

# Bootstrap 4 Flex（弹性）布局

Bootstrap 4 通过 flex 类来控制页面的布局。

## 弹性盒子(flexbox)

Bootstrap 3 与 Bootstrap 4 最大的区别就是 Bootstrap 4 使用弹性盒子来布局，而不是使用浮动来布局。

弹性盒子是 CSS3 的一种新的布局模式，更适合响应式的设计，如果你还不了解 flex，可以阅读我们的 [CSS3 弹性盒子\(Flex Box\)](#) 教程

*注意：IE9 及其以下版本不支持弹性盒子，所以如果你需要兼容 IE8-9，请使用 Bootstrap 3。*

以下实例使用 d-flex 类创建一个弹性盒子容器，并设置三个弹性子元素：

### 实例

```
<div class="d-flex p-3 bg-secondary text-white">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
```

[尝试一下 »](#)

创建显示在同一行上的弹性盒子容器可以使用 d-inline-flex 类：

### 实例

```
<div class="d-inline-flex p-3 bg-secondary text-white">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
```

[尝试一下 »](#)

## 水平方向

`.flex-row` 可以设置弹性子元素水平显示，这是默认的。

使用 `.flex-row-reverse` 类用于设置右对齐显示，即与 `.flex-row` 方向相反。

### 实例

```
<div class="d-flex flex-row bg-secondary">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
```

```
</div>
<div class="d-flex flex-row-reverse bg-secondary">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
```

[尝试一下 »](#)

## 垂直方向

`.flex-column` 类用于设置弹性子元素垂直方向显示, `.flex-column-reverse` 用于翻转子元素：

### 实例

```
<div class="d-flex flex-column">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
<div class="d-flex flex-column-reverse">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
```

[尝试一下 »](#)

## 内容排列

`.justify-content-*` 类用于修改弹性子元素的排列方式，\* 号允许的值有：**start (默认)**, **end**, **center**, **between** 或 **around**：

### 实例

```
<div class="d-flex justify-content-start">...</div>
<div class="d-flex justify-content-end">...</div>
<div class="d-flex justify-content-center">...</div>
<div class="d-flex justify-content-between">...</div>
<div class="d-flex justify-content-around">...</div>
```

[尝试一下 »](#)

## 等宽

`.flex-fill` 类强制设置各个弹性子元素的宽度是一样的：

### 实例

```
<div class="d-flex">
<div class="p-2 bg-info flex-fill">Flex item 1</div>
<div class="p-2 bg-warning flex-fill">Flex item 2</div>
```

```
<div class="p-2 bg-primary flex-fill">Flex item 3</div>
</div>
```

[尝试一下 »](#)

## 扩展

`.flex-grow-1` 用于设置子元素使用剩下的空间。以下实例中前面两个子元素只设置了它们所需要的空间，最后一个获取剩余空间。：

### 实例

```
<div class="d-flex">
<div class="p-2 bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary flex-grow-1">Flex item 3</div>
</div>
```

[尝试一下 »](#)

**提示：**使用 `.flex-shrink-1` 用于设置子元素的收缩规则。

## 排序

`.order` 类可以设置弹性子元素的排序，从 `.order-1` 到 `.order-12`，数字越低权重越高（`.order-1` 排在 `.order-2` 之前）：

### 实例

```
<div class="d-flex bg-secondary">
<div class="p-2 bg-info order-3">Flex item 1</div>
<div class="p-2 bg-warning order-2">Flex item 2</div>
<div class="p-2 bg-primary order-1">Flex item 3</div>
</div>
```

[尝试一下 »](#)

## 外边距

`.mr-auto` 类可以设置子元素右外边距为 `auto`，即 `margin-right: auto!important;`，`.ml-auto` 类可以设置子元素左外边距为 `auto`，即 `margin-left: auto!important;`：

### 实例

```
<div class="d-flex bg-secondary">
<div class="p-2 mr-auto bg-info">Flex item 1</div>
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 bg-primary">Flex item 3</div>
</div>
<div class="d-flex bg-secondary">
<div class="p-2 bg-info">Flex item 1</div>
```

```
<div class="p-2 bg-warning">Flex item 2</div>
<div class="p-2 ml-auto bg-primary">Flex item 3</div>
</div>
```

[尝试一下 »](#)

## 包裹

弹性容器中包裹子元素可以使用以下三个类：.flex-nowrap (默认), .flex-wrap 或 .flex-wrap-reverse。设置 flex 容器是单行或者多行。

### 实例

```
<div class="d-flex flex-wrap">..</div>
<div class="d-flex flex-wrap-reverse">..</div>
<div class="d-flex flex-nowrap">..</div>
```

[尝试一下 »](#)

## 内容对齐

我们可以使用 .align-content-\* 来控制在垂直方向上如何去堆叠子元素，包含的值有：.align-content-start (默认), .align-content-end, .align-content-center, .align-content-between, .align-content-around 和 .align-content-stretch。

这些类在只有一行的弹性子元素中是无效的。

### 实例

```
<div class="d-flex flex-wrap align-content-start">..</div>
<div class="d-flex flex-wrap align-content-end">..</div>
<div class="d-flex flex-wrap align-content-center">..</div>
<div class="d-flex flex-wrap align-content-around">..</div>
<div class="d-flex flex-wrap align-content-stretch">..</div>
```

[尝试一下 »](#)

## 子元素对齐

如果要设置单行的子元素对齐可以使用 .align-items-\* 类来控制，包含的值有：.align-items-start, .align-items-end, .align-items-center, .align-items-baseline, 和 .align-items-stretch (默认)。

### 实例

```
<div class="d-flex flex-wrap align-content-start">..</div>
<div class="d-flex flex-wrap align-content-end">..</div>
<div class="d-flex flex-wrap align-content-center">..</div>
<div class="d-flex flex-wrap align-content-around">..</div>
<div class="d-flex flex-wrap align-content-stretch">..</div>
```

[尝试一下 »](#)

## 指定子元素对齐

如果要设置指定子元素对齐可以使用 `.align-self-*` 类来控制，包含的值有：`.align-self-start`, `.align-self-end`, `.align-self-center`, `.align-self-baseline`, 和 `.align-self-stretch` (默认)。

### 实例

```
<div class="d-flex bg-light" style="height:150px">
<div class="p-2 border">Flex item 1</div>
<div class="p-2 border align-self-start">Flex item 2</div>
<div class="p-2 border">Flex item 3</div>
</div>
```

尝试一下 »

## 响应式 flex 类

我们可以根据不同的设备，设置 flex 类，从而实现页面响应式布局，以下表格中的 \* 号可以的值有：sm, md, lg 或 xl, 对应的是小型设备、中型设备，大型设备，超大型设备。

类	实例	实现
弹性容器		
<code>.d-*-flex</code>	根据不同的屏幕设备创建弹性盒子容器	尝试一下
<code>.d-*-inline-flex</code>	根据不同的屏幕设备创建行内弹性盒子容器	尝试一下
方向		
<code>.flex-*-row</code>	根据不同的屏幕设备在水平方向显示弹性子元素	尝试一下
<code>.flex-*-row-reverse</code>	根据不同的屏幕设备在水平方向显示弹性子元素，且右对齐	尝试一下
<code>.flex-*-column</code>	根据不同的屏幕设备在垂直方向显示弹性子元素	尝试一下
<code>.flex-*-column-reverse</code>	根据不同的屏幕设备在垂直方向显示弹性子元素，且方向相反	尝试一下
内容对齐		
<code>.justify-content-*-start</code>	根据不同屏幕设备在开始位置显示弹性子元素 (左对齐)	尝试一下
<code>.justify-content-*-end</code>	根据不同屏幕设备在尾部显示弹性子元素 (右对齐)	尝试一下

<code>.justify-content-*-center</code>	根据不同屏幕设备在 flex 容器中居中显示子元素	尝试一下
<code>.justify-content-*-between</code>	根据不同屏幕设备使用 "between" 显示弹性子元素	尝试一下
<code>.justify-content-*-around</code>	根据不同屏幕设备使用 "around" 显示弹性子元素	尝试一下
等宽		
<code>.flex-*-fill</code>	根据不同的屏幕设备强制等宽	尝试一下
扩展		
<code>.flex-*-grow-0</code>	不同的屏幕设备不设置扩展	
<code>.flex-*-grow-1</code>	不同的屏幕设备设置扩展	
收缩		
<code>.flex-*-shrink-0</code>	不同的屏幕设备不设置收缩	
<code>.flex-*-shrink-1</code>	不同的屏幕设备设置收缩	
包裹		
<code>.flex-*-nowrap</code>	不同的屏幕设备不设置包裹元素	尝试一下
<code>.flex-*-wrap</code>	不同的屏幕设备设置包裹元素	尝试一下
<code>.flex-*-wrap-reverse</code>	不同的屏幕设备反转包裹元素	尝试一下
内容排列		
<code>.align-content-*-start</code>	根据不同屏幕设备在起始位置堆叠元素	尝试一下
<code>.align-content-*-end</code>	根据不同屏幕设备在结束位置堆叠元素	尝试一下
<code>.align-content-*-center</code>	根据不同屏幕设备在中间位置堆叠元素	尝试一下
<code>.align-content-*-around</code>	根据不同屏幕设备，使用 "around" 堆叠元素	尝试一下

.align-content--stretch	根据不同屏幕设备，通过伸展元素来堆叠	尝试一下
元素对齐		
.align-items--start	根据不同屏幕设备，让元素在头部显示在同一行。	尝试一下
.align-items--end	根据不同屏幕设备，让元素在尾部显示在同一行。	尝试一下
.align-items--center	根据不同屏幕设备，让元素在中间位置显示在同一行。	尝试一下
.align-items--baseline	根据不同屏幕设备，让元素在基线上显示在同一行。	尝试一下
.align-items--stretch	根据不同屏幕设备，让元素延展高度并显示在同一行。	尝试一下
单独一个子元素的对齐方式		
.align-self--start	据不同屏幕设备，让单独一个子元素显示在头部。	尝试一下
.align-self--end	据不同屏幕设备，让单独一个子元素显示在尾部	尝试一下
.align-self--center	据不同屏幕设备，让单独一个子元素显示在居中位置	尝试一下
.align-self--baseline	据不同屏幕设备，让单独一个子元素显示在基线位置	尝试一下
.align-self--stretch	据不同屏幕设备，延展一个单独子元素	尝试一下

 点我分享笔记