

# 后盾网 人人做后盾

[www.houdunwang.com](http://www.houdunwang.com)

## Flex 弹性布局

后盾网 2011-2016

---

## 介绍

- 弹性盒布局模型（Flexible Box Layout）是 CSS3 规范中提出的一种新的布局方式。该布局模型的目的是提供一种更加高效的方式来对容器中的条目进行布局、对齐和分配空间。这种布局方式在条目尺寸未知或动态时也能工作。这种布局方式已经被主流浏览器所支持，可以在 Web 应用开发中使用。

## 实现方式

- `display: box, box-{*}` 是 2009 年的 flexbox 版本
- `display: flexbox`, 或者函数 `flex()`, 是 2011 年比较奇怪的杂糅
- `display: flex, flex-{*}`, 是目前正在使用的 2012 年发布的标准

# Flex 弹性布局

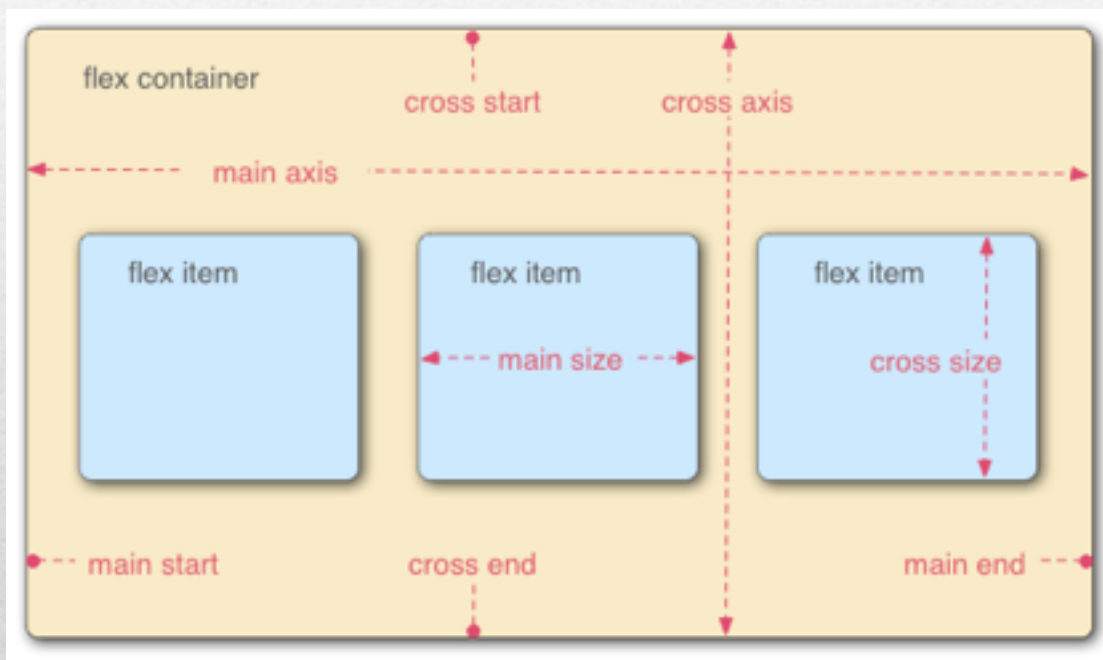
---

容器默认存在两根轴：水平的主轴（**main axis**）和垂直的交叉轴（**cross axis**）。

主轴的开始位置（与边框的交叉点）叫做**main start**，结束位置叫做**main end**；

交叉轴的开始位置叫做**cross start**，结束位置叫做**cross end**。

项目默认沿主轴排列。单个项目占据的主轴空间叫做**main size**，占据的交叉轴空间叫做**cross size**。



## Flex弹性布局

## display:flex 设置容器具有弹性特性

- 任何一个容器都可以指定为Flex布局。
- Webkit内核的浏览器，必须加上-webkit前缀。
- 采用Flex布局的元素，称为Flex容器（flex container），简称"容器"。它的所有子元素自动成为容器成员，称为Flex项目（flex item），简称"项目"。

### 示例

- `div{`
- `display: -webkit-flex;`
- `display: flex;`
- `}`

# 弹性容器

---



```
*{margin:0;padding:0}
#main
{
width:220px;
height:300px;
border:1px solid black;
display:flex;
/*可以注释掉测试*/
flex-direction: column-reverse;
}
```

```
#main div
{
/*让其均匀分配*/
flex:1;
}
```

```
<div id="main">
  <div style="background-color:coral;">后盾</div>
  <div style="background-color:lightblue;">快学</div>
  <div style="background-color:lightgreen;">论坛</div>
</div>
```

## flex-direction属性决定主轴的方向

- flex-direction: row 水平排列
- flex-direction: row-reverse 水平反转排列
- flex-direction: column 垂直排列
- flex-direction: column-reverse 垂直反转排列



# 容器属性

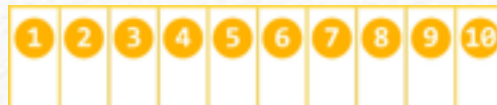
```
#main {  
  width: 200px;  
  height: 200px;  
  border: 1px solid #c3c3c3;  
  display: flex;  
  flex-wrap: wrap ;  
}
```

```
#main div {  
  width: 50px;  
  height: 50px;  
}
```

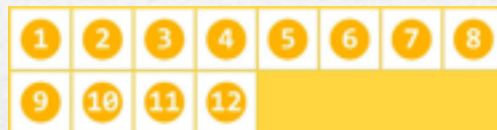
```
<div id="main">  
  <div style="background-color:coral;">A</div>  
  <div style="background-color:lightblue;">B</div>  
  <div style="background-color:khaki;">C</div>  
  <div style="background-color:pink;">D</div>  
  <div style="background-color:lightgrey;">E</div>  
  <div style="background-color:lightgreen;">F</div>  
</div>
```

## flex-wrap 如果排不下，如何换行

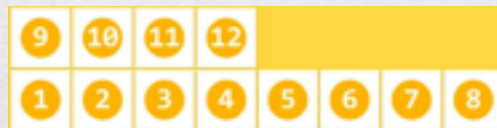
- flex-wrap: nowrap 不换行（默认）



- flex-wrap: wrap 换行，第一行在上方



- flex-wrap: wrap-reverse 换行，第一行在下方

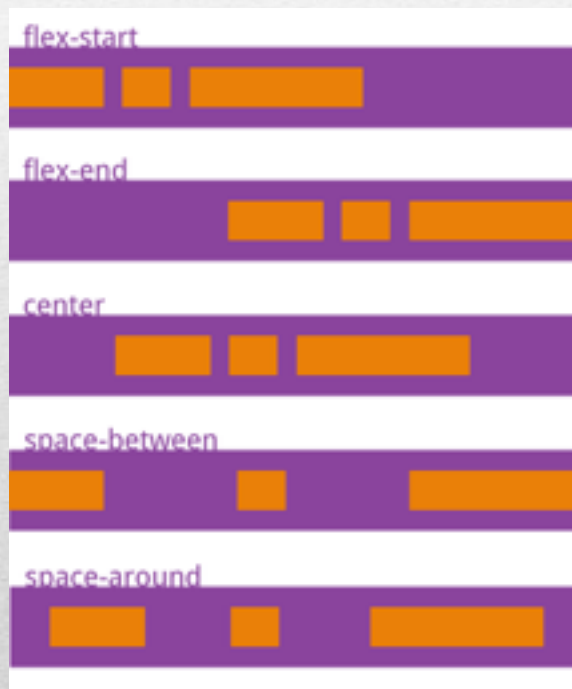


# 容器属性

## justify-content属性定义了项目的水平对齐方式

- justify-content: flex-start 左对齐（默认）
- justify-content: flex-end 右对齐
- justify-content: center 居中对齐
- justify-content: space-between 两端对齐，项目之间的间隔都相等
- justify-content: space-around 每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍

```
#main {  
  width: 400px;  
  height: 150px;  
  border: 1px solid #c3c3c3;  
  display: flex;  
  justify-content: space-between ;  
}  
#main div {width: 70px;height: 70px;}  
<div id="main">  
  <div style="background-color:coral;"></div>  
  <div style="background-color:lightblue;"></div>  
  <div style="background-color:khaki;"></div>  
  <div style="background-color:pink;"></div>  
</div>
```



## 项目的水平对齐方式

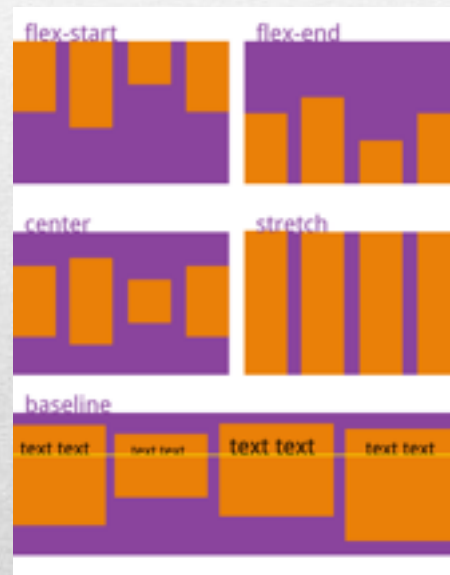


## align-items属性定义项目在垂直轴上如何对齐

- align-items: flex-start 顶端对齐
- align-items: flex-end 底部对齐
- align-items: center 中间对齐
- align-items: stretch 项目未设置高度或设为auto，将占满整个容器的高度（默认）

```
#main{
width:220px;
height:300px;
border:1px solid black;
display:flex;
align-items:center ;
}

#main div{flex: 1;}
<div id="main">
  <!-- 容器中必须有内容 -->
  <div style="background-color:coral;">A</div>
  <div style="background-color:lightblue;">B</div>
</div>
```



## 项目的垂直对齐方式



## align-content属性定义项目在交叉轴上如何对齐

- align-content属性定义了多根轴线的对齐方式。
- 容器内必须有多行的项目，该属性才能渲染出效果(用谷歌测试)

### 语法

- align-content: flex-start 顶部对齐
- align-content: flex-end 底部对齐
- align-content: center 中间对齐
- align-content: space-between 项目位于各行之间留有空白的容器内。
- align-content: space-around 项目位于各行之前、之间、之后都留有空白的容器内。
- align-content: stretch 默认值。项目被拉伸以适应容器。



## 多行元素的对齐设置

```
#main {
  width: 200px;
  height: 200px;
  border: 1px solid #c3c3c3;
  display: flex;

  /*如果注释, 下面的align-content就不管用了*/
  flex-wrap: wrap;
  align-content: center;
}
#main div {
  width: 50px;
  height: 50px;
}
<div id="main">
  <div style="background-color: coral;">A</div>
  <div style="background-color: lightblue;">B</div>
  <div style="background-color: khaki;">C</div>
  <div style="background-color: pink;">D</div>
  <div style="background-color: lightgrey;">E</div>
  <div style="background-color: lightgreen;">F</div>
</div>
```

## 多行元素的对齐设置

---

**order**属性定义项目的排列顺序。数值越小，排列越靠前，默认为0。

```
#main {  
  width: 400px;  
  height: 150px;  
  border: 1px solid #c3c3c3;  
  display: flex;  
}  
#main div {  
  width: 70px;  
  height: 70px;  
}  
div#one    {order:2;}  
div#two    {order:3;}  
div#three  {order:1;}  
  
<div id="main">  
  <div style="background-color:coral;" id="one"></div>  
  <div style="background-color:lightgreen;" id="two"></div>  
  <div style="background-color:pink;" id="three"></div>  
</div>
```

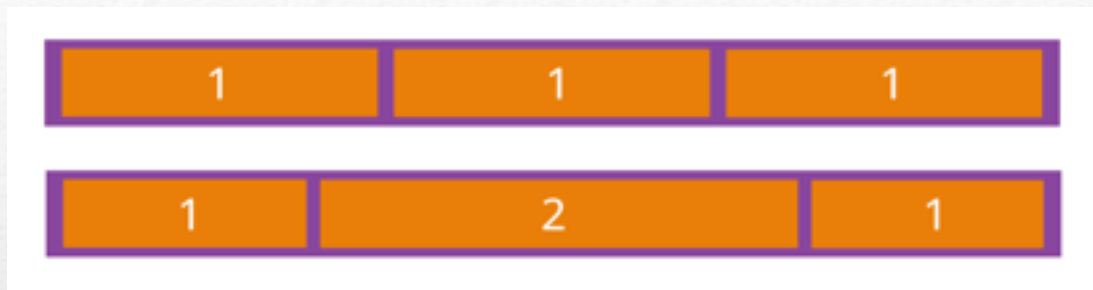
## 项目属性

---



**flex-grow**属性定义项目的放大比例，默认为0，即如果存在剩余空间，也不放大。

```
#main {  
  width: 350px;  
  height: 100px;  
  border: 1px solid #c3c3c3;  
  display: flex;  
}  
#main div:nth-of-type(1) {flex-grow: 1;}  
/*让id为main子元素div的很轻松的扩展到别的3倍*/  
#main div:nth-of-type(2) {flex-grow: 3;}  
#main div:nth-of-type(3) {flex-grow: 1;}  
#main div:nth-of-type(4) {flex-grow: 1;}  
  
<div id="main">  
  <div style="background-color:coral;"></div>  
  <div style="background-color:lightblue;"></div>  
  <div style="background-color:khaki;"></div>  
  <div style="background-color:pink;"></div>  
</div>
```



## 项目属性

## flex-basis 属性规定灵活项目的初始长度

- 如果元素不是灵活的项目，则 flex-basis 属性不起作用
- 值可以是一个长度单位或者一个百分比，规定灵活项目的初始长度

# 项目属性

---

**flex-shrink**属性定义了项目的缩小比例，默认为1，所以在空间不足情况下，该项目才将缩小。

```
#main {
  width: 300px;
  height: 100px;
  border: 1px solid #c3c3c3;
  display: flex;
}
#main div {flex-basis: 200px;}
/*子元素每个200px总共600px，父级元素300px
溢出量为300px，abc要消化这300的溢出
比例是1:2:1，那么300/4是75
所以a承担75,b承担150,c承担75
最终用基本宽度200px依次减去值就可以算出了
所以a是125,b是50,c是125*/
#main div:nth-of-type(2) {flex-shrink:2;}

<div id="main">
  <div style="background-color:coral;">a</div>
  <div style="background-color:lightblue;">b</div>
  <div style="background-color:khaki;">c</div>
</div>
```

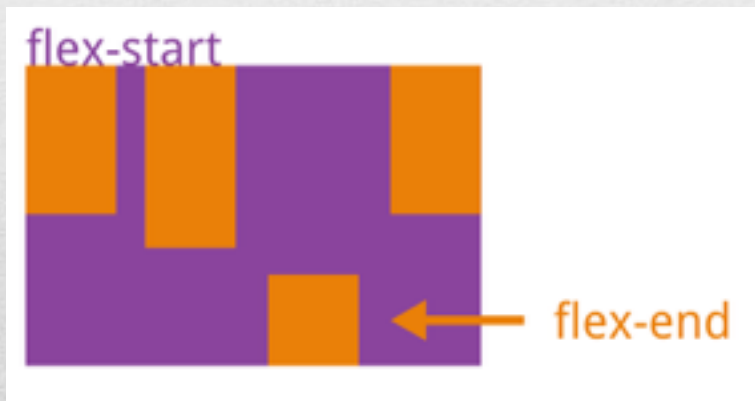
## 项目属性

---



**align-self**属性允许单个项目有与其他项目不一样的对齐方式，可覆盖**align-items**属性

- align-items: flex-start 顶端对齐
- align-items: flex-end 底部对齐
- align-items: center 中间对齐
- align-items: stretch 项目未设置高度或设为auto，将占满整个容器的高度（默认）



## 项目属性

---

```
#main{
width:220px;
height:300px;
border:1px solid black;
display:flex;
/*上面对齐*/
align-items:flex-start;
}
#main div{flex: 1;}
#love
{
/*让指定的元素居中对齐*/
align-self:center;
}
<div id="main">
  <div style="background-color:coral;">亲人</div>
  <div style="background-color:lightblue;" id="love">爱人</div>
  <div style="background-color:lightgreen;">朋友</div>
</div>
```

## 项目属性

---