**[你真的了解word-wrap和word-break的区别吗？](http://www.cnblogs.com/2050/archive/2012/08/10/2632256.html)**

这两个东西是什么，我相信至今还有很多人搞不清，只会死记硬背的写一个word-wrap:break-word;word-break:break-all;这样的东西来强制断句，又或者是因为这两个东西实在是太拗口了，长得又差不多，导致连背都很难背下来。

那它们到底是什么呢？我在mozilla的官网上找到如下的解释：

[**word-wrap**](https://developer.mozilla.org/en-US/docs/CSS/word-wrap)

[word-wrap](http://images.cnblogs.com/cnblogs_com/2050/201208/20120810172525868.jpg)

[word-break](https://developer.mozilla.org/en-US/docs/CSS/word-break)

[word-break](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101725384531.jpg)

我们看到两个解释中都出现了 break lines within words 这样的词汇，说明它们都跟单词内断句又关。然后我们试着翻译一下上面的两段英文：

**word-wrap:**

css的 word-wrap 属性用来标明是否允许浏览器在单词内进行断句，这是为了防止当一个字符串太长而找不到它的自然断句点时产生溢出现象。

**word-break:**

css的 word-break 属性用来标明怎么样进行单词内的断句。

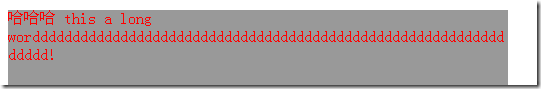
貌似从上面两个读着都蛋疼的翻译中找到了一些区别：word-wrap 强调的是是否允许单词内断句，而word-break强调的则是怎么样来进行单词内的断句。

说到这里，好像依然不是很明朗，好吧，表达能力差的孩纸真是伤不起啊，只能用些实例来补救了。

首先，何谓单词内断句？当然这里指的都是西文单词。

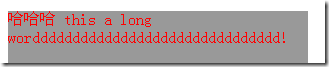
[](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101725401791.png)

这是没有单词内断句的情况，我们看到那个单词是在是太长了，所以它溢出了包裹它的容器。

[](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101725426544.png)

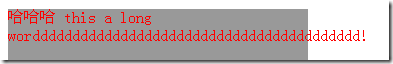
这是进行了单词内断句的情况，就是一个单词被断作了两行。

先要明确一点，不加word-wrap或word-break的时候，就是浏览器默认的时候，如果有一个单词很长，导致一行中剩下的空间已经放不下它时，则浏览器会把这个单词挪到下一行去：

[](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101725443347.png)

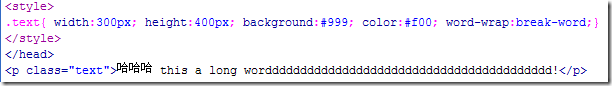
word..d 这个我们创造出来的单词本来应该紧接在 long 后面的，但是long后剩下的空间已经不够了，而单词默认是不能断开的，所以浏览器就只好把它屈尊下移了。

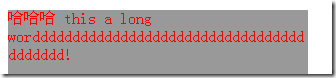
这个长单词还不算变态的，因为至少它没有长到超过包裹它的元素的长度，但是如果超过了呢？

[](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101725519731.png)

如果超过了就像前面我们提到过的，它会溢出它的父容器外，因为这时它是不允许被截断的，那就只能冲出去咯。

这个时候word-wrap就能派上用场了。我们给这段文字加上word-wrap:break-word看看会怎么样

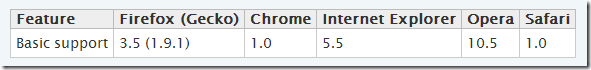
[](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101725534865.png)

[](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101725584369.png)

哈哈，你给我滚回去吧，变态的长单词，即使你断了也无所谓。

这样，为了防止长单词溢出，就在它的内部断句了。这就是 word-wrap:break-word 的功能。

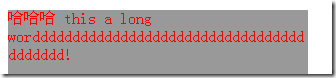
在看看word-wrap的浏览器支持情况：

[](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101725593299.png)

很好，几乎所有浏览器都支持。

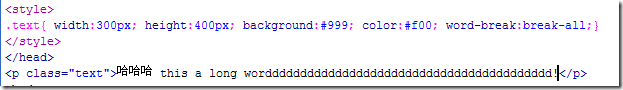
好吧，你现在可以说了，都有了word-wrap这个东西，那还要 word-break 来干什么鸟？

万恶的资本家总是想要榨干劳动者的一切，你看，下图中的long后面不是还有一段空间吗，难道就这样放着？太浪费了。。。

[](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101725584369.png)

是啊，long后面那里足可以放好几个短的单词了，都是那个臭变态的长单词！

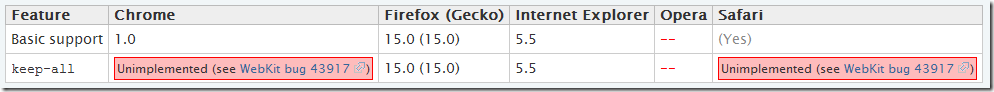
所以IE真的很体贴，不要再骂它啦，考虑到不浪费一点点空间的问题，它创造出了 word-break 这个东西。现在我们来看看使用 word-break:break-all; 后会怎么样。

[](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101726018957.png)

[01](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101726031267.png)

看看发生什么了？那个变态的长单词并没有被挪到下一行，而是直接放在了long后面，然后在父容器的右边界断开了。所以，它没有浪费空间了哦。

再看下word-break:break-all的浏览器支持情况：

[](http://images.cnblogs.com/cnblogs_com/2050/201208/201208101726092799.png)

除了opera外，其他都支持耶（火狐也从不支持改为支持了）!

最开始我们说了，word-wrap 是用来决定允不允许单词内断句的，如果不允许的话长单词就会溢出。最重要的一点是它还是会首先尝试挪到下一行，看看下一行的宽度够不够，不够的话就进行单词内的断句。

而word-break:break-all则更变态，因为它断句的方式非常粗暴，它不会尝试把长单词挪到下一行，而是直接进行单词内的断句。这也可以解释为什么说它的作用是决定用什么方式来断句的，那就是——用了word-break:break-all，就等于使用粗暴方式来断句了。总之一句话，如果您想更节省空间，那就用word-break:break-all就对了！

但无论如何，单词内的断句都会对西文的可读性产生一定的影响，有时候这点就要注意了。

ps:网上有些文章说，word-wrap:break-word 对长串英文不起作用，其实这是非常错误的，word-wrap:break-word照样能把一个长串英文或数字拆成多行。事实上，word-wrap:break-word与word-break:break-all共同点是都能把长单词强行断句，不同点是word-wrap:break-word会首先起一个新行来放置长单词，新的行还是放不下这个长单词则会对长单词进行强制断句；而word-break:break-all则不会把长单词放在一个新行里，当这一行放不下的时候就直接强制断句了

当我们设置我的的div，或者其它文本框固定宽度之后，文本内容过多就会出文本溢出（显示在区域外面，不换行的情况）。

这时我们可以使用Css中的几个属于来解。有以下的三个属于可以解决问题：

1，word-break 属性规定自动换行的处理方法。

word-break: normal|break-all|keep-all;

|  |  |
| --- | --- |
| **值** | **描述** |
| normal | 使用浏览器默认的换行规则。 |
| break-all | 允许在单词内换行。 |
| keep-all | 只能在半角空格或连字符处换行。 |

2，overflow 属性规定当内容溢出元素框时发生的事情。

|  |  |
| --- | --- |
| **值** | **描述** |
| visible | 默认值。内容不会被修剪，会呈现在元素框之外。 |
| hidden | 内容会被修剪，并且其余内容是不可见的。 |
| scroll | 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。 |
| auto | 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。 |
| inherit | 规定应该从父元素继承 overflow 属性的值。 |

3，white-space 属性设置如何处理元素内的空白。

|  |  |
| --- | --- |
| **值** | **描述** |
| normal | 默认。空白会被浏览器忽略。 |
| pre | 空白会被浏览器保留。其行为方式类似 HTML 中的 <pre> 标签。 |
| nowrap | 文本不会换行，文本会在在同一行上继续，直到遇到 <br> 标签为止。 |
| pre-wrap | 保留空白符序列，但是正常地进行换行。 |
| pre-line | 合并空白符序列，但是保留换行符。 |
| inherit | 规定应该从父元素继承 white-space 属性的值。 |

我使用的是 word-break:break-all来实现文本的换行，因为我们字符是中文导致换行会出现问题。使用overflow会出滚动条，如果是

允许有滚动条可以使用。white-space是属于文本里的空白字符，如果文本里含有很多的空白字符我们可以考虑加上该属性。