Layout manager that allows the user to flip（用指轻弹）left and right through pages of data. You supply（提供一个PagerAdapter的实现类）an implementation of a [PagerAdapter](file:///E:\android\adt-bundle-windows-x86-20131030\sdk\docs\reference\android\support\v4\view\PagerAdapter.html) to generate the pages that the view shows.

Note this class is currently under early design and development. The API will likely change in later updates of the compatibility library, requiring changes to the source code of apps when they are compiled against the newer version.

ViewPager is most often used in conjunction(结合) with [Fragment](file:///E:\android\adt-bundle-windows-x86-20131030\sdk\docs\reference\android\app\Fragment.html), which is a convenient way to supply and manage the lifecycle of each page. There are standard adapters implemented for using fragments with the ViewPager, which cover the most common use cases. These are [FragmentPagerAdapter](file:///E:\android\adt-bundle-windows-x86-20131030\sdk\docs\reference\android\support\v4\app\FragmentPagerAdapter.html) and [FragmentStatePagerAdapter](file:///E:\android\adt-bundle-windows-x86-20131030\sdk\docs\reference\android\support\v4\app\FragmentStatePagerAdapter.html); each of these classes have simple code showing how to build a full user interface with them.

ViewPager与actionbar结合会有更复杂的用法。

Here is a more complicated example of ViewPager, using it in conjuction with [ActionBar](file:///E:\android\adt-bundle-windows-x86-20131030\sdk\docs\reference\android\app\ActionBar.html) tabs. You can find other examples of using ViewPager in the API 4+ Support Demos and API 13+ Support Demos sample code.

public class ActionBarTabsPager extends Activity {  
    ViewPager mViewPager;  
    TabsAdapter mTabsAdapter;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        mViewPager = new ViewPager(this);  
        mViewPager.setId(R.id.pager);  
        setContentView(mViewPager);  
  
        final ActionBar bar = getActionBar();  
        bar.setNavigationMode(ActionBar.NAVIGATION\_MODE\_TABS);  
        bar.setDisplayOptions(0, ActionBar.DISPLAY\_SHOW\_TITLE);  
  
        mTabsAdapter = new TabsAdapter(this, mViewPager);  
        mTabsAdapter.addTab(bar.newTab().setText("Simple"),  
                CountingFragment.class, null);  
        mTabsAdapter.addTab(bar.newTab().setText("List"),  
                FragmentPagerSupport.ArrayListFragment.class, null);  
        mTabsAdapter.addTab(bar.newTab().setText("Cursor"),  
                CursorFragment.class, null);  
  
        if (savedInstanceState != null) {  
            bar.setSelectedNavigationItem(savedInstanceState.getInt("tab", 0));  
        }  
    }  
  
    @Override  
    protected void onSaveInstanceState(Bundle outState) {  
        super.onSaveInstanceState(outState);  
        outState.putInt("tab", getActionBar().getSelectedNavigationIndex());  
    }  
  
    /\*\*  
     \* This is a helper class that implements the management of tabs and all  
     \* details of connecting a ViewPager with associated TabHost.  It relies on a  
     \* trick.  Normally a tab host has a simple API for supplying a View or  
     \* Intent that each tab will show.  This is not sufficient for switching  
     \* between pages.  So instead we make the content part of the tab host  
     \* 0dp high (it is not shown) and the TabsAdapter supplies its own dummy  
     \* view to show as the tab content.  It listens to changes in tabs, and takes  
     \* care of switch to the correct paged in the ViewPager whenever the selected  
     \* tab changes.  
     \*/  
    public static class TabsAdapter extends FragmentPagerAdapter  
            implements ActionBar.TabListener, ViewPager.OnPageChangeListener {  
        private final Context mContext;  
        private final ActionBar mActionBar;  
        private final ViewPager mViewPager;  
        private final ArrayList<TabInfo> mTabs = new ArrayList<TabInfo>();  
  
        static final class TabInfo {  
            private final Class<?> clss;  
            private final Bundle args;  
  
            TabInfo(Class<?> \_class, Bundle \_args) {  
                clss = \_class;  
                args = \_args;  
            }  
        }  
  
        public TabsAdapter(Activity activity, ViewPager pager) {  
            super(activity.getFragmentManager());  
            mContext = activity;  
            mActionBar = activity.getActionBar();  
            mViewPager = pager;  
            mViewPager.setAdapter(this);  
            mViewPager.setOnPageChangeListener(this);  
        }  
  
        public void addTab(ActionBar.Tab tab, Class<?> clss, Bundle args) {  
            TabInfo info = new TabInfo(clss, args);  
            tab.setTag(info);  
            tab.setTabListener(this);  
            mTabs.add(info);  
            mActionBar.addTab(tab);  
            notifyDataSetChanged();  
        }  
  
        @Override  
        public int getCount() {  
            return mTabs.size();  
        }  
  
        @Override  
        public Fragment getItem(int position) {  
            TabInfo info = mTabs.get(position);  
            return Fragment.instantiate(mContext, info.clss.getName(), info.args);  
        }  
  
        @Override  
        public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {  
        }  
  
        @Override  
        public void onPageSelected(int position) {  
            mActionBar.setSelectedNavigationItem(position);  
        }  
  
        @Override  
        public void onPageScrollStateChanged(int state) {  
        }  
  
        @Override  
        public void onTabSelected(Tab tab, FragmentTransaction ft) {  
            Object tag = tab.getTag();  
            for (int i=0; i<mTabs.size(); i++) {  
                if (mTabs.get(i) == tag) {  
                    mViewPager.setCurrentItem(i);  
                }  
            }  
        }  
  
        @Override  
        public void onTabUnselected(Tab tab, FragmentTransaction ft) {  
        }  
  
        @Override  
        public void onTabReselected(Tab tab, FragmentTransaction ft) {  
        }  
    }  
}