

## 第 9 天【媒体查询和多列】

### 主要内容

- 1、媒体查询
- 2、多列

### 学习目标

节数	知识点	要求
第一节（媒体查询）	概念	了解
	优缺点	了解
	设置步骤	掌握
第二节	响应式布局案例	掌握
第三节（多列）	定义	了解
	属性	了解

## 一、媒体查询

### 1.1 概念

一说到响应式设计，肯定离不开媒体查询 media。一般认为媒体查询是 CSS3 的新增内容，实际上 CSS2 已经存在了，CSS3 新增了媒体属性和使用场景(IE8-浏览器不支持)。

Responsive design，意在实现不同屏幕分辨率的终端上浏览网页的不同展示方式。通过响应式设计能使网站在手机和平板电脑上有更好的浏览阅读体验。



### 1.2 响应式布局的优缺点

#### 1.2.1 优点

- (1) 面对不同分辨率设备灵活性强
- (2) 能够快捷解决多设备显示适应问题

#### 1.2.2 缺点

- (1) 兼容各种设备工作量大，效率低下
- (2) 代码累赘，会出现隐藏无用的元素，加载时间加长

### 1.3 响应式设计的步骤

#### 1.3.1 设置 meta 标签

大多数移动浏览器将 HTML 页面放大为宽的视图 (viewport) 以符合屏幕分辨率。你可以使用视图的 meta 标签来进行重置。下面的视图标签告诉浏览器，使用设备的宽度作为视图宽度并禁止初始的缩放。在 <head> 标签里加入这个 meta 标签。

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
```

参数解释：

**width = device-width**：宽度等于当前设备的宽度

**initial-scale**：初始的缩放比例（默认设置为 1.0）

**minimum-scale**：允许用户缩放到的最小比例（默认设置为 1.0）

**maximum-scale**：允许用户缩放到的最大比例（默认设置为 1.0）

**user-scalable**：用户是否可以手动缩放（默认设置为 no，因为我们不希望用户放大缩小页面）

### 1.3.2 通过媒介查询来设置 Media Queries

Media Queries 是响应式设计的核心。

Media Queries 能在不同的条件下使用不同的样式，使页面在不同终端设备下达到不同的渲染效果。

具体来说,Media Queries 的使用方法如下。

内部引入：

**@media 媒体类型 and （媒体特性）{你的样式}**

外部引入：

**<link rel="stylesheet" href="" media="screen and （宽度）"/>**

媒体特性是通过 min/max 来表示大于等于或小于做为逻辑判断，

而不是使用小于( < )和大于( > )这样的符号来判断。接下来一起来看看 Media Queries 在实际项目中常用的方式。

#### 1. 最大宽度 max-width

“max-width” 是媒体特性中最常用的一个特性，其意思是指媒体类型小于或等于指定的宽度时，样式生效。如：

```
@media screen and (max-width:480px){  
  
  .ads {  
  
    display:none;  
  
  }  
}
```

```
}
```

上面表示的是：当屏幕小于或等于 480px 时,页面中的广告区块（.ads）都将被隐藏。

## 2.最小宽度 min-width

“min-width”与“max-width”相反,指的是媒体类型大于或等于指定宽度时,样式生效。

```
@media screen and (min-width:900px){  
    .wrapper {  
        width: 980px;  
    }  
}
```

上面表示的是：当屏幕大于或等于 900px 时，容器“.wrapper”的宽度为 980px。

## 3.多个媒体特性使用

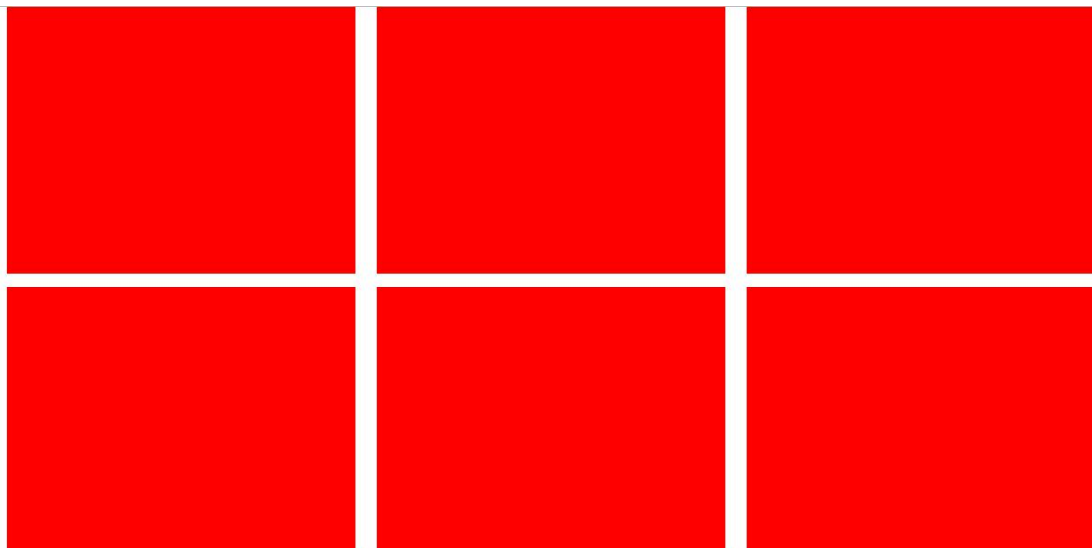
Media Queries 可以使用关键词“and”将多个媒体特性结合在一起。也就是说，一个 Media Query 中可以包含 0 到多个表达式，表达式又可以包含 0 到多个关键字，以及一种媒体类型。

当屏幕在 600px~900px 之间时，body 的背景色渲染为 “#f5f5f5”，如下所示。

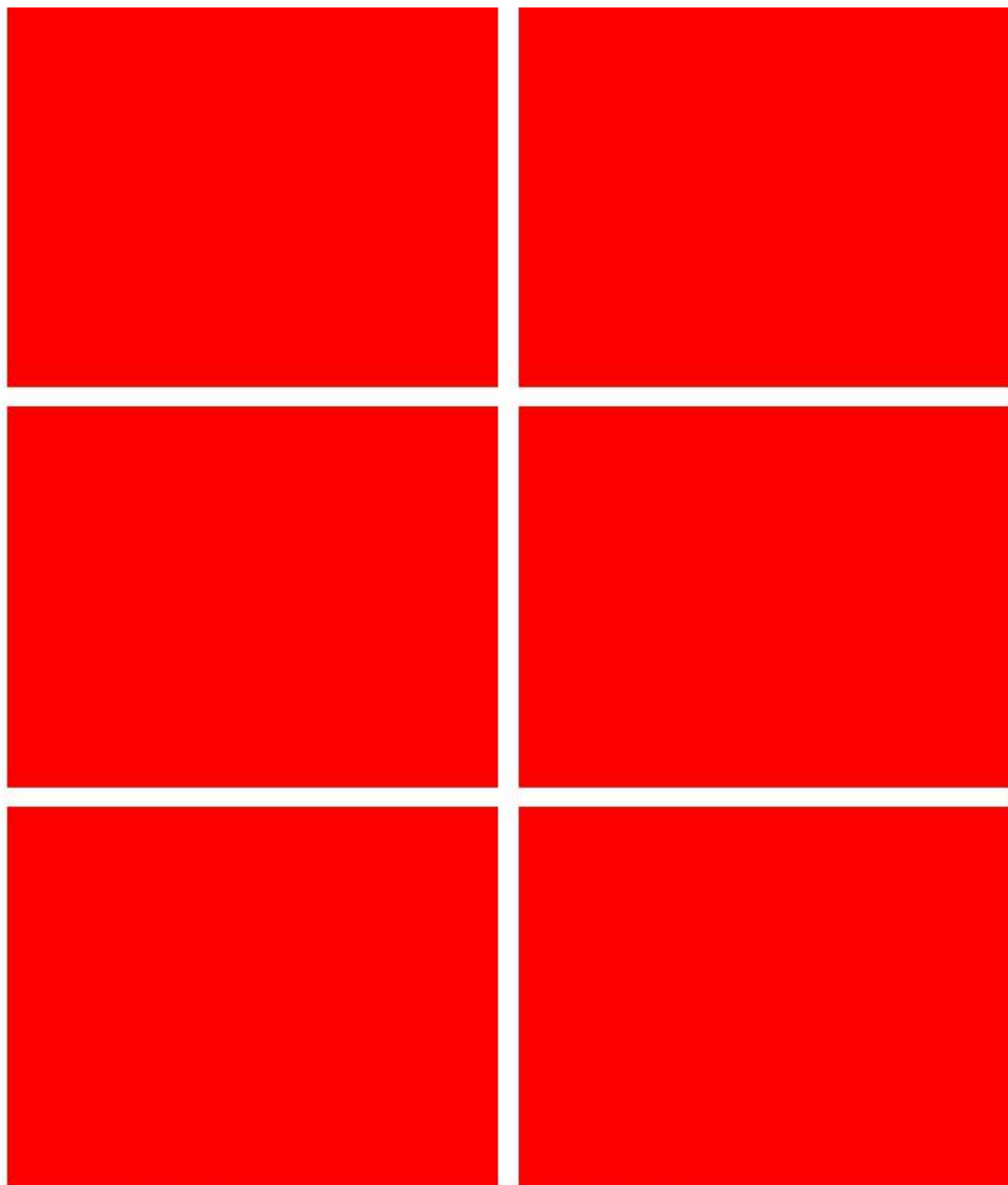
```
@media screen and (min-width:600px) and (max-width:900px){  
  
    body {background-color:#f5f5f5;}  
  
}
```

## 1.4 实例

PC 端



iPad 端



移动端





```
<div class="container">
```

```
  <div> </div>
```

```
  <div> </div>
```

```
  <div> </div>
```

```
  <div> </div>
```

```
  <div> </div>
```

```
  <div> </div>
```

```
</div>
```

```
.container{
```

```
  width: 85%;
```

```
  margin: 0 auto;
```

```
}
```

```
.container>div{
```

```
  height: 300px;
```

```
  background-color: red;
```

```
  margin-bottom: 15px;
```

```
  float: left;
```

```
}
```

```
@media screen and (min-width: 992px){
```

```
  .container>div{
```

```
    width: 32%;
```

```

        margin-right: 2%;

    }

    .container>div:nth-child(3n){

        margin-right: 0;

    }

}

@media screen and (min-width: 768px) and (max-width: 991px){

    .container>div{

        width: 49%;

    }

    .container>div:nth-child(2n){

        margin-left: 2%;

    }

}

@media screen and (max-width: 767px){

    .container>div{

        width: 100%;

    }

}

```

## 二、多列

### 2.1 定义

CSS3 可以将文本内容设计成像报纸一样的多列布局，如下实例：

菜鸟教程 - 学的不仅是技术，更是梦想！ 菜鸟教程(www.runoob.com)提供了最全的 编程技术基础教程，介绍了HTML、CSS、	Javascript、Python、Java、Ruby、C、P HP、MySQL等各种编程语言的基础知	识。同时本站中也提供了大量的在线实 例，通过实例，您可以更好的学习编程。
---	--	---

## 2.2 属性

### 2.2.1 创建多列

column-count 属性指定了需要分割的列数。

以下实例将 <div> 元素中的文本分为 3 列：

```
div {
  column-count: 3;
}
```

#### 2.2.2 列与列之间的间隙

column-gap 属性指定了列与列间的间隙。

以下实例指定了列与列间的间隙为 40 像素：

```
div {
  column-gap: 40px;
}
```

#### 2.2.1 列边框

( 1 ) column-rule-style 属性指定了列与列间的边框样式：

```
div {
  column-rule-style: solid;
}
```

( 2 ) column-rule-width 属性指定了两列的边框厚度：

```
div {
  column-rule-width: 1px;
}
```

( 3 ) column-rule-color 属性指定了两列的边框颜色：

```
div {
  column-rule-color: lightblue;
}
```

```
}
```

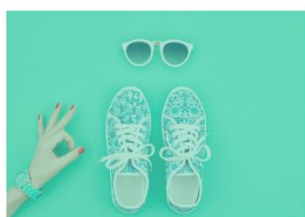
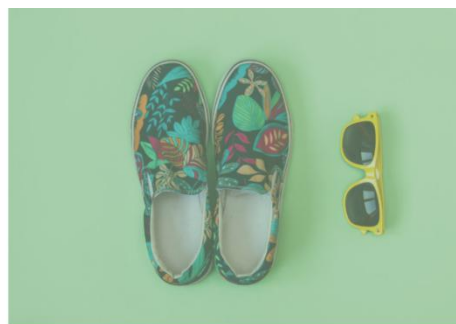
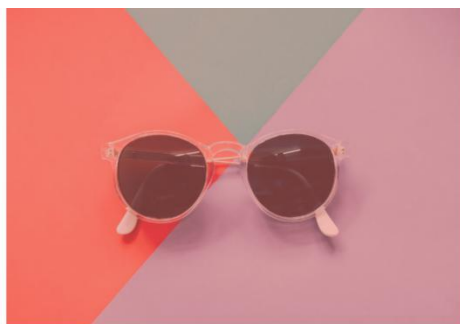
(4) column-rule 属性是 column-rule-\* 所有属性的简写。

以下实例设置了列直接的边框的厚度，样式及颜色：

```
div {
  column-rule: 1px solid lightblue;
}
```

### 三、作业

Pc 端



Windows 7 旗舰版 64位



移动端：

