# setOffscreenPageLimit方法

1. 官方文档：

void setOffscreenPageLimit (int limit)

文档说明：

Set the number of pages that should be retained to either side of the current page in the view hierarchy in an idle state. Pages beyond this limit will be recreated from the adapter when needed.（大意：设置预加载N个页面）

百度翻译：

设置应保留在空闲状态的视图层次结构中的当前页的任何一方的页数。

Google翻译：

设置的该应保留在视图层级中的当前页面的任一侧处于空闲状态的页数。

详细介绍：

This is offered as an optimization. If you know in advance the number of pages you will need to support or have lazy-loading mechanisms in place on your pages, tweaking this setting can have benefits in perceived smoothness of paging animations and interaction. If you have a small number of pages (3-4) that you can keep active all at once, less time will be spent in layout for newly created view subtrees as the user pages back and forth.

Google翻译：

这是提供作为优化。如果你事先知道的页数，你需要的地方在您的网页支持或有延迟加载机制，调整这个设置可以在寻呼动画和互动感知的平滑收益。如果你有一小部分，你可以保持活跃一下子页（3-4），更少的时间将在布局新创建的视图子树作为用户页面来回花了。

**注意：**

You should keep this limit low, especially if your pages have complex layouts. This setting defaults to 1.

Google翻译 ：

**您应该保持此限制较低，特别是如果你的网页有复杂的布局。此设置默认为1。**

**参数：**

|  |  |
| --- | --- |
| Parameters | |
| limit | int: How many pages will be kept offscreen in an idle state. |

1. **关系**：

# ViewPager

public class ViewPager   
extends [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup.html)

|  |  |  |  |
| --- | --- | --- | --- |
| [java.lang.Object](https://developer.android.com/reference/java/lang/Object.html) | | | |
| ↳ | [android.view.View](https://developer.android.com/reference/android/view/View.html) | | |
|  | ↳ | [android.view.ViewGroup](https://developer.android.com/reference/android/view/ViewGroup.html) | |
|  |  | ↳ | android.support.v4.view.ViewPager |

1. 源码：

private static final int *DEFAULT\_OFFSCREEN\_PAGES* = 1;//默认加载数

public void setOffscreenPageLimit(int limit) {  
 if (limit < *DEFAULT\_OFFSCREEN\_PAGES*) {  
 Log.*w*(*TAG*, "Requested offscreen page limit " + limit + " too small; defaulting to " +  
 *DEFAULT\_OFFSCREEN\_PAGES*);  
 limit = *DEFAULT\_OFFSCREEN\_PAGES*;  
 }  
 if (limit != mOffscreenPageLimit) {  
 mOffscreenPageLimit = limit;  
 populate();  
 }  
}

private int mCurItem; // Index of currently displayed page.

void populate() {  
 populate(mCurItem);  
}