

[首页](#) [ANDROID](#) [互联网](#) [杂乱无章](#) [科技资讯](#) [程序员人生](#) [程序员笑话](#) [编程技术](#) [网址导航](#)

5.2.1 Fragment实例精讲——底部导航栏的实现(方法1)

分类 [Android 基础入门教程](#)

Android 基础入门教程(Q群号 : 153836263)

[1.0 Android基础入门教程](#)

[1.0.1 2015年最新Android基...](#)

[1.1 背景相关与系统架构分析](#)

[1.2 开发环境搭建](#)

[1.2.1 使用Eclipse + ADT + S...](#)

[1.2.2 使用Android Studio开...](#)

[1.3 SDK更新不了问题解决](#)

[1.4 Genymotion模拟器安装](#)

[1.5.1 Git使用教程之本地仓...](#)

[1.5.2 Git之使用GitHub搭建...](#)

[1.6 .9\(九妹\)图片怎么玩](#)

[1.7 界面原型设计](#)

[1.8 工程相关解析\(各种文件...](#)

[1.9 Android程序签名打包](#)

[1.11 反编译APK获取代码&...](#)

[2.1 View与ViewGroup的概念](#)

[2.2.1 LinearLayout\(线性布局\)](#)

[2.2.2 RelativeLayout\(相对布...](#)

[2.2.3 TableLayout\(表格布局\)](#)

[2.2.4 FrameLayout\(帧布局\)](#)

[2.2.5 GridLayout\(网格布局\)](#)

[2.2.6 AbsoluteLayout\(绝对...](#)

[2.3.1 TextView\(文本框\)详解](#)

[2.3.2 EditText\(输入框\)详解](#)

[2.3.3 Button\(按钮\)与ImageB...](#)

[2.3.4 ImageView\(图像视图\)](#)

[2.3.5.RadioButton\(单选按钮...](#)

[2.3.6 开关按钮ToggleButton...](#)

[2.3.7 ProgressBar\(进度条\)](#)

[2.3.8 SeekBar\(拖动条\)](#)

[2.3.9 RatingBar\(星级评分条\)](#)

[2.4.1 ScrollView\(滚动条\)](#)

本节引言：

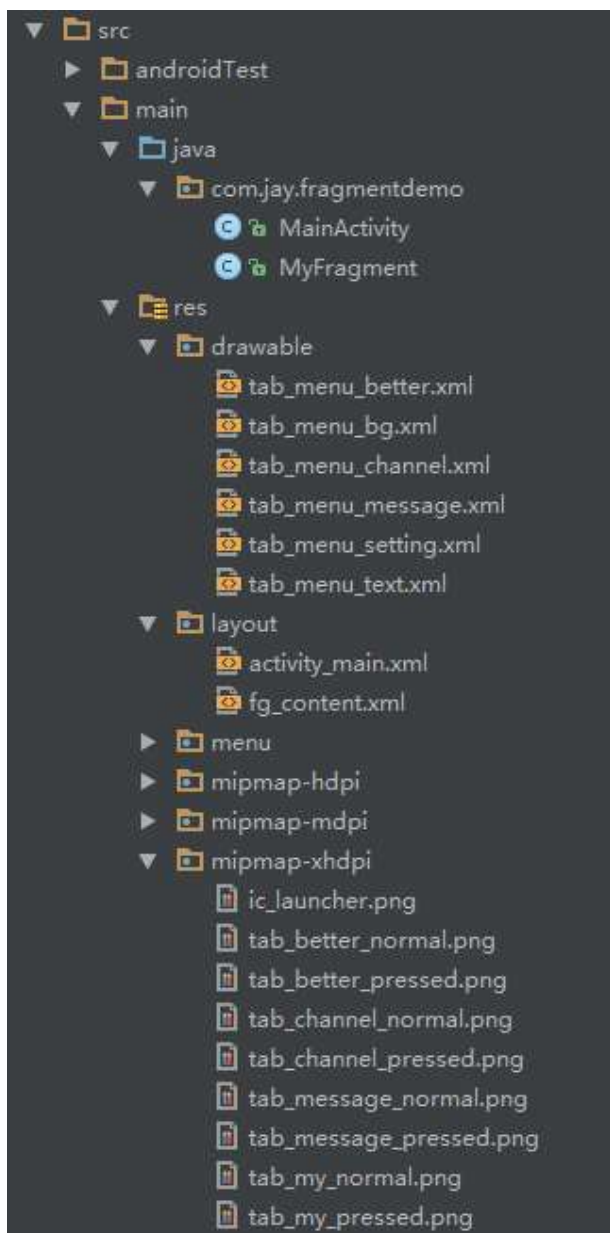
在上一节中我们对Fragment进行了一个初步的了解，学习了概念，生命周期，Fragment管理与Fragment事务，以及动态与静态加载Fragment。从本节开始我们会讲解一些Fragment在实际开发中的一些实例！而本节给大家讲解的是底部导航栏的实现！而基本的底部导航栏方法有很多种，比如全用TextView做，或者用RadioButton，又或者使用TabLayout + RadioButton，当然复杂的情况还是得走外层套布局的方法！本节我们用TextView来做一个底部导航栏的效果，也熟悉下Fragment的使用！好的，开始本节内容！

1.要实现的效果图以及工程目录结构：

先看看效果图吧：



接着看看我们的工程的目录结构：



2.实现流程：

Step 1：写下底部选项的一些资源文件

我们从图上可以看到，我们底部的每一项点击的时候都有不同的效果是吧！我们是通过是否selected来判定的！我们要写的资源文件有：首先是图片，然后是文字，接着是背景！

图片Drawable资源：**tab_menu_channel.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@mipmap/tab_channel_pressed" android:state_selected="true" />
    <item android:drawable="@mipmap/tab_channel_normal" />
</selector>
```

- 2.4.2 Date & Time组件(上)
- 2.4.3 Date & Time组件(下)
- 2.4.4 Adapter基础讲解
- 2.4.5 ListView简单实用
- 2.4.6 BaseAdapter优化
- 2.4.7 ListView的焦点问题
- 2.4.8 ListView之checkbox错...
- 2.4.9 ListView的数据更新问题
- 2.5.0 构建一个可复用的自定...
- 2.5.1 ListView Item多布局的...
- 2.5.2 GridView(网格视图)的...
- 2.5.3 Spinner(列表选项框)...
- 2.5.4 AutoCompleteTextVie...
- 2.5.5 ExpandableListView(...
- 2.5.6 ViewPager(翻转视图)...
- 2.5.7 Toast(吐司)的基本使用
- 2.5.8 Notification(状态栏通...
- 2.5.9 AlertDialog(对话框)详解
- 2.6.0 其他几种常用对话框基...
- 2.6.1 PopupWindow(悬浮框...
- 2.6.2 菜单(Menu)
- 2.6.3 ViewPager的简单使用
- 2.6.4 DrawerLayout(官方侧...
- 3.1.1 基于监听的事件处理机制
- 3.2 基于回调的事件处理机制
- 3.3 Handler消息传递机制浅析
- 3.4 TouchListener PK OnTo...
- 3.5 监听EditText的内容变化
- 3.6 响应系统设置的事件(Co...
- 3.7 AsyncTask异步任务
- 3.8 Gestures(手势)
- 4.1.1 Activity初学乍练
- 4.1.2 Activity初窥门径
- 4.1.3 Activity登堂入室
- 4.2.1 Service初涉
- 4.2.2 Service进阶
- 4.2.3 Service精通
- 4.3.1 BroadcastReceiver牛...
- 4.3.2 BroadcastReceiver庖...
- 4.4.1 ContentProvider初探

其他三个照葫芦画瓢！

文字资源：`tab_menu_text.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/and
roid">
    <item android:color="@color/text_yellow" android:state_sele
cted="true" />
    <item android:color="@color/text_gray" />
</selector>
```

背景资源：`tab_menu_bg.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/and
roid">
    <item android:state_selected="true">
        <shape>
            <solid android:color="#FFC4C4C4" />
        </shape>
    </item>
    <item>
        <shape>
            <solid android:color="@color/transparent" />
        </shape>
    </item>
</selector>
```

Step 2：编写我们的Activity布局

`activity_main.xml`:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/...
es/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <RelativeLayout
        android:id="@+id/ly_top_bar"
        android:layout_width="match_parent"
        android:layout_height="48dp"
        android:background="@color/bg_topbar">

        <TextView
            android:id="@+id/txt_topbar"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_centerInParent="true"
            android:gravity="center"
```

- 4.4.2 ContentProvider再探...
- 4.5.1 Intent的基本使用
- 4.5.2 Intent之复杂数据的传递
- 5.1 Fragment基本概述
- 5.2.1 Fragment实例精讲——...
- 5.2.2 Fragment实例精讲——...
- 5.2.3 Fragment实例精讲——...
- 5.2.4 Fragment实例精讲——...
- 5.2.5 Fragment实例精讲——...
- 6.1 数据存储与访问之——文...
- 6.2 数据存储与访问之——S...
- 6.3.1 数据存储与访问之——...
- 6.3.2 数据存储与访问之——...
- 7.1.1 Android网络编程要学...
- 7.1.2 Android Http请求头与...
- 7.1.3 Android HTTP请求方...
- 7.1.4 Android HTTP请求方...
- 7.2.1 Android XML数据解析
- 7.2.2 Android JSON数据解析
- 7.3.1 Android 文件上传
- 7.3.2 Android 文件下载（1）
- 7.3.3 Android 文件下载（2）
- 7.4 Android 调用WebService
- 7.5.1 WebView(网页视图)基...
- 7.5.2 WebView和JavaScrip...
- 7.5.3 Android 4.4后WebVie...
- 7.5.4 WebView文件下载
- 7.5.5 WebView缓存问题
- 7.5.6 WebView处理网页返...
- 7.6.1 Socket学习网络基础准备
- 7.6.2 基于TCP协议的Socket...
- 7.6.3 基于TCP协议的Socket...
- 7.6.4 基于UDP协议的Socke...
- 8.1.1 Android中的13种Draw...
- 8.1.2 Android中的13种Draw...
- 8.1.3 Android中的13种Draw...
- 8.2.1 Bitmap(位图)全解析 P...
- 8.2.2 Bitmap引起的OOM问题
- 8.3.1 三个绘图工具类详解
- 8.3.2 绘图类实战示例

```

        android:textSize="18sp"
        android:textColor="@color/text_topbar"
        android:text="信息"/>

```

<View

```

        android:layout_width="match_parent"
        android:layout_height="2px"
        android:background="@color/div_white"
        android:layout_alignParentBottom="true"/>

```

</RelativeLayout>

<LinearLayout

```

        android:id="@+id/ly_tab_bar"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:layout_alignParentBottom="true"
        android:background="@color/bg_white"
        android:orientation="horizontal">

```

<TextView

```

        android:id="@+id/txt_channel"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="@drawable/tab_menu_bg"
        android:drawablePadding="3dp"
        android:drawableTop="@drawable/tab_menu_channel"
        android:gravity="center"
        android:padding="5dp"
        android:text="@string/tab_menu_alert"
        android:textColor="@drawable/tab_menu_text"
        android:textSize="16sp" />

```

<TextView

```

        android:id="@+id/txt_message"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="@drawable/tab_menu_bg"
        android:drawablePadding="3dp"
        android:drawableTop="@drawable/tab_menu_message"
        android:gravity="center"
        android:padding="5dp"
        android:text="@string/tab_menu_profile"
        android:textColor="@drawable/tab_menu_text"
        android:textSize="16sp" />

```

<TextView

```

        android:id="@+id/txt_better"
        android:layout_width="0dp"
        android:layout_height="match_parent"

```

8.3.3 Paint API之—— Mask...

8.3.4 Paint API之—— Xferm...

8.3.5 Paint API之—— Xferm...

8.3.6 Paint API之—— Xferm...

8.3.7 Paint API之—— Xferm...

8.3.8 Paint API之—— Xferm...

8.3.9 Paint API之—— Color...

8.3.10 Paint API之—— Colo...

8.3.11 Paint API之—— Colo...

8.3.12 Paint API之—— Path...

8.3.13 Paint API之—— Sha...

8.3.14 Paint几个枚举/常量值...

8.3.15 Paint API之—— Type...

8.3.16 Canvas API详解(Part 1)

8.3.17 Canvas API详解(Part...

8.3.18 Canvas API详解(Part...

8.4.1 Android动画合集之帧...

8.4.2 Android动画合集之补...

8.4.3 Android动画合集之属...

8.4.4 Android动画合集之属...

9.1 使用SoundPool播放音...

9.2 MediaPlayer播放音频与...

9.3 使用Camera拍照

9.4 使用MediaRecord录音

10.1 TelephonyManager(电...

10.2 SmsManager(短信管理...

10.3 AudioManager(音频管...

10.4 Vibrator(振动器)

10.5 AlarmManager(闹钟服务)

10.6 PowerManager(电源服...

10.7 WindowManager(窗口...

10.8 LayoutInflater(布局服务)

10.9 WallpaperManager(壁...

10.10 传感器专题(1)——相...

10.11 传感器专题(2)——方...

10.12 传感器专题(3)——加...

10.12 传感器专题(4)——其...

10.14 Android GPS初涉

11.0 《2015最新Android基...

```

        android:layout_weight="1"
        android:background="@drawable/tab_menu_bg"
        android:drawablePadding="3dp"
        android:drawableTop="@drawable/tab_menu_better"
        android:gravity="center"
        android:padding="5dp"
        android:text="@string/tab_menu_pay"
        android:textColor="@drawable/tab_menu_text"
        android:textSize="16sp" />

```

<TextView

```

        android:id="@+id/txt_setting"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="@drawable/tab_menu_bg"
        android:drawablePadding="3dp"
        android:drawableTop="@drawable/tab_menu_setting"
        android:gravity="center"
        android:padding="5dp"
        android:text="@string/tab_menu_setting"
        android:textColor="@drawable/tab_menu_text"
        android:textSize="16sp"/>

```

</LinearLayout>

<View

```

        android:id="@+id/div_tab_bar"
        android:layout_width="match_parent"
        android:layout_height="2px"
        android:background="@color/div_white"
        android:layout_above="@id/ly_tab_bar"/>

```

<FrameLayout

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/ly_top_bar"
        android:layout_above="@id/div_tab_bar"
        android:id="@+id/ly_content">

```

</FrameLayout>

</RelativeLayout>

代码解析：

首先定义顶部标题栏的样式，48dp的LinearLayout中间加上一个TextView作为标题！

接着定义一个大小为56dp的LinearLayout对其底部，在这个里面加入四个TextView，比例1:1:1:1，并且设置相关属性，接着在这个LinearLayout上加一条线段！

最后以标题栏和底部导航栏为边界，写一个FrameLayout，宽高

match_parent , 用做Fragment的容器！

PS : 这里四个TextView属性是重复的，你也可以自行抽取出来，编写一个style，设置下~

Step 3 : 隐藏顶部导航栏

意外发现以前的在Activity中调

用requestWindowFeature(Window.FEATURE_NO_TITLE);可以隐藏手机 自带顶部导航栏，但是写demo时候发现会报错，即使这句话写在了 setContentView()之前！可能是因为 继承的是AppCompatActivity而非 Activity类！

当然以前的getSupportActionBar().hide()隐藏掉ActionBar，但是他还是会在界面上！ 最后还有一种方法就是自己编写一个style，然后在

AndroidManifest.xml中为Application设置这个Theme：

接着AndroidManifest.xml设置下theme属性：

```
android:theme="@style/Theme.AppCompat.NoActionBar"
```

PS : 上述"良心代码"由好程序员曹神赞助~

Step 4 : 创建一个Fragment的简单布局与类：

fg_content.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    android:orientation="vertical" android:layout_width="match_
parent"
    android:layout_height="match_parent"
    android:background="@color/bg_white">

    <TextView
        android:id="@+id/txt_content"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:text="呵呵呵"
        android:textColor="@color/text_yellow"
        android:textSize="20sp"/>

</LinearLayout>
```

MyFragment.java:

```
/**
 * Created by Coder-pig on 2015/8/28 0028.
 */
```

```

public class MyFragment extends Fragment {

    private String content;

    public MyFragment(String content) {
        this.content = content;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fg_content, contain
er, false);
        TextView txt_content = (TextView) view.findViewById(R.i
d.txt_content);
        txt_content.setText(content);
        return view;
    }
}

```

代码解析：

就是简单的重写了一个onCreateView()方法，其他方法可以按需重写!

Step 5：编写MainActivity.java

先说说我们要考虑的一些关键问题：

Fragment什么时候初始化和add到容器中？什么时候hide和show？

如何让TextView被选中？选中一个TextView后，要做一些什么操作？

刚进入MainActivity怎么样让一个TextView处于Selected的状态？

嗯，接下来——解答上面这些问题：

我们选中TextView后对对应的Fragment进行判空，如果为空，初始化，并添加到容器中；而hide的话，我们定义一个方法hide所有的Fragment，每次触发点击事件就先调用这个hideAll方法，讲所有Fragment隐藏起来，然后如果TextView对应的Fragment不为空，我们就将这个Fragment显示出来；

这个我们通过点击事件来实现，点击TextView后先重置所有TextView的选中状态为false，然后设置点击的TextView的选中状态为true；

这个更简单，我们是通过点击事件来设置选中的，那么在onCreate()方法里加个触发点击事件的方法不就可以了嘛~ `txt_channel.performClick();`

逻辑都弄懂了，直接上代码咯：

MainActivity.java:

```

/**
 * Created by Coder-pig on 2015/8/28 0028.
 */
public class MainActivity extends AppCompatActivity implements

```

```

View.OnClickListener{

    //UI Object
    private TextView txt_topbar;
    private TextView txt_channel;
    private TextView txt_message;
    private TextView txt_better;
    private TextView txt_setting;
    private FrameLayout ly_content;

    //Fragment Object
    private MyFragment fg1,fg2,fg3,fg4;
    private FragmentManager fManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_main);
        fManager = getFragmentManager();
        bindViews();
        txt_channel.performClick();    //模拟一次点击，既进去后选择
        第一项
    }

    //UI组件初始化与事件绑定
    private void bindViews() {
        txt_topbar = (TextView) findViewById(R.id.txt_topbar);
        txt_channel = (TextView) findViewById(R.id.txt_channel)
;
        txt_message = (TextView) findViewById(R.id.txt_message)
;
        txt_better = (TextView) findViewById(R.id.txt_better);
        txt_setting = (TextView) findViewById(R.id.txt_setting)
;
        ly_content = (FrameLayout) findViewById(R.id.ly_content
);

        txt_channel.setOnClickListener(this);
        txt_message.setOnClickListener(this);
        txt_better.setOnClickListener(this);
        txt_setting.setOnClickListener(this);
    }

    //重置所有文本的选中状态
    private void setSelected(){
        txt_channel.setSelected(false);
        txt_message.setSelected(false);
        txt_better.setSelected(false);
        txt_setting.setSelected(false);
    }

    //隐藏所有Fragment

```



```
private void hideAllFragment(FragmentTransaction fragmentTransaction){
    if(fg1 != null) fragmentTransaction.hide(fg1);
    if(fg2 != null) fragmentTransaction.hide(fg2);
    if(fg3 != null) fragmentTransaction.hide(fg3);
    if(fg4 != null) fragmentTransaction.hide(fg4);
}

@Override
public void onClick(View v) {
    FragmentTransaction fTransaction = fManager.beginTransaction();
    hideAllFragment(fTransaction);
    switch (v.getId()){
        case R.id.txt_channel:
            setSelected();
            txt_channel.setSelected(true);
            if(fg1 == null){
                fg1 = new MyFragment("第一个Fragment");
                fTransaction.add(R.id.ly_content, fg1);
            }else{
                fTransaction.show(fg1);
            }
            break;
        case R.id.txt_message:
            setSelected();
            txt_message.setSelected(true);
            if(fg2 == null){
                fg2 = new MyFragment("第二个Fragment");
                fTransaction.add(R.id.ly_content, fg2);
            }else{
                fTransaction.show(fg2);
            }
            break;
        case R.id.txt_better:
            setSelected();
            txt_better.setSelected(true);
            if(fg3 == null){
                fg3 = new MyFragment("第三个Fragment");
                fTransaction.add(R.id.ly_content, fg3);
            }else{
                fTransaction.show(fg3);
            }
            break;
        case R.id.txt_setting:
            setSelected();
            txt_setting.setSelected(true);
            if(fg4 == null){
                fg4 = new MyFragment("第四个Fragment");
                fTransaction.add(R.id.ly_content, fg4);
            }else{
                fTransaction.show(fg4);
            }
    }
}
```

```
        break;
    }
    fTransaction.commit();
}
}
```

3.代码下载：

FragmentDemo.zip：[FragmentDemo.zip 下载](#) 声明：图片素材来自 App：**better**，本代码只做演示，并无用于商业用途！

4.本节小结

本节给大家讲解了如何使用一个LinarLayout + 四个TextView 实现一个底部导航栏以及 Fragment add，hide，show的逻辑~还是蛮简单的，最后要感谢小猪秘密基地的基神，B神，还有好程序员曹神给我的一些指点！万分感谢，仅以此篇纪念小猪重返装逼界，嗯，重返应用层，嘿嘿，本节就到这里，谢谢~



<div>在线实例</div> <ul style="list-style-type: none">· HTML 实例· CSS 实例· JavaScript 实例· Ajax 实例· jQuery 实例· XML 实例· Java 实例	<div>字符集&工具</div> <ul style="list-style-type: none">· HTML 字符集设置· HTML ASCII 字符集· HTML ISO-8859-1· HTML 实体符号· HTML 拾色器· JSON 格式化工具	<div>最新更新</div> <ul style="list-style-type: none">· TIOBE 2015 年12...· Swift 正式开源· PHP 7 正式发布· Shell 编程快速入门· Shell 文件包含· Shell 输入/输出...· Shell printf 命令	<div>站点信息</div> <ul style="list-style-type: none">· 意见反馈· 免责声明· 关于我们· 文章归档	<div>关注微信</div> <div></div> <div>Copyright © 2013-2015 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1</div>
---	---	--	---	--