

编程规范的形成在于细节的积累

1、如何不让EditText获取焦点

在EditText的父级控件中找一个，设置成
android:focusable="true"
android:focusableInTouchMode="true"

2、如何让ScrollView保持全屏 尤其对于RelativeLayout

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/scroller"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:fillViewport="true" >
```

3、Activity Fragment嵌套的时候如何保证最上层Fragment才退出，其次要注意，在最顶层的Fragment不要把自己添加到backStack中，否则返回就会出现空白，并注意，该规则在整个App中保持一致。

```
FragmentManager manager = getFragmentManager();
if(manager.getBackStackEntryCount() > 0){
    manager.popBackStack();
}
else{
    getActivity().finish();
}
}
```

4、xmlns:android的作用

xmlns:android这是一个XML命名空间，告诉Android开发工具你准备使用Android命名空间里的一些通用属性。

在所有Android XML设计文件中，最外层的标记必须使用这个属性。有了它，你就可以alt+/作为提示，不该输入什么，什么是对的，什么是错的，也可以理解为语法文件。或者语法判断器什么的。

5、

百度地图-点击事件问题

地图上marker或其他对象的点击事件都会传到map对象，所以很多人在使用API时发现我绑定了map和一个marker的click事件，发现点击marker的时候，不仅marker的click事件会被触发，map的click事件也会被触发。API会将事件向上传递，实际上点击任何覆盖物都会向上传递到map。那问题是怎么区分呢？map的click事件的事件参数e中包含一个名为overlay的属性，所以只需要在事件处理函数中判断overlay是否存在即可区分

```
[javascript]
01. map.addEventListener("click", function(e) { //点击事件
02. //alert(e.point.lng + ", " + e.point.lat);
03. if(!e.overlay){
04.     //alert("不是覆盖物")
05. }
```

```

04.         alert("aaaaaaa");
05.     }
06. }

```

6、ANDROID中EDITTEXT如何定位光标位置

代码： `edittext.setSelection(int);`
 范例：
`et.setText(content);` //设置EditText控件的内容
`et.setSelection(content.length());` //将光标移至文字末尾

7、判断是否为空

```

public static boolean isEmpty(CharSequence str) {
    if (str == null || str.length() == 0)
        return true;
    else
        return false;
}

```

8、确保TextView居中

`android:includeFontPadding="false"`

9、LinearLayout基本可以满足任何需求、所有xml属性里面设置的属性，都可以通过代码来实现

10、如何利用自定义的Attrs，首先定义，其次xml引用，最后代码中获取。

第一步：

```

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

    <declare-styleable name="CircleImageView">
        <attr name="border_width" format="dimension" />
        <attr name="border_color" format="color" />
    </declare-styleable>

    <declare-styleable name="slide_switch">
        <attr name="left_text" format="reference" />
        <attr name="right_text" format="reference" />
        <attr name="select_color" format="reference"></attr>
        <attr name="unselect_color" format="reference"></attr>
    </declare-styleable>

</resources>

```

第二步：

```

<com.netease.engagement.widget.SlideSwitchView
    xmlns:slide_view="http://schemas.android.com/apk/res/com.netease.date"
    android:id="@+id/slide_switch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    slide_view:left_text="@string/rec_yuanfen_tab_voice"
    slide_view:right_text="@string/rec_yuanfen_tab_text"
    slide_view:select_color="@color/white"
    slide_view:unselect_color="@color/content_text">
</com.netease.engagement.widget.SlideSwitchView>

```

第三步：

```

*      初始化，主要设置文字颜色、动画时长
*/
public void init(Context context, AttributeSet attrs) {

    mPosition = Position.LEFT;
}

```

```

rootView = LayoutInflater.from(context).inflate(
    R.layout.view_slide_switch_layout, this, true);

bgTextView = (TextView) rootView.findViewById(R.id.bg_purple);
lefTextView = (TextView) rootView.findViewById(R.id.left_text);
righTextView = (TextView) rootView.findViewById(R.id.right_text);
contentView = rootView.findViewById(R.id.content);

selectColor = Color.WHITE;
unSelectColor = Color.GRAY;
disapperingDuration = 100;
appearingDuration = 100;
slideDuration = 200;

```

11、公用一个dialog，一个消失后，同一个加载新布局出现

```

@Override
public void onModifyProfileSuccess(int transactionId, final PortraitInfo obj) {
    if (mChangeAvatarTid != transactionId)
        return;
    stopWaiting();

    if (mTipDialog.isShowing()) {
        mTipDialog.dismiss();
    }

    // 因为公用同一个Dialog，需要延时，否则无法正常显示
    if (lobj.isOpenFate) {

        new Handler(mContext.getMainLooper()).postDelayed(new Runnable() {

            @Override
            public void run() {
                showGotoOpenYuanFen();
            }
        }, 300);
        return;
    }
}

```

12、dialog的布局大小设定，一般至少需要dialog的宽度全屏，如果需要透明，我们在布局里面利用padding或者magin来做。这样比较好控制，一定选择Floating，否则需要自己处理居中的问题，有时候为了点击外部可以取消，我们没办法全屏。

13、View何时初始化的问题：Fragment为例，渲染之后，合理的位置，做合理的事情

```

@Override
public void onViewCreated(View view, Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    initView(view);

    // view渲染成功后，再请求，万一无效呢
    doGetYuanfen();

    //init 功能逻辑代码
}

```

14、计算ListView实际滚动距离

```

public int getScrollY() { //获取滚动距离 View c = listView.getChildAt(0); if (c
== null) { return 0; } int firstVisiblePosition =

```

```
listView.getFirstVisiblePosition(); int top = c.getTop(); int headerHeight = 0;
if (firstVisiblePosition >= 1) { headerHeight = listView.getHeight(); } return
-top + firstVisiblePosition * c.getHeight() + headerHeight; }
```

15、EditTextView设置焦点获取：

这三个属性必须同时设置：

```
private EditText passwde = null;
• passwde.setFocusable(true);
• passwde.setFocusableInTouchMode(true);
• passwde.requestFocus();
```

16、设置光标位置

修改 EditText 中的光标位置

Android 的 EditText 控件默认获取焦点的时候, 插入光标是在第一个位置的, 如果EditText中设置了文本, 这个时候光标是在文本的最前面, 而不是文本的最后. 为了方便用户使用, 需要把光标移动到文本最后, 但是EditText 没有提供 `setCursor` 函数. 经查看其文档 发现 `setSelection` 函数可以完成该功能. 如下:

```
01. EditText et = ...
02. String text = "text";
03. et.setText(text);
04. et.setSelection(text.length());
05.
```

17、WeakHashMap和HashMap的区别

- WeakHashMap, 此种Map的特点是, 当除了自身有对key的引用外, 此key没有其他引用那么此map会自动丢弃此值, 见实例: 此例子中声明了两个Map对象, 一个是HashMap, 一个是WeakHashMap, 同时向两个map中放入a、b两个对象, 当HashMap remove掉a 并且将a、b都指向null时, WeakHashMap中的a将自动被回收掉。出现这个状况的原因是, 对于a对象而言, 当HashMap remove掉并且将a指向null后, 除了WeakHashMap中还保存a外已经没有指向a的指针了, 所以WeakHashMap会自动舍弃掉a, 而对于b对象虽然指向了null, 但HashMap中还有指向b的指针

实际对象的弱引用, 这意味着只要该对象没有被强对象引用就有可能被垃圾回收机制回收对应的Key-value

```
/**
 * WeakHashMap只是保留对象的弱引用
 * @author Administrator
 */
public class WeakHashMapTest {
    public static void main(String[] args) {
        WeakHashMap whm= new WeakHashMap();
        //三个key-value中的key 都是匿名对象, 没有强引用指向该实际对象
        whm.put(new String("语文"),new String("优秀"));
        whm.put(new String("数学"), new String("及格"));
        whm.put(new String("英语"), new String("中等"));
        //增加一个字符串的强引用
        whm.put("java", new String("特别优秀"));
        System.out.println(whm);
        //通知垃圾回收机制来进行回收
        System.gc();
        System.runFinalization();
        //再次输出whm
        System.out.println("第二次输出whm:"+whm);
    }
}
```

```
}
```

```
}
```

```
第二次输出whm:{java=特别优秀}
```