(Drawable, Drawable, Drawable, Drawable)



# レイアウト構築について

## ■ 画像

### XML

```
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@mipmap/ic_launcher_round"  
    android:scaleType="center"  
    android:adjustViewBounds="true"  
    android:contentDescription="@string/app_name"/>
```

### レイアウト表示

My Application



### ■ src : 画像設定

setImageResource(int)

### ■ adjustViewBounds : 縦横比維持設定

setAdjustViewBounds(boolean)

### ■ scaleType : リサイズ/配置設定

setScaleType(ImageView.ScaleType)

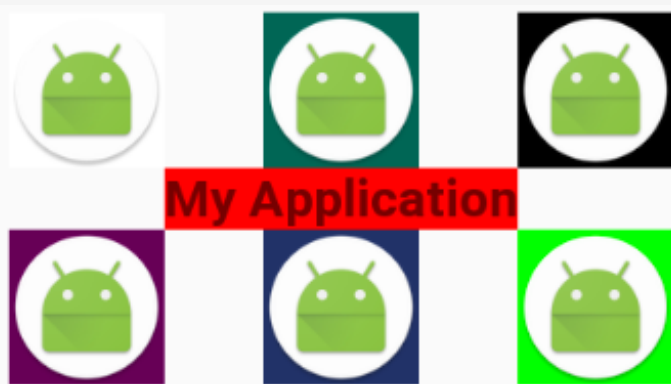
ScaleType

- matrix : Matrixでの描画
- fitXY : 縦横最大表示、アスペクト比無視
- fitStart : 左上端に寄せる
- fitCenter : 中央に寄せる
- fitEnd : 右下端に寄せる
- center : 中央に寄せる
- centerCrop : 余白含め中央寄せ
- centerInside : 余白除く中央寄せ

# レイアウト構築について

## ■ レイアウト : RelativeLayout 相対配置

### レイアウト表示



### ■ 設定項目

- layout\_above
- layout\_alignBaseline
- layout\_alignTop
- layout\_alignBottom
- layout\_alignLeft
- layout\_alignRight
- layout\_alignParentTop
- layout\_alignParentBottom
- layout\_alignParentLeft
- layout\_alignParentRight
- layout\_alignWithParentIfMissing
- layout\_below
- layout\_centerHorizontal
- layout\_centerInParent
- layout\_centerVertical
- layout\_toLeftOf
- layout\_toRightOf

# レイアウト構築について

## ■ レイアウト : LinearLayout 一列配置

### XML

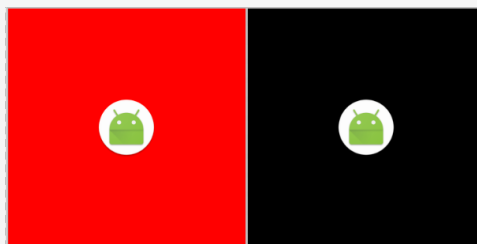
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <ImageView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:adjustViewBounds="true"
        android:background="#f00"
        android:scaleType="center"
        android:src="@mipmap/ic_launcher_round"/>

    <ImageView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:adjustViewBounds="true"
        android:background="#000"
        android:scaleType="center"
        android:src="@mipmap/ic_launcher_round"/>

</LinearLayout>
```

### レイアウト表示



### ■ orientation : 縦横設定

vertical | horizontal

setOrientation(int)

### ■ layout\_weight : 比率指定

数値での設定



# レイアウト構築について

## ■ View

xml

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="horizontal">

  <View
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="#0f0"
    android:visibility="visible"/>

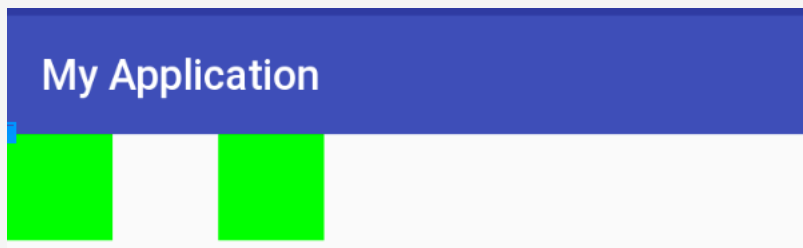
  <View
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="#0f0"
    android:visibility="invisible"/>

  <View
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="#0f0"
    android:visibility="gone"/>

  <View
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="#0f0"
    android:visibility="visible"/>

</LinearLayout>
```

レイアウト表示



■ background : 背景色設定

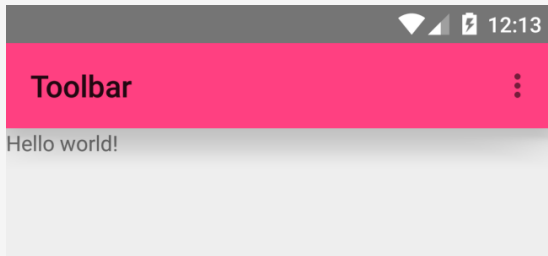
■ visibility : 表示設定

setVisibility(int)



# レイアウト構築について

## ■ 様々なView



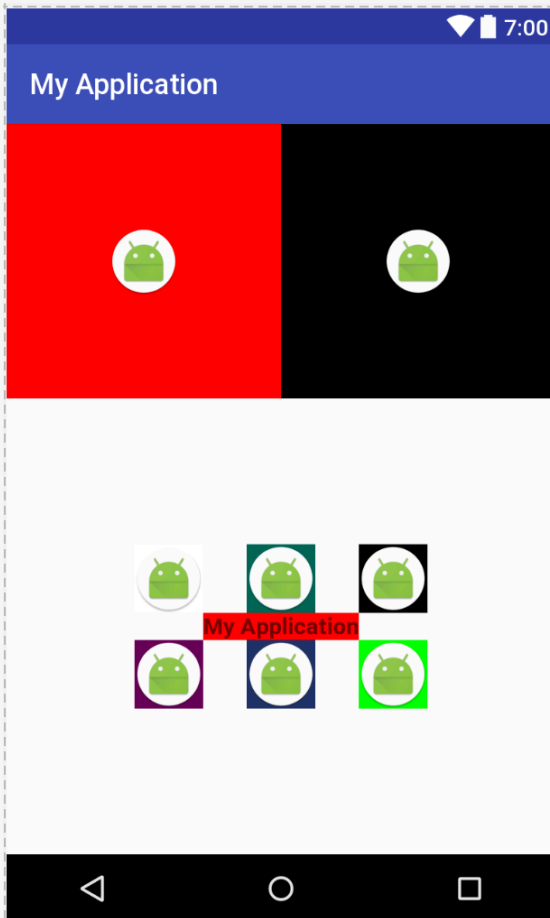
- ViewPager
- ListView
- GridView
- ScrollView
- Toolbar
- CheckBox
- RadioButton
- Spinner etc...





# レイアウト構築について

## ■ 下記を作っていきます



### 使用するView

- LinerLayout
- RelativeLayout
- ImageView
- TextView

# レイアウト構築について

## ■ activity\_main.xml

とりあえずxmlを書きましょう！  
ただ別紙参照です！

パワポに入らない。。。。

# レイアウト構築について:activity\_main.xml

## ■ LinearLayout

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <ImageView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:adjustViewBounds="true"
        android:background="#f00"
        android:scaleType="center"
        android:src="@mipmap/ic_launcher_round"/>

    <ImageView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:adjustViewBounds="true"
        android:background="#000"
        android:scaleType="center"
        android:src="@mipmap/ic_launcher_round"/>

</LinearLayout>
```

layout\_width=0dp  
weight=1  
がポイント

# レイアウト構築について:activity\_main.xml

## ■ RelativeLayout

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:background="#f00"
        android:gravity="center"
        android:text="@string/app_name"
        android:textSize="16sp"
        android:textStyle="bold"/>

    <ImageView
        android:id="@+id/image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/text"
        android:layout_toEndOf="@id/text"
        android:adjustViewBounds="true"
        android:background="#0f0"
        android:contentDescription="@string/app_name"
        android:scaleType="center"
        android:src="@mipmap/ic_launcher_round"/>
```

layout\_centerInParent=true  
layout\_below="@id/text"  
toEndOf="@id/text"  
toStartOf="@id/text"  
等がポイント



# イベントについて

## ■ よく活用するイベント

```
View view = findViewById(R.id.top);
view.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(MainActivity.this, "onClick",
    }
});

view.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View view) {
        Toast.makeText(MainActivity.this, "onLongClick",
    }
});

view.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        Log.d("OnTouchListener", "Action = " + motionEvent.getAction());
        return false;
    }
});
```

※別紙参照

■ OnClickListener : クリックイベント

■ OnTouchListener : タッチイベント

■ OnItemClickListener : リストクリックイベント

など

# リスト表示について

## ■ シンプルなリスト表示

My Application	
Position 0	
Position 1	
Position 2	
Position 3	
Position 4	
Position 5	

- ListViewを設定・取得
- ArrayAdapterを設定
- ListViewにAdapterを設定



# リスト表示について

## ■ ListViewを設定・取得

activity\_main.xml

```
<ListView  
    android:id="@+id/list"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"/>
```

MainActivity.java

```
ListView listView = (ListView) findViewById(R.id.list);
```

■ xmlにListViewを設定

■ javaファイルでListViewを取得

# リスト表示について

## ■ ArrayAdapterを設定

MainActivity.java

```
ArrayAdapter<String> adapter = new ArrayAdapter<>(this,  
                                                android.R.layout.simple_list_item_1);  
for(int i = 0; i < 20; i++){  
    adapter.add("Position " + i);  
}
```

### ■ Adapterインスタンスを生成

### ■ Adapterに値を設定

#### ポイント1

レイアウト設定

Android.R.layout.simple\_list\_item\_1

#### ポイント2

値の追加

adapter.add("Position " + i);



# リスト表示について

## ■ ListViewにAdapterを設定

MainActivity.java

```
listView.setAdapter(adapter);
```

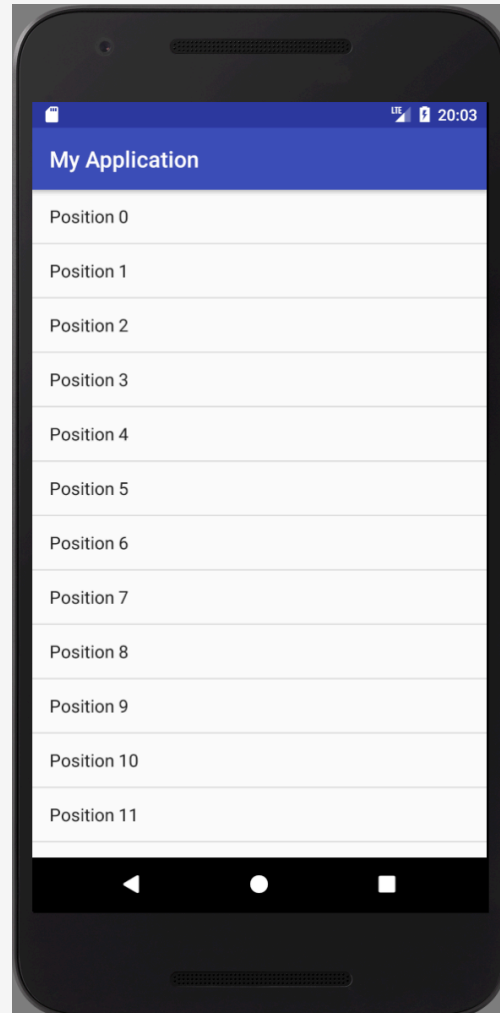
```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
    }  
});
```

- `setAdapter(adapter);`
- ついでにアイテムクリック



# リスト表示について

■ 完成！



# リスト表示について：注意点

- テキスト表示であれば簡単
- カスタマイズすると一度はハマる
- だが基本的にカスタマイズが求められる
- Adapterにも色々な種類がある
  - ArrayAdapter
  - SimpleAdapter
  - BaseAdapter ※一番おすすめ！

- レイアウト構築の基本がわかった
  - Layout, Text, Image etc...
- 主要となるイベント設定がわかった
- 簡単なリスト表示方法がわかった
- List表示についてなんとなく、ハマるポイントが存在しているという事実を知った