



第8章 Bootstrap响应式布局





学习要点

- 1 • 响应式Web设计和视口概念
- 2 • 弹性盒布局
- 3 • Bootstrap框架下载与部署
- 4 • Bootstrap布局容器
- 5 • Bootstrap栅格系统
- 6 • Bootstrap组件及公共样式



响应式Web设计

随着移动产品的日益丰富，出现了各种屏幕尺寸的手机、平板等移动设备。在响应式Web设计出现之前，对于同样的内容分别针对每一种尺寸的屏幕独立开发一个网站，成本非常高。因此，响应式Web设计应运而生，它可以让一个网站同时适配多种屏幕和多个设备，同时让网站的布局和功能随用户的设备屏幕大小而随时变化。

视口

实现响应式Web设计，首先要了解的是视口的概念。视口(viewport)最早是苹果公司推出 iPhone时发明的，为的是让iPhone的小屏幕尽可能完整显示整个网页。在移动端浏览器中存在两种视口，一种是可见视口，即设备大小；另一种是视窗视口，即网页宽度。如图8-1所示，设备屏幕是411px的宽度，浏览器中，411px的屏幕宽度能够展示980px宽度内容。那么，411px就是可见视口宽度，980px就是视窗视口的宽度。



为了显示更多内容，浏览器会经过viewport的默认缩放将网页等比例缩小。但是，为了让用户能够看清设备中的内容，通常情况下，并不使用默认的viewport进行展示，而是自定义配置视口的属性，使这个缩小比例更加适当。HTML5中，<meta>标签用于配置视口属性。

```
<meta name=' viewport' content=' user-scalable=no, width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0' >
```

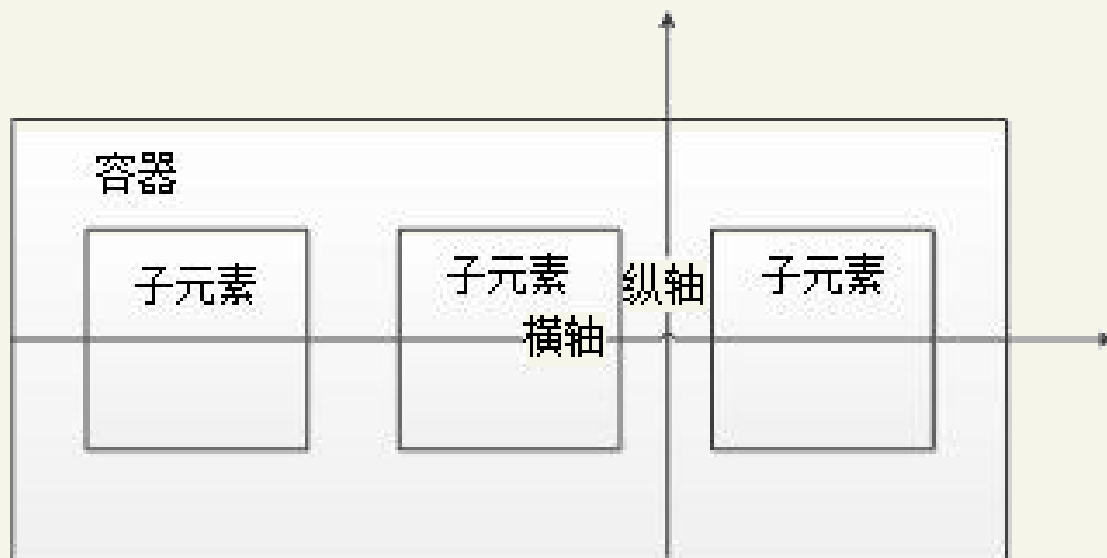
属性	取值	描述
width	正整数或 device-width	定义视口宽度，单位 px； 取值 device-width 说明视窗视口与可见视口宽度相同
height	正整数或 device-height	定义视口高度，单位 px，一般不用
user-scalable	yes/no	是否允许用户手动缩放页面，默认值 yes
initial-scale	[0.0-10.0]	初始缩放值
minimum-scale	[0.0-10.0]	缩小最小比例，必须小于或等于 maximum-scale 设置
maximum-scale	[0.0-10.0]	放大最大比例，必须大于或等于 minimum-scale 设置



弹性盒布局（**flexible box**）可以轻松地创建响应式网页布局，采用了弹性盒布局的元素称之为弹性盒容器，它的目的在于通过弹性的方式来对齐和分布弹性盒容器中内容的空间，使其能适应不同屏幕，为盒模型提供最大的灵活性，并且不需要使用浮动，也不会弹性盒容器与其内容之间产生外边距塌陷，是一种非常灵活的布局方式。

8.2.1 弹性盒布局结构

了解弹性盒布局的结构，如图8-2所示。



8.2.2 弹性盒容器属性

1) display

display用于指定目标元素为弹性盒容器，其值有两个：flex和inline-flex，如果目标元素为行内元素则使用inline-flex。在代码demo8-1.html中，.flex-container的css增加display:flex，保存代码后，查看效果。

2) flex-direction

flex-direction属性决定弹性盒容器内子元素的排列方向，取值如表8-1所示。

表 8-1 flex-direction 的排列方向

取值	意义
row	默认值，主轴为水平方向，起点在左端
row-reverse	主轴为水平方向，起点在右端
column	主轴为垂直方向，起点在上沿
column-resverse	主轴为垂直方向，起点在下沿



8.2.2 弹性盒容器属性

3) flex-wrap

默认情况下，子元素都排在轴线上，flex-wrap属性定义如果一条轴线排不下，如何换行，相应属性值如表8-2所示。

表8-2 flex-wrap属性值

取值	意义
nowrap	默认值，不换行
wrap	换行，第一行在上方
wrap-reverse	换行并且反转，第一行在下方



8.2.2 弹性盒容器属性

4) flex-flow

flex-flow是flex-direction属性和flex-wrap属性的简写形式，默认值为row nowrap。在代码demo8-1.html中，加多数个.flex-item元素结构，并且在.flex-container的css增加flex-flow: column wrap，保存代码后，查看效果，用户自行操作，操作完成后还原代码。

5) justify-content

justify-content属性定义了子元素在主轴上的对齐方式，如表8-3所示。

表8-3 justify-content属性值

取值	意义
flex-start	默认值，左对齐
flex-end	右对齐
center	居中
space-between	两端对齐，子元素之间的间隔都相等
space-around	每个子元素两侧的间隔相等，因此子元素之间的间隔比子元素与边框的间隔大一倍



8.2.2 弹性盒容器属性

6) align-items

align-items属性定义子元素在垂直于主轴方向上的对齐方式，如表8-4所示。

表8-4 align-items属性值

取值	意义
flex-start	垂直于主轴方向起点对齐
flex-end	垂直于主轴方向终点对齐
center	垂直于主轴方向中间位置对齐
baseline	子元素的第一行文字的基线对齐
stretch	默认值，如果子元素未设置高度或设为 auto，将占满整个容器的高度



8.2.3 弹性盒容器子元素属性

1) order

order属性定义子元素的排列顺序。默认值为0，数值越小，排列越靠前。例如，将A、B、C的order值分别改为2、3、1，保存代码查看效果。

2) flex-grow

flex-grow属性定义子元素的放大比例，默认值为0，也就是说如果存在剩余空间，也不放大。如果所有子元素的**flex-grow**属性都为1，则它们将剩余空间等分。如果一个子元素的**flex-grow**属性为2，其它子元素都为1，则前者占据的剩余空间将比其它子元素多一倍。

3) flex-shrink

flex-shrink属性定义了子元素的缩小比例，默认值为1，也就是说如果空间不足，该子元素将缩小。如果所有子元素的**flex-shrink**属性都为1，当空间不足时，都将等比例缩小。如果一个子元素的**flex-shrink**的属性为0，其它子元素都为1，当空间不足时，前者不缩小。



8.2.3 弹性盒容器子元素属性

4) flex-basis

flex-basis属性定义了分配多余空间之前，子元素占据的主轴空间。浏览器根据这个属性，计算主轴中是否有多余空间。默认值为**auto**，也就是子元素的本来大小。也可以设置成具体值（比如**200px**），那么子元素将占据固定空间。

5) flex

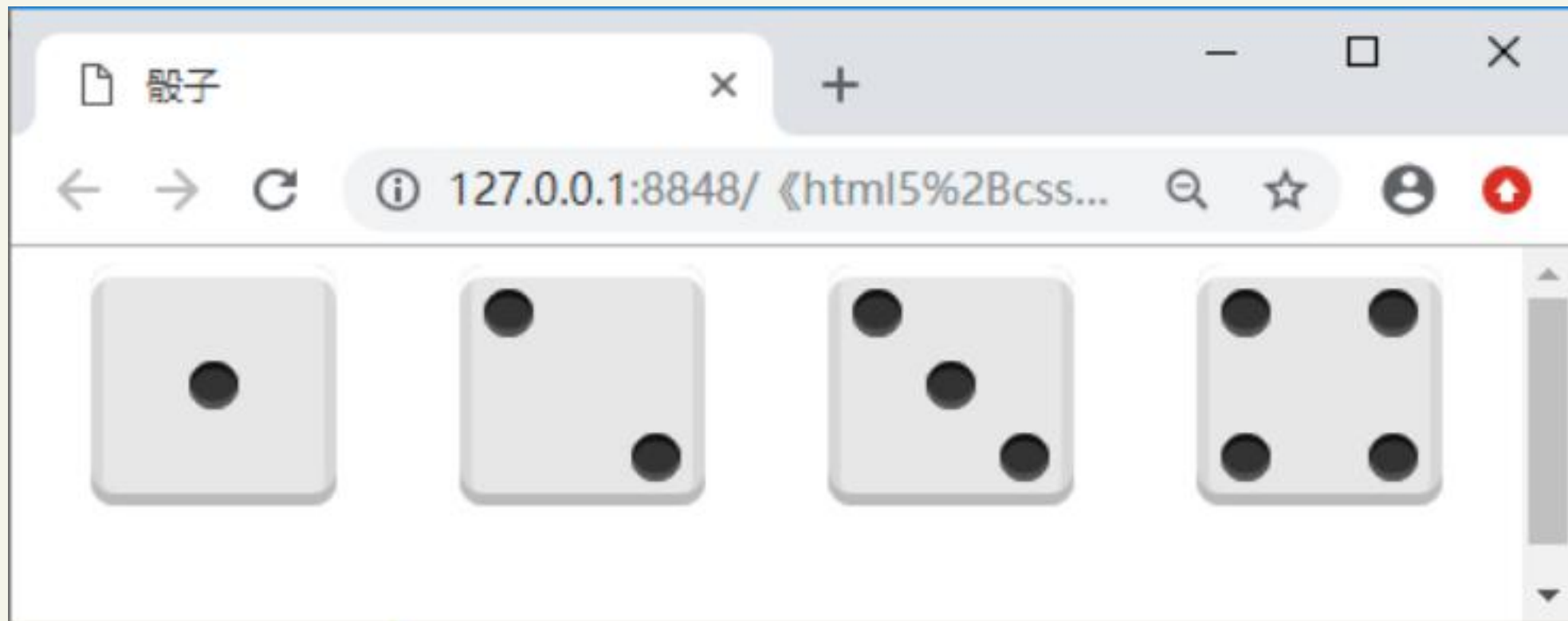
flex属性是**flex-grow**，**flex-shrink**和**flex-basis**的简写形式，默认值为**0 1 auto**。后两个属性可选。例如，设A的**flex: 2 1 500px**，B与C设**flex:1**后保存代码查看效果。

6) align-self

align-self属性允许子元素有与其它子元素不一样的对齐方式，可覆盖容器的**align-items**属性，默认值为**auto**，表示继承父元素的**align-items**属性，如果没有父元素，则等同于**stretch**。例如，将B、C的CSS增加**align-self: stretch**;



利用弹性盒布局属性制作”骰子”（案例8-2），效果如下：

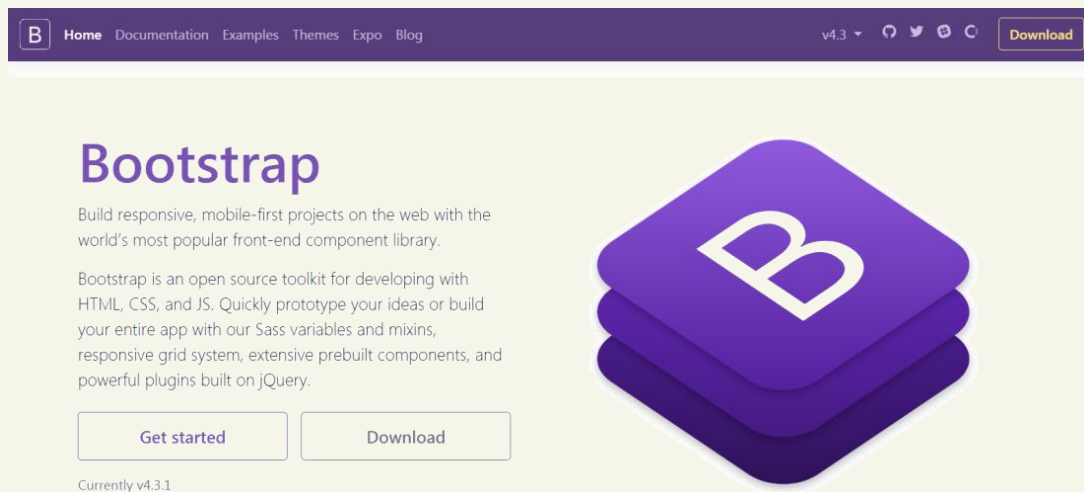


8.3.1 Bootstrap下载与快速部署

Bootstrap是目前较受欢迎的前端框架，可以满足响应式页面的开发需求，并且遵循移动优先的原则，它是基于HTML、CSS、JAVASCRIPT等前端技术基础上实现的。

Bootstrap中预定了一套CSS样式和一套对应的jQuery代码，应用时只需提供固定的HTML结构，添加**Bootstrap**中提供的class名称，就可以完成指定效果的实现。

Bootstrap官方网站地址：<https://getbootstrap.com>，读者可前去下载最新版本。本教材在编写时它的最新版本是v4.3.1。官网首页如图8-6所示。



8.3.1 Bootstrap下载与快速部署

Bootstrap v4.3.1的使用需要有4个文件支持，所以建议读者直接使用CDN方式嵌入HTML文件的方式完成快速部署，而无需下载，避免占用服务器资源，加快页面响应速度，当然前提条件是网络通畅。

CDN方式快速部署Bootstrap v4.3.1有4个文件、视口和头部规范。

1.1行CSS文件

2.3行JS文件

3.视口

4.文件头部规范

具体代码请参看技术开发手册，技术开发手册中提供了最精简的模板“Hello, World!”，包含了以上各方面，可直接复制并粘贴至本地HTML文件中使用。至此，Bootstrap v4.3.1（以下简称Bootstrap）安装完成。



8.3.2 栅格系统

栅格系统用于通过放置在**容器**中的一系列**行**（row）与**列**（column）所组成的**格子**来创建网页布局，网页内容就放置在格子中，不仅可以让网页的信息呈现美感与易读，网页布局也更加灵活与规范。

1.Container容器

Container容器是Bootstrap框架的最基本元素，这是启用整个栅格系统必不可少的前置条件。容器中包含行（.row），行中包含列（.col）。Container包含两个样式：

- .container样式设置一个响应式的、居中的容器
- .container-fluid样式设置的是一个宽度为100%的最大合法宽度的容器



8.3.2 栅格系统

2.栅格系统

Bootstrap包含了一个强大的移动优先的栅格系统，它是基于一个12列的布局，有5种响应尺寸（对应不同的屏幕），完全基于flex弹性盒布局构建，并结合自己预定义的CSS、JavaScript类，来创建各种形状和尺寸的布局。

Bootstrap使用ems和rems来定义大多数属性的规格大小，px用于全局层面栅格断点和容器宽度，Bootstrap栅格系统在各种屏幕和设备上的参数如表8-5所示。

表 8-5 栅格系统在各种屏幕和设备上的参数

	超小屏幕 <576px	小屏幕 ≥576px	中等屏幕 ≥768px	大屏幕 ≥992px	超大屏幕 ≥1200px
.container 最大宽度	None (auto)	540px	720px	960px	1140px
类前缀	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
列数 (column)	12				
列间隔	30px (每列两侧各占 15px)				



8.3.2 栅格系统

2. 栅格系统

Bootstrap栅格系统通过一系列包含内容的行和列来创建页面布局，其基本工作原理如下：

- 由Container类来进行网页布局
- Container中必须创建行(.row)
- 网页内容必须放在列(.col)内，且只有列才是行的直接子元素
- 每列都有水平的padding值，用于控制它们之间的间隔
- .col-sm-5中的数字表示在小屏幕尺寸中这列要占用的列数是5列，每行最多12列
- 如果只定义一个屏幕规格即可向上覆盖所有设备，向下如果没有定义则默认为12栅格。



8.3.3 组件及公用样式

组件是Bootstrap中已经封装好的一组功能块，以实现具体的功能，如导航栏和轮播图，用户只需要替换相应的网页内容即可使用。公用样式则是在整个Bootstrap系统中已经定义好的类，可以被全局调用，用户只需要引用类名即可实现具体效果，如颜色类。

1. 导航栏

导航栏是网页中一个非常重要的基础组件。Bootstrap提供了响应式导航条的样例，它在小屏幕上水平导航栏会切换为垂直。其所需的样式如下：

- `.navbar`类创建一个标准的导航栏
- `.navbar-expand-xl|lg|md|sm`类来创建响应式的导航栏（大屏幕水平铺开，小屏幕垂直堆叠）
- `.navbar-brand`用于定义公司或产品的品牌图标。
- 导航栏上使用`<ul>`元素并添加`.navbar-nav`类来定义导航栏上的选项
- 每个`<li>`元素上添加`.nav-item`类，`<a>`元素上使用`.nav-link`类
- 创建折叠导航栏，按钮上添加`.navbar-toggler`，增加`data-toggle='collapse'`，`data-target='#thetarget'`属性，然后在设置了`.collapse navbar-collapse`类的div上包裹导航内容(ul)，div元素上的id匹配按钮.data-target上的指定id



8.3.3 组件及公用样式

2.轮播图

轮播图类似一个循环播放的幻灯片，使用了CSS的3D变形转换和JavaScript制作。轮播图所使用的类说明表8-6所示。

表8-6 图片轮播类说明

类	描述
.carousel	创建一个轮播
.carousel-indicators	为轮播添加一个指示符，就是轮播图底下的一个个小点，轮播的过程可以显示目前是第几张图
.carousel-inner	添加切切换的图片
.carousel-item	指定每个图片的内容
.carousel-control-prev	添加左侧按钮，点击会返回上一张
.carousel-control-next	添加右侧按钮，点击会切换到下一张
.carousel-control-prev-icon	与.carousel-control-prev一起使用，设置左侧按钮
.carousel-control-next-icon	与.carousel-control-next一起使用，设置右侧按钮



8.3.3 组件及公用样式

3.Bootstrap中的弹性盒布局

Bootstrap v4与Bootstrap v3的最大区别之一是Bootstrap v4使用了弹性盒Flex布局，而Bootstrap v3使得的是float浮动布局方式。在Bootstrap v4中如果需要使用flex，可以通过一整套灵活的类来管理，不推荐使用display: flex，那会带来不必要的影响。

- .d-flex, .d-lg-flex等，表示启用栅格布局
- .flex-row, .flex-column, .flex-row-reverse等设置弹性盒子元素的排列方式
- .justify-content-start, .justify-content-center等设置子元素在主轴的对齐位置
- .align-items-start, .align-items-stretch等设置子元素在垂直轴方向的对齐位置



8.3.3 组件及公用样式

4.颜色

Bootstrap中，提供了很多可供全局使用的公共样式，其中颜色类是比较频繁使用的一种。Bootstrap定义的文字颜色类，目的在于通过颜色传达意义，表达不同的模块。这些颜色类包括以下：

.text-primary, .text-secondary, .text-success, .text-danger, .text-warning, .text-info, .text-light .text-dark, text-body, .text-muted, .text-white, .text-black-50 and .text-white-50



8.3.4 Bootstrap响应式布局案例

图8-12 大屏幕尺寸效果图



图8-13 中等屏幕以下尺寸效果图

