

那些不常见，但却非常实用的css属性(整理不易)



爱前端不爱恋爱
微信公众号：web前端学习圈，关注领取全套50G前端系统教程

关注她

592 人赞同了该文章

1、-webkit-line-clamp

可以把 块容器 中的内容限制为指定的行数。并且在超过行数后，在最后一行显示"..."

这是正常的展示

How do I manage verified tables and mapping
tables when I need to verify the fields my upload
data or do tables mapping?

```
display: -webkit-box; /* 值必须为-webkit-box或者-webkit-inline-box*/  
-webkit-box-orient: vertical; /* 值必须为vertical*/  
-webkit-line-clamp: 2; /* 值为数字，表示一共显示几行*/  
overflow: hidden;
```

这是加了 line-clamp 后的展示

How do I manage verified tables and mapping
tables when I need to verify the fields my uplo...

<https://developer.mozilla.org/zh-CN/docs/Web/CSS/-webkit-line-clamp>

2、all

将除却 unicode-bidi 与 direction 之外的所有属性重设至其初始值，或继承值。

```
all: unset; /*initial | inherit | unset*/
```

initial 改变该元素所有属性的值至初始值。inherit 改变该元素的值 至 从父元素继承的值
unset 如果该元素的属性的值是可继承的，重置为父元素的继承的值，反之则改变为初始值。

3、box-decoration-break

<https://developer.mozilla.org/zh-CN/docs/Web/CSS/box-decoration-break>

4、caret-color

用来定义插入光标（caret）的颜色，这里说的插入光标，就是那个在网页的可编辑器区域内，用来指示用户的输入具体会插入到那里的那个一闪一闪的形似竖杠 | 的东西。

```
caret-color: red;
```



Enter text in the field to see the caret:

5、clip-path / shape-outside

clip-path 属性使用裁剪方式创建元素的可显示区域。区域内的部分显示，区域外的隐藏。类似的裁剪还有 svg 的 clipPath。

clip-path 的取值有以下几种图形

- inset(xxx): 裁剪为长方形

circle(xx): 裁剪为原型

ellipse(xxx): 裁剪为椭圆

polygon(xx): 裁剪为多边形

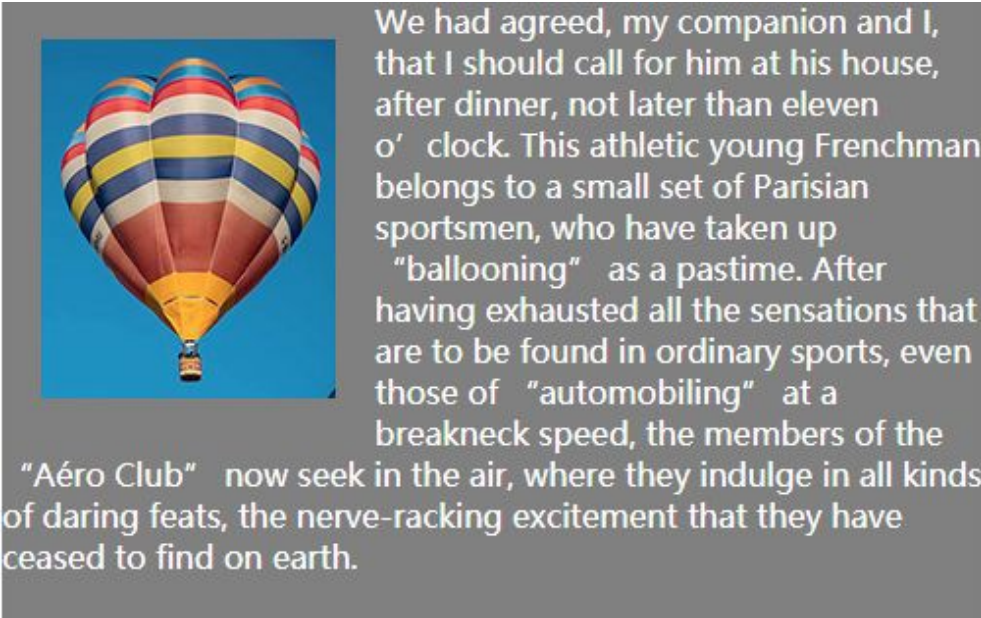
none: 不裁剪

例子前的准备

```
<div
  style="width: 500px; height: 500px;text-align: left;background-color:gray;color:whit
>
  
  We had agreed, my companion and I, that I should call for him at his house,
  after dinner, not later than eleven o'clock. This athletic young Frenchman
  belongs to a small set of Parisian sportsmen, who have taken up "ballooning"
  as a pastime. After having exhausted all the sensations that are to be found
  in ordinary sports, even those of "automobiling" at a breakneck speed, the
  members of the "Aéro Club" now seek in the air, where they indulge in all
  kinds of daring feats, the nerve-racking excitement that they have ceased to
  find on earth.
</div>
```

不裁剪

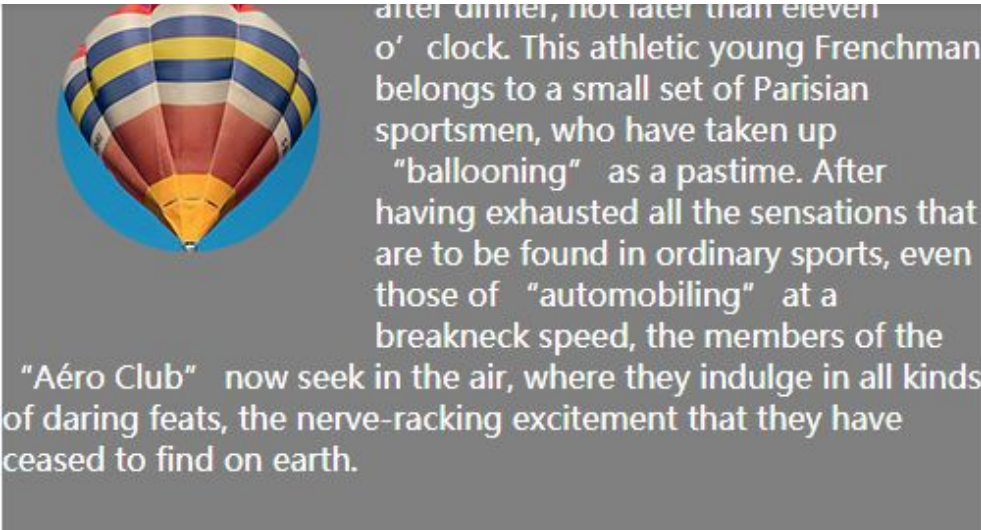
clip-path: none;



圆形裁剪

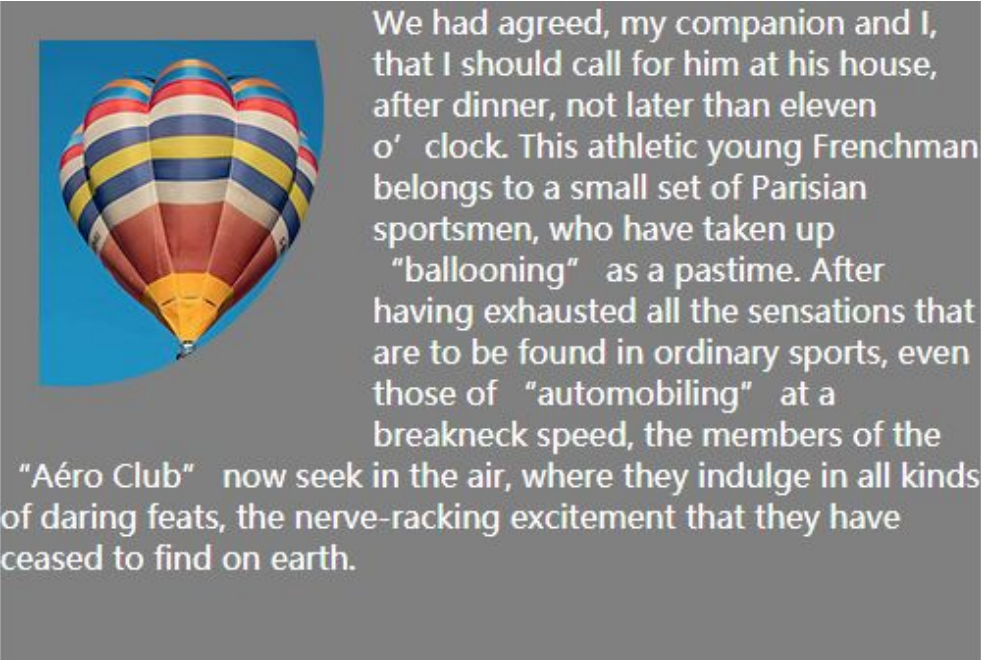
clip-path: circle(40%); //半径为40%，圆心默认为中心





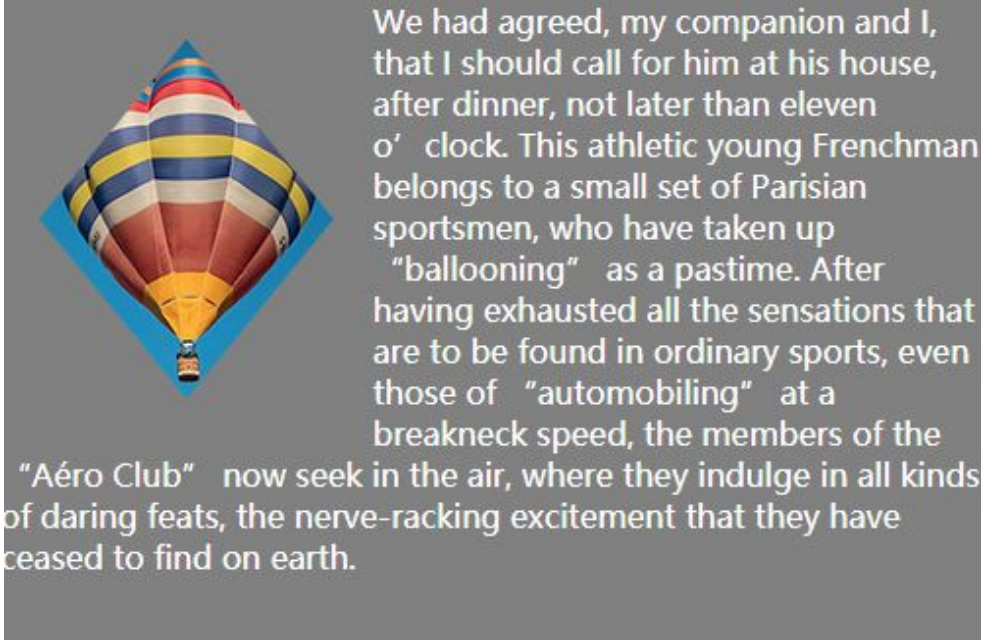
椭圆裁剪

```
clip-path: ellipse(130px 140px at 10% 20%);
```



多边形裁剪

```
clip-path: polygon(50% 0, 100% 50%, 50% 100%, 0 50%);
```

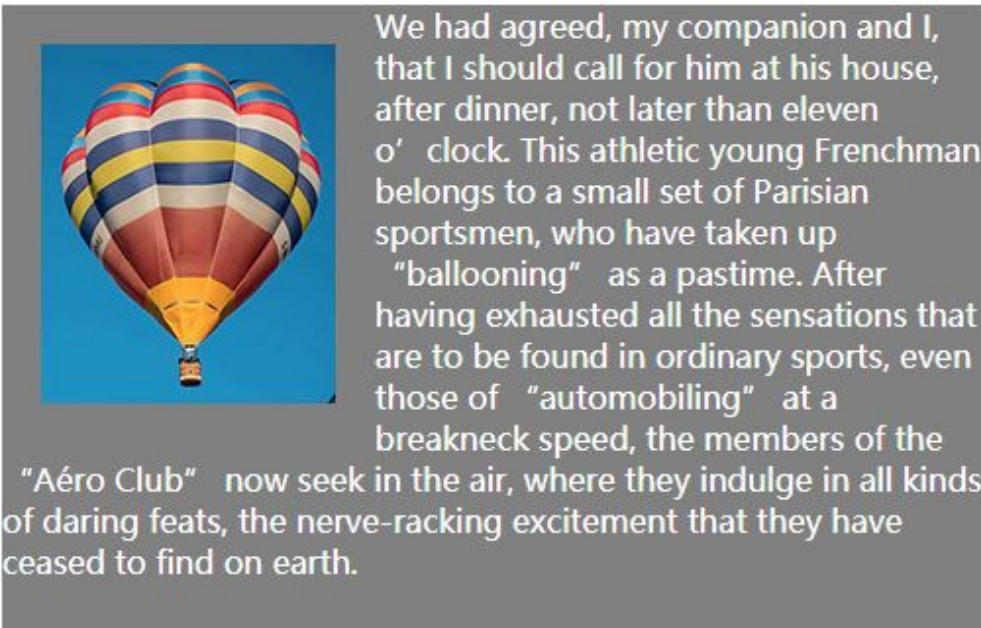


我们可以看到上面图形如何裁剪，外面的看不见的框始终是矩形，也就是说文字始终是按矩形的样式在周围环境。

那么有办法让文字紧紧的贴在裁剪的图形周围呢？
有的，使用 shape-outside 属性

shape-outside 定义了一个可以是 非矩形的形状，相邻的内联内容应围绕该形状进行包裹。默认情况下，内联内容包围其矩形边距；

默认矩形环绕

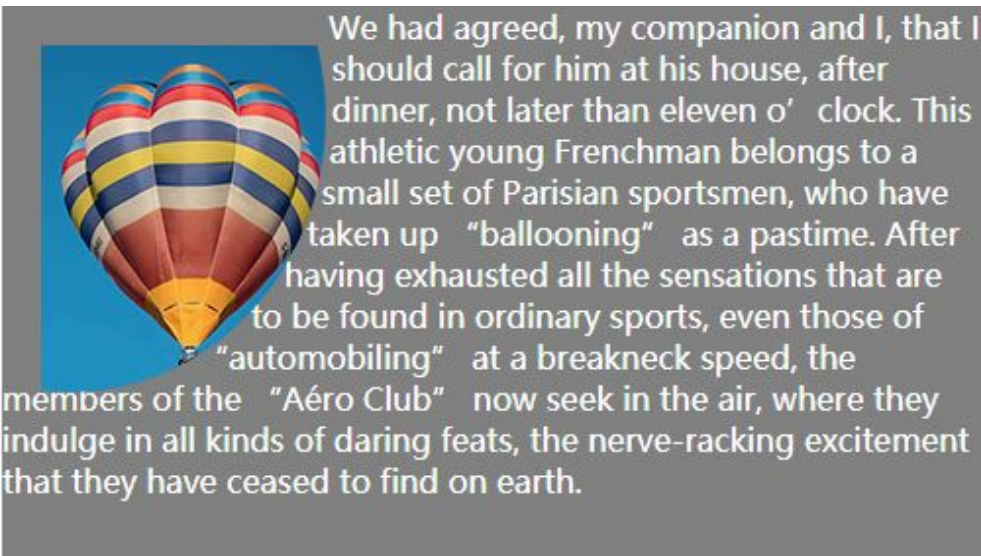


圆形环绕

```
clip-path: circle(40%);  
shape-outside: circle(40%);
```

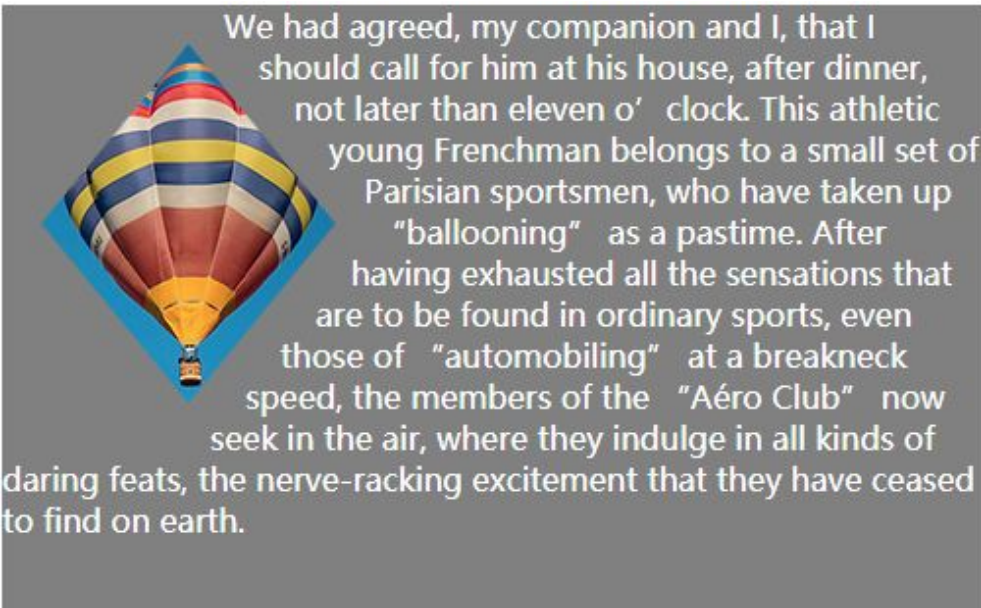
椭圆环绕

```
clip-path: ellipse(130px 140px at 10% 20%);  
shape-outside: ellipse(130px 140px at 20% 20%);
```



多变性环绕

```
clip-path: polygon(50% 0, 100% 50, 50% 100, 0 50%);  
shape-outside: polygon(50% 0, 100% 50, 50% 100, 0 50%);
```



当然你不一定在圆形裁剪上，使用圆形环绕，你可以在圆形裁剪上使用多边形环绕，在多边形裁剪上，使用菱形环绕。总之这两个之间没关系，不是两两匹配的。都随你便，自己定义



object-fit 属性指定可替换元素的内容应该如何适应到其使用的高度和宽度确定的框。object-position 属性来指定被替换元素的内容对象在元素框内的对齐方式。

注: 其中可替换元素有 iframe, video, embed, img, 还有一些在特性情况也是可替换元素, option, audio, canvas, object

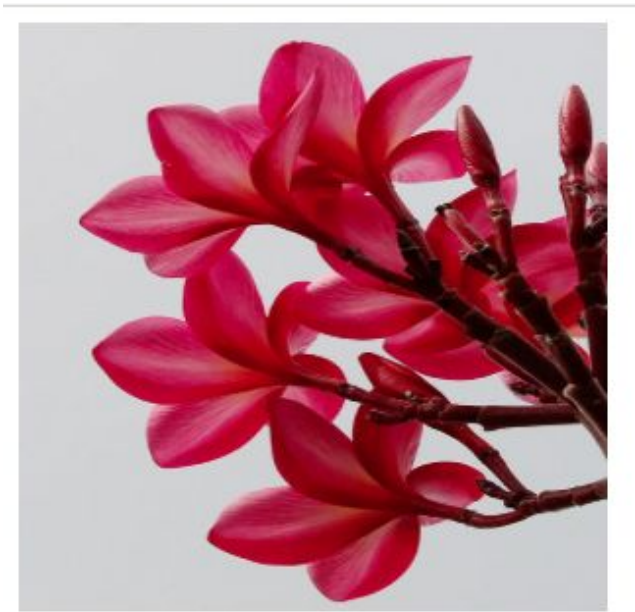
例子前的准备

```
<div style="width: 300px; height: 300px; background-color:gray;">
  <img class="clip-mode" style="height: 100%; width: 100%;" src="https://interactive
</div>
```

图片一定能要设置为宽高 100%，即不能超过父容器，才可以设置 object-fit，否则没意义。

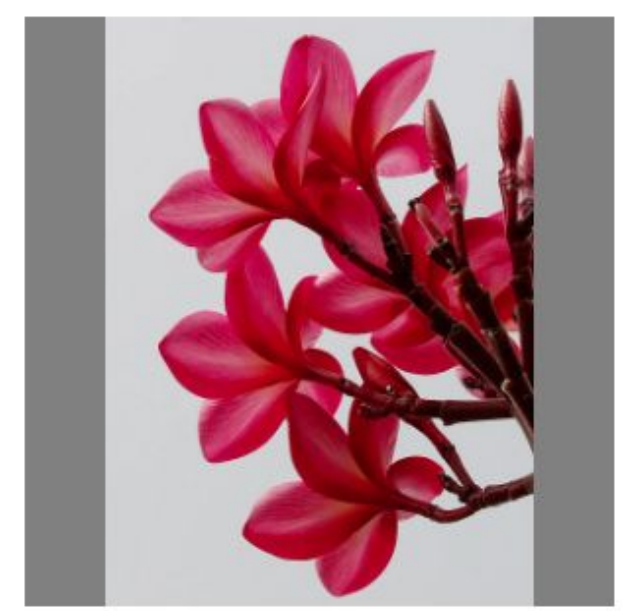
fill 填满 图片会拉变形，宽度和高度都被拉到父容器的 100%，以适应父容器

```
object-fit: fill;
```



contain 包容 图片不会变形，图片会按照自身比例进行缩放，整个图片放入父容器中，较短的边会出现自动填充的空白。

```
object-fit: contain;
```



cover 覆盖 图片不会变形，图片会按照自身比例进行缩放，整个图片放入父容器中，按照图片最短的边，纳入父容器为基准。较长的边会溢出

```
object-fit: cover;
```

none 和父容器的宽高没关系。展示其图片最原始的宽高比，以自身图片的“中心”为基点，放置到父容器的“中心”位置。



scale-down 内容的尺寸与 none 或 contain 中的一个相同，取决于它们两个之间谁得到的对象尺寸会更小一些。

```
object-fit: scale-down;
```

如果图片比父容器尺寸大，那么按照 contain 的效果，如果图片比父容器小，那么按照 none 的效果。

在上面 object-fit 的展示中，我们发现可替换元素的对齐方式都是自动的。

比如 object-fit: fill; 的左上角和父容器的左上角对齐。

object-fit: none;的中心和父容器的中心对齐等等。

但是我们想手动更改对齐方式呢？？

可以使用 **object-position** 属性， 规定了可替换元素的内容，在其内容框中的位置。

object-position: 10px 10px; 可以设置 px， 第一个值代表距离父容器左边的距离， 第二个值代表距离父容器顶部的距离。 只有一个数值则只代表距离父容器左侧的距离。 也可以设置%数值， 但此时只有某一边有空白才会起作用， 如果没有空白， 刚好铺满父元素， 则不起作用。 设置 px 就没有这样的问题， 任何之后都会起作用。 object-position: right top;可以设置关键字， 第一个值关键字可设置 (left|center|right) ， 第二个关键字可设置(top|center|bottom),此时不表示距离左侧或者顶部的距离， 而表示放置在父元素的什么位置。

```
object-fit: fill;
object-position: 50px 50px;    //距离左侧10px， 距离顶部10%
```

加 object-position 前后对比图:



```
object-fit: contain;
object-position: right top;    //停留子 右上方
```

加 object-position 前后对比图:



7、font-stretch

为字体定义一个正常或经过伸缩变形的字体外观，它仅仅意味着当有多种字体可供选择时，会为字体选择最适合的大小。

normal 默认字体

semi-condensed, condensed, extra-condensed, ultra-condensed 小于默认字体

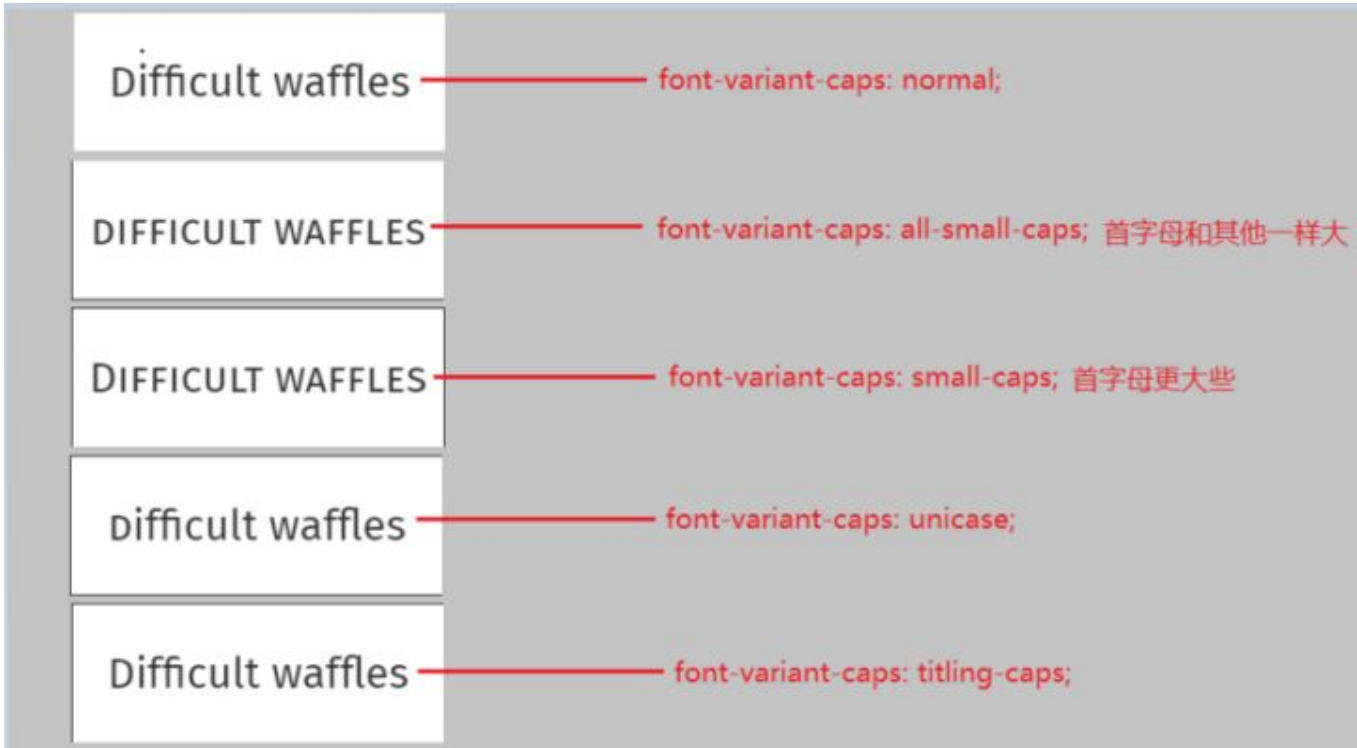
semi-expanded, expanded, extra-expanded, ultra-expanded 大于默认字体



8、font-variant-caps

可以控制大写字母特殊字符的使用。

- normal 关闭一切特殊字符变体的使用。
- small-caps 允许小型大写字母的使用（OpenType 特性：smcp）。小型大写字母指使用大写形式，但尺寸与对应小写字母相同的字母。
- all-small-caps 将大小写字母全部转化为小型大写字母。（OpenType 特性: c2sc, smcp）。
- petite-caps 允许特小型大写字母的使用（OpenType 特性: pcap）。
- all-petite-caps 将大小写字母全部转化为小型大写字母。（OpenType 特性: c2pc, pcap）。
- unicase 允许将大写字母转化为小型大写字母与普通小写字母的混用（OpenType 特性: unic）。
- titling-caps 允许首字母大写（OpenType 特性: titl）。大写字母变体字符通常被设计成与用于小写字母。在标题序列中，如果均使用大写字母，可能会带来过于强烈的视觉效果。首字母大写即用来应对这种情况。



9、font-variant-east-asian

控制对东亚字符(如日语和汉语,韩语等)使用替代符号。





10、max-content / min-content / fill-available / fit-content

这几个值都可用在 width, height, min-width, min-height, max-width 和 max-height 属性上。

display 必须为 inline-block 或者 block，否则上面的值不起作用。

fill-available

元素撑满可用空间。参考的基准为父元素有多宽多高。

类似子元素的 div 撑满父元素的宽，fill-available 不仅可以撑满宽还能撑满高。

例子前的代码

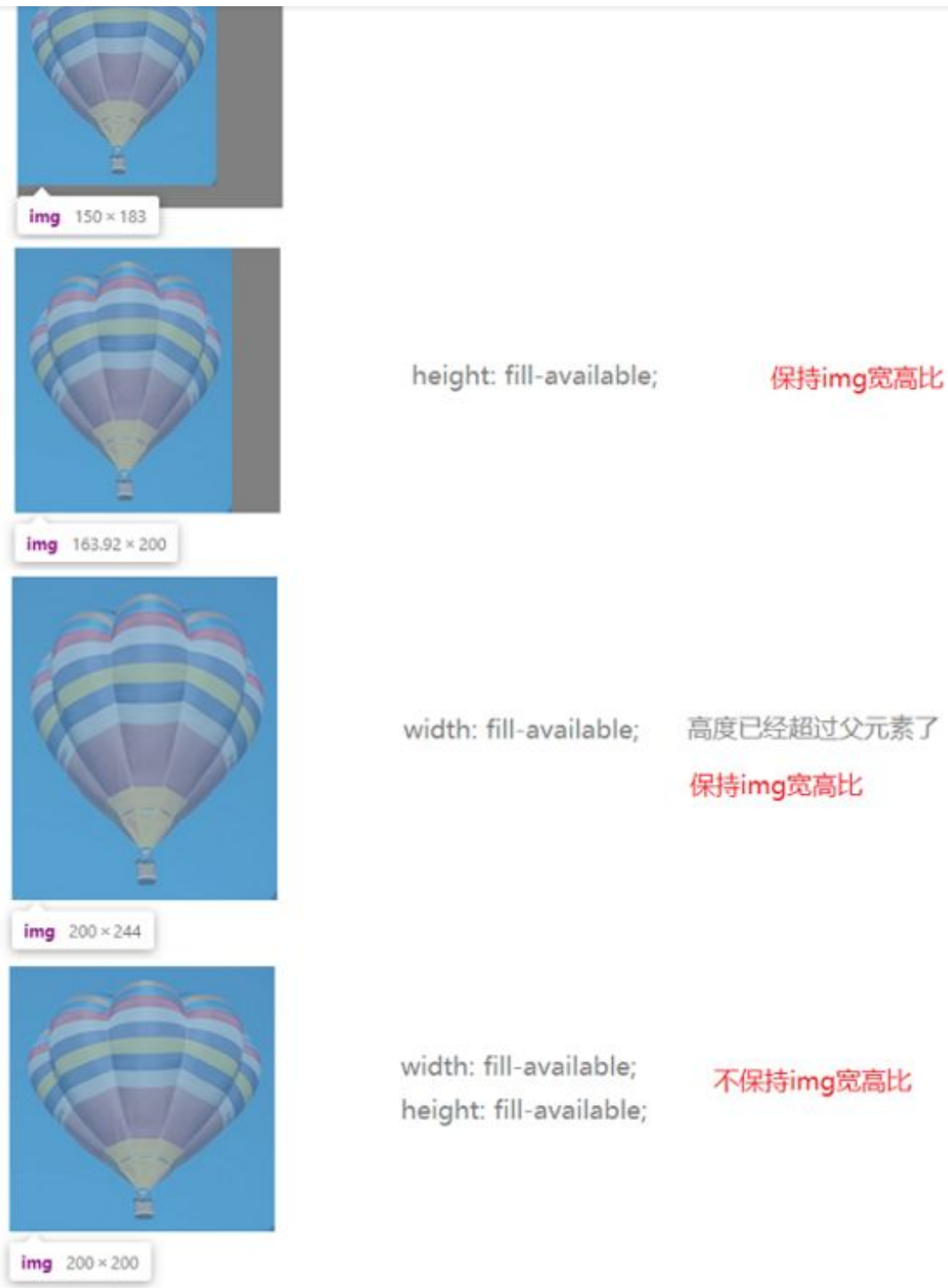
```
<div style="width: 300px; height: 100px; background-color:gray;">
  <span style="display:inline-block;background-color: burlywood;"
    >这是子元素的内容</span>
  <div>
    </div>
</div>
```

给 span 上设置 fill-available 时的不同表现



假如里面的有个元素，是 img 呢？它也是 inline-block，应该也满足情况。





我们可以看到 `img` 和 `span` 的不同在于，设置 `width` 或者 `height` 其中一个时，整个 `image` 会按照自身比例缩放。

max-content

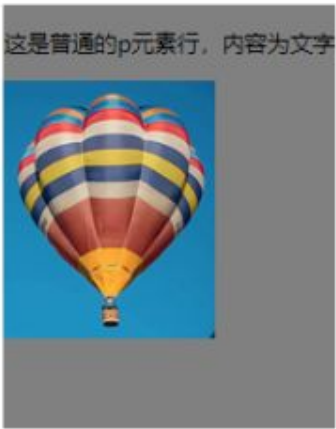
它的宽度或者高度，会自动调整为，刚刚好容纳下子元素中那个长度最长(按照文字不换行时计算)的元素即可。
参考的基准为子元素有多宽多高。

```
<div class="parent">
  <div class="current" style="width: 200px; height: 300px; background-color:gray;">
    <p>这是普通的p元素行，内容为文字</p>
    <img src="https://interactive-examples.mdn.mozilla.net/media/examples/balloon-smal
  </div>
</div>
```

给 `current div` 设置 `max-content` 时得不同表现。



正常



width: max-content;

按照里面最长的文字行的
宽度为自己的宽度



height: max-content;

在改变writing-mode
为纵向后，发现也是按
照最长的列为自己的高
度

min-content

它的宽度或者高度，会自动调整为，刚刚好容纳下子元素中那个“最小宽度值”最大的元素即可，剩余超长的要么换行，要么溢出
参考的基准为子元素“最小宽度值”有多宽多高。

什么是“最小宽度值”？
比如图片，最小宽度值，就是图片原始的宽高；如果是一串中文，则最小宽度值为单个汉字的宽高；如果是一串英文，则最小宽度值为里面单词最长的那个。



这是p元素行



width: min-content; 汉字和图片的最小宽度值相比，自然图片更长，所以采用图片的

这是普通的p元素行，内容为文字



width: min-content; 虽然整个汉字长，但是就最小宽度值而言，还是图片更长，所以采用图片的

pneumonoultramicroscopicsilicovolcanoconiosis, 英语中最长单词，意思为尘肺病



width: min-content; 图片和英语单词相比，最小宽度值更长的为这个英语单词，所以采用英文单词的宽度

fit-content

表示元素自动伸缩到内容的宽度，和 max-content 的区别为，都是刚刚好容纳下子元素中那个长度最长的元素即可。

不同的是 max-content 在计算时按照文字不换行时计算，如果超过父元素，则不换行，直接产生滚动条； 而 fit-content 在超过父元素后，换行，不产生滚动条。

pneumonoultramicroscopicsilicovolcanoconiosis, 在我超过父元素后，max-content和fit-content有不同的表现



width: max-content; 不换行，直接往后排，产生横向滚动条了

pneumonoultramicroscopicsilicovolcanoconiosis, 在我超过父元素后，max-content和fit-content有不同的表现



width: fit-content; 在父容器放不下时换行了，没有滚动条

11、fit-content()

这个不同于上面的 fit-content 值，它是函数，用在 grid 布局当中。

(略)

具体查看 grid 布局



比较简单，可以查看

resize
[developer.mozilla.org](#)



13、scroll-behavior

当用户通过 API 触发滚动操作时，CSS 属性 scroll-behavior 为一个滚动框指定滚动行为，平滑到达还是立即到达

可以查看
[官方文档](#)

14、max() / min()

max 在两者这件取最大；min 函数在两者之间取最小。

max, min 可以用在任何可以设置 <length>, <frequency>, <angle>, <time>, <percentage>, <number>, or <integer> 等值的地方

```
width: max(50vw, 300px);
width: min(50vw, 300px);
```

15、clamp()

clamp 函数会在定义的 三个值之间，取中间的那个值，是大小在中间，不是位置在中间，

```
clamp(MIN, VAL, MAX)
```

和 max, min 函数一样，可以用在任何可以设置 <length>, <frequency>, <angle>, <time>, <percentage>, <number>, or <integer> 等值的地方

```
font-size: clamp(1px, 3px, 2px);    //中间值为2px
font-size: clamp(1px, 3px, 5px);    //中间值为3px
font-size: clamp(4px, 3px, 5px);    //中间值为4px
width: clamp(200rem, 25vw, 150px);  //在这三个之间取中间值
```

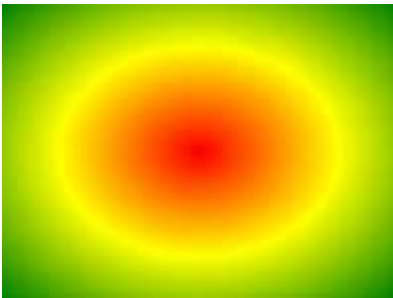
16、conic-gradient()

在渐变中我们知道，有：

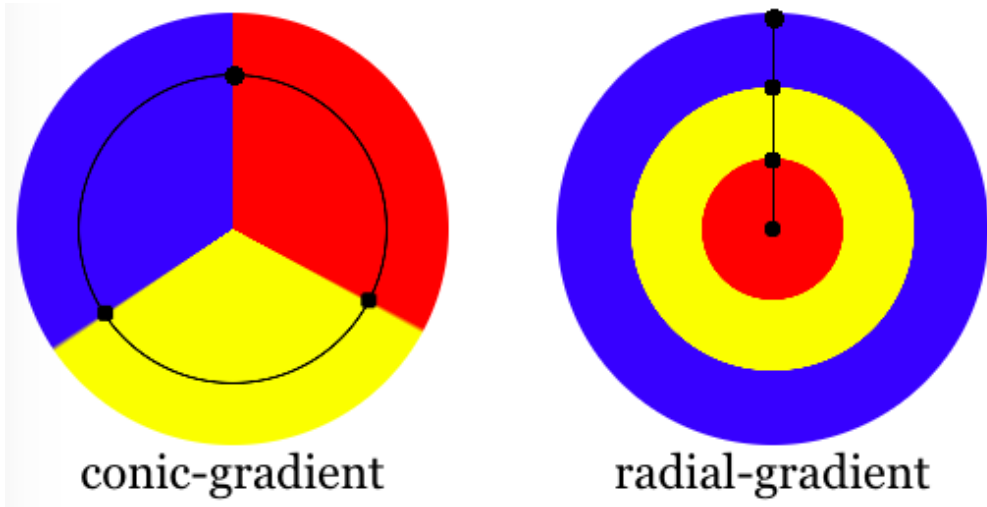
- 线性渐变,linear-gradient，从一个方向，沿着另一个方向，颜色线性过渡。



- 径向基渐变，radial-gradient，从某一个点，沿圆环向外慢慢辐射。



这是锥形和径向渐变的区别图



```
background: conic-gradient(red, orange, black, green, blue);
```

默认从时钟 12 点，按照顺时针旋转



18、:out-of-range / :in-range

比较简单，可以查看官方文档

[:out-of-range](#)

[:in-range](#)

19、writing-mode

定义了文本水平或垂直排布以及在块级元素中文本的书写方向

horizontal-tb 表示水平书写，从上(top)到下(bottom)书写vertical-rl 表示垂直书写，从右(right)向左(left)(古人书法就这么写的)
vertical-lr 表示垂直书写，从左向右
注意没有 horizontal-bt,不要杜撰

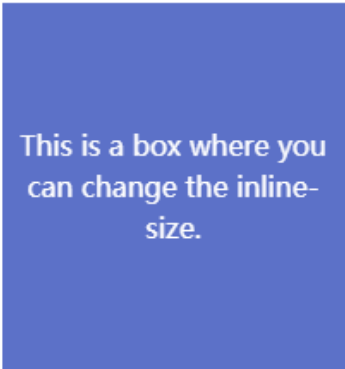
	script	script	script
horizontal-tb	我家没有电脑。	Example text	1994年に至っては
vertical-lr	我家没有电脑。	Example text	1994年に至っては
vertical-rl	我家没有电脑。	Example text	1994年に至っては

20、inline-size

和元素的 width，height 效果一样，都会改变盒子的大小。但是会覆盖 width，height 值。

不同在于，width 是绝对的水平方向，height 是绝对的竖直方向；

而 inline-size 是相对的水平方向，可通过 writing-mode 模式，改变方向



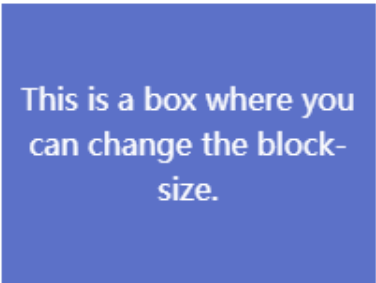
默认，inline-size和width方向一致，覆盖width



inline-size: 120px;
writing-mode: vertical-rl;inline-size和height方向一致，覆盖height

21、block-size

类似于 inline-size，只不过和 inline-size 刚好相反，block-size 默认设置的是 height 方向的值。



默认，block-size方向和height方向一致，因为block元素width铺满父元素的宽，覆盖height



block-size: 150px;
writing-mode: vertical-rl;此时block-size和width方向一致，覆盖width

原文链接：[那些不常见，但却非常实用的css属性\(整理不易\) - 个人文章 - SegmentFault 思否](#)

发布于 06-08

前端开发 前端工程师 程序员

▲ 赞同 592 ▼ 22 条评论 分享 喜欢 收藏 申请转载 ...

文章被以下专栏收录



web前端学习圈
web前端学习方法/知识干货/实战案例等，每天更新

关注专栏

推荐阅读

你未必知道的49个CSS知识点

本文的每一条，都是我曾经发过的掘金沸点，其中有很多条超过了百赞（窃喜）。鉴于时不时有童鞋翻我以前的沸点，因此，本文收集了个人目前发过的所有CSS知识点动图，以便阅读。需要说明的...

爱前端不爱... 发表于web前端...

那些你总是记不住但又总是要用的css

有没有经常遇到一些样式每次写都要用百度呢？我收集了一些我平时经常要用到的但又总是记不住的样式。一、设置input 的placeholder 的字体样式input::-webkit-input-placeholder { /* Chrome...

爱前端不爱... 发表于web前端...

55 个提高你备片段

这篇文章会记CSS 片段，使我们解决许多的，墙裂建议日后查找 附许多优质文章可爱前端不爱...

22 条评论

切换为时间排序

写下你的评论...



YISEN

06-09



1



宋运

06-09

收藏了先，不知道兼容性如何

赞



阿健大叔 回复 宋运

06-09

可以考虑用在只需要支持chrome或者electron上[捂嘴]

赞



宋运 回复 阿健大叔

06-09

哈哈，猜到如此



1



邪气满满

06-09

第一个只支持chrome还是算了吧

2



	safari edge	
	<div><div>👍 赞</div></div>	
	<div><div><div><div></div><div>panda 回复 邪气满满</div></div><div>现在chrome, edge, safari, 火狐都支持啊</div><div><div>👍 赞</div></div></div></div>	06-11
	<div><div><div><div></div><div>时过境迁</div></div><div>最重要的是要能兼容ie8及更高的版本啊</div><div><div>👍 2</div></div></div></div>	06-09
	<div><div><div><div></div><div>dyson</div></div><div>虽然写的很好,但是我还是要指出 请交代一下 兼容性!!!</div><div><div>👍 2</div></div></div></div>	06-09
	<div><div><div><div></div><div>Jason cui</div></div><div>科普css不加兼容性的都是耍流氓[手动狗头]!</div><div><div>👍 3</div></div></div></div>	06-09
	<div><div><div><div></div><div>任人</div></div><div>兼容都不说, 跟耍流氓有啥区别</div><div><div>👍 赞</div></div></div></div>	06-09
	<div><div><div><div></div><div>孟起</div></div><div>兼容性不好</div><div><div>👍 赞</div></div></div></div>	06-09
	<div><div><div><div></div><div>eei Wee</div></div><div>学到了</div><div><div>👍 赞</div></div></div></div>	06-09
	<div><div><div><div></div><div>守桥留蒲</div></div><div>太屌了, 全看完了~</div><div><div>👍 赞</div></div></div></div>	06-09
	<div><div><div><div></div><div>消肿</div></div><div>break-before/break-after/break-inside ——控制打印页面换页的特性, 连chrome都没有实现好, 前几天刚踩坑。</div><div><div>👍 赞</div></div></div></div>	06-09
	<div><div><div><div></div><div>看猴戏</div></div><div>非常感谢, 刚好用上</div><div><div>👍 赞</div></div></div></div>	06-10
	<div><div><div><div></div><div>Darling</div></div><div>感谢, 收藏夹里吃灰去吧</div><div><div>👍 赞</div></div></div></div>	06-10
	<div><div><div><div></div><div>张帆</div></div><div>干货, 收藏了</div><div><div>👍 赞</div></div></div></div>	06-10
	<div><div><div><div></div><div>guorui</div></div><div>支持[赞]</div><div><div>👍 赞</div></div></div></div>	06-11
	<div><div><div><div></div><div>云沧海</div></div><div>收藏了</div><div><div>👍 赞</div></div></div></div>	07-02

来台仕很里安

👍 赞



刃舞

08-12

好活，赏

👍 赞