

**使用 1.17 版本的 flowfram 宏包为报纸，小册子或杂志
创建流框**

**Creating Flow Frames for Posters,
Brochures or Magazines using flowfram.sty version 1.17**

原作者 Nicola L. C. Talbot / 2014-9-30

译 玻璃 / 2020-4-13

Dr Nicola Talbot

Dickimaw Books

<http://www.dickimaw-books.com/>

1 介绍	1
1.1 宏包选项	1
1.2 浮动体	2
1.3 draft 选项	2
1.4 章	3
1.5 框架堆叠顺序	3
1.6 HTML	4
2 定义新的框架	5
2.1 流框架	5
2.1.1 提前结束流框架	6
2.2 静态框架	7
2.2.1 重要注意事项	10
2.3 动态框架	10
2.3.1 将章标题置于动态框架中	12
2.3.2 在动态框架中放置页眉和页脚	13
2.3.3 续文	13
2.3.4 重要注意事项	14
3 修改框架属性	17
3.1 非矩形框架	23
3.2 即时打开和关闭框架	25
4 位置和尺寸	31
4.1 确定版心的位置	31
4.2 确定框架的尺寸和位置	32
4.3 相对位置	32
5 预定义布局	35
5.1 栏样式	35
5.2 带有附加框架的栏样式	35
5.3 从右到左的栏	40

目 录

5.4	背景特效	40
5.4.1	垂直条效果	41
5.4.2	水平条效果	41
5.4.3	背景框架	42
5.4.4	垂直和水平线	42
6	缩略图和小目录	45
6.1	缩略图	45
6.2	小型目录	47
7	全局设置	49
7.1	宏	49
7.2	长度	49
7.3	计数器	51
8	故障排除	53
8.1	一般查询	53
8.2	意外的输出	54
8.3	错误消息	56
	词汇表	59

本文档是 `flowfram` 宏包的用户手册。想要进一步了解软件包细节的高级用户应该阅读 `flowfram.pdf`。示例文件位于路径 $\langle\text{TEXMF}\rangle/\text{doc}/\text{latex}/\text{flowfram}/\text{samples}/$ 下, 其中 $\langle\text{TEXMF}\rangle$ 指此包的 $\text{T}_{\text{E}}\text{X}$ 根目录。(本文件位于 $\langle\text{TEXMF}\rangle/\text{doc}/\text{latex}/\text{flowfram}/$, 执行 `texdoc ffuserguide` 命令可调出本文件。)

`flowfram` 是 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 的宏包, 用于在文档中创建文本框架, 以便文档环境的内容按照定义顺序从一个框架流向下一个框架。这对创建不符合标准一栏或两栏布局的海报或杂志或任何其他形式的文档特别有用。如果你希望使用图形用户界面设置文档布局, 则有一个 `flowframtk`¹ 的可选辅助应用程序。

`flowfram` 包试图让 $\text{T}_{\text{E}}\text{X}$ 做一些它最初设计不想做的事情。它会修改输出例程, 可能并不总是按需要执行。由于 $\text{T}_{\text{E}}\text{X}$ 输出例程的异步特性, 对段落跨度不等的框架, 必须格外小心。(见 8.2 节)

`flowfram` 包提供三种由用户指定尺寸和位置的框架: 流框架、静态框架和动态框架²。文档环境的主要内容按定义顺序从一个流框架流向下一个流框架, 而静态和动态框架的内容则使用第 3 章中描述的命令进行显式设置。请注意, 除非另有说明, 否则所有坐标都是相对版心 (`typeblock`, `text frame`) 左下角的坐标。如果有双面文档, 则根据 `\oddsidemargin` 和 `\evensidemargin` 的值, 版心的位置可能会有所不同, 并且除非另有说明, 否则所有框架都将相应地移动。

此程序包仅使用有限数量的类文件和程序包进行了测试。因为它修改了输出例程, 所以它很可能与其它任何也执行此操作的包 (例如 `longtable`) 冲突。

你应该在 `hyperref` 和任何颜色包 (例如 `color`) 之后加载 `flowfram`。

1.1 宏包选项

pages 确定页列表是指由页计数器给定的页码(`pages=relative`)还是指绝对页码(`pages=absolute`)。默认是 `relative`, 以确保向后兼容, 但如果你有一个重置页面计数器的文档, 则最好使用 `pages=absolute`。

draft 开启草稿模式 (见 1.3 节)。

final 关闭草稿模式 (默认)。

thumbtabs 控制目录缩略图, 到 6.1 节中查看细节。

¹<http://www.dickimaw-books.com/apps/flowframtk/>。
²想自定义框架形状? 查看 3.1。

1. 介绍

1.1	宏包选项	1
1.2	浮动体	2
1.3	draft 选项	2
1.4	章	3
1.5	框架堆叠顺序	3
1.6	HTML	4

本章简要介绍了包, 包选项和各种框架类型。

LR 使用第 5.1 节中描述的列样式布局时，从左到右定义流框架。

RL 使用第 5.1 节中描述的列样式布局时，从右到左定义流框架。

rotate 可具有 **true**（可在缩略图中旋转文本）或 **false**（在缩略图中堆叠文本）值，默认是 **true**。

color 可具有 **true**（允许框架具有颜色设置）或 **false**（禁用框架设置中的颜色），默认值为 **true**。

verbose 可具有 **true** 或 **false** 值，默认是 **false**，提供用以协助调试。

1.2 浮动体

标准的 **figure** 和 **table** 环境在流框架中的行为与往常一样，它们的星型版本 **figure*** 和 **table*** 的行为与 **figure** 和 **table** 没有区别。³

浮动体（如图和表）只能在流框架中进行。但是，这个包提供了额外的环境：**staticfigure** 和 **statictable**，可在静态框架和动态框架中使用。与它们的图和表对应项不同，它们是固定的，因此不采用可选的位置说明符。**\caption** 和 **\label** 命令可以像往常一样在 **staticfigure** 和 **statictable** 中使用，但是请记住，如果框架显示在多个页面上，那么最终可能会出现多个定义的标签。

1.3 draft 选项

flowfram 包具有选项 **draft**，该选项将为已定义的每个框架绘制边界框。在每个边界框的右下角（除了表示类型块的边界框），将显示一个标记，格式为 “[*T*):*idn*];*idl*]”，其中 *T* 是表示框架类型的首字母，*idn* 是框架的标识号 (IDN)，*idl* 是该框架的标识标签 (IDL)。(*T*) 的值为：**F**（流框架）、**S**（静态框架）或 **D**（动态框架）。标记 “[*M*):*idn*]” 表示的边界框是 IDN 为 *idn* 的流框边注占用的区域。请注意，即使框架已旋转，边界框也不会旋转。

如果要显示或隐藏特定类型的边界框，可以使用以下命令之一：

- **\showtypeblocktrue** 显示版心的边界框。

³这是因为流框架的布局是任意。

- `\showtypeblockfalse` 不显示版心的边界框。
- `\showmarginstrue` 显示边注的边框。
- `\showmarginsfalse` 不显示边注的边框。
- `\showframebboxtrue` 显示框架边界框。
- `\showframebboxfalse` 不显示框架边界框。

可以使用以下命令查看当前页的布局（无论是否设置了 `draft` 选项）：

```
\flowframshowlayout
```

`flowfram` 包还为无法处理颜色或旋转特殊对象的预览器提供了 `color=false` 和 `rotate=false` 选项。（否则，最终可能会出现黑色大矩形而不是你希望的浅背景颜色，从而使文本变得模糊）

1.4 章

如果定义了 `\chapter` 命令，`flowfram` 包将修改其定义，以便为每个章节的第一页将页面样式设置为 `\chapterfirstpagestyle`。此命令默认为 `|plain|`，这是章节第一页的常用页面样式。如果要使用其他页面样式，则需要将 `\chapterfirstpagestyle` 重定义为相关页面样式的名称。在调用 `\clearpage` 和 `\cleardoublepage` 之前，章节定义的开头使用了

```
\ffprechapterhook
```

章节标题可放置在动态框架中（如本文档所示），详见 2.3.1 小节。

1.5 框架堆叠顺序

每页上的内容按以下顺序排列：

1. 该页定义的每个静态框架按 IDN 升序。
2. 该页定义的每个流框架按 IDN 升序。

3. 该页定义每个动态框架按 IDN 升序。

4. 使用了“draft”宏包选项的边框。

此顺序可用于确定是否要将页面上的其它内容覆盖或作为参考底图。请注意，框架之间不会相互作用。如果有两个或多个重叠的框架，则每个框架的文本不会尝试环绕其它框架，而只是覆盖它们⁴。

1.6 HTML

`flowfram` 包现在附带了一个 \LaTeX 2HTML 样式的文件 `flowfram.perl`。但是，此样式文件不是用来模拟 `flowfram` 包，而是为了方便从 \LaTeX 源创建纯 HTML 文档。所有与框架相关的信息都忽略。默认情况下，将忽略任何静态或动态框架的内容，但可以使用

```
\HTMLset{showstaticcontents}{1}
```

来显示静态框架的内容，或使用

```
\HTMLset{showdynamiccontents}{1}
```

来显示动态框架的内容（在 `html` 包中定义了 `\HTMLset`）。请注意，这会将文本放置在文档中设置内容的位置。此样式文件不创建 HTML 框架。因此，它可以用于创建 PDF 文档的可访问版本，例如此文档的 HTML 版本 `ffuserguide.HTML`。

⁴可以有任意形状的框架吗？见 3.1 节

2.1 流框架

流动框架是框架的必要及主要类型，文档环境的文本将按定义顺序从一个框架流向下一个框架。每个流框都有其宽度、高度、在页面上的位置，以及可选的边框。要定义一个新的流框架使用

```
\newflowframe[⟨页面列表⟩]{⟨宽度⟩}{⟨高度⟩}{⟨x 坐标⟩}{⟨y 坐标⟩}[⟨label⟩]
```

其中 ⟨宽度⟩ 是框架的宽度，⟨高度⟩ 是框架的高度，(x,y) 是该块左下角相对于版心左下角的位置¹。第一个可选参数 ⟨页面列表⟩ 指这个框架出现的页面，第二个可选参数 ⟨label⟩ 指框架的标识标签 IDL。

页面列表可以由关键字指定：all、odd、even 或 none，也可以由单独页码或页面范围的逗号分隔列表指定。如果省略 ⟨页面列表⟩，则假定为 all。页面范围可以是封闭范围（例如 2-8）或开放范围（例如 <10 或 >5）。例如：<3,5,7-11,>15 表示第 1、2、5、7、8、9、10、11 页以及大于第 15 页的所有页。默认情况下，这些页码是指页面计数器²的整数值，因此如果有一个页面 i 和一个页面 1 或一个页面 a，它们都将具有相同的布局（除非在两个页面之间的某个位置更改了页面列表设置）。

从版本 1.4 开始，如果使用宏包选项 pages=absolute，那么页面列表中的数字就是绝对页码。在这种情况下，第 1 页仅指文档的第一页，而不管文档中是否有其他第 1 页或第 i 页。

每个框架都有自己的唯一 IDN，与定义它的顺序相对应。因此，要定义的第一个流框架具有 IDN 1，第二个流框架具有 IDN 2，依此类推。然后，当你要修改框架的设置时，可以使用此编号来标识框架。或者，可以使用最后一个可选参数 ⟨label⟩ 为框架指定唯一的 IDL。

可以使用以下命令从给定流框架的 IDN 中检索给定流框架的 IDL：

```
\getflowlabel{⟨idn⟩}
```

相反，可以使用以下命令从给定流框架的 IDL 中检索给定流框架的 IDN：

```
\getflowid{⟨cmd⟩}{⟨idl⟩}
```

其中 ⟨cmd⟩ 是用于存储框架 IDN 的控制序列。例如：

The label for the first flow frame is "\getflowlabel{1}". The flow frame labelled

¹如果要从绝对页面坐标转换为相对于类型块的坐标，请参阅第 54 页的问题 10。

²为什么我不能使用页码格式？请参阅第 53 页的问题 3。

2. 定义新的框架

2.1	流框架	5
2.1.1	提前结束流框架	6
2.2	静态框架	7
2.2.1	重要注意事项	10
2.3	动态框架	10
2.3.1	将章标题置于动态框架中	12
2.3.2	在动态框架中放置页眉和页脚	13
2.3.3	续文	13
2.3.4	重要注意事项	14

本章介绍如何定义新框架，以及如何识别和设置框架内容。另请参阅第 5 章

"main" has IDN \verb|.

将会得到：The label for the first flow frame is "main". The flow frame labelled "main" has IDN 1.。（可以查看 7.3 节。）

请注意，`\getflowlabel` 不执行任何检查来确定提供的 IDN 是否有效，但如果未定义提供的 IDN，`\getflowid` 将生成错误。

默认情况下，流框架将不具有边界，但星形形式

```
\newflowframe*[\langle 页面列表 \rangle][\langle 宽度 \rangle][\langle 高度 \rangle][\langle x 坐标 \rangle][\langle y 坐标 \rangle][\langle label \rangle]
```

将在流框架周围放置一个边界。（如果需要不同的边框，请参见第 3 章。）

请注意，如果文档继续超出定义的流框架（例如，流框架仅在第 1 页到第 10 页上定义，但文档包含 11 页），则将定义单个流框架，模拟所有后续页的一系列模式。

在本文中，我们使用了命令

```
\newflowframe{0.6\textwidth}{\textwidth}{0pt}{0pt}[main]
```

来定义主流框架³（即这个框架）。

2.1.1 提前结束流框架

可以使用以下命令之一强制文本立即移动到下一个定义的流框架：`\newpage`、`\pagebreak` 或 `\framebreak`。前两种模式的工作方式与标准的两列模式类似。当段落跨越两个不同宽度的流框架时，需要最后一个，即 `\framebreak`，因为 \TeX 的输出例程在前一个框架的最后一个段落结束之前不会调整为新的 `\hsize` 值。因此，新流框架开始处的段落结尾将保留前一流框架的宽度。

如果一个段落跨越两个宽度不等的流框架而不使用 `\framebreak`，则会发出警告。如果舍入错误（例如，如果框架是使用 `flowframtk` 或 `jpgfdraw` 创建的）导致框架宽度的细微差异，则可以调整公差以抑制这些警告。默认公差为 2pt。若要更改此设置，请将长度寄存器的公差设置为所需的公差。例如，要抑制宽度差小于 3pt 的警告，请执行

```
\setlength{\fftolerance}{3pt}
```

如果你想开始一个新页面，而不是简单地移动到下一个框架，请使用命令 `\clearpage`，或者对于双面文档，从下一个奇数页执行 `\cleardoublepage` 开始。

³偶数页的框架位置由第 3 章定义的 `\setflowframe` 命令来设置。

2.2 静态框架

静态框架是文本既不流入也不流出的矩形区域。必须显式设置内容，一旦设置，静态框架的内容将在每个页面上保持不变，直到显式更改为止。因此，可以使用静态框架使公司徽标等出现在每页的同一位置。

从版本 1.03 开始，现在可以使用具有非矩形内容的静态框架，有关详细信息，请参见 3.1 节。

定义一个新的静态框架使用命令：

```
\newstaticframe*[\langle 页面列表 \rangle]{\langle 宽度 \rangle}{\langle 高度 \rangle}{\langle x 坐标 \rangle}{\langle y 坐标 \rangle}[\langle label \rangle]
```

其中与 `\newflowframe` 框架一样，`\langle 宽度 \rangle` 是框架的宽度，`\langle 高度 \rangle` 是框架的高度，`(x,y)` 是该框架相对于版心左下角的位置。第一个可选参数 `\langle 页面列表 \rangle` 指此静态框架应该出现的页面列表，最后一个可选参数 `\langle label \rangle` 是唯一的文本 IDL，可以使用它来标识此框架。如果未指定标签，则可以通过其唯一的 IDN 引用此框架。要定义的第一个静态框架的 IDN 为 1，第二个 IDN 为 2，依此类推。

可以使用以下命令从给定静态框架的 IDN 中检索该框架的 IDL：

```
\getstaticlabel{\langle idn \rangle}
```

相反，也可以使用以下命令从其 IDL 中检索给定静态框架的 IDN：

```
\getstaticid{\langle cmd \rangle}{\langle idl \rangle}
```

其中 `\langle cmd \rangle` 是用于存储框架的 IDN 的控制序列。例如：

```
The label for the first static frame is "\getstaticlabel{1}". The static frame
labelled "backleft" has IDN \getstaticid{\myid}{backleft}\myid.
```

可以得到：The label for the first static frame is "backleft". The static frame labelled "backleft" has IDN 1。

请注意，`\getstaticlabel` 不执行任何检查来确定所提供的 IDN 是否有效，但是如果未定义所提供的 IDL，`\getstaticid` 将生成错误。

与 `\newflowframe` 框架一样，有一个星型版本

```
\newflowframe*[\langle 页面列表 \rangle]{\langle 宽度 \rangle}{\langle 高度 \rangle}{\langle x 坐标 \rangle}{\langle y 坐标 \rangle}[\langle label \rangle]
```

它将在该静态框架周围放置一个边界。

要设置特定静态框架的内容，可以使用 `staticcontents` 环境：

```
\begin{staticcontents}{\langle IDN \rangle}
\langle 内容 \rangle
\end{staticcontents}
```

其中 $\langle IDN \rangle$ 是与该静态框架关联的唯一 IDN, $\langle 内容 \rangle$ 是静态框架的内容, 或者使用以下命令:

```
\setstaticcontents{\langle IDN \rangle}{\langle 内容 \rangle}
```

会做同样的事情。

环境和命令都有可用的星号版本, 使你能够通过其关联的 IDL 而不是 IDN 来标识静态框架:

```
\begin{staticcontents*}{\langle IDL \rangle}
\langle 内容 \rangle
\end{staticcontents*}
```

或等效地:

```
\setstaticcontents*{\langle IDL \rangle}{\langle 内容 \rangle}
```

在 `staticcontents` 和 `staticcontents*` 的主体中, 或在 `\setstaticcontents` 和 `\setstaticcontents*` 的第二个参数中, 可以使用以下命令移动到另一个静态框架:

```
\continueonframe[\langle 延续文本 \rangle]{\langle id \rangle}
```

如果使用 `staticcontents*` 或 `\setstaticcontents*`, 则 $\langle id \rangle$ 表示下一个静态框架的 IDL, 否则 $\langle id \rangle$ 表示下一个静态框架的 IDN。可选参数指定要放置在第一个静态框架末尾的一些延续文本。例如, 假设我定义了两个标记为 “frame1” 和 “frame2” 的静态框架, 那么

```
\begin{staticcontents*}{frame1}
Some text in the first frame. (Let's
assume this frame is somewhere on the
left half of the page.)
\continueonframe[Continued on the right]{frame2}
This is some text in the second frame. (Let's
assume this frame is somewhere on the
right half of the same page.)
\end{staticcontents*}
```

相当于

```
\begin{staticcontents*}{frame1}
Some text in the first frame. (Let' s
assume this frame is somewhere on the
left half of the page.)
\ffcontinuedtextlayout{Continued on the right}
\end{staticcontents*}
\begin{staticcontents*}{frame2}\par\noindent
This is some text in the second frame. (Let' s
assume this frame is somewhere on the
right half of the same page.)
\end{staticcontents*}
```

其中

```
\ffcontinuedtextlayout{<text>}
```

控制延续文本的显示方式。用于显示延续文本的字体由

```
\ffcontinuedtextfont{<text>}
```

指定。

请注意，这假定在两个框架之间的转换中似乎没有段落中断。如果要段落分段，则需要
在`\continueonframe` 前后显式放置一个段落分段。例如：

```
\begin{staticcontents*}{frame1}
Some text in the first frame. (Let' s
assume this frame is somewhere on the
left half of the page.)

\continueonframe[Continued on the right]{frame2}

This is some text in the second frame. (Let' s
assume this frame is somewhere on the
right half of the same page.)
\end{staticcontents*}
```

2.2.1 重要注意事项

- 当设置静态框架的内容时，内容将立即被排版并存储在一个框中，直到将内容放在页面上为止。这意味着，如果使用在整个文档中变化的任何信息（例如页码），则设置静态框架内容时的当前值将是使用的值。
- 然而，如果在静态框架内使用 `\label` 命令，则每次显示静态框架时，标签信息将被写入辅助文件，直到该框架的内容已经改变。这意味着你可能最终会得到多重定义的标签。

2.3 动态框架

动态框架类似于静态框架，只是它的内容在每页上重新排版。（静态框架将其内容存储在存储框中，而动态框架将其内容存储在宏中。⁴）

从版本 1.03 开始，可以使用具有非矩形内容的动态框架，有关详细信息，请参见 3.1 节。

要创建新的动态框架，请使用以下命令：

```
\newdynamicframe[⟨页面列表⟩]{⟨宽度⟩}{⟨高度⟩}{⟨x 坐标⟩}{⟨y 坐标⟩}[⟨label⟩]
```

参数与 `\newflowframe` 和 `\newstaticframe` 的参数完全相同。同样，每个动态框架都有一个关联的唯一 IDN，对于要定义的第一个动态框架，从 1 开始，还可以使用最后一个可选参数 `⟨label⟩` 设置唯一 IDL。

可以使用以下方法从给定动态框架的 IDN 中检索该框架的 IDL：

```
\getdynamiclabel{⟨idn⟩}
```

相反，可以使用：

```
\getdynamicid{⟨cmd⟩}{⟨idl⟩}
```

从给定动态框架的 IDL 中检索该框架的 IDN，其中 `⟨cmd⟩` 是用于存储框架的 IDN 的控制序列。

例如：

```
The label for the first dynamic frame is "\getdynamiclabel{1}".
The dynamic frame labelled "chaphead" has IDN
\getdynamicid{\myid}{chaphead}\myid.
```

⁴这意味着可以在静态内容环境的主体中具有抄录文本，但不能在动态内容环境的正文中（参见第 2.3.1 页附近）。

可以得到：The label for the first dynamic frame is "chaphead". The dynamic frame labelled "chaphead" has IDN 1。

请注意，`\getdynamiclabel` 不执行任何检查来确定提供的 IDN 是否有效，但如果未定义提供的 IDL，则 `\getdynamicid` 将生成错误。

与其他框架类型一样，还有一个星号版本

```
\newdynamicframe*[\langle 页面列表 \rangle]{\langle 宽度 \rangle}{\langle 高度 \rangle}{\langle x 坐标 \rangle}{\langle y 坐标 \rangle}[\langle label \rangle]
```

这将在动态框架周围放置一个边界。例如，在本文档中，我们使用了

```
\newdynamicframe{0.38\textwidth}{\textwheight}{0.62\textwidth}{0pt}[chaphead]
```

它将会在奇数页的右边和偶数页的左边创建框架。（偶数页的位置会使用第 3 章中定义的命令 `\setdynamicframe` 来设置。）

使用以下命令设置动态框架的内容：

```
\setdynamiccontents{\langle id \rangle}{\langle 内容 \rangle}
```

其中 $\langle id \rangle$ 是与该动态框架关联的唯一 IDN， $\langle 内容 \rangle$ 是动态框架的内容。或者，如果已将 IDL 分配给动态框架，则可以使用星号版本：

```
\setdynamiccontents*{\langle idl \rangle}{\langle 内容 \rangle}
```

与大多数 L^AT_EX 命令一样，不能在 $\langle 内容 \rangle$ 中使用抄录文本。

从版本 1.09 开始，还可以使用 `dynamiccontents` 环境设置内容：

```
\begin{dynamiccontents}{\langle id \rangle}
\langle 内容 \rangle
\end{dynamiccontents}
```

或 `dynamiccontents*` 环境：

```
\begin{dynamiccontents*}{\langle label \rangle}
\langle 内容 \rangle
\end{dynamiccontents*}
```

请注意，不能在 `dynamiccontents` 或 `dynamiccontents*` 环境中使用抄录文本。

你还可以使用以下任一方法将文本附加（追加）到动态框架：

```
\appenddynamiccontents{<id>}{<内容>}
```

或：

```
\appenddynamiccontents*{<label>}{<内容>}
```

2.3.1 将章标题置于动态框架中

如果定义了章节，可以使用命令使章节标题显示在动态框架中

```
\dfchaphead{<IDN>}
```

其中 $\langle IDN \rangle$ 是动态框架的 IDN。如果要使用 IDL 而不是 IDN，则此命令还有一个星号版本。例如，在本文档中，我们使用了以下命令：

```
\dfchaphead*{chaphead}
```

如果使用 $\backslash dfchaphead$ ，则可以通过重新定义

```
\DFchapterstyle{<title>}
```

来调整章节标题的格式，重定义

```
\DFschapterstyle{<title>}
```

来调整没有编号的章节格式。例如，本文档按如下方式重新定义了这些命令：

```
\renewcommand{\DFchapterstyle}[1]{%
  {\raggedright\sffamily\bfseries\Huge\color{blue}\thechapter. #1\par}%
}
\renewcommand{\DFschapterstyle}[1]{%
  {\raggedright\sffamily\bfseries\Huge\color{blue} #1\par}%
}
```

没有办法将其它类型的章节标题放置在动态框架中。

2.3.2 在动态框架中放置页眉和页脚

可以使用命令

```
\makedfheaderfooter
```

将页眉和页脚放置在动态框架中。这将创建两个具有 IDL 为 `header` 和 `footer` 的动态框架。页面样式将像往常一样使用，但是你可以使用 `\setdynamicframe` 移动或调整页眉和页脚的大小（在第 3 章中介绍）。

2.3.3 续文

在 `dynamiccontents` 或 `dynamiccontents*` 的主体中，可以使用以下命令移动到另一个动态框架：

```
\continueonframe[⟨延续文本⟩]{id}
```

如果此命令出现在 `dynamiccontents*` 中，`⟨id⟩` 将引用新框架的 IDL，否则它将引用新框架的 IDN。可选参数指定要放置在第一个动态框架末尾的一些延续文本。例如，假设我定义了两个标记为 “frame1” 和 “frame2” 的动态框架，那么

```
\begin{dynamiccontents*}{frame1}
Some text in the first frame. (Let' s
assume this frame is somewhere on the
left half of the page.)
\continueonframe[Continued on the right]{frame2}
This is some text in the second frame. (Let' s
assume this frame is somewhere on the
right half of the same page.)
\end{dynamiccontents*}
```

等价于

```
\begin{dynamiccontents*}{frame1}
Some text in the first frame. (Let' s
assume this frame is somewhere on the
left half of the page.)
```

```

\ffcontinuedtextlayout{Continued on the right}
\end{dynamiccontents*}
\begin{dynamiccontents*}{frame2}\par\noindent
This is some text in the second frame. (Let' s
assume this frame is somewhere on the
right half of the same page.)
\end{dynamiccontents*}

```

其中

```
\ffcontinuedtextlayout{<text>}
```

控制延续文本的显示方式，用于显示延续文本的字体由

```
\ffcontinuedtextfont{<text>}
```

控制。

请注意，这假定在两个框架之间的转换中似乎没有段落中断。如果要段落分段，则需要用 `continueonframe` 前后显式放置一个段落分段。例如：

```

\begin{dynamiccontents*}{frame1}
Some text in the first frame. (Let' s
assume this frame is somewhere on the
left half of the page.)

\continueonframe[Continued on the right]{frame2}

This is some text in the second frame. (Let' s
assume this frame is somewhere on the
right half of the same page.)
\end{dynamiccontents*}

```

2.3.4 重要注意事项

- 不能在动态框架中使用逐字文本。这包括在 `dynamiccontents` 和 `dynamiccontents*` 环境的主体。

- `\continueonframe` 不能用于任何设定动态框架内容命令的参数，例如设置动态框架。
- 动态框架是在所有静态和流动框架之后绘制在页面上的。



定义流框架、静态框架和动态框架后，可以更改它们的属性。这三种类型的框架大多具有相同的属性命令，但有些是特定用于某一类型的。

使用以下命令之一修改流框架属性：

```
\setflowframe{⟨idn 列表⟩}{⟨键-值对列表⟩}
```

或者星型版本：

```
\setflowframe*{⟨idl 列表⟩}{⟨键-值对列表⟩}
```

或者可以使用以下方法设置所有流框架的属性：

```
\setallflowframes{⟨键-值对列表⟩}
```

使用以下命令之一修改静态框架属性：

```
\setstaticframe{⟨idn 列表⟩}{⟨键-值对列表⟩}
```

或者星型版本：

```
\setstaticframe*{⟨idl 列表⟩}{⟨键-值对列表⟩}
```

或者可以使用以下方法设置所有静态框架的属性：

```
\setallstaticframes{⟨键-值对列表⟩}
```

使用以下命令之一修改动态框架属性：

```
\setdynamicframe{⟨idn 列表⟩}{⟨键-值对列表⟩}
```

或者星型版本：

```
\setdynamicframe*{⟨idl 列表⟩}{⟨键-值对列表⟩}
```

或者可以使用以下方法设置所有动态框架的属性：

3. 修改框架属性

3.1 非矩形框架 23

3.2 即时打开和关闭框架 25

本章介绍如何修改框架属性，例如尺寸和位置。

```
\setalldynamicframes{⟨键-值对列表⟩}
```

在上述每一种情况下, ⟨*idn* 列表⟩ 可以是关键字之一: **all**、**odd** 或 **even** (表示该类型的所有框架、IDN 为奇数的该类型的框架或 IDN 为偶数的该类型的框架), 也可以是用逗号分隔的 IDN 号列表或 IDN 范围。

对于星型版本, ⟨*idl* 列表⟩ 应该是 *idl* 的逗号分隔列表。请注意, 不能使用上述关键字或在 ⟨*idl* 列表⟩ 包含范围。

⟨键-值对列表⟩ 参数必须是以逗号分隔的 ⟨键⟩=⟨值⟩ 列表所指示要修改的属性。如果 ⟨值⟩ 包含一个或多个逗号或等号, 请确保将其分组。可用值如下:

width=⟨长度⟩ 这是框架的宽。

height=⟨长度⟩ 这是框架的高。

x=⟨长度⟩ 框架在其上定义的所有页的 *x* 坐标。

y=⟨长度⟩ 框架在其上定义的所有页的 *y* 坐标。

evenx=⟨长度⟩ 定义框架在偶数页的 *x* 坐标, 但前提是文档是双面文档。

例如, 在本文档中, 我们使用了

```
\setflowframe*{main}{evenx=0.4\textwidth}  
\setdynamicframe*{chaphead}{evenx=0pt}
```

在偶数页上分别切换包含文档文本的流框架和章节标题的动态框架的位置。
可以使用以下命令交换奇数和偶数值:

```
\ffswapoddeven{⟨IDN⟩}
```

用于流框架;

```
\sfswapoddeven{⟨IDN⟩}
```

用于静态框架;

```
\dfswapoddeven{<IDN>}
```

用于动态框架。这些命令都有星型版本，用于使用框架的 IDL 而不是 IDN。

eveny=<长度> 框架的 y 坐标，用于定义框架在所有偶数页的位置，但前提是文档是双面文档。

oddx=<长度> 定义框架在奇数数页的 x 坐标，但前提是文档是双面文档。

oddy=<长度> 定义框架在奇数数页的 y 坐标，但前提是文档是双面文档。

valign=<位置> 更改静态或动态框架内容的垂直对齐方式。<pos> 值可以是：c、t 或 b 之一。静态框架的默认值为 c，动态框架的默认值为 t。此键不适用于流框架。

label=<text> 为框架分配 IDL。（如果在第一次定义框架时未指定标签，则将其指定一个与其 IDN 相同的标签。）提供此键以允许用户标记由第 5 章中描述的某些预定义布局命令生成的框架。

border=<样式> 框架周围边框的样式，这可以采用以下值：**none**（无边框）、**plain**（普通边框）或不带前一个反斜杠的 L^AT_EX 框架生成命令的名称。（我们承认符号有点混乱，框架生成命令是在其参数周围放置某种边界的命令，例如 `\fbox`，或者如果你使用的是 `fancybox` 包：`\doublebox`、`\ovalbox`、`\Ovalbox` 和 `\shadowbox`。）**fbox** 的值相当于 **plain**。
例如，要使第一个静态框架具有椭圆形边框，则执行以下操作：

```
\setstaticframe{1}{border=ovalbox}
```

或者可以定义自己的边界框：

```
\newcommand{\greenyellowbox}[1]{\fcolorbox{green}{yellow}{#1}}
\setstaticframe{1}{border=greenyellowbox}
```

下一个示例使用 `tikz` 包定义一个花式框架，因此需要使用：

```
\usepackage{tikz}
\usetikzlibrary{snakes}
```

边框命令定义如下：

```

\newlength\fancywidth
\newlength\fancyheight
\newlength\fancydepth

\newcommand{\fancyborder}[1]{%
\settowidth{\fancywidth}{#1}%
\settoheight{\fancyheight}{#1}%
\settodepth{\fancydepth}{#1}%
\addtolength{\fancyheight}{\fancydepth}%
\hspace{-\flowframesep}%
\tikz[baseline=0pt]{%
\draw[snake=bumps,raise snake=\flowframesep,
line width=\flowframerule]
(0pt,0pt) rectangle (\fancywidth,\fancyheight);
}}

```

这会生成一个凹凸的边框，但它使用 `\flowframesep` 确定边框和文本之间的间隙，并使用 `\flowframerule` 设置线宽。这样可以确保正确计算偏移量（见下文）。此新边框现在可以应用于框架：

```
\setstaticframe{1}{border=fancyborder}
```

offset=〈偏移量〉 边框偏移量，如果是用户定义的边框。这是从左侧边框的外边缘到边框内文本边框的左边缘的距离。`flowfram` 包能够计算下列已知的框架生成命令的边框：`\fbox`、`\ovalbox`、`\ovalbox`、`\doublebox` 和 `\shadowbox`。对于所有其他边界，假定偏移量为 `-\flowframesep-\flowframerule`。如果定义了自己的框架生成命令，则可能需要显式指定偏移量，否则流/静态/动态框架最终可能会向右或向左移动。但是，如果要计算，上面的示例可以计算它们自己的偏移，例如：

```

\newcommand{\thickgreenyellowbox}[1]{%
{\setlength{\fboxsep}{5pt}\setlength{\fboxrule}{6pt}%
\fcolorbox{green}{yellow}{#1}}}

```

然后必须指定偏移量。在本例中，偏移量为 `-5pt-6pt=-11pt`，因此需要执行以下操作：


```
\setstaticframe{1}{border=thickgreenyellowbox,offset=-11pt}
```

bordercolor=⟨颜色⟩ 使用标准制框命令的边框颜色。颜色可以指定为 **green**，也可以包括颜色模型，像 `[rgb]{0,1,0}`。例如：

```
\setallflowframes{border=doublebox,bordercolor=[rgb]{1,0,0.5}}
```

textcolor=⟨颜色⟩ 框架的文本颜色。同样，颜色可以指定为 **green**，也可以包括颜色模型，例如 `[rgb]{0,1,0}`。

backcolor=⟨颜色⟩ 框架的背景色。同样，颜色可以指定为 **green**，也可以包括颜色模型，例如 `[rgb]{0,1,0}`。请注意，背景颜色仅延伸到边界框，而不是边界。如果希望它延伸到边界，则需要定义自己的边界类型（见上文）。

pages=⟨页面列表⟩ 应该出现的页面列表。它可以有以下值：**all**、**even**、**odd** 或 **none**（如果有多个具有相同编号的页，后者会那一点开始删除框架），也可以是一个单独的单页列表或页范围。例如：

```
\setdynamicframe{1}{pages={1,5,8-10}}
```

列表中的数字要么是指页面计数器的整数值（与包选项 **pages=relative** 一起使用时），要么是绝对页码（与包选项 **pages=absolute** 一起使用时）。

从版本 1.14 开始，还有一种快速设置页面列表的方法，它不需要像 `\setflowframe` 命令那样的 `⟨key⟩=⟨value⟩` 格式：

```
\flowsetpagelist{⟨idn⟩}{⟨页面列表⟩}  
\dynamicsetpagelist{⟨idn⟩}{⟨页面列表⟩}  
\staticsetpagelist{⟨idn⟩}{⟨页面列表⟩}
```

注意，这些命令没有星型版本。第一个参数必须是单个 *IDN*。另见 3.2 节。

excludepages=⟨页面列表⟩（1.14 版的新版本）不应出现框架的页码列表，用逗号分隔。注意，这会覆盖 **pages** 键给出的任何页面。对于此键，⟨页面列表⟩ 只能包含逗号分隔的数字。不允许使用范围。例如：

```
\setdynamicframe{1}{excludepages={7}}
\setdynamicframe{1}{pages={1-10}}
```

这将使动态框架出现在第 1 至 6 页和第 8 至 10 页。

同样，还有一种快速设置排除页面列表的方法，它不需要像 `\setflowframe` 命令那样的 `<key>=<value>` 格式：

```
\flowsetexclusion{<idn>}{<页面列表>}
\dynamicsetexclusion{<idn>}{<页面列表>}
\staticsetexclusion{<idn>}{<页面列表>}
```

或者可以使用以下方法添加到排除页面列表：

```
\flowaddexclusion{<idn>}{<页面列表>}
\dynamicaddexclusion{<idn>}{<页面列表>}
\staticaddexclusion{<idn>}{<页面列表>}
```

注意，这些命令没有星型版本。第一个参数必须是单个 *IDN*。另见 3.2 节。

hide=<布尔值> 如果设置此值，则无论页面或排除设置如何，静态或动态框架都将隐藏。（1.16 版本的新命令。）

hidethis=<布尔值> 类似于 **hide**，但输出例程总是将其重置回 **false**，因此它只影响当前页。（1.16 版本的新命令。）

margin=<*side*> 流框架边注的位置，其值为 **left**，**right**，**inner** 或 **outer**，此设置仅适用于流框架。

clear=<布尔值> 如果设置了此值，则静态或动态框架将在下一页开始时清除，否则仅在下一次出现 `\setstaticcontents` 或 `\verbstaticcontents` 环境或 `\setdynamiccontents` 时清除，具体取决于框架类型。默认情况下不设置此值，此设置不适用于流动框架。
例如，为了防止章节标题在每页上重新出现，我们使用了以下命令：

```
\setdynamicframe*{chaphead}{clear}
```

如果要将标签放入静态或动态框架中，应使用 **clear** 键来防止对标签进行多次定义。

style=<命令> 这是不带反斜杠的命令的名称，该命令将应用于指定动态框架的内容。命令可以是声明，例如：

```
\setalldynamicframes{style=large}
```

这将以字体大小为 **large** 设置所有动态框架的内容。也可以是采用单个参数的命令，例如：

```
\setalldynamicframes{style=textbf}
```

这将使所有动态框架的文本以粗体显示。若要取消设置样式，请执行 **style=none**。此设置仅适用于动态框架。

angle=<*n*> 将框架内容旋转 *n* 度（1.02 版本的新命令）。请注意，边界框不会出现旋转。

shape=<形状命令> 为静态框架或动态框架（1.03 版本的新命令）的内容定义形状。如果 <形状命令> 为 **relax**，则不会应用段落形状。详见 3.1 节。

3.1 非矩形框架

从 1.03 版本开始，可以指定非矩形的静态或动态框架（但不能指定流框架）。请注意，尽管框架的形状设置不同，边界框仍将显示为矩形。可以使用 T_EX 的 **\parshape** 命令，也可以使用 Donald Arseneau 的 **shapepar** 包定义的 **\shapepar**/**\Shapepar** 命令（如果使用 **\shapepar** 或 **\Shapepar**，请记住包含 **shapepar** 包）。

\shapepar 或 **\Seabepar** 命令在可使用的形状类型中提供了更大的灵活性。但是，请注意 **shapepar** 文档中给出的建议。

\parshape 使用 **\parshape**，不能在框架的中间，顶部或底部有切口，但是可以在框架的左侧或右侧有切口。当与形状键一起用于静态或动态框架时，**\par** 和分节命令的效果将被修改，以允许段落形状扩展到单个段落之外，并允许分节命令（但不包括 **\chapter** 或 **\part**）。

\shapepar/**\Shapepar** 使用 **\shapepar** 或 **\Shapepar**，你可能想设计切口，但其中可能没有任何分节命令，段落分隔符，垂直间距或数学运算。你可以使用 **\simpar** 模拟段落中断，但是不建议这样做。形状的大小取决于文本的数量，因此形状会随着添加或删除文本而扩大或缩小。通常，**\Shapepar** 比 **\shapepar** 更适合用作框架形状。有关这些命令的更多详细信息，请参见 **shapepar** 文档。

This is an example of a dynamic frame with a non-rectangular shape. This zigzag shape was specified using the shape key setting in `\setstaticframe`. The `\parshape` command was used to set the shape. Using the shape key rather than explicitly using `\parshape` within the `dynamiccontents` environment means that I can have paragraph breaks, sectioning commands, and even some mathematics

$$E = mc^2 \quad (3.1)$$

whilst retaining the shape.

This example has a more complicated shape that can not be generated using T_EX's `\parshape` command, so `\shapepar` was used instead. Note that this document must include the `shapepar` package in this instance, whereas no extra packages are required to use `\parshape`. No mathematics or sectioning commands are allowed here. The shape will expand as more text is added to it.

若要将框架还原为其默认矩形设置, 请使用 `shape=\relax`。对于不熟悉 T_EX `\parshape` 命令的用户, 语法如下:

`\parshape=n i1 l1 i2 l2 ... in ln`

其中 n 是 (i_j, l_j) 对的数目, i_j 指定第 j 行的左缩进, l_j 指定第 j 行的长度。

左上角的动态框架使用以下方式分配为锯齿形:

```
\setdynamicframe*{chaphead}{shape={\parshape=20
0.6\linewidth 0.4\linewidth 0.5\linewidth 0.4\linewidth
0.4\linewidth 0.4\linewidth 0.3\linewidth 0.4\linewidth
0.2\linewidth 0.4\linewidth 0.1\linewidth 0.4\linewidth
0pt 0.4\linewidth 0.1\linewidth 0.4\linewidth
0.2\linewidth 0.4\linewidth 0.3\linewidth 0.4\linewidth
0.4\linewidth 0.4\linewidth 0.5\linewidth 0.4\linewidth
0.6\linewidth 0.4\linewidth 0.5\linewidth 0.4\linewidth
0.4\linewidth 0.4\linewidth 0.3\linewidth 0.4\linewidth
0.2\linewidth 0.4\linewidth 0.1\linewidth 0.4\linewidth
0pt 0.4\linewidth 0.1\linewidth 0.4\linewidth
}}
```

`\shapepar` 和 `\shapepar` 的语法更复杂, 有关详细信息, 请参阅 `shapepar` 文档。一般来说就是:

`\shapepar{<形状>}`

`shapepar` 包有四个预定义的形状: `\squareshape`、`\diamondshape`、`heartshape` 和 `\nutshape`。

左下角的静态框架使用以下命令指定了一个心形:

```
\setstaticframe*{shapedb}{shape={\shapepar\heartshape},pages={24}}
```

要将框架重置回其原始矩形形状, 请执行以下操作:

```
\setstaticframe*{shapedb}{shape=\relax}
```

`flowfram` 包当前不支持任何其他段落形状生成命令, 任何其他命令必须在框架的内容中显式使用。

3.2 即时打开和关闭框架

在文档环境中修改页面列表（或页面排除列表）的风险是很大的。在输出例程查找下一框架之前，此列表必须是最新的。为了使这一点更容易，从版本 1.14 开始，有一些命令可以帮助你做到这一点。如果要使用这些命令，最好使用 `package` 选项 `pages=absolute`。

下一次使用输出例程时，本节中描述的命令将更新页面列表（可能还有排除列表）。它们被设计用来打开或关闭下一页或下一奇数页上的框架。因此，你需要注意这些命令的位置。例如，如果你有一个双面文档，并且执行了以下操作：

```
\dynamicswitchonnexodd{1}  
\mainmatter  
\chapter{Introduction}
```

这将使 IDN 为 1 的动态框架在第 1 章的第一页可见。但是，如果你这样做

```
\mainmatter  
\dynamicswitchonnexodd{1}  
\chapter{Introduction}
```

将产生不同的效果，因为 `\mainmatter` 发出了一个 `\cleardoublepage`，因此打开动态框架的命令与第 1 章开头的页面相同。这意味着动态框架将不会出现，直到之后的奇数页（第 3 页）。

这些命令都具有相同的语法，其中一个参数可以是逗号分隔的列表。星型版本使用 IDL，非星型版本使用 IDN。

```
\flowswitchonnexodd{<IDN 列表>}  
\flowswitchonnexodd*{<IDL 列表>}
```

从下一页开始开启列出的流框架。

```
\flowswitchoffnnext{<IDN 列表>}  
\flowswitchoffnnext*{<IDL 列表>}
```

从下一页开始关闭列出的流框架。

```
\flowswitchonnexodd{<IDN 列表>}
```

`\flowswitchonnextodd*{⟨IDL 列表⟩}`

从下一个奇数页开始打开列出的流框架。

`\flowswitchoffnextodd{⟨IDN 列表⟩}`
`\flowswitchoffnextodd*{⟨IDL 列表⟩}`

从下一个奇数页开始关闭列出的流框架。

`\flowswitchonnextonly{⟨IDN 列表⟩}`
`\flowswitchonnextonly*{⟨IDL 列表⟩}`

仅在下一页上开启列出的流框架。

`\flowswitchoffnextonly{⟨IDN 列表⟩}`
`\flowswitchoffnextonly*{⟨IDL 列表⟩}`

仅在下一页上关闭列出的流框架。

`\flowswitchonnextoddonly{⟨IDN 列表⟩}`
`\flowswitchonnextoddonly*{⟨IDL 列表⟩}`

仅在下一奇数页上开启列出的流框架。

`\flowswitchoffnextoddonly{⟨IDN 列表⟩}`
`\flowswitchoffnextoddonly*{⟨IDL 列表⟩}`

仅在下一奇数页上关闭列出的流框架。

`\dynamicswitchonnext{⟨IDN 列表⟩}`
`\dynamicswitchonnext*{⟨IDL 列表⟩}`

从下一页开始开启列出的动态框架。

```
\dynamicswitchoffnext{<IDN 列表>}  
\dynamicswitchoffnext*{<IDL 列表>}
```

从下一页开始关闭列出的动态框架。

```
\dynamicswitchonnextodd{<IDN 列表>}  
\dynamicswitchonnextodd*{<IDL 列表>}
```

从下一奇数页开始开启列出的动态框架。

```
\dynamicswitchoffnextodd{<IDN 列表>}  
\dynamicswitchoffnextodd*{<IDL 列表>}
```

从下一奇数页开始关闭列出的动态框架。

```
\dynamicswitchonnextonly{<IDN 列表>}  
\dynamicswitchonnextonly*{<IDL 列表>}
```

仅在下一页上打开列出的动态框架。

```
\dynamicswitchoffnextonly{<IDN 列表>}  
\dynamicswitchoffnextonly*{<IDL 列表>}
```

仅在下一页上关闭列出的动态框架。

```
\dynamicswitchonnextoddonly{<IDN 列表>}  
\dynamicswitchonnextoddonly*{<IDL 列表>}
```

仅在下一奇数页上打开列出的动态框架。

```
\dynamicswitchoffnextoddonly{<IDN 列表>}  
\dynamicswitchoffnextoddonly*{<IDL 列表>}
```

仅在下一奇数页上关闭列出的动态框架。

```
\staticswitchonnext{<IDN 列表>}  
\staticswitchonnext*{<IDL 列表>}
```

从下一页开始开启列出的静态框架。

```
\staticswitchoffnext{<IDN 列表>}  
\staticswitchoffnext*{<IDL 列表>}
```

从下一页开始关闭列出的静态框架。

```
\staticswitchonnextodd{<IDN 列表>}  
\staticswitchonnextodd*{<IDL 列表>}
```

从下一奇数页开始开启列出的静态框架。

```
\staticswitchoffnextodd{<IDN 列表>}  
\staticswitchoffnextodd*{<IDL 列表>}
```

从下一奇数页开始关闭列出的静态框架。

```
\staticswitchonnextonly{<IDN 列表>}  
\staticswitchonnextonly*{<IDL 列表>}
```

仅在下一页上打开列出的静态框架。

```
\staticswitchoffnextonly{<IDN 列表>}  
\staticswitchoffnextonly*{<IDL 列表>}
```

仅在下一页上关闭列出的静态框架。

```
\staticswitchonnextoddonly{<IDN 列表>}  
\staticswitchonnextoddonly*{<IDL 列表>}
```

仅在下一奇数页上打开列出的静态框架。


```
\staticswitchoffnextoddoneonly{⟨IDN 列表⟩}  
\staticswitchoffnextoddoneonly*{⟨IDL 列表⟩}
```

仅在下一奇数页上关闭列出的静态框架。

`flowfram` 包随附一个示例文件 `sample-pages.tex`，该文件使用其中一些命令。



本章介绍可用于确定框架位置和尺寸的一些命令。有关这些命令或此处未列出的其他命令的详细信息，请参阅随附文档 `flowfram.pdf`。

4.1 确定版心的位置

如前所述，创建新框架时，必须指定它们相对于版心的位置，但如果要将框架放置在距纸张边缘一定距离的位置，该怎么办？`flowfram` 包提供以下命令，用于计算从版心到纸张边界的距离：

```
\computeleftedgeodd{<长度>}
```

计算（奇数）页左边缘相对于版心左侧的位置，并将结果存储在 `<长度>` 中。

```
\computeleftedgeeven{<长度>}
```

如上所述，但对于偶数页。

```
\computetopedge{<长度>}
```

这将计算页面的上边缘（相对于版心的底部），并将结果存储在 `<长度>` 中。

```
\computebottomedge{<长度>}
```

这将计算页面的下边缘（相对于版心的底部），并将结果存储在 `<长度>` 中。

```
\computerightedgeodd{<长度>}
```

计算（奇数）页右边缘相对于版心左侧的位置，并将结果存储在 `<长度>` 中。

```
\computerightedgeeven{<长度>}
```

如上所述，但对于偶数页。

请注意，在所有情况下，`<长度>` 必须是 `LATEX` 长度命令。

例如，如果要创建一个框架，其左下角距页面左边缘一英寸，距页面下边缘半英寸（这假定奇数页和偶数页具有相同的页边距）：

4. 位置和尺寸

4.1 确定版心的位置	31
4.2 确定框架的尺寸和位置	32
4.3 相对位置	32

本章介绍了一些用于确定框架位置和尺寸的命令。

```
% 定义两个新的长度来表示 x 和 y 坐标
\newlength{\myX}
\newlength{\myY}
% 计算从版心到纸张边缘的距离
\computeleftedgeodd{\myX}
\computebottomedge{\myY}
% 添加绝对坐标以获得坐标
% 相对于版心
\addtolength{\myX}{1in}
\addtolength{\myY}{0.5in}
```

4.2 确定框架的尺寸和位置

可以使用以下命令之一确定框架的尺寸和位置：

- `\getstaticbounds{<IDN>}`
- `\getstaticbounds*{<IDL>}`
- `\getflowbounds{<IDN>}`
- `\getflowbounds*{<IDL>}`
- `\getdynamicbounds{<IDN>}`
- `\getdynamicbounds*{<IDL>}`

对于每个命令，星型版本以 IDL 作为参数，而非星型版本以 IDN 作为参数。每个命令都将相关信息存储在长度 `\ffareawidth`、`\ffareaheight`、`\ffareax` 和 `\ffareay` 中。

有关其他相关命令，请参阅随附文档 `flowfram.pdf` 中的“确定尺寸和位置”部分。

4.3 相对位置

要打印一框架与另一框架的相对位置，请执行以下操作：

```
\relativeframelocation{<类型 1>}{<idn1>}{<类型 2>}{<idn2>}
```

其中 <类型 1> 和 <idn1> 表示第一个框架的类型和 IDN，<类型 2> 和 <idn2> 表示第二个框架的类型和 IDN。还有一个星型版本：

```
\relativeframelocation*{<类型 1>}{<idl1>}{<类型 2>}{<idl2>}
```

其中 <类型 1> 和 <idl1> 表示第一个框架的类型和 IDL，<类型 2> 和 <idl2> 表示第二个框架的类型和 IDL。以上两个命令都将打印以下命令之一：

- \FFaboveleft 如果第一个框架在第二个框架的左上方。
- \FFaboveright 如果第一个框架在第二个框架的右上方。
- \FFabove 如果第一个框架在第二个框架上方。
- \FFbelowleft 如果第一个框架在第二个框架的左下方。
- \FFbelowright 如果第一个框架在第二个框架的右下方。
- \FFbelow 如果第一个框架在第二个框架的下方。
- \FFleft 如果第一个框架在第二个框架的左侧。
- \FFright 如果第一个框架在第二个框架的右侧。
- \FFoverlap 如果两个框架重叠。

如果第一个框架的下边缘高于第二个框架的上边缘，则认为框架高于另一框架。

如果第一个框架的上边缘低于第二个框架的下边缘，则认为该框架低于另一框架。

如果第一个框架的右边缘位于第二个框架的左边缘的左侧，则认为该框架位于另一框架的左侧。

如果第一个框架的左边缘位于第二个框架的右边缘的右侧，则认为框架位于另一框架的右侧。

请注意，无论两个框架中的任何一个是否显示在当前页上，都会为当前页选取相对位置。如果当前页是奇数页，则将比较奇数页的框架设置，否则将比较偶数页的框架设置。但是请记住，每页的第一段保留了段落开头的设置，因此，如果奇数页和偶数页的框架有不同的位置，则 `\relativeframelocation` 可能使用错误的页设置（如果在页面开头使用）。

例如，此文档定义了一个标记为 `main` 的流框架（此框架）和一个用于显示章节标题的标记为 `chaphed` 的动态框架，以下代码

The dynamic frame is
`\relativeframelocation*{dynamic}{chaphead}{flow}{main}`
 of the flow frame.

将得到: The dynamic frame is on the left of the flow frame.。

对于同一类型的框架, 有一些快捷命令:

```
\reldynamicloc{<idn1>}{<idn2>}
```

这相当于

```
\relativeframelocation{dynamic}{<idn1>}{dynamic}{<idn2>}
```

```
\relstaticloc{<idn1>}{<idn2>}
```

这相当于

```
\relativeframelocation{static}{<idn1>}{static}{<idn2>}
```

```
\relflowloc{<idn1>}{<idn2>}
```

这相当于

```
\relativeframelocation{flow}{<idn1>}{flow}{<idn2>}
```

上面的每个命令都有一个星号版本, 使用 IDL 而不是 IDN。

对于同一页上的框架, 这些命令可以在 `\continueonframe` 的可选参数中使用。例如:

```
\begin{dynamiccontents}{1}
Some text in the first dynamic frame that goes on for
quite a bit longer than this example.
\continueonframe[continued \reldynamicloc{2}{1}]{2}
This text is in the second dynamic frame which is
somewhere on the same page.
\end{dynamiccontents}
```

有关确定一框架与另一框架的相对位置的其他命令, 请参阅随附文档 `flowfram.pdf` 中的“确定一框架与另一框架的相对位置”部分。

`flowfram` 包有许多命令，可以在预定义的布局中创建框架。这些命令只能在导言区中使用。

5.1 栏样式

重新定义了标准的 \LaTeX 命令 `\onecolumn` 和 `\twocolumn`，以创建一个或两个流框，这些流框填充了彼此分隔的整个类型块（在 `\twocolumn` 的情况下），间隔为 `\columnsep` 宽度。这些流框架的高度可能没有版心高，因为它们的高度被调整为 `\baselineskip` 的整数倍。你可以使用以下命令关闭此自动调整：

```
\ffvadjustfalse
```

`onecolumn` 和 `twocolumn` 命令还接受一个可选参数，该参数是为其定义流框架的页面列表。除了 `onecolumn` 和 `twocolumn` 之外，还定义了以下命令：

```
\Ncolumn[⟨页面⟩]{⟨n⟩}
```

这将创建 n 列流框架，每个流框架之间的间隔为 `\columnsep`。

```
\onecolumninarea[⟨页面⟩]{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这将创建一个流框架来填充给定区域，并调整高度，使其为 `\baselineskip` 的整数倍。

```
\twocolumninarea[⟨页面⟩]{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这将创建两个以 `\columnsep` 的距离分隔的列流框，填充指定的整个区域，并再次调整高度，使其成为 `\baselineskip` 的整数倍。列之间以 `\columnsep` 隔开。

```
\Ncolumninarea[⟨页面⟩]{⟨n⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这是 `\twocolumninarea` 的更一般形式，它生成 n 个流框架而不是两个。

5.2 带有附加框架的栏样式

除了上面定义的栏样式流框架之外，还可以定义 $\langle n \rangle$ 个栏，并在其上方或下方生成一个附加框架。栏与额外框架之间的垂直间隙约为 $1\backslash\text{vcolumnsep}$ 。在以下每个定义中，参数 $\langle \text{页面} \rangle$ 是为其定义

5. 预定义布局

5.1	栏样式	35
5.2	带有附加框架的栏样式	35
5.3	从右到左的栏	40
5.4	背景特效	40
5.4.1	垂直条效果	41
5.4.2	水平条效果	41
5.4.3	背景框架	42
5.4.4	垂直和水平线	42

本章介绍创建以预定义布局排列的框架的命令。

框架的页面列表, $\langle n \rangle$ 是所需的栏数, $\langle \text{类型} \rangle$ 是在列的上方或下方移动的框架类型 (可能是以下之一: **flow**、**static** 或 **dynamic**)。新框架应填充的区域由 $\langle \text{宽度} \rangle$, $\langle \text{高度} \rangle$ (区域的宽度和高度) 和 $\langle x \rangle$, $\langle y \rangle$ (区域左下角相对于版心的左下角的位置) 定义。

栏顶部或底部的附加框架的高度由 $\langle H \rangle$ 给出。

```
\onecolumntopinarea[ $\langle \text{页面} \rangle$ ]{ $\langle \text{类型} \rangle$ }{ $\langle H \rangle$ }{ $\langle \text{宽度} \rangle$ }{ $\langle \text{高度} \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }
```

这将在其上方创建一个带有 $\langle \text{类型} \rangle$ 框架的流框架, 从而填充指定的区域。

```
\twocolumntopinarea[ $\langle \text{页面} \rangle$ ]{ $\langle \text{类型} \rangle$ }{ $\langle H \rangle$ }{ $\langle \text{宽度} \rangle$ }{ $\langle \text{高度} \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }
```

这将创建两栏流框架, 并在其上方设置一个 $\langle \text{类型} \rangle$ 框架, 填充指定的区域。

```
\Ncolumntopinarea[ $\langle \text{页面} \rangle$ ]{ $\langle \text{类型} \rangle$ }{ $\langle n \rangle$ }{ $\langle H \rangle$ }{ $\langle \text{宽度} \rangle$ }{ $\langle \text{高度} \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }
```

这将创建 n 栏流框架, 并在其上方设置一个 $\langle \text{类型} \rangle$ 框架, 填充指定的区域。

```
\onecolumnbottominarea[ $\langle \text{页面} \rangle$ ]{ $\langle \text{类型} \rangle$ }{ $\langle H \rangle$ }{ $\langle \text{宽度} \rangle$ }{ $\langle \text{高度} \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }
```

这将创建一个在其下方带有 $\langle \text{类型} \rangle$ 框架的流框架, 填充指定的区域。

```
\twocolumnbottominarea[ $\langle \text{页面} \rangle$ ]{ $\langle \text{类型} \rangle$ }{ $\langle H \rangle$ }{ $\langle \text{宽度} \rangle$ }{ $\langle \text{高度} \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }
```

这将创建两栏流框架, 并在其下方设置一个 $\langle \text{类型} \rangle$ 框架, 填充指定的区域。

```
\Ncolumnbottominarea[ $\langle \text{页面} \rangle$ ]{ $\langle \text{类型} \rangle$ }{ $\langle n \rangle$ }{ $\langle H \rangle$ }{ $\langle \text{宽度} \rangle$ }{ $\langle \text{高度} \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }
```

这将创建 n 栏流框架, 并在其下方设置一个 $\langle \text{类型} \rangle$ 框架, 填充指定的区域。

以下命令是上述内容的特殊情况:

```
\onecolumnStopinarea[ $\langle \text{页面} \rangle$ ]{ $\langle H \rangle$ }{ $\langle \text{宽度} \rangle$ }{ $\langle \text{高度} \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }
```

这相当于:

```
\onecolumntopinarea[ $\langle \text{页面} \rangle$ ]{static}{ $\langle H \rangle$ }{ $\langle \text{宽度} \rangle$ }{ $\langle \text{高度} \rangle$ }{ $\langle x \rangle$ }{ $\langle y \rangle$ }
```



```
\onecolumnDtopinarea[⟨页面⟩]{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：

```
\onecolumnntopinarea[⟨页面⟩]{dynamic}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

```
\onecolumnntop[⟨页面⟩]{⟨类型⟩}{⟨H⟩}
```

类似于 \onecolumnntopinarea，其中的区域指的是整个版心。

```
\onecolumnStop[⟨页面⟩]{⟨H⟩}
```

这相当于：\onecolumnntop[⟨页面⟩]{static}{⟨H⟩}。

```
\onecolumnDtop[⟨页面⟩]{⟨H⟩}
```

这相当于：\onecolumnntop[⟨页面⟩]{dynamic}{⟨H⟩}。

```
\twocolumnStopinarea[⟨页面⟩]{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：\twocolumnntopinarea[⟨页面⟩]{static}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}

```
\twocolumnDtopinarea[⟨页面⟩]{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：\twocolumnntopinarea[⟨页面⟩]{dynamic}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}

```
\twocolumnntop[⟨页面⟩]{⟨类型⟩}{⟨H⟩}
```

类似于 \twocolumnntopinarea，其中的区域指的是整个版心。

```
\twocolumnStop[⟨页面⟩]{⟨H⟩}
```

这相当于：\twocolumnntop[⟨页面⟩]{static}{⟨H⟩}。

```
\twocolumnDtop[⟨页面⟩]{⟨H⟩}
```

这相当于：\twocolumnntop[⟨页面⟩]{dynamic}{⟨H⟩}。

```
\NcolumnStopinarea[⟨页面⟩]{⟨n⟩}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：\Ncolumnstopinarea[⟨页面⟩]{static}{⟨n⟩}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}。

```
\NcolumnDtopinarea[⟨页面⟩]{⟨n⟩}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：\Ncolumnstopinarea[⟨页面⟩]{dynamic}{⟨n⟩}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}。

```
\Ncolumnstop[⟨页面⟩]{⟨类型⟩}{⟨n⟩}{⟨H⟩}
```

类似于 \Ncolumnstopinarea，当然其中的区域指的是整个的版心。

```
\NcolumnStop[⟨页面⟩]{⟨n⟩}{⟨H⟩}
```

这相当于：\Ncolumnstop[⟨页面⟩]{static}{⟨n⟩}{⟨H⟩}。

```
\NcolumnDtop[⟨页面⟩]{⟨n⟩}{⟨H⟩}
```

这相当于：\Ncolumnstop[⟨页面⟩]{dynamic}{⟨n⟩}{⟨H⟩}。

```
\onecolumnSbottominarea[⟨页面⟩]{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：

```
\onecolumnbottominarea[⟨页面⟩]{static}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

```
\onecolumnDbottominarea[⟨页面⟩]{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：

```
\onecolumnbottominarea[⟨页面⟩]{dynamic}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

```
\onecolumnbottom[⟨页面⟩]{⟨类型⟩}{⟨H⟩}
```

类似于 \onecolumnbottominarea，其中的区域指的是整个版心。

```
\onecolumnSbottom[⟨页面⟩]{⟨H⟩}
```

这相当于：\onecolumnbottom[⟨页面⟩]{static}{⟨H⟩}。

```
\onecolumnDbottom[⟨页面⟩]{⟨H⟩}
```

这相当于：\onecolumnbottom[⟨页面⟩]{dynamic}{⟨H⟩}。

```
\twocolumnSbottominarea[⟨页面⟩]{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：\twocolumnbottominarea[⟨页面⟩]{static}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}

```
\twocolumnDbottominarea[⟨页面⟩]{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：\twocolumnbottominarea[⟨页面⟩]{dynamic}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}

```
\twocolumnbottom[⟨页面⟩]{⟨类型⟩}{⟨H⟩}
```

类似于\twocolumnbottominarea，其中的区域指的是整个版心。

```
\twocolumnSbottom[⟨页面⟩]{⟨H⟩}
```

这相当于：\twocolumnbottom[⟨页面⟩]{static}{⟨H⟩}。

```
\twocolumnDbottom[⟨页面⟩]{⟨H⟩}
```

这相当于：\twocolumnbottom[⟨页面⟩]{dynamic}{⟨H⟩}。

```
\NcolumnSbottominarea[⟨页面⟩]{⟨n⟩}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：\Ncolumnbottominarea[⟨页面⟩]{static}{⟨n⟩}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}。

```
\NcolumnDbottominarea[⟨页面⟩]{⟨n⟩}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}
```

这相当于：\Ncolumnbottominarea[⟨页面⟩]{dynamic}{⟨n⟩}{⟨H⟩}{⟨宽度⟩}{⟨高度⟩}{⟨x⟩}{⟨y⟩}。

```
\Ncolumnbottom[⟨页面⟩]{⟨类型⟩}{⟨n⟩}{⟨H⟩}
```

类似于 `\Ncolumnbottominarea`，当然其中的区域指的是整个的版心。

```
\NcolumnSbottom[⟨页面⟩]{⟨n⟩}{⟨H⟩}
```

这相当于：`\Ncolumnbottom[⟨页面⟩]{static}{⟨n⟩}{⟨H⟩}`。

```
\NcolumnDbottom[⟨页面⟩]{⟨n⟩}{⟨H⟩}
```

这相当于：`\Ncolumnbottom[⟨页面⟩]{dynamic}{⟨n⟩}{⟨H⟩}`。

5.3 从右到左的栏

上面定义的命令的默认行为是从左到右创建流框架。但是，如果你是从右到左排版，则可能更喜欢以相反的顺序定义流框架。这可以通过包选项 `RL` 来完成。或者，你可以使用以下命令：

```
\lefttorightcolumnsfalse
```

在使用诸如 `\twocolumn` 或 `\Ncolumn` 之类的命令之前。可使用以下命令切换回从左到右的栏：

```
\lefttorightcolumnstrue
```

5.4 背景特效

静态框架可用于产生背景。有许多命令可以创建可用作背景的静态框架。在以下定义中，`⟨页面⟩` 是定义的静态框架的页面列表（`all` 为默认值）。对于垂直条：`⟨xoffset⟩` 是框架水平移动的值（默认为 `0pt`），`⟨W1⟩` 是第一个框架的宽度，颜色由 `⟨C1⟩` 和 IDL `⟨L1⟩` 指定，依此类推，直到 `⟨Wn⟩` 为第 `⟨n⟩` 个框架的宽度，其颜色由 `⟨Cn⟩` 和 IDL `⟨Ln⟩` 指定。对于垂直条：`⟨yoffset⟩` 是框架垂直移动的值（默认为 `0pt`），`⟨H1⟩` 是第一框架的高度，颜色由 `⟨C1⟩` 和 IDL `⟨L1⟩` 指定，依此类推，直到 `⟨Hn⟩` 为第 `⟨n⟩` 个框架的高度，其颜色由 `⟨Cn⟩` 和 IDL `⟨Ln⟩` 指定。

注意：与以前的命令不同，这些命令都是相对于实际页面的，而不是相对于版心。因此，`x` 偏移量为 `0pt` 表示第一个垂直框架与页面的左边缘齐平，而 `y` 偏移量为 `0pt` 表示第一个水平框架与页面的底部边缘齐平。这是因为背景往往会覆盖整个页面，而不仅仅是整个版心。

颜色规范必须完全用大括号括起来，例如 `[rgb]1,0,1` 而不是 `[rgb]1,0,1`。

5.4.1 垂直条效果

```
\vtwotone[⟨页面列表⟩][⟨x 偏移量⟩]{⟨W1⟩}{⟨C1⟩}{⟨L1⟩}{⟨W2⟩}{⟨C2⟩}{⟨L2⟩}
```

定义两个静态框架以创建两种颜色的垂直条纹。（此命令用于在本文档标题页上创建彩色背景。）

```
\vNtone[⟨页面列表⟩][⟨x 偏移量⟩]{⟨n⟩}{⟨W1⟩}{⟨C1⟩}{⟨L1⟩}...{⟨Wn⟩}{⟨Cn⟩}{⟨Ln⟩}
```

这类似于 `\vtwotone`，但是是 n 个静态框架而不是两个。

```
\vtwotonebottom[⟨页面列表⟩][⟨x 偏移量⟩]{⟨H⟩}{⟨W1⟩}{⟨C1⟩}{⟨L1⟩}{⟨W2⟩}{⟨C2⟩}{⟨L2⟩}
```

这与 `\vtwotone` 相似，但是静态框架的高度仅为 $\langle H \rangle$ ，而不是页面的整个高度。框架沿着页面底部边缘对齐。

```
\vtwotonetop[⟨页面列表⟩][⟨x 偏移量⟩]{⟨H⟩}{⟨W1⟩}{⟨C1⟩}{⟨L1⟩}{⟨W2⟩}{⟨C2⟩}{⟨L2⟩}
```

这与 `\vtwotone` 相似，但是静态框架的高度仅为 $\langle H \rangle$ ，而不是页面的整个高度。框架沿着页面顶部边缘对齐。（此命令用于在本文档每页的顶部创建边框效果。使用了两个 `\vtwotonetop` 命令，一个用于偶数页，另一个用于奇数页。）

```
\vNtonebottom[⟨页面列表⟩][⟨x 偏移量⟩]{⟨n⟩}{⟨W1⟩}{⟨C1⟩}{⟨L1⟩}...{⟨Wn⟩}{⟨Cn⟩}{⟨Ln⟩}  
\vNtonetop[⟨页面列表⟩][⟨x 偏移量⟩]{⟨n⟩}{⟨W1⟩}{⟨C1⟩}{⟨L1⟩}...{⟨Wn⟩}{⟨Cn⟩}{⟨Ln⟩}
```

这是用于 n 个框架的通用版本。

5.4.2 水平条效果

```
\htwotone[⟨页面列表⟩][⟨y 偏移量⟩]{⟨H1⟩}{⟨C1⟩}{⟨L1⟩}{⟨H2⟩}{⟨C2⟩}{⟨L2⟩}
```

这定义了两个静态框架以创建两种颜色的水平条纹。

```
\hNtone[⟨页面列表⟩][⟨y 偏移量⟩]{⟨n⟩}{⟨H1⟩}{⟨C1⟩}{⟨L1⟩}...{⟨Hn⟩}{⟨Cn⟩}{⟨Ln⟩}
```

这类似于 `\htwotone`，但是是 n 个静态框架而不是两个。

```
\htwotoneleft[⟨页面列表⟩][⟨y 偏移量⟩]{⟨W⟩}{⟨H1⟩}{⟨CI⟩}{⟨LI⟩}{⟨H2⟩}{⟨C2⟩}{⟨L2⟩}
```

这类似于 `\htwotone`，但是静态框架只有 $\langle W \rangle$ 宽，而不是页面的整个宽度。框架沿页面的左边缘对齐。

```
\htwotoneright[⟨页面列表⟩][⟨y 偏移量⟩]{⟨W⟩}{⟨H1⟩}{⟨CI⟩}{⟨LI⟩}{⟨H2⟩}{⟨C2⟩}{⟨L2⟩}
```

这类似于 `\htwotone`，但是静态框架只有 $\langle W \rangle$ 宽，而不是页面的整个宽度。框架沿页面的右边缘对齐。

```
\hNtoneleft[⟨页面列表⟩][⟨y 偏移量⟩]{⟨W⟩}{⟨n⟩}{⟨H1⟩}{⟨CI⟩}{⟨LI⟩}...{⟨Hn⟩}{⟨Cn⟩}{⟨Ln⟩}  
\hNtoneright[⟨页面列表⟩][⟨y 偏移量⟩]{⟨W⟩}{⟨n⟩}{⟨H1⟩}{⟨CI⟩}{⟨LI⟩}...{⟨Hn⟩}{⟨Cn⟩}{⟨Ln⟩}
```

这是用于 n 个框架的通用版本。

5.4.3 背景框架

要制作一个覆盖整个页面的静态框架，请使用：

```
\makebackgroundframe[⟨页面⟩][⟨IDL⟩]
```

请注意，应该在创建任何其他静态框架之前创建此框架，因为如果给定背景色，它将掩盖在其之前创建的所有静态框架。

5.4.4 垂直和水平线

你可以使用以下命令在两个框架之间创建垂直或水平线：

```
\insertvrule[⟨y top⟩][⟨y bottom⟩]{⟨T1⟩}{⟨IDN1⟩}{⟨T2⟩}{⟨IDN2⟩}
```

这将创建一个新的静态框架，该框架介于 IDN 为 $\langle IDN1 \rangle$ 的 $\langle T1 \rangle$ 框架和 IDN 为 $\langle IDN2 \rangle$ 的 $\langle T2 \rangle$ 框架之间，并在其中放置一条垂直线，从最高框架的最高点到最低框架的最低点。第一个可选参数 $\langle y top \rangle$ （默认值为 `0pt`）将直线扩展到最高点以上，第二个可选参数 $\langle y bottom \rangle$ （默认值为 `0pt`）将直线扩展到最低点以下。如果任一可选参数为负，则将缩短直线而不是扩展直线。直线粗细由命令

```
\ffcolumnseprule
```

给出。请注意，此更改自 1.09 版开始，1.09 之前的版本使用 `\columnseprule`。

`\insertvrule` 绘制的垂直直线是使用以下命令创建的：

```
\ffvrule{offset}{width}{height}
```

这可以通过重定义来得到更好看的直线（请参见下文）。

```
\inserthrule[⟨x left⟩][⟨x right⟩]{⟨T1⟩}{⟨IDN1⟩}{⟨T2⟩}{⟨IDN2⟩}
```

这将创建一个新的静态框架，该框架适合 IDN 为 $\langle IDN1 \rangle$ 的 $\langle T1 \rangle$ 框架和 IDN 为 $\langle IDN2 \rangle$ 的 $\langle T2 \rangle$ 框架，并在其中放置一条水平直线，从左框架的最左点延伸到右框架的最右点。第一个可选参数 $\langle x left \rangle$ （默认值为 `0pt`）将直线向左扩展最左边的点，第二个可选参数 $\langle x right \rangle$ （默认值为 `0pt`）将直线扩展最右边的点。如果任一可选参数为负，则将缩短直线而不是扩展直线。直线的高度由

```
\ffcolumnseprule
```

给出。

`\inserthrule` 绘制的水平直线使用以下命令创建：

```
\ffhrule{offset}{width}{height}
```

这可以通过重定义来得到更好看的直线。

`\ffcolumnseprule` 的默认值为 `2pt`。`\insertvrule` 和 `\inserthrule` 都有星型版本，可让你通过 IDL（而不是 IDN）来标识框架。框架类型 $\langle T1 \rangle$ 和 $\langle T2 \rangle$ 可以是以下关键字之一：`flow`，`static` 或 `dynamic`。

可以重新定义命令 `\ffruleddeclarations`，以设置直线绘制方式的声明。此命令最可能的用途是设置直线颜色。例如：

```
\twocolumnStop{2in}
```

```
\renewcommand{\ffruleddeclarations}{\color{red}}
```

```
\insertvrule{flow}{1}{flow}{2}
```

```
\renewcommand{\ffruleddeclarations}{\color{blue}}
```

```
\inserthrule{static}{1}{flow}{1}
```

这将创建一个包含两栏（流框架 1 和 2）以及上面的静态框架的布局。将红色垂直直线放置在流框架 1 和 2 之间的静态框架中，将蓝色水平直线放置在静态框架和第一个流框架之间。（但是，水平直线将跨越两个流框架，因为那是静态框架的宽度。）

在以下示例中，已将直线重新定义为锯齿形（可通过 `tikz` 包获得）：

```
\usepackage{flowfram}
\usepackage{tikz}
\usetikzlibrary{snakes}

\twocolumnStop

\renewcommand{\ffvrule}[3]{%
\hfill
\tikz{\draw[snake=zigzag,line width=#2,yshift=-#1] (0,0) -- (0pt,#3);}%
\hfill\mbox{}}

\insertvrule{flow}{1}{flow}{2}

\renewcommand{\ffhrule}[3]{%
\tikz{\draw[snake=zigzag,line width=#3,xshift=-#1] (0,0) -- (#2,0pt);}}

\inserthrule{static}{1}{flow}{1}
```


6.1 缩略图

在此页面的右侧，有一个带有章号的蓝色矩形。这是一个缩略图，它使你可以快速浏览页面时大致了解文档中的下落。实际上，每个缩略图都是一个动态框架，你可以使用包选项 `thumbtabs` 来控制是否在缩略图中显示数字和（或）标题。这是一个 `key=value` 选项，其中值可以是 `title`（显示标题但不显示数字—默认值），`number`（显示数字但不显示标题），`both`（显示数字和标题）和 `none`（不显示数字或标题）的值之一。

如果想要文档有缩略图，可以在文档导言区使用命令

```
\makethumbtabs[⟨y 偏移量⟩][⟨高度⟩][⟨章节类型⟩]
```

默认情况下，最上方的缩略图与版心的顶部处于同一水平，但可以使用第一个可选参数 `⟨y 偏移量⟩` 垂直移动。每个缩略图的高度将为 `⟨高度⟩`，并将与类型 `⟨章节类型⟩` 相对应。如果省略 `⟨章节类型⟩`，则在定义 `\chapter` 命令时将使用章，否则将使用节。缩略图宽度由长度

```
\thumbtabwidth
```

给出，其默认值为 `1cm`。`\thumbtabindex` 命令将在当前页面上显示缩略图索引（所有缩略图）。然后，你需要使用

```
\enablethumbtabs
```

来启用各个缩略图选项卡，并使用

```
\disablethumbtabs
```

使其消失。你可以使用命令

```
\tocandthumbtabindex
```

而不是命令 `\tableofcontents` 和 `\thumbtabindex` 将目录与缩略图对齐¹。如果使用 `hyperref` 包，则缩略图索引上的文本将是指向文档相应部分的超链接。请注意，在使用 `\tocandthumbtabindex` 时，可能需要垂直向上或向下移动缩略图，以确保它们与目录正确对齐。

缩略图选项卡上的文本格式由命令

¹但是只有在页面上有足够的空间时才这样做。

6. 缩略图和小目录

6.1 缩略图 45

6.2 小型目录 47

本章介绍如何创建缩略图和小目录，例如本文档中使用的那些。

```
\thumbtabindexformat
```

用于缩略图选项卡索引条目，而

```
\thumbtabformat
```

用于单个缩略图选项卡。默认情况下，缩略图上的文本将被旋转，但是由于某些预览器未实现旋转，因此提供了包选项 `norotate`，它将垂直放置字母。这看起来不如旋转文本。另请注意，旋转链接时，某些预览器不会将超链接放置在正确的位置，因此这也可能会引起问题。

可以使用命令

```
\setthumbtab{<n>}{<键值对列表>}
```

来更改缩略图属性，其中 $\langle n \rangle$ 是从 1（对于顶部的缩略图）开始到由计数器 `maxthumbtabs`（对于底部的缩略图）给出的值的编号。请注意，这些数字与关联框架的 IDN 不相关。你也可以使用关键字 `all` 而不是 $\langle n \rangle$ 来指示新属性应用于所有缩略图。

要仅更改缩略图索引的设置，请使用

```
\setthumbtabindex{<n>}{<键值对列表>}
```

这两个命令的 \langle 键值对列表 \rangle 与 `\setdynamicframe` 的 \langle 键值对列表 \rangle 相同。同样， $\langle n \rangle$ 可以是缩略图索引或关键字 `all`。

默认情况下，缩略图具有灰色背景。在本文档中，我们使用了

```
\setthumbtab{1}{backcolor=[rgb]{0.15,0.15,1}}
\setthumbtab{2}{backcolor=[rgb]{0.2,0.2,1}}
\setthumbtab{3}{backcolor=[rgb]{0.25,0.25,1}}
\setthumbtab{4}{backcolor=[rgb]{0.3,0.3,1}}
\setthumbtab{5}{backcolor=[rgb]{0.35,0.35,1}}
\setthumbtab{6}{backcolor=[rgb]{0.4,0.4,1}}
\setthumbtab{7}{backcolor=[rgb]{0.45,0.45,1}}
\setthumbtab{8}{backcolor=[rgb]{0.5,0.5,1}}
```

将缩略图背景颜色更改为蓝色。

我们还使用以下方法更改了缩略图文本的样式：

```
\newcommand{\thumbtabstyle}[1]{\hypersetup{linkcolor=white}%  
\textbf{\large\sffamily #1}}}  
\setthumbtab{all}{style=thumbtabstyle,textcolor=white}
```

请注意，该样式使用 `\hypersetup`² 更改超链接文本的颜色，因为超链接将覆盖文本颜色。

6.2 小型目录

在本文档中，每章标题之后都有该章的小型目录。要启用小型目录，请使用以下命令

```
\enableminitoc[<章节类型>]
```

默认节类型与缩略图选项卡使用的节类型相同。

如果你希望小型目录出现在动态框架中，则可以使用

```
\appenddfminitoc{<IDN>}
```

其中，`<IDN>` 是适当的动态框架的 IDN。如果要使用 IDL 代替 IDN，也可以使用带星号的版本。

例如，在本文档中，我们在导言区中使用了以下命令

```
\appenddfminitoc*{chaphead}
```

它已将小型目录放置到 IDL 为 `chaphead` 的动态框架中。

小型目录文本的样式由命令

```
\minitocstyle{<内容>}
```

指定，其中的参数是小型目录内容的设置。如果要更改小型目录，可以重新定义此命令。小型目录之前的间隙由长度

```
\beforeminitocskip
```

给出，而小型目录之后的间隙由长度

²由 `hyperref` 宏包定义

```
\afterminitocskip
```

给出，这些长度可以使用命令 `\setlength` 进行更改。

7.1 宏

以下宏可以使用 `\renewcommand` 修改：

- `\setffdraftcolor`
在草稿模式下显示时，这将设置边框的颜色。默认值为：`\color[gray]{0.8}`。例如，如果你想要更深的灰色，请执行以下操作：

```
\renewcommand{\setffdraftcolor}{\color[gray]{0.3}}
```

- `\setffdrafttypeblockcolor`
在草稿模式下显示时，可以设置字体块的边框的颜色。默认值为：`\color[gray]{0.9}`。例如，如果要使用中等灰色，请执行以下操作：

```
\renewcommand{\setffdrafttypeblockcolor}{\color[gray]{0.5}}
```

- `\fflabelfont`
这将在草稿模式下设置边框标记的字体样式。默认值为：`\small\sffamily`。例如，如果你想要更大的字体，请执行以下操作：

```
\renewcommand{\fflabelfont}{\large\sffamily}
```

- `\ffruleddeclarations`
这将设置影响使用 `\insertvrule` 和 `\inserthrule` 创建的直线的声明。默认定义不执行任何操作。有关更多详细信息，请参见 5.4.4 小节。

- `\ffcontinuedtextfont<text>`
这将 `<text>` 设置为延续字体样式。默认定义为 `\emph{\small <text>}`。有关更多详细信息，请参见 2.2 节和 2.3 节。

7.2 长度

以下是长度，可以使用 `\setlength` 进行更改：

7. 全局设置

7.1 宏	49
7.2 长度	49
7.3 计数器	51

本节描述了 *flowfram* 包使用的样式宏，长度和计数器。

- `\fflabelsep`
这是从边界框右侧放置边界框标记的距离。默认值为 `1pt`。
- `\flowframesep`
对于标准边框类型，这是框架文本与其边框之间的间隙。
- `\flowframerule`
如果是使用框架制作命令给定的边框，该命令使用 `\fboxsep` 设置边框宽度（例如 `\fbox`），则这就是边框的宽度。
- `\sdfparindent`
这是静态或动态框架内的段落缩进。默认值为 `0pt`。
- `\vcolumnsep`
这是使用 `\Ncolumn` 等这些命令时顶部框架和列框架之间的近似垂直距离。（可以将流框架的高度调整为 `\baselineskip` 的整数倍。）
- `\columnsep`
这是使用 `\Ncolumn` 或 `\Ncolumn` 等命令时列框架之间的水平距离。
- `\ffcolumnseprule`
这是使用 `\insertvrule` 创建的垂直直线的宽度或使用 `\inserthrule` 创建的水平直线的高度。
- `\beforeminitocskip`
这是小型目录之前的垂直距离。
- `\afterminitocskip`
这是小型目录之后的垂直距离。
- `\fftolerance`
当段落跨越两个宽度不相等的流框架时，输出例程将发出警告，除非宽度差小于 `\fftolerance` 的值。

7.3 计数器

以下是可以通过 `\value{<计数器名>}` 或通过 `\the<计数器名>` 访问的计数器。但是，不应修改这些计数器的值。

maxflow 到目前为止已定义的流框架总数。

thisframe 存储当前流框架的 IDN。你可以使用以下方式标记和引用 IDN：

```
\labelflowid{<标签>}
```

这类似于标准 `\label` 命令，但将始终引用当前流框架的 IDN。然后可以使用 `\ref{<标签>}` 进行引用。请注意，即使在静态或动态框架的内容中使用，它始终将引用当前流框架。在给定页面的给定流框架中，请勿使用多个 `\labelflowid` 实例，否则你会收到“multiply defined references”警告。

displayedframe 存储当前显示的流框架的索引。如果所有流框架都显示在当前页面上，则这些 IDN 相同，但是如果某些流框架被隐藏，则值可能会不同。你可以使用

```
\labelflow{<标签>}
```

并使用 `\ref{<标签>}` 在文档中的其他位置引用它。例如，如果你使用列布局，则可能需要执行以下操作：

```
This text is about hippos\labelflow{hippos}.
```

```
% Somewhere else in the document  
See column~\ref{hippos} on page~\pageref{hippos}  
for information on hippos.
```

在给定页面的给定流框架中，请勿使用多个 `\labelflow` 实例，否则你会收到“multiply defined references”警告。请注意，即使在静态或动态框架的内容中使用 `\labelflow`，它也始终引用当前的流框架。

maxstatic 到目前为止已定义的静态框架总数。

maxdynamic 到目前为止已定义的动态框架总数。

maxthumbs 缩略图的总数。

absolute page 绝对页码。

有关常见问题的最新列表, 请访问 <http://www.dickimaw-books.com/faqs/flowframfaq.html>。如果你有一个此处未解决的问题, 请首先尝试。如果那不能回答你的问题, 请尝试在 StackExchange (<http://tex.stackexchange.com/>), L^AT_EX 社区论坛 (<http://latex-community.org/forum/>) 或 comp.text.tex 新闻组。通常, 在那些地方回答问题的速度比通过电子邮件发送给我的查询要快得多, 后者往往会在我的收件箱中遗失。

8.1 一般查询

1. 如果仅在例如 1–10 页上定义了所有流框, 那么如果我添加一些额外的文本以使文档超过 10 页会发生什么情况?
输出例程将创建一个新的流框, 并使用它。
2. 我可以在页面列表中使用格式化的页码吗?
不能。
3. 为什么不能?
当输出例程以一个流框架结束时, 它将寻找该页面上定义的下一个流框架。如果没有剩余内容, 它将搜索所有已定义流框的页面列表, 以查看下一页是否在该范围内。如果在该页面上未定义任何内容, 它将输出该页面, 然后尝试下一页。这引起了两个问题:
 - (a) L^AT_EX 不是千里眼。如果当前在第 14 页上, 而在下一页上, 页码更改为 A, 则直到到达该点为止, 它都无法知道。因此, 它正在寻找在第 15 页而不是在 A 页上定义的流框架。
 - (b) L^AT_EX 如何分辨页面 C 是否位于页面 A 和 D 之间? 它将需要一种算法, 该算法可以将带格式的数字转换回整数。鉴于存在多种不同的格式化计数器值的方式 (标准的罗马和字母格式除外), 因此无法编写算法来对某种任意格式进行格式化。
4. 我可以有一个任意形状的框架吗?
你可以将某些不规则形状分配给静态或动态框架 (请参阅 3.1 节)。请注意, 边界框仍将显示为矩形, 具有给定框架的尺寸, 该尺寸可能与分配的形状不对应。此功能不适用于流框。
5. 为什么流框架中的文本出现在静态框架或动态框架中?
(原文太长了, 看原文吧! 这一大段简直比做一篇阅读理解还难。。)

8. 故障排除

8.1 一般查询	53
8.2 意外的输出	54
8.3 错误消息	56

如果你在使用 *flowfram* 包时遇到任何问题, 请查阅本章。

6. 为什么我会收到很多 `hbox` 的消息？
可能是因为框架较窄。框架狭窄时，最好使用左对齐格式。
7. 为什么我不断收到多重定义的警告？
可能是显示在一页以上的静态或动态框架中使用了 `\label`。尝试使用 `clear` 确保始终在每页末尾清除边框。
8. 如果我使用切换到两栏模式的命令或环境（例如 `theindex`），会发生什么？
从 1.01 版开始，在序言之外出现的任何 `\onecolumn` 或 `\twocolumn` 命令都将打印可选参数的内容，并发出警告。建议设置自己的框架以在索引中使用。有关示例，请参见本文档的源代码 `ffuserguide.tex`。
9. 如何更改静态或动态框架内容的垂直对齐方式？
使用 `\setstaticframe` 或 `\setdynamicframe`（1.03 版的新功能）中的 `valign`。
10. 如何计算距页面边缘而不是版心的距离？
查看 4.1 节。
11. 是否可以使用 GUI 来简化框架的创建？
是的，可以从以下网站下载 `flowframtk`: <http://www.dickimaw-books.com/apps/flowframtk/>。

8.2 意外的输出

1. 我的流框架开始处的线宽错误。
如果有不同宽度的流框架，则会出现此问题。因为 `\hsize` 的更改直到段落中断才生效。因此，如果有一个跨越两个流框架的段落，则在第二个流框架的开头的段落结尾将保留在前一个流框架的底部的段落开头的宽度。可以通过在发生框架中断的位置插入 `\framebreak` 来解决此问题（请参阅 2.1.1 节）。
2. 添加边框时，我的框架向右移动。
如果使用了 `flowfram` 程序包无法识别的边框，则可能会发生这种情况。你需要使用 `offset` 设置偏移量（请参见第 3 章）。

3. 每页右侧都有一个垂直的白色条带。

如果你有一个 A4 文档，并且 `ghostscript` 将信件尺寸作为默认纸张尺寸，则可能会发生这种情况。你可以通过编辑文件 `gs_init.ps` 来更改默认的纸张尺寸。更改：

```
% Optionally choose a default paper size other than U.S. letter.  
% (a4)
```

为

```
% Optionally choose a default paper size other than U.S. letter.  
(a4)
```

4. 我没有得到任何输出。

你所有的流框架都是空的。`TEX` 在将文本放入流框架之前，不会将框架放在页面上。因此，如果流框架中没有文字，它将不会输出页面。如果只希望填充静态框架或动态框架，而其他地方什么都没有，则只需 `\mbox{}\clearpage`。这样会将零面积的不可见物体放入流框架，但这足以使 `TEX` 确信该文档包含一些文本。

5. 没有出现最后一页。

参阅上一个答案。

6. 我的静态或动态框架内没有段落缩进。

静态或动态框架中的段落缩进由长度 `\sdfparindent` 控制，该长度默认情况下设置为 `0pt`。为了使缩进与流框架使用的缩进相同，请在导言区中放置以下内容：

```
\setlength{\sdfparindent}{\parindent}
```

7. 我的节编号顺序错误。

请记住，在页面输出之前，不会设置动态框架的内容，并且内容将按照 IDN 的顺序进行设置，因此，如果在动态框架中出现任何分节命令，则可能无法将它们设置为同一部分，按照它们在输入文件中的顺序排列。

8. 当我使用 `\parshape` 时，静态或动态框架的内容已向左移动。

如果你的 `\parshape` 设置超出了线宽，则会发生这种情况。例如：

`\parshape=1 0.4\linewidth 0.7\linewidth`

这指定总长度为 `1.1\linewidth` 的行，该行太长。

8.3 错误消息

1. Illegal unit of measure (pt inserted)

所有长度必须有单位。定义新框架时，切记要包括单位。如长度：`width`, `height`, `x`, `y` 和 `offset`¹。

2. Missing number, treated as zero

\LaTeX 需要一个数字。有许多可能的原因：

- (a) 你使用的是 IDL 而不是 IDN。如果要通过标签来引用框架，则需要记住使用 `\set<类型>frame` 命令的星型版本，或者在设置静态框架或动态框架的内容时使用。
- (b) 指定页面列表时，将关键字与页面范围混合在一起。例如：`1, even` 无效。

3. Flow frame IDL '*<label>*' already defined

每种框架类型中的所有 IDL 必须唯一。对于静态框架和动态框架的重复 IDL，也存在类似的错误消息。

4. Can't find flow frame id

指定了不存在的流框架 IDL。对于静态框架和动态框架也有类似的错误消息。检查以确保正确拼写了标签，并检查是否使用了正确的框架类型命令。（例如，如果静态框架 IDL 为 `mylabel`，而你尝试执行 `\setflowframe*{mylabel}{<选项>}`，则会出现此错误，因为 `mylabel` 引用静态框架而不是流框架。）

5. Key 'clear' is boolean

`clear` 只能具有 `true` 或 `false` 值。

6. Key 'clear' not available

`clear` 仅适用于静态或动态框架。

¹`offset` 也可以是 `computer` 值。

7. Key 'style' not available
style 仅适用于动态框架。
8. Key 'margin' not available
margin 仅适用于流框架。
9. Key 'shape' not available
shape 仅适用于静态或动态框架。
10. Dynamic frame style '*style*' not defined
指定的样式 *style* 必须是不带反斜杠的命令名称。你可能输入了错误的名称，或者忘记了定义命令。
11. Argument of \fbox has an extra }
如果你写的是 `border=\fbox` 而不是 `border=fbox`，则会发生此错误。切记不要包括初始反斜杠。
12. Not in outer par mode
静态或动态框架中不能有浮动体（例如图形，表格或边注）。如果要在静态或动态框架内使用图形或表格，请使用 `staticfigure` 或 `statictable`。
13. Somethings wrong---maybe missing \item
假设所有列表类型的环境都以 `\item` 开头，则可能是由于上一次运行中的 `toc`（目录），`ttb`（thumbtab）或 `aux`（辅助）文件出了问题。尝试删除它们，然后重试。
14. No room for a new \skip
你已超过 $\text{T}_{\text{E}}\text{X}$ 的 256 个寄存器限制。使用 `etex` 包。
15. \verb illegal in command argument
通常，不能在命令参数中使用抄录文本。此规则适用于 `flowfram` 包定义的所有命令。另请参见下文。
16. I get \verb illegal in command argument when using verbatim text inside the dynamiccontents environment.
不能在动态内容环境的已加星标或未加星标的版本中使用抄录文本。（请参阅第 11 页。）



bounding box

框架的边界框是为该框架的内容分配的区域。但是，文本可能无法完全填充该区域，并且文本可能会溢出该区域。

dynamic frame

文本固定在其中的框架，但是每次显示该框架时都要重新排版内容。

flow frame

文档中的框架，以使文档环境的内容按照定义的顺序从一个框架流向下一个框架。每页上至少必须有一个流框架。

frame making command

\LaTeX 命令，在其参数周围放置某种边框。例如：`\fbox`。

frame

页面的矩形区域，可以在其中放置文本（不要与制框命令相混淆）。共有三种类型：流框架，静态框架和动态框架。

identification label (IDL)

分配给框架的唯一标签，使你可以按标签而不是 IDN 引用框架。

identification number (IDN)

分配给每个框架的唯一编号，可在修改框架外观时用来标识框架。例如：如果定义了 3 个流框架，2 个静态框架和 1 个动态框架，则流框架将具有 IDN 1、2 和 3，静态框架将具有 IDN 1 和 2，而动态框架将具有 IDN 1。

page list

页面列表。它可以是单个关键字：`all`，`odd`，`even` 或 `none`，也可以是单个页码或页码范围的逗号分隔列表。例如：`<3,5,7-11,>15` 表示第 1, 2, 5, 7, 8, 9, 10, 11 页以及第 15 页之后的所有页。这些数字默认通过十进制值引用页计数器。要使它们引用绝对页码，请使用包选项 `pages=absolute`。

page range

页面范围可以是闭合的，例如 `5-10`，或开放的，例如 `<7` 或 `>9`。

static frame

固定文本的框架。内容是固定的，直到通过 `\setstaticcontents` 中的 `clear` 明确更改或删除为止。

typeblock

文本主体所在的页面区域。该区域的宽度和高度由 `\textwidth` 和 `\textheight` 给出。



