**• 前言：不同框架的定位与适用场景**

**• 组件的理解和分类**

**• 变化侦测机制**

**• 路由**

**• 状态和数据流管理**

**• 数据抓取和同步**

**• CSS 管理方案**

**• 构建工具链**

**• 同构/服务端渲染**

**• 跨平台渲染**

**• 类型系统**

**• 构建时优化**

**• 运行时优化**

**• Web Components 和框架的关系**

**• Web Assembly 和框架的关系**

**响应式 Web 设计**

响应式 Web 设计是一个让用户通过各种尺寸的设备浏览网站获得良好的视觉效果的方法。例如，您先在计算机显示器上浏览一个网站，然后在智能手机上浏览，智能手机的屏幕尺寸远小于计算机显示器，但是你却没有感觉到任何差别，两者的用户体验几乎一样。

响应式 Web 设计工作原理

为了应用响应式 Web 设计，您需要创建一个包含适应各种设备尺寸样式的 CSS。一旦页面在特定的设备上加载，该页面上使用了各种字体和 Web 开发技术，比如媒体查询（Media Queries），此时，会先检测设备的视口大小，然后加载特定于设备的样式。

Bootstrap 3 的设计目标是移动设备优先，然后才是桌面设备。这实际上是一个非常及时的转变，因为现在越来越多的用户使用移动设备。

为了让 Bootstrap 开发的网站对移动设备友好，确保适当的绘制和触屏缩放，需要在网页的 head 之中添加 viewport meta 标签，如下所示：

<meta name="viewport" content="width=device-width, initial-scale=1.0">

width 属性控制设备的宽度。假设您的网站将被带有不同屏幕分辨率的设备浏览，那么将它设置为 device-width 可以确保它能正确呈现在不同设备上。

initial-scale=1.0 确保网页加载时，以 1:1 的比例呈现，不会有任何的缩放。

在移动设备浏览器上，通过为 viewport meta 标签添加 user-scalable=no 可以禁用其缩放（zooming）功能。

通常情况下，maximum-scale=1.0 与 user-scalable=no 一起使用。这样禁用缩放功能后，用户只能滚动屏幕，就能让您的网站看上去更像原生应用的感觉。

响应式图像

<img src="..." class="img-responsive" alt="响应式图像">

通过添加 img-responsive class 可以让 Bootstrap 3 中的图像对响应式布局的支持更友好。

接下来让我们看下这个 class 包含了哪些 css 属性。

在下面的代码中，可以看到img-responsive class 为图像赋予了 max-width: 100%; 和 height: auto; 属性，可以让图像按比例缩放，不超过其父元素的尺寸。

.img-responsive {

display: block;

height: auto;

max-width: 100%;

}

这表明相关的图像呈现为 block。当您把元素的 display 属性设置为 block，以块级元素显示。

设置 height:auto，相关元素的高度取决于浏览器。

设置 max-width 为 100% 会重写任何通过 width 属性指定的宽度。这让图片对响应式布局的支持更友好。

如果需要让使用了 .img-responsive 类的图片水平居中，请使用 .center-block 类，不要用 .text-center。

什么是网格（Grid）？

在平面设计中，网格是一种由一系列用于组织内容的相交的直线（垂直的、水平的）组成的结构（通常是二维的）。它广泛应用于打印设计中的设计布局和内容结构。在网页设计中，它是一种用于快速创建一致的布局和有效地使用 HTML 和 CSS 的方法。

简单地说，网页设计中的网格用于组织内容，让网站易于浏览，并降低用户端的负载。

什么是 Bootstrap 网格系统（Grid System）？

Bootstrap 官方文档中有关网格系统的描述：

Bootstrap 包含了一个响应式的、移动设备优先的、不固定的网格系统，可以随着设备或视口大小的增加而适当地扩展到 12 列。它包含了用于简单的布局选项的预定义类，也包含了用于生成更多语义布局的功能强大的混合类。

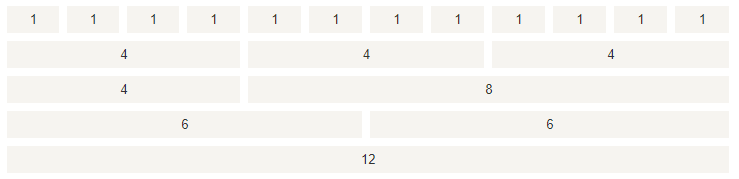
让我们来理解一下上面的语句。Bootstrap 3 是移动设备优先的，在这个意义上，Bootstrap 代码从小屏幕设备（比如移动设备、平板电脑）开始，然后扩展到大屏幕设备（比如笔记本电脑、台式电脑）上的组件和网格。

移动设备优先策略

内容：决定什么是最重要的。

布局：优先设计更小的宽度。基础的 CSS 是移动设备优先，媒体查询是针对于平板电脑、台式电脑。

渐进增强：随着屏幕大小的增加而添加元素。响应式网格系统随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。



Bootstrap 网格系统（Grid System）的工作原理

网格系统通过一系列包含内容的行和列来创建页面布局。下面列出了 Bootstrap 网格系统是如何工作的：

行必须放置在 .container class 内，以便获得适当的对齐（alignment）和内边距（padding）。

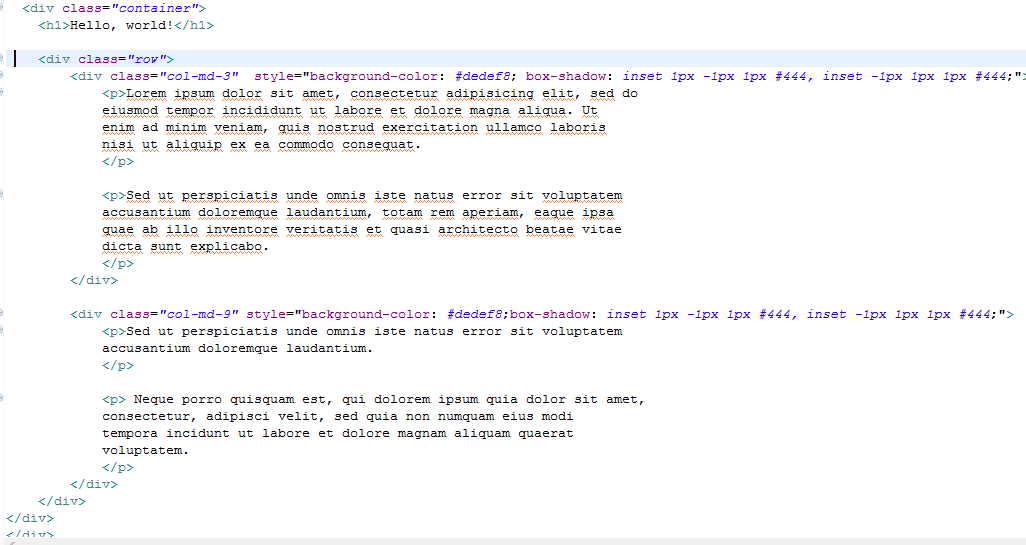
使用行来创建列的水平组。

内容应该放置在列内，且唯有列可以是行的直接子元素。

预定义的网格类，比如 .row 和 .col-xs-4，可用于快速创建网格布局。LESS 混合类可用于更多语义布局。

列通过内边距（padding）来创建列内容之间的间隙。该内边距是通过 .rows 上的外边距（margin）取负，表示第一列和最后一列的行偏移。

网格系统是通过指定您想要横跨的十二个可用的列来创建的。例如，要创建三个相等的列，则使用三个 .col-xs-4。



细节

<div class="container">...</div> 元素被添加，确保居中和最大宽度。

一旦添加了容器，接下来您需要考虑以行为单位。添加 <div class="row">...</div>，并在行内添加列 <div class="col-md-6"></div>。

网格中的每一行是由 12 个单元组成的，您可以使用这些单元定义列的尺寸。在我们的实例中，有两个列，每个列由 6 个单元组成，即 6+6=12。

您可以尝试其他更多的选项，比如 <div class="col-md-3"></div> 和 <div class="col-md-9"></div> 或 <div class="col-md-7"></div> 和 <div class="col-md-5"></div>。

您可以尝试一下，但要确保总和总是 12。

现在 Bootstrap 在中型设备中，会查找带有 md 的类，并使用它们。在大型设备中，会查找带有 lg 的类，并使用它们。在本实例中，我们的 2 个 div 将从 50%/50% 分割转变为 33%/66%。



现在，给我们提供了 3 种不同的列布局，分别适用于三种设备。在手机上，它将是左边 25% 右边 75% 的布局。在平板电脑上，它将是 50%/50% 的布局。在大型视口的设备上，它将是 33%/66% 的布局。



响应式的列重置

以下实例包含了4个网格，但是我们在小设备浏览时无法确定网格显示的位置。

为了解决这个问题，可以使用 .clearfix class和 响应式实用工具来解决，如下面实例所示：





为了在内容中嵌套默认的网格，请添加一个新的 .row，并在一个已有的 .col-md-\* 列内添加一组 .col-md-\* 列。被嵌套的行应包含一组列，这组列个数不能超过12（其实，没有要求你必须占满12列）。

