**flex-shrink**

* **版本：CSS3**
* 媒体：视觉
* [Issues](https://github.com/doyoe/css-handbook/issues)
* [Pull Requests](https://github.com/doyoe/css-handbook/pulls)

Base Browsers: IE6.0+, Firefox2.0+, Chrome4.0+, Safari6.0+, Opera15.0+

**语法：**

**flex-shrink**：[<number>](http://css.doyoe.com/values/numeric/number.htm)

**默认值**：1

**适用于**：flex子项

**继承性**：无

**动画性**：是

**计算值**：指定值

**取值：**

[<number>](http://css.doyoe.com/values/numeric/number.htm)：

用数值来定义收缩比率。不允许负值

**说明：**

**设置或检索弹性盒的收缩比率。**

* 根据弹性盒子元素所设置的收缩因子作为比率来收缩空间。
* **示例：a,b,c将按照1:1:3的比率来收缩空间**

**HTML Code：**

<ul class="flex">

<li>a</li>

<li>b</li>

<li>c</li>

</ul>

.flex{display:flex;width:400px;margin:0;padding:0;list-style:none;}

.flex li{width:200px;}

.flex li:nth-child(3){flex-shrink:3;}

flex-shrink的默认值为1，如果没有显示定义该属性，将会自动按照默认值1在所有因子相加之后计算比率来进行空间收缩。

本例中c显式的定义了flex-shrink，a,b没有显式定义，但将根据默认值1来计算，可以看到总共将剩余空间分成了5份，其中a占1份，b占1份，c占3分，即1:1:3

我们可以看到父容器定义为400px，子项被定义为200px，相加之后即为600px，超出父容器200px。那么这么超出的200px需要被a,b,c消化

通过收缩因子，所以加权综合可得200\*1+200\*1+200\*3=1000px；  
于是我们可以计算a,b,c将被移除的溢出量是多少：  
a被移除溢出量：(200\*1/1000)\*200，即约等于40px  
b被移除溢出量：(200\*1/1000)\*200，即约等于40px  
c被移除溢出量：(200\*3/1000)\*200，即约等于120px  
最后a,b,c的实际宽度分别为：200-40=160px, 200-40=160px, 200-120=80px

* 对应的脚本特性为**flexGrow**。

