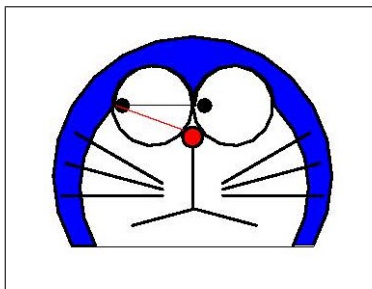


# 目录

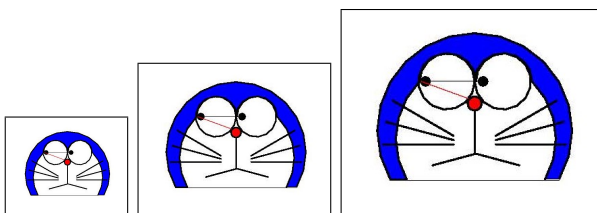
<b>1 图片参数</b>	<b>2</b>
1.1 图片原始大小	2
1.2 scale 表示图片缩放倍数 (0.3 倍/0.5 倍/0.7 倍)	2
1.3 width 表示图片宽度 (3cm/0.3 倍文本宽度)	2
1.4 height 表示图片高度 (1cm/0.1 倍页高)	2
1.5 angle 表示图片逆时针旋转角度, origin 表示图片旋转中心, 默认值为 l (左), 可以设置为 r (右)、c (中)、t (顶)、b (底)、B (基线)	3
1.5.1 绕左侧中心逆时针旋转 90 度	3
1.5.2 绕左侧中心顺时针旋转 90 度	3
1.5.3 绕右侧中心逆时针旋转 90 度	4
1.5.4 绕中央中心逆时针旋转 90 度	4
1.5.5 绕中心顶端中心逆时针旋转 90 度	5
1.5.6 绕底部中心逆时针旋转 90 度	5
1.5.7 绕基线中心逆时针旋转 90 度	6
1.6 设置图片到边框的距离 (原始尺寸/1cm)	6
<b>2 在文本中插入图片</b>	<b>7</b>
2.1 不使用浮动体环境	7
2.2 使用浮动体环境	7
<b>3 图文混排</b>	<b>7</b>
<b>4 并排插入图片</b>	<b>8</b>
4.1 并排插入共享标题的图片	8
4.2 并排插入拥有小标题的图片	9
4.3 并排插入拥有子编号的图片	11
4.3.1 subfig 宏包	11
4.3.2 subcaption 宏包	11
4.4 并排插入不同比例的图片	13
4.5 结论	15

## 1 图片参数

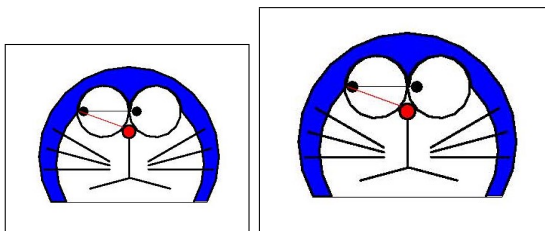
### 1.1 图片原始大小



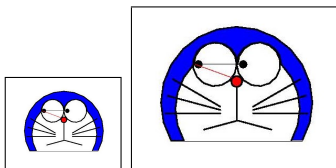
### 1.2 scale 表示图片缩放倍数（0.3 倍/0.5 倍/0.7 倍）



### 1.3 width 表示图片宽度（3cm/0.3 倍文本宽度）

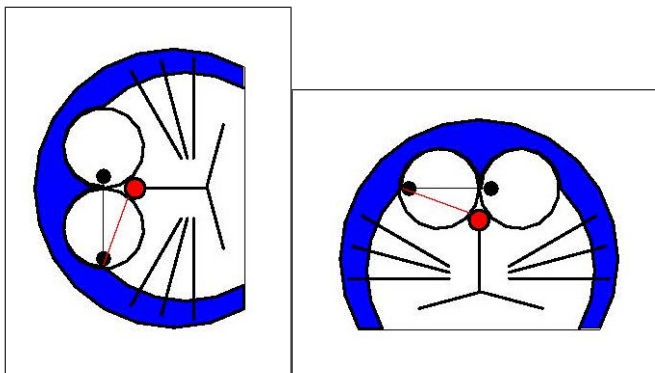


### 1.4 height 表示图片高度（1cm/0.1 倍页高）

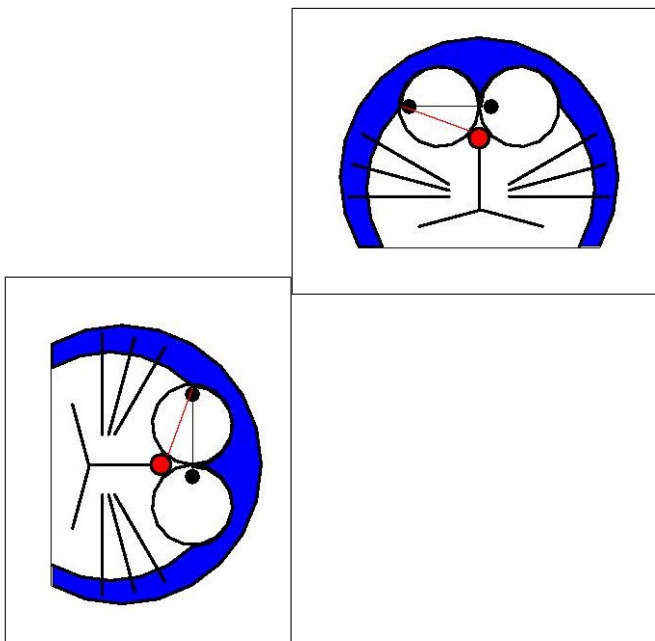


1.5 angle 表示图片逆时针旋转角度, origin 表示图片旋转中心, 默认值为 l (左), 可以设置为 r (右)、c (中)、t (顶)、b (底)、B (基线)

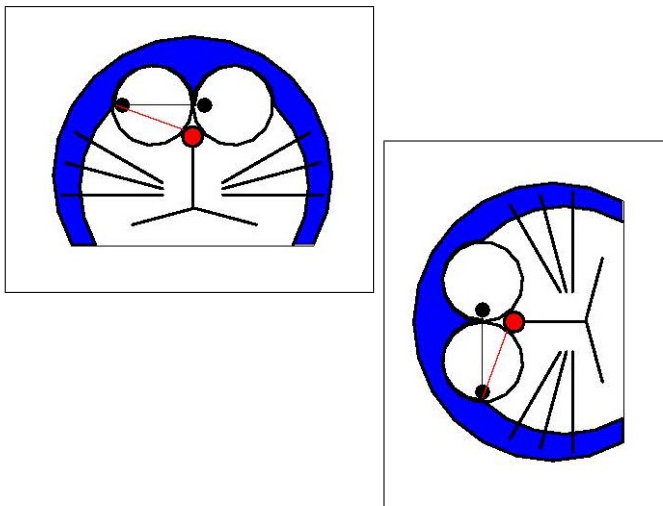
1.5.1 绕左侧中心逆时针旋转 90 度



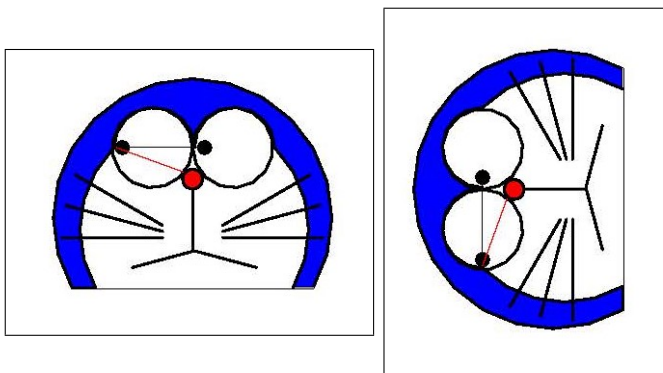
1.5.2 绕左侧中心顺时针旋转 90 度



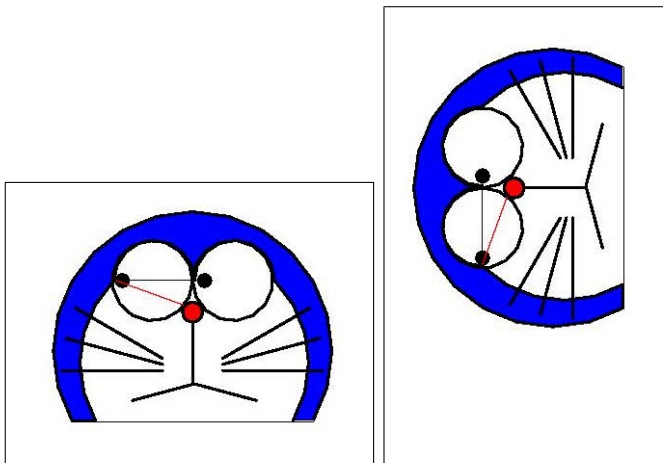
1.5.3 绕右侧中心逆时针旋转 90 度



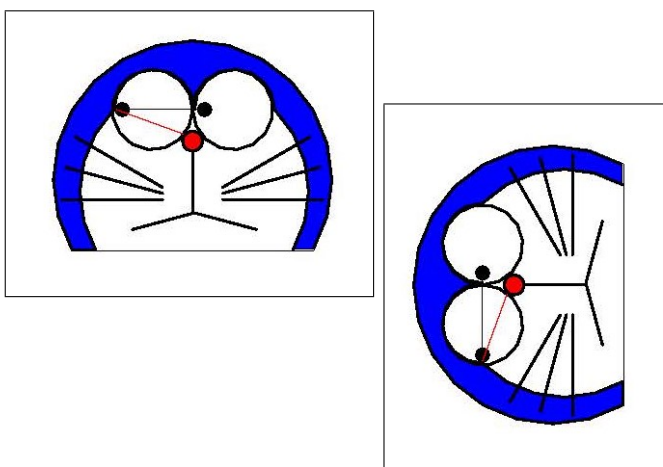
1.5.4 绕中央中心逆时针旋转 90 度



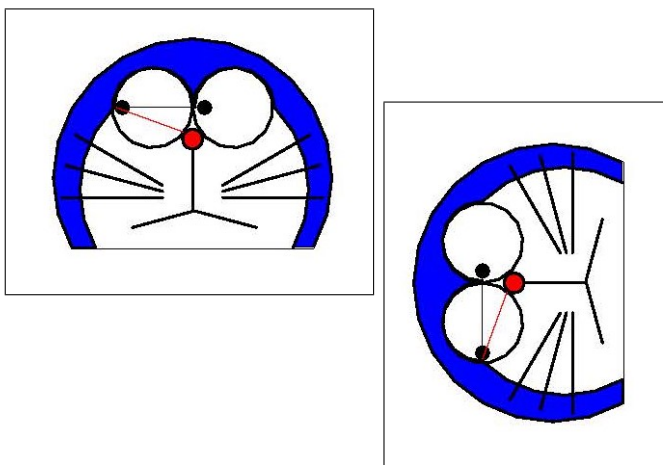
1.5.5 绕中心顶端中心逆时针旋转 90 度



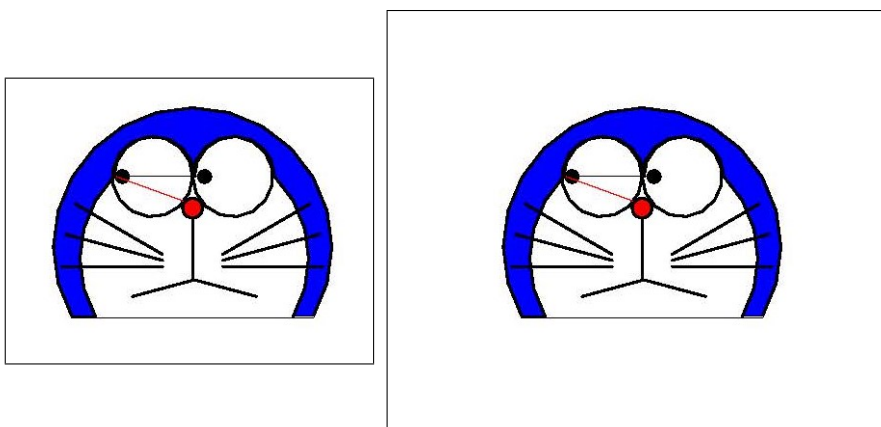
1.5.6 绕底部中心逆时针旋转 90 度



### 1.5.7 绕基线中心逆时针旋转 90 度

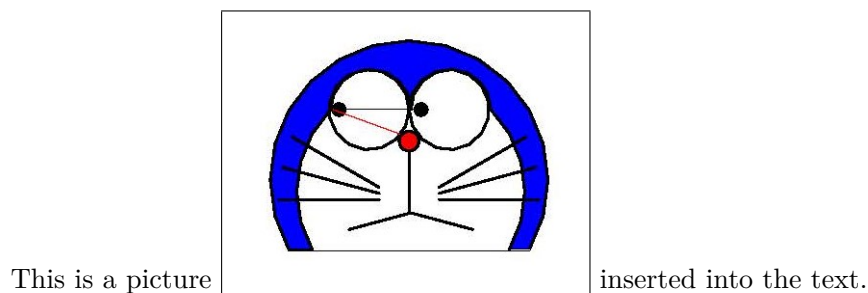


### 1.6 设置图片到边框的距离 (原始尺寸/1cm)



## 2 在文本中插入图片

### 2.1 不使用浮动体环境



### 2.2 使用浮动体环境

This is a picture independent of the text.

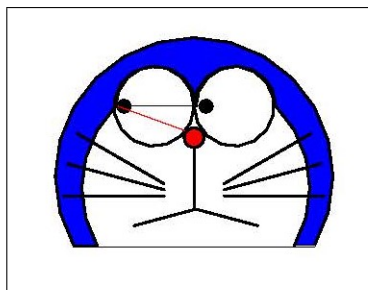
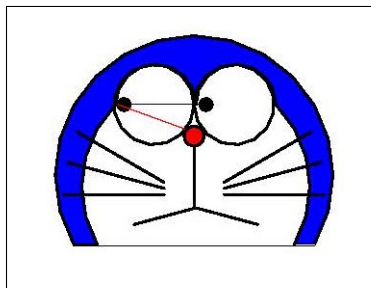


图 1: doraemon\_figure

## 3 图文混排

Texte texte texte texte texte texte texte texte texte texte  
texte texte texte texte texte texte texte texte... Texte texte texte texte  
texte texte texte texte texte texte texte texte texte texte texte  
texte texte texte...



Texte texte texte texte texte texte  
texte texte texte texte texte texte texte  
texte texte texte texte texte texte texte...  
Texte texte texte texte texte texte

图 2: doraemon\_wrapfigure

```

texte texte texte texte texte texte texte
texte texte texte texte texte texte... Texte
texte texte texte texte texte texte texte
texte texte texte texte texte texte texte
texte texte texte texte texte... Texte texte
texte texte texte texte texte texte texte
texte texte texte texte texte texte texte...

```

## 4 并排插入图片

### 4.1 并排插入共享标题的图片

使用  $\text{\LaTeX}$  原生命令中的 `figure` 环境和 `graphicx` 宏包提供的 “`\includegraphics[options]{name}`” 就可以完成简单的并排图片的插入，一般会在可选参数中设置 `width` 为文本宽度的倍数，如果不设置可选参数，则会将图片按照原始大小排列在一起，但是这很可能导致因为图片总宽度超过文本宽度而无法在一行上并排放置，下面展示不设置可选参数（3）和在可选参数中设置 `width` 为 0.25 倍文本宽度（4）两种情况的打印效果。

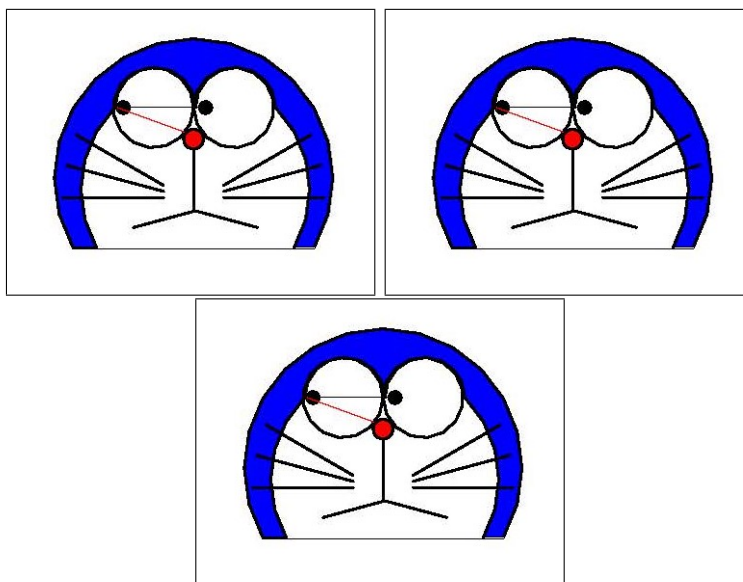


图 3: doraemon\_figure\_graphicx\_horizontal (1)



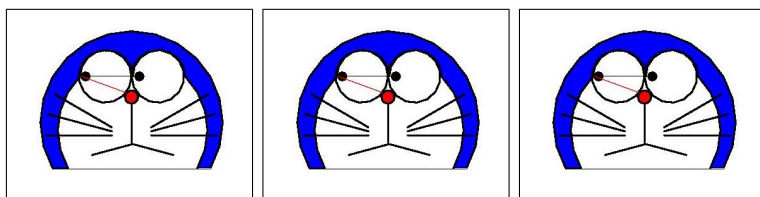


图 4: doraemon\_figure\_graphicx\_horizontal (2)

在 “`\includegraphics[options]{name}`” 命令之间插入 “`\hfill`” 命令，可以在图与图之间自动插入一段距离，使得两边的图片分别位于文本的左右边界（5）。

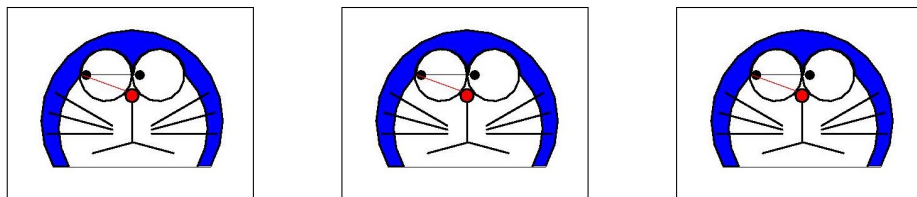


图 5: doraemon\_figure\_graphicx\_horizontal (3)

如果在 “`\includegraphics[options]{name}`” 命令之间插入 “`\\`” 符号，就可以达到像表格中一样的换行效果，使图片纵向排列（6）。

还可以在 “`\\`” 符号的后面加上用中括号括起来的距离长度，比如 “`[2ex]`”，来增加图片之间的距离（7）。

## 4.2 并排插入拥有小标题的图片

但是上述这种设置方式没有办法为每个小图设置小标题<sup>1</sup>，此时使用 `minipage` 环境将每个小图包起来，就可以在各自的 `minipage` 环境中通过 “`\caption{}`” 命令设置小标题（4.3.2）。注意，`minipage` 环境需要设置表示边框宽度的必选参数（通常就是需要设置的图片的宽度），否则编译会报错。相应地，原来 “`\includegraphics[options]{name}`” 命令的可选参数中的 `width` 的值应该增大，因为此时该命令放置在 `minipage` 环境中，可选参数

<sup>1</sup>情况其实有些复杂，应该分成两种情况来讨论：如果是并排插入图片，在每一条 “`\includegraphics[options]{name}`” 命令之后插入 “`\caption{}`” 命令，编译不会报错，但是会导致最后带标题的图片变成纵向排列；如果是纵向插入图片，在每一条 “`\includegraphics[options]{name}`” 命令之后，“`\\`” 符号之前插入 “`\caption{}`” 命令，编译会报错。

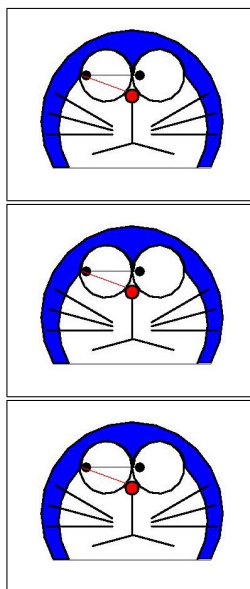


图 6: doraemon\_figure\_graphicx\_vertical (1)

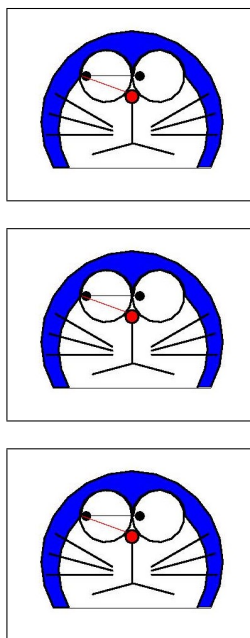


图 7: doraemon\_figure\_graphicx\_vertical (2)

表示的不再是原来文本宽度的倍数，而是 minipage 环境中文本宽度的倍数，一般改为单倍的文本宽度。

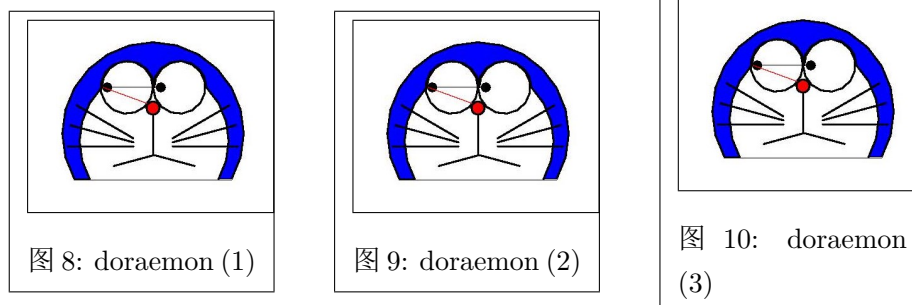


图 11: doraemon\_minipage\_caption

### 4.3 并排插入拥有子编号的图片

上述这种设置方式会导致小图的编号接着前面大图的编号往下继续编号，而没有重新编号。为了让每个并排插入的小图能够拥有自己的子编号，让子编号重新编号，让小图共享的母编号接着前面大图的编号往下继续编号，有两种宏包可以选择。

#### 4.3.1 subfig 宏包

subfig 宏包的的前身是 subfigure 宏包，后者现已不再使用。subfig 宏包提供了 “\subfloat[subcaption]{body}” 命令，其原理类似 minipage 环境，可选参数用来输入小图标题，必选参数用来输入 “\includegraphics[options]{name}” 命令，“\subfloat[subcaption]{body}” 命令没有用来控制图片宽度的参数，图片的宽度通过嵌套在其中的 “\includegraphics[options]{name}” 命令来控制(??)。

#### 4.3.2 subcaption 宏包

subcaption 宏包提供了三种方式来解决上述问题。第一种方式是通过将表 ( ) 中的 “\caption{ }” 命令替换为 “\subcaption{ }” 命令，就可以使小图重新编号 (12)。

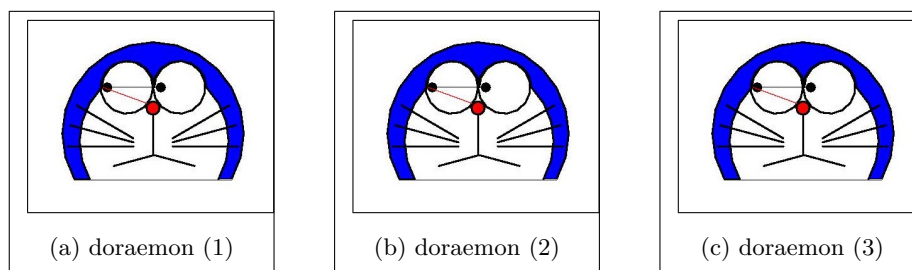


图 12: doraemon\_minipage\_subcaption

第二种方式是通过 “`\subcaptionbox{heading}[width]{content}`” 命令来完成，其原理类似 minipage 环境，第一个必选参数用来输入小图标题（而在 “`\subfloat[subcaption]{body}`” 命令种这是一个可选参数），第二个可选参数用来输入边框的宽度，第三个必选参数用来输入 “`\includegraphics[options]{name}`” 命令。注意，和在 minipage 环境中不同，“`\includegraphics[options]{name}`” 命令中表示图片宽度的可选参数在 “`\subcaptionbox{heading}[width]{content}`” 命令中仍然是以文本宽度作为基准的，因此不需要改变其大小。另外，根据 subcaption 宏包的官方手册显示，“`\subcaptionbox{heading}[width]{content}`” 的第二个可选参数表示的是边框的宽度，默认值就是其中图片的宽度，但是经检验，将其设置为图片宽度（13）和不设置这一可选参数（14）两种情况下，前者得到的效果是有略微差异的，前者和使用 “`\subcaption{}`” 命令（12）的效果相同，后者和使用 “`\subfloat[subcaption]{body}`” 命令（??）的效果相同。

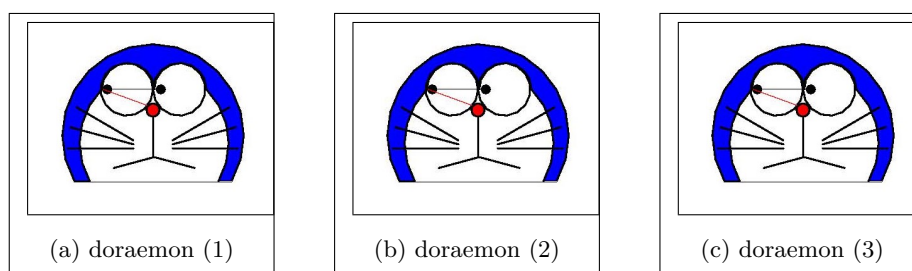


图 13: doraemon\_subcaptionbox (1)

第三种方式是通过 subfigure 环境（注意，该环境并非来自已经不再使用的 subfigure 宏包）来完成，其原理和 minipage 环境几乎完全相同，同样需要输入代表边框宽度的必选参数，并且其中 “`\includegraphics[options]{name}`”

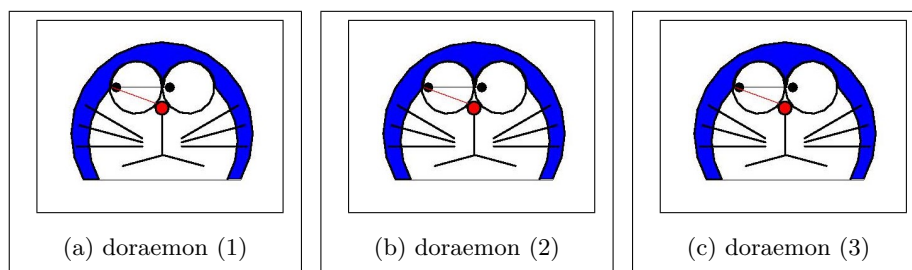


图 14: doraemon\_subcaptionbox (2)

命令的可选参数表示的宽度需要增大，因为此时 “`\textwidth`” 代表的文本宽度变成了 subfigure 创建的边框中的文本宽度，一般改为单倍的文本宽度。另外还有一点有意思的是，在 subfigure 中不管使用 “`\caption{}`” 命令还是 “`\subcaption{}`” 命令，小图都会重新编号，效果没有区别 (15)。

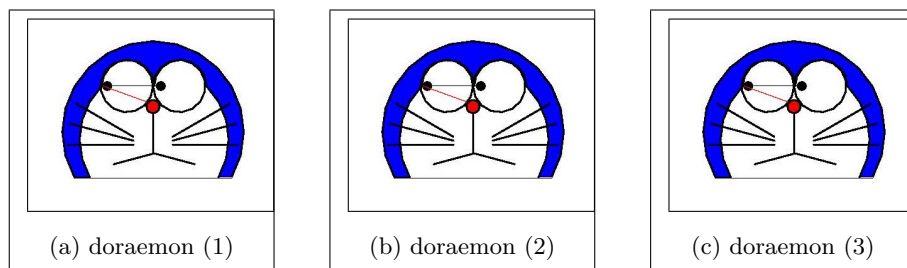


图 15: doraemon\_subfigure

#### 4.4 并排插入不同比例的图片

最后还可以讨论一下并排插入不同比例的图片的情况。首先有一个值得注意的现象，那就是如果在上述的命令中，如果在所有图片的 “`\includegraphics[options]{name}`” 命令的可选参数中同时输入 width 和 scale 参数，最后会以 width 参数作为排版的依据，而会忽视设置的 scale 参数。下面展示通过上面的各种方法设置不同比例图片的排版效果。

- 直接通过 figure 环境和 “`\includegraphics[options]{name}`” 命令 (16)
- figure 环境嵌套 minipage 环境，并且使用 “`\caption`” 命令 (20)
- 使用 subfig 宏包的 “`\subfloat[sub-caption]{body}`” 命令 (??)
- figure 环境嵌套 minipage 环境，并且使用 “`\subcaption`” 命令 (21)

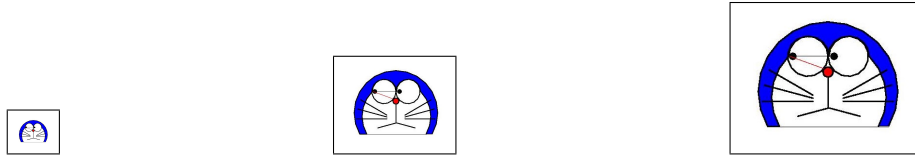


图 16: doraemon\_figure\_graphicx\_scale

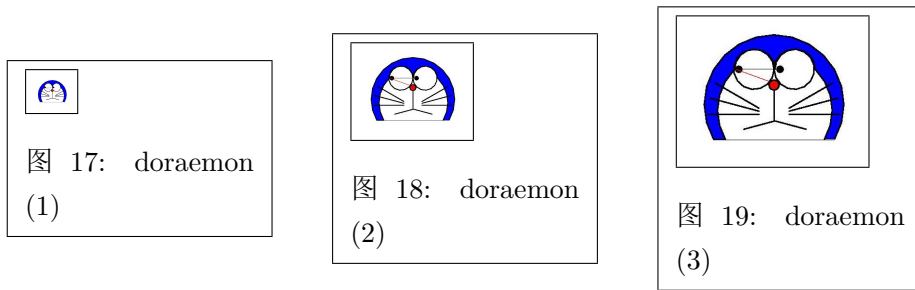


图 20: doraemon\_minipage\_caption\_scale

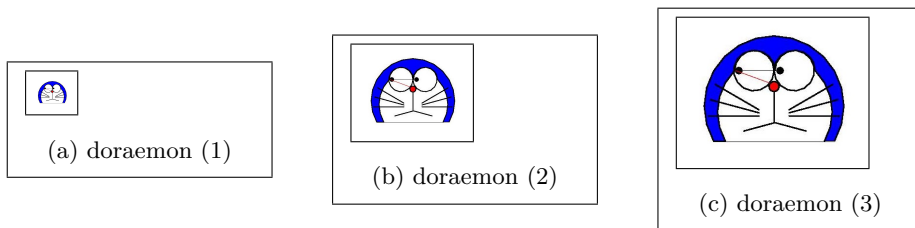


图 21: doraemon\_minipage\_subcaption\_scale

使用 subcaption 宏包的 “`\subcaptionbox{heading}[width]{content}`” 命令 (22, 23)

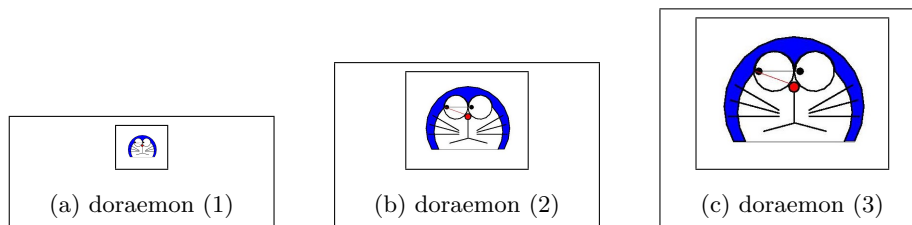


图 22: doraemon\_subcaptionbox\_scale (1)

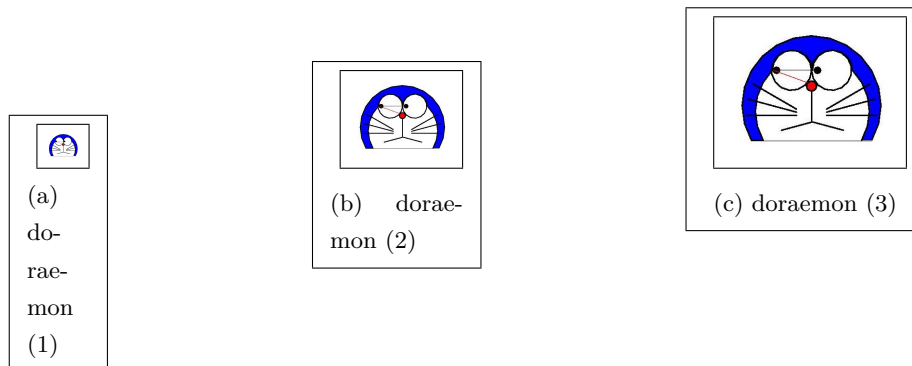


图 23: doraemon\_subcaptionbox\_scale (2)

使用 subcaption 宏包的 subfigure 环境 (24)

## 4.5 结论

通过以上各命令的效果，可以发现，排版效果最好的两种方式是：

- 使用 subcaption 宏包的 “`\subcaptionbox{heading}[width]{content}`” 命令，并且为其设置了相等的边框宽度 (22)
- 使用 subcaption 宏包的 subfigure 环境 (24)

使用 minipage 环境 (20, 21) 即使设置了相等的边框宽度，也会由于边框的下端没有对齐而导致排版效果不如以上两种方式美观（或许 minipage

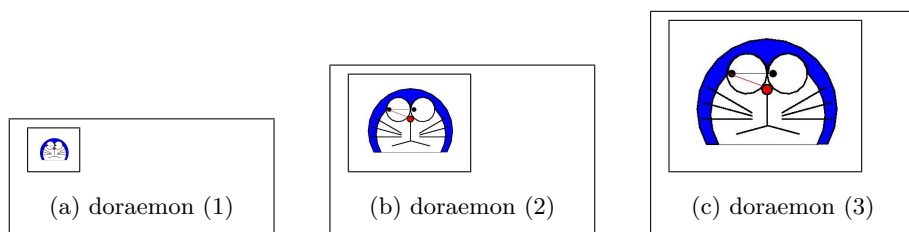


图 24: doraemon\_subfigure

环境有相关参数可以调整这一效果), 而使用 subfig 宏包的 “\subfloat[subcaption]{body}” 命令 (??), 则会由于无法设置图片的宽度, 从而和使用 subcaption 宏包的 “\subcaptionbox{heading}[width]{content}” 命令, 但是却没有为其设置边框宽度 (23) 一样, 导致最后的边框宽度不相等, 排版效果最差。

看来, 如果要并排插入图片, 最好选择使用 subcaption 宏包的 “\subcaptionbox{heading}[width]{content}” 命令或者 subfigure 环境。



## 插图

1	doraemon_figure . . . . .	7
2	doraemon_wrapfigure . . . . .	7
3	doraemon_figure_graphicx_horizontal (1) . . . . .	8
4	doraemon_figure_graphicx_horizontal (2) . . . . .	9
5	doraemon_figure_graphicx_horizontal (3) . . . . .	9
6	doraemon_figure_graphicx_vertical (1) . . . . .	10
7	doraemon_figure_graphicx_vertical (2) . . . . .	10
8	doraemon (1) . . . . .	11
9	doraemon (2) . . . . .	11
10	doraemon (3) . . . . .	11
11	doraemon_minipage_caption . . . . .	11
12	doraemon_minipage_subcaption . . . . .	12
13	doraemon_subcaptionbox (1) . . . . .	12
14	doraemon_subcaptionbox (2) . . . . .	13
15	doraemon_subfigure . . . . .	13
16	doraemon_figure_graphicx_scale . . . . .	14
17	doraemon (1) . . . . .	14
18	doraemon (2) . . . . .	14
19	doraemon (3) . . . . .	14
20	doraemon_minipage_caption_scale . . . . .	14
21	doraemon_minipage_subcaption_scale . . . . .	14
22	doraemon_subcaptionbox_scale (1) . . . . .	15
23	doraemon_subcaptionbox_scale (2) . . . . .	15
24	doraemon_subfigure . . . . .	16