HTML5 CSS3盒子模型

六 、CSS3盒子模型

盒子阴影 box-shadow

- h-shadow 必需。水平阴影的位置。允许负值。
- v-shadow 必需。垂直阴影的位置。允许负值。
- blur 可选。模糊距离。
- spread 可选。阴影的尺寸。
- color 可选。阴影的颜色。请参阅 CSS 颜色值。
- inset 可选。将外部阴影 (outset) 改为内部阴影。

弹性盒子模型

display:flex

给父级设置一个display:flex属性，子元素设置flex相关属性才可以自动分配宽高。

flex – 给需要设置弹性布局的子元素设置该属性

- flex-grow：占父元素的剩余空间的多少
  比如这个例子：

```
<ul class="flex">
    <li>a</li>
    <li>b</li>
    <li>c</li>
</ul>

.flex{display:flex;width:600px;margin:0;padding:0;list-style:none;}
.flex li:nth-child(1){width:200px;}
.flex li:nth-child(2){flex-grow:1;width:50px;}
.flex li:nth-child(3){flex-grow:3;width:50px;}
```

```
flex-grow的默认值为0，如果没有显示定义该属性，是不会拥有分配剩余空间权利的。
本例中b,c两项都设置的定义了flex-grow，flex容器的剩余空间分成了4份，其中b占1份，c占3分，即1:3
flex容器的剩余空间长度为：600-200-50-50=300px，所以最终a,b,c的长度分别为：
a: 50+(300/4)=200px
b: 50+(300/4*1)=125px
a: 50+(300/4*3)=275px
```

- flex-shrink： 默认值是1。占据超出父级容器的宽度的百分比。如果所有的子元素的宽度相加没有超过父级的在宽度，则次属性无效。

```
<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
```

- flex-basis：和width一样，他的默认值为auto，把上面几个例子换成flex-basis也是一样的。工作中最好用flex-basis，更符合规范。

总结：

如果父级的空间足够：flex-grow有效，flex-shrink无效。

如果父级的空间不够：flex-shrink 有效，flex-grow无效。

flex属性是以上三者的集合，一般设置为flex：1

其他属性，设置给父级元素

- flex-wrap：wrap;子元素在必要的时候换行显示。默认值是 `nowrap` ，还有一个值是 `wrap-reverse`
- flex-direction：规定主轴的方向（水平与垂直）

```
1.      row：主轴与行内轴方向作为默认的书写模式。即横向从左到右排列（左对齐）。
2.      row-reverse：对齐方式与row相反。
3.      column：主轴与块轴方向作为默认的书写模式。即纵向从上往下排列（顶对齐）。
4.      column-reverse：对齐方式与column相反。
```

- flex-flow： `flex-wrap  flex-direction` ;

- align-content：设置子元素的整体对齐方式
- 此值必须子元素占据多行

1. flex-start：各行向弹性盒容器的起始位置堆叠。弹性盒容器中第一行的侧轴起始边界紧靠住该弹性盒容器的侧轴起始边界，之后的每一行都紧靠住前面一行。
2. flex-end：各行向弹性盒容器的结束位置堆叠。弹性盒容器中最后一行的侧轴起结束界紧靠住该弹性盒容器的侧轴结束边界，之后的每一行都紧靠住前面一行。
3. center：各行向弹性盒容器的中间位置堆叠。各行两两紧靠住同时在弹性盒容器中居中对齐，保持弹性盒容器的侧轴起始内容边界和第一行之间的距离与该容器的侧轴结束内容边界与第最后一行之间的距离相等。（如果剩下的空间是负数，则各行会向两个方向溢出的相等距离。）
4. space-between：各行在弹性盒容器中平均分布。如果剩余的空间是负数或弹性盒容器中只有一行，该值等效于'flex-start'。在其它情况下，第一行的侧轴起始边界紧靠住弹性盒容器的侧轴起始内容边界，最后一行的侧轴结束边界紧靠住弹性盒容器的侧轴结束内容边界，剩余的行则按一定方式在弹性盒窗口中排列，以保持两两之间的空间相等。
5. space-around：各行在弹性盒容器中平均分布，两端保留子元素与子元素之间间距大小的一半。如果剩余的空间是负数或弹性盒容器中只有一行，该值等效于'center'。在其它情况下，各行会按一定方式在弹性盒容器中排列，以保持两两之间的空间相等，同时第一行前面及最后一行后面的空间是其他空间的一半。
6. stretch：各行将会伸展以占用剩余的空间。如果剩余的空间是负数，该值等效于'flex-start'。在其它情况下，剩余空间被所有行平分，以扩大它们的侧轴尺寸。 （默认值）

- align-items: 定义flex子项在flex容器的当前行的侧轴（纵轴）方向上的对齐方式。

1. flex-start：弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行的侧轴起始边界。
2. flex-end：弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行的侧轴结束边界。
3. center：弹性盒子元素在该行的侧轴（纵轴）上居中放置。（如果该行的尺寸小于弹性盒子元素的尺寸，则会向两个方向溢出相同的长度）。
4. baseline：如弹性盒子元素的行内轴与侧轴为同一条，则该值与'flex-start'等效。其它情况下，该值将参与基线对齐。
5. stretch：如果指定侧轴大小的属性值为'auto'，则其值会使项目的边距盒的尺寸尽可能接近所在行的尺寸，但同时会遵照'min/max-width/height'属性的限制。

- justify-content：设置盒子在主轴方向上的对齐方式

1. flex-start：弹性盒子元素将向行起始位置对齐。该行的第一个子元素的主起始位置的边界将与该行的主起始位置的边界对齐，同时所有后续的伸缩盒项目与其前一个项目对齐。
2. flex-end：弹性盒子元素将向行结束位置对齐。该行的第一个子元素的主结束位置的边界将与该行的主结束位置的边界对齐，同时所有后续的伸缩盒项目与其前一个项目对齐。
3. center：弹性盒子元素将向行中间位置对齐。该行的子元素将相互对齐并在行中居中对齐，同时第一个元素与行的主起始位置的边距等同与最后一个元素与行的主结束位置的边距（如果剩余空间是负数，则保持两端相等长度的溢出）。
4. space-between：弹性盒子元素会平均地分布在行里。如果最左边的剩余空间是负数，或该行只有一个子元素，则该值等效于'flex-start'。在其它情况下，第一个元素的边界与行的主起始位置的边界对齐，同时最后一个元素的边界与行的主结束位置的边距对齐，而剩余的伸缩盒项目则平均分布，并确保两两之间的空白空间相等。
5. space-around：弹性盒子元素会平均地分布在行里，两端保留子元素与子元素之间间距大小的一半。如果最左边的剩余空间是负数，或该行只有一个伸缩盒项目，则该值等效于'center'。在其它情况下，伸缩盒项目则平均分布，并确保两两之间的空白空间相等，同时第一个元素前的空间以及最后一个元素后的空间为其他空白空间的一半。

其他属性，设置给子元素

- align-self: 定义flex子项单独在侧轴（纵轴）方向上的对齐方式。取值与 align-items 一样。不过多了一个auto值：

1. auto：如果'align-self'的值为'auto'，则其计算值为元素的父元素的'align-items'值，如果其没有父元素，则计算为'stretch'。

- order：设置弹性盒子的顺序

允许调整大小

reisze：

1. both
2. none
3. horizontal
4. vertical

倒影

box-reflect：a b c；

1. a:left/right/above/below
2. b:距离本体多远
3. c:遮盖层

怪异盒子模型

box-sizing

1. content-box ： 默认值 盒子总宽=内(width)+padding+border
2. border-box ： 怪异盒模型 盒子的总宽=width，会对应得缩小内容部分
3. 如果border+padding > width，盒子总宽=border+padding，内容部分为0

多列布局（加上兼容性前缀）

column-width：每列的最小宽度
column-count：列数
columns： column-width column-count ；规定列的宽度和列数。
column-gap：列之间间隙的大小
column-rule：列之间的边框。值与border一样的
column-span：none/all 设置给子元素，规定这个元素跨不夸列

1. // 下面两个只兼容谷歌
2. -webkit-column-break-before: 设置或检索对象之前是否断行。设定给子元素，子元素之前是否另起一列 auto/always/avoid
3. auto：既不强迫也不禁止在元素之前断行并产生新列
4. always：总是在元素之前断行并产生新列
5. avoid：避免在元素之前断行并产生新列
6. -webkit-column-break-after: 设定给子元素，子元素之后

禁止文字选中

user-select：none；
-moz-user-select:none;
-ms-user-select:none;