

用法：类似于ListView

1、布局声明

```
<!--轮播控件-->
<android.support.v4.view.ViewPager
    android:id="@+id/vp_guidedvp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</android.support.v4.view.ViewPager>
```

2、代码中获取控件并设置数据

```
mVpGuide = (ViewPager) findViewById(R.id.vp_guidedvp);
```

设置数据

```
private static final int[] IMG_GUIDE = {R.drawable.guide_1, R.drawable.guide_2, R.drawable.guide_3};
private List<ImageView> IMGARR = new ArrayList<>();

for (int i = 0; i < IMG_GUIDE.length ; i++) {
    //创建图片对象
    ImageView imageView = new ImageView(getApplicationContext());
    //设置图片对象的数据
    imageView.setImageResource(IMG_GUIDE[i]);
    //将图片对象添加到集合
    IMGARR.add(imageView);
}
//设置数据到ViewPager里
mMyAdapter = new MyAdapter();
mVpGuide.setAdapter(mMyAdapter);
```

创建数据适配器

```
class MyAdapter extends PagerAdapter{
    @Override
    public int getCount() {
        //获取条目总数
        return IMG_GUIDE.length;
    }
    @Override
    public boolean isViewFromObject(View view, Object object) {
        return view==object;
    }
    @Override
    public Object instantiateItem(ViewGroup container, int position) {
        //获取集合中数据
        ImageView imageView = IMGARR.get(position);
        //将获取到的数据设置到ViewPager中去
        container.addView(imageView);
        //返回ViewPager显示
        return imageView;
    }
    @Override
    public void destroyItem(ViewGroup container, int position, Object object) {
        container.removeView((View) object);
    }
}
```

控制左右滑动，写 ViewPager的滑动监听方法

```

mVpGuide.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
    @Override    //这个方法会在屏幕滚动过程中不断被调用
    //position和当前页面index是一致的
    //positionOffset是当前页面滑动比例，如果页面向右翻动，这个值不断变大，最后在趋近1的情况后突变为0。如果页面向左
    //positionOffsetPixels是当前页面滑动像素，变化情况和positionOffset一致。
    public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {
        //获取小红点在滑动的过程中的实际值
        float v = mIvRedSlide*(position + positionOffset);
        //获取布局参数
        FrameLayout.LayoutParams params = (FrameLayout.LayoutParams) mIvRed.getLayoutParams();
        //设置布局参数设置小红点的移动值
        params.leftMargin = (int) (v+0.5f);
        //请求布局重新给小红点布局
        mIvRed.getParent().requestLayout();
    }
    //参数position，代表哪个页面被选中当用手指滑动翻页的时候，如果翻动成功了（滑动的距离够长）
    //手指抬起来就会立即执行这个方法，position就是当前滑动到的页面
    @Override
    public void onPageSelected(int position) {
        if(position==(IMG_GUIDE.length-1)){
            mButGuide.setVisibility(View.VISIBLE);
        }else{
            mButGuide.setVisibility(View.GONE);
        }
    }
    @Override
    public void onPageScrollStateChanged(int state) {
    }
});
mIvRed.getViewTreeObserver().addOnGlobalLayoutListener(new ViewTreeObserver.OnGlobalLayoutListener() {
    @Override
    public void onGlobalLayout() {
        mIvRed.getViewTreeObserver().removeOnGlobalLayoutListener(this);
        //获取小红点滑动的距离（两个白点之间的距离）
        mIvRedSlide = mLinear.getChildAt(1).getLeft() - mLinear.getChildAt(0).getLeft();
    }
});
}

```

该滑动事件案例模型图

