**CSS3 @media 查询**

**实例**

如果文档宽度小于 300 像素则修改背景颜色(background-color):

@media screen and (max-width: 300px) {  
    body {  
        background-color:lightblue;  
    }  
}

**定义和使用**

使用 @media 查询，你可以针对不同的媒体类型定义不同的样式。

@media 可以针对不同的屏幕尺寸设置不同的样式，特别是如果你需要设置设计响应式的页面，@media 是非常有用的。

当你重置浏览器大小的过程中，页面也会根据浏览器的宽度和高度重新渲染页面。

**浏览器支持**

表格中的数字表示支持 @media 规则的第一个浏览器的版本号。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 规则 |  |  |  |  |  |
| @media | 21 | 9 | 3.5 | 4.0 | 9 |

**CSS 语法**

@media not|only mediatype and (mediafeature and|or|not mediafeature) {

CSS-Code;

}

not, and, 和 only 可用于联合构造复杂的媒体查询，您还可以通过用逗号分隔多个媒体查询，将它们组合为一个规则。

not, only 和 and 关键字含义:

* **not**: not 运算符用于否定媒体查询，如果不满足这个条件则返回 true，否则返回 false。 如果出现在以逗号分隔的查询列表中，它将仅否定应用了该查询的特定查询。 如果使用 not 运算符，则还必须指定媒体类型。
* **only**: only 运算符仅在整个查询匹配时才用于应用样式，并且对于防止较早的浏览器应用所选样式很有用。 当不使用 only 时，旧版本的浏览器会将 screen and (max-width: 500px) 简单地解释为 screen，忽略查询的其余部分，并将其样式应用于所有屏幕。 如果使用 only 运算符，则还必须指定媒体类型。
* **,** (逗号) 逗号用于将多个媒体查询合并为一个规则。 逗号分隔列表中的每个查询都与其他查询分开处理。 因此，如果列表中的任何查询为 true，则整个 media 语句均返回 true。 换句话说，列表的行为类似于逻辑或 or 运算符。
* **and**: and 操作符用于将多个媒体查询规则组合成单条媒体查询，当每个查询规则都为真时则该条媒体查询为真，它还用于将媒体功能与媒体类型结合在一起。

媒体类型（Media types）描述设备的一般类别。除非使用 not 或 only 逻辑操作符，媒体类型是可选的，并且会（隐式地）应用 all 类型。

你也可以针对不同的媒体使用不同样式文件 :

<!-- 宽度大于 900px 的屏幕使用该样式 -->

<link rel="stylesheet" media="screen and (min-width: 900px)" href="widescreen.css">

<!-- 宽度小于或等于 600px 的屏幕使用该样式 -->

<link rel="stylesheet" media="screen and (max-width: 600px)" href="smallscreen.css">

....

**媒体类型**

|  |  |
| --- | --- |
| **值** | **描述** |
| all | 用于所有设备 |
| aural | 已废弃。用于语音和声音合成器 |
| braille | 已废弃。 应用于盲文触摸式反馈设备 |
| embossed | 已废弃。 用于打印的盲人印刷设备 |
| handheld | 已废弃。 用于掌上设备或更小的装置，如PDA和小型电话 |
| print | 用于打印机和打印预览 |
| projection | 已废弃。 用于投影设备 |
| screen | 用于电脑屏幕，平板电脑，智能手机等。 |
| speech | 应用于屏幕阅读器等发声设备 |
| tty | 已废弃。 用于固定的字符网格，如电报、终端设备和对字符有限制的便携设备 |
| tv | 已废弃。 用于电视和网络电视 |

**媒体功能**

|  |  |
| --- | --- |
| **值** | **描述** |
| aspect-ratio | 定义输出设备中的页面可见区域宽度与高度的比率 |
| color | 定义输出设备每一组彩色原件的个数。如果不是彩色设备，则值等于0 |
| color-index | 定义在输出设备的彩色查询表中的条目数。如果没有使用彩色查询表，则值等于0 |
| device-aspect-ratio | 定义输出设备的屏幕可见宽度与高度的比率。 |
| device-height | 定义输出设备的屏幕可见高度。 |
| device-width | 定义输出设备的屏幕可见宽度。 |
| grid | 用来查询输出设备是否使用栅格或点阵。 |
| height | 定义输出设备中的页面可见区域高度。 |
| max-aspect-ratio | 定义输出设备的屏幕可见宽度与高度的最大比率。 |
| max-color | 定义输出设备每一组彩色原件的最大个数。 |
| max-color-index | 定义在输出设备的彩色查询表中的最大条目数。 |
| max-device-aspect-ratio | 定义输出设备的屏幕可见宽度与高度的最大比率。 |
| max-device-height | 定义输出设备的屏幕可见的最大高度。 |
| max-device-width | 定义输出设备的屏幕最大可见宽度。 |
| max-height | 定义输出设备中的页面最大可见区域高度。 |
| max-monochrome | 定义在一个单色框架缓冲区中每像素包含的最大单色原件个数。 |
| max-resolution | 定义设备的最大分辨率。 |
| max-width | 定义输出设备中的页面最大可见区域宽度。 |
| min-aspect-ratio | 定义输出设备中的页面可见区域宽度与高度的最小比率。 |
| min-color | 定义输出设备每一组彩色原件的最小个数。 |
| min-color-index | 定义在输出设备的彩色查询表中的最小条目数。 |
| min-device-aspect-ratio | 定义输出设备的屏幕可见宽度与高度的最小比率。 |
| min-device-width | 定义输出设备的屏幕最小可见宽度。 |
| min-device-height | 定义输出设备的屏幕的最小可见高度。 |
| min-height | 定义输出设备中的页面最小可见区域高度。 |
| min-monochrome | 定义在一个单色框架缓冲区中每像素包含的最小单色原件个数 |
| min-resolution | 定义设备的最小分辨率。 |
| min-width | 定义输出设备中的页面最小可见区域宽度。 |
| monochrome | 定义在一个单色框架缓冲区中每像素包含的单色原件个数。如果不是单色设备，则值等于0 |
| orientation | 定义输出设备中的页面可见区域高度是否大于或等于宽度。 |
| resolution | 定义设备的分辨率。如：96dpi, 300dpi, 118dpcm |
| scan | 定义电视类设备的扫描工序。 |
| width | 定义输出设备中的页面可见区域宽度。 |

**实例**

使用 @media 查询来制作响应式设计：

@media only screen and (max-width: 500px) {  
    .gridmenu {  
        width:100%;  
    }  
  
    .gridmain {  
        width:100%;  
    }  
  
    .gridright {  
        width:100%;  
    }  
}

## 注意：screen可以省略 实例

/\* 将 body 的背景色设置为棕褐色 \*/

body {

background-color: tan;

}

/\* 在小于或等于 992 像素的屏幕上，将背景色设置为蓝色 \*/

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

body {

background-color: blue;

}

}

/\* 在 600 像素或更小的屏幕上，将背景色设置为橄榄色 \*/

@media screen and (max-width: 600px) {

body {

background-color: olive;

}

}

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  body {  background-color: tan;  color: black;  }  /\* 在宽度为 992px 或更小的屏幕上，背景颜色为蓝色 \*/  @media screen and (max-width: 992px) {  body {  background-color: blue;  color: white;  }  }  /\* 在宽度为 600px 或更小的屏幕上，背景色为橄榄色 \*/  @media screen and (max-width: 600px) {  body {  background-color: olive;  color: white;  }  }  </style>  </head>  <body>  <h1>请调整浏览器窗口大小以查看效果！</h1>  <p>默认情况下，文档的背景色为棕褐色。如果屏幕尺寸为 992px 或更小，则颜色将变为蓝色。如果小于或等于 600px，它将变为橄榄色。</p>  </body>  </html> |

您想知道我们为什么要精确使用 992px 和 600px 吗？它们就是我们所称的设备的“典型断点”（typical breakpoints）。您可以在我们的 [响应式 Web 设计教程](https://www.w3school.com.cn/css/css_rwd_intro.asp) 中学习有关典型断点的更多知识。

**菜单的媒体查询**

在本例中，我们使用媒体查询来创建响应式导航菜单，该菜单在不同的屏幕尺寸上会有所不同。

**大型屏幕：**

* [Home](javascript:void(0))
* [News](javascript:void(0))
* [Contact](javascript:void(0))
* [About](javascript:void(0))

**小型屏幕：**

* [Home](javascript:void(0))
* [News](javascript:void(0))
* [Contact](javascript:void(0))
* [About](javascript:void(0))

**实例**

/\* navbar 容器 \*/

.topnav {

overflow: hidden;

background-color: #333;

}

/\* Navbar 链接 \*/

.topnav a {

float: left;

display: block;

color: white;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

/\* 在宽度为 600 像素或更小的屏幕上，使菜单链接彼此堆叠，而不是并排 \*/

@media screen and (max-width: 600px) {

.topnav a {

float: none;

width: 100%;

}

}

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>  \* {    box-sizing: border-box;  }  /\* 设置顶部导航栏样式 \*/  .topnav {    overflow: hidden;    background-color: #333;  }  /\* 设置 topnav 链接的样式 \*/  .topnav a {    float: left;    display: block;    color: #f2f2f2;    text-align: center;    padding: 14px 16px;    text-decoration: none;  }  /\* 悬停时改变颜色 \*/  .topnav a:hover {    background-color: #ddd;    color: black;  }  /\* 在宽度为 600 像素或更小的屏幕上，使菜单链接彼此堆叠，而不是并排 \*/  @media screen and (max-width: 600px) {    .topnav a {      float: none;      width: 100%;    }  }  </style>  </head>  <body>  <h1>响应式导航菜单</h1>  <p>请调整浏览器窗口的大小以查看效果：当屏幕小于 600 像素时，导航菜单将垂直显示，而不是水平显示。</p>  <div class="topnav">    <a href="#">Link</a>    <a href="#">Link</a>    <a href="#">Link</a>  </div>  </body>  </html> |

**列的媒体查询**

媒体查询的常见用法是创建弹性布局。在本例中，我们创建了一个布局，该布局在四列、两列和全宽列之间变化，具体取决于不同的屏幕尺寸：

**大型屏幕：**

**中等屏幕：**

**小型屏幕：**

**实例**

/\* 创建彼此相邻浮动的四个相等的列 \*/

.column {

float: left;

width: 25%;

}

/\* 在 992p x或更小的屏幕上，从四列变为两列 \*/

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

.column {

width: 50%;

}

}

/\* 在宽度小于或等于 600 像素的屏幕上，使各列堆叠，而不是并排 \*/

@media screen and (max-width: 600px) {

.column {

width: 100%;

}

}

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>  \* {    box-sizing: border-box;  }  /\* 创建彼此相邻浮动的四个等列 \*/  .column {    float: left;    width: 25%;    padding: 20px;  }  /\* 清除列后的浮动 \*/  .row:after {    content: "";    display: table;    clear: both;  }  /\* 在 992px 或更小的屏幕上，从四列变为两列 \*/  @media screen and (max-width: 992px) {    .column {      width: 50%;    }  }  /\* 在宽度小于或等于 600px 的屏幕上，使列堆叠在一起，而不是彼此相邻 \*/  @media screen and (max-width: 600px) {    .column {      width: 100%;    }  }  </style>  </head>  <body>  <h1>响应式四列布局</h1>  <p><b>请调整浏览器窗口的大小以查看响应效果。</b>在宽度小于或等于 992px 的屏幕上，列的大小将从四列调整为两列。在宽度为 600px 或更小的屏幕上，这些列将堆叠在一起，而不是彼此相邻。</p>  <div class="row">    <div class="column" style="background-color:#aaa;">      <h2>Column 1</h2>      <p>Some text..</p>    </div>      <div class="column" style="background-color:#bbb;">      <h2>Column 2</h2>      <p>Some text..</p>    </div>      <div class="column" style="background-color:#ccc;">      <h2>Column 3</h2>      <p>Some text..</p>    </div>      <div class="column" style="background-color:#ddd;">      <h2>Column 4</h2>      <p>Some text..</p>    </div>  </div>  </body>  </html> |

**提示：**更现代的创建列布局方法是使用 CSS Flexbox（请参见下面的例子）。但是，Internet Explorer 10 以及更早版本不支持它。如果需要 IE6-10 的支持，请使用浮动（如上所示）。

如需学习有关弹性框布局模块的更多知识，请学习 [CSS Flexbox](https://www.w3school.com.cn/css/css3_flexbox.asp) 这一章。

如需学习有关响应式 Web 设计的更多知识，请学习我们的 [响应式 Web 设计教程](https://www.w3school.com.cn/css/css_rwd_intro.asp)。

**实例**

/\* 弹性盒的容器 \*/

.row {

display: flex;

flex-wrap: wrap;

}

/\* 创建四个相等的列 \*/

.column {

flex: 25%;

padding: 20px;

}

/\* 在 992px 或更小的屏幕上，从四列变为两列 \*/

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

.column {

flex: 50%;

}

}

/\* 在宽度小于或等于 600 像素的屏幕上，使各列堆叠，而不是并排 \*/

@media screen and (max-width: 600px) {

.row {

flex-direction: column;

}

}

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>  \* {    box-sizing: border-box;  }  /\* 弹性框的容器 \*/  .row {    display: flex;    flex-wrap: wrap;  }  /\* 创建四个等列 \*/  .column {    flex: 25%;    padding: 20px;  }  /\* 在 992px 或更小的屏幕上，从四列变为两列 \*/  @media screen and (max-width: 992px) {    .column {      flex: 50%;    }  }  /\* 在宽度小于或等于 60px 的屏幕上，使列堆叠在一起，而不是彼此相邻 \*/  @media screen and (max-width: 600px) {    .row {      flex-direction: column;    }  }  </style>  </head>  <body>  <h1>响应式四列弹性布局</h1>  <p><b>请调整浏览器窗口的大小以查看响应效果。</b>在宽度小于或等于 992px 的屏幕上，列的大小将从四列调整为两列。在宽度为 600px 或更小的屏幕上，这些列将堆叠在一起，而不是彼此相邻。</p>  <div class="row">    <div class="column" style="background-color:#aaa;">      <h2>Column 1</h2>      <p>Some text..</p>    </div>      <div class="column" style="background-color:#bbb;">      <h2>Column 2</h2>      <p>Some text..</p>    </div>      <div class="column" style="background-color:#ccc;">      <h2>Column 3</h2>      <p>Some text..</p>    </div>      <div class="column" style="background-color:#ddd;">      <h2>Column 4</h2>      <p>Some text..</p>    </div>  </div>  </body>  </html> |

**用媒体查询隐藏元素**

媒体查询的另一种常见用法是在不同屏幕尺寸上隐藏元素：

在小屏幕上我会隐藏。

**实例**

/\* 如果屏幕尺寸为600像素或更小，请隐藏该元素 \*/

@media screen and (max-width: 600px) {

div.example {

display: none;

}

}

|  |
| --- |
| **<!DOCTYPE html>**  **<html>**  **<head>**  **<meta name="viewport" content="width=device-width, initial-scale=1">**  **<style>**  **div.example {**  **background-color: yellow;**  **padding: 20px;**  **}**  **@media screen and (max-width: 600px) {**  **div.example {**  **display: none;**  **}**  **}**  **</style>**  **</head>**  **<body>**  **<h1>隐藏不同屏幕尺寸的元素</h1>**  **<div class="example">Example DIV.</div>**  **<p>当浏览器的宽度为 600 像素或更小时，隐藏 div 元素。请调整浏览器窗口的大小以查看效果。</p>**  **</body>**  **</html>** |

**用媒体查询改变字体**

您还可以使用媒体查询来更改不同屏幕尺寸上的元素的字体大小：

可变的字体大小。

**实例**

/\* 如果屏幕尺寸超过 600 像素，把 <div> 的字体大小设置为 80 像素 \*/

@media screen and (min-width: 600px) {

div.example {

font-size: 80px;

}

}

/\* 如果屏幕大小为 600px 或更小，把 <div> 的字体大小设置为 30px \*/

@media screen and (max-width: 600px) {

div.example {

font-size: 30px;

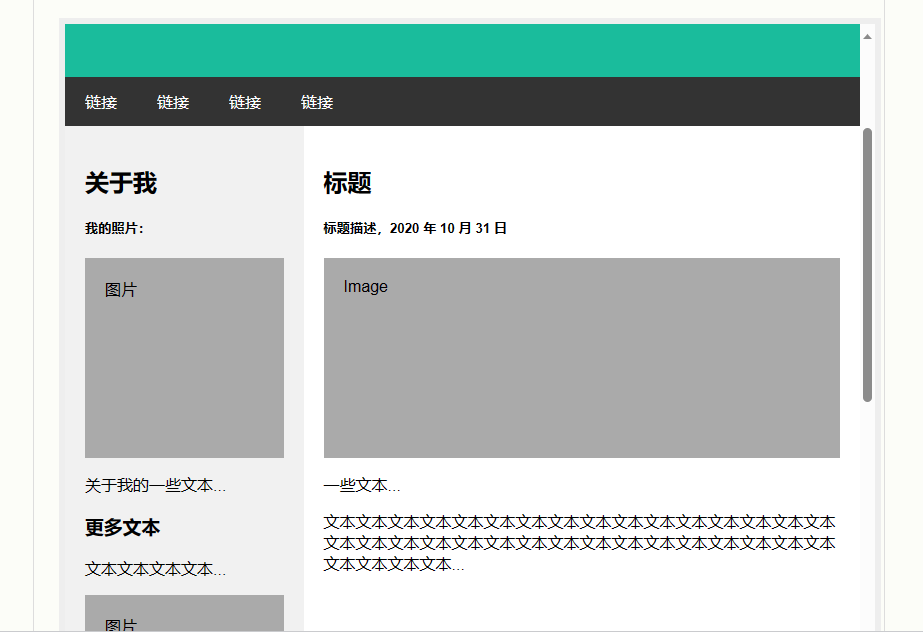
}

}

|  |
| --- |
| **<!DOCTYPE html>**  **<html>**  **<head>**  **<meta name="viewport" content="width=device-width, initial-scale=1">**  **<style>**  **div.example {**  **background-color: lightgrey;**  **padding: 20px;**  **}**  **@media screen and (min-width: 600px) {**  **div.example {**  **font-size: 80px;**  **}**  **}**  **@media screen and (max-width: 600px) {**  **div.example {**  **font-size: 30px;**  **}**  **}**  **</style>**  **</head>**  **<body>**  **<h1>在不同的屏幕尺寸上更改元素的字体大小</h1>**  **<div class="example">Example DIV.</div>**  **<p>当浏览器的宽度为 600 像素或更小时，将 DIV 的字体大小设置为 30px。当它是 601 像素或更宽时，将字体大小设置为 80 像素。请调整浏览器窗口的大小以查看效果。</p>**  **</body>**  **</html>** |

**弹性网站**

在本例中，我们将媒体查询与 flexbox 一起使用，以创建响应式网站，其中包含弹性导航栏和弹性内容。

****

**实例**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Page Title</title>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>  \* {    box-sizing: border-box;  }  /\* 设置 body 的样式 \*/  body {    font-family: Arial;    margin: 0;  }  /\* 页眉/LOGO \*/  .header {    padding: 60px;    text-align: center;    background: #1abc9c;    color: white;  }  /\* 设置顶部导航栏的样式 \*/  .navbar {    display: flex;    background-color: #333;  }  /\* 设置导航栏链接的样式 \*/  .navbar a {    color: white;    padding: 14px 20px;    text-decoration: none;    text-align: center;  }  /\* 悬停时改变颜色 \*/  .navbar a:hover {    background-color: #ddd;    color: black;  }  /\* 列容器 \*/  .row {    display: flex;    flex-wrap: wrap;  }  /\* 创建两个彼此相邻的不相等的列 \*/  /\* 侧栏/左侧列 \*/  .side {    flex: 30%;    background-color: #f1f1f1;    padding: 20px;  }  /\* 主列 \*/  .main {    flex: 70%;    background-color: white;    padding: 20px;  }  /\* 伪图像，仅用于此例 \*/  .fakeimg {    background-color: #aaa;    width: 100%;    padding: 20px;  }  /\* 页脚 \*/  .footer {    padding: 20px;    text-align: center;    background: #ddd;  }  /\* 响应式布局 - 当屏幕宽度小于 700 像素时，使两列相互堆叠，而不是彼此相邻 \*/  @media screen and (max-width: 700px) {    .row, .navbar {      flex-direction: column;    }  }  </style>  </head>  <body>  <!-- Note -->  <div style="background:yellow;padding:5px">    <h4 style="text-align:center">请调整浏览器窗口的大小以查看响应效果。</h4>  </div>  <!-- Header -->  <div class="header">    <h1>My Website</h1>    <p>With a <b>flexible</b> layout.</p>  </div>  <!-- Navigation Bar -->  <div class="navbar">    <a href="#">Link</a>    <a href="#">Link</a>    <a href="#">Link</a>    <a href="#">Link</a>  </div>  <!-- The flexible grid (content) -->  <div class="row">    <div class="side">      <h2>About Me</h2>      <h5>Photo of me:</h5>      <div class="fakeimg" style="height:200px;">Image</div>      <p>Some text about me in culpa qui officia deserunt mollit anim..</p>      <h3>More Text</h3>      <p>Lorem ipsum dolor sit ame.</p>      <div class="fakeimg" style="height:60px;">Image</div><br>      <div class="fakeimg" style="height:60px;">Image</div><br>      <div class="fakeimg" style="height:60px;">Image</div>    </div>    <div class="main">      <h2>TITLE HEADING</h2>      <h5>Title description, Dec 7, 2017</h5>      <div class="fakeimg" style="height:200px;">Image</div>      <p>Some text..</p>      <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>      <br>      <h2>TITLE HEADING</h2>      <h5>Title description, Sep 2, 2017</h5>      <div class="fakeimg" style="height:200px;">Image</div>      <p>Some text..</p>      <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>    </div>  </div>  <!-- Footer -->  <div class="footer">    <h2>Footer</h2>  </div>  </body>  </html> |

**方向：人像 / 风景**

媒体查询还可以用于根据浏览器的方向更改页面的布局。

您可以设置一组 CSS 属性，这些属性仅在浏览器窗口的宽度大于其高度时才适用，即所谓的横屏：

**实例**

如果方向处于横向模式，则使用浅蓝背景色：

@media only screen and (orientation: landscape) {

body {

background-color: lightblue;

}

}

|  |
| --- |
| <!DOCTYPE htm <!DOCTYPE html>  <html>  <head>  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <style>  body {    background-color: lightgreen;  }  @media only screen and (orientation: landscape) {    body {      background-color: lightblue;    }  }  </style>  </head>  <body>  <p>请调整浏览器窗口的大小。如果此文档的宽度大于高度，背景色为“浅蓝色”，否则为“浅绿色”。</p>  </body>  </html> |

**最小宽度到最大宽度**

您还可以使用 max-width 和 min-width 属性设置最小宽度和最大宽度。

例如，当浏览器的宽度在 600 到 900 像素之间时，更改 <div> 元素的外观：

**实例**

@media screen and (max-width: 900px) and (min-width: 600px) {

div.example {

font-size: 50px;

padding: 50px;

border: 8px solid black;

background: yellow;

}

}

|  |
| --- |
| <!DOCTYPE html> <!DOCTYPE html>  <html>  <head>  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>  @media screen and (max-width: 900px) and (min-width: 600px) {    div.example {      font-size: 50px;      padding: 50px;      border: 8px solid black;      background: yellow;    }  }  </style>  </head>  <body>  <h1>在不同的屏幕尺寸上更改 DIV 的外观</h1>  <div class="example">Example DIV.</div>  <p>当浏览器的宽度在 600 到 900 像素之间时，更改 DIV 的外观。<b>请调整浏览器窗口大小以查看效果。</b></p>  </body>  </html> |

使用附加值：在下面的例子中，我们使用逗号（类似 OR 运算符）将附加的媒体查询添加到已有媒体查询中：

**实例**

/\* 当宽度在 600 像素到 900 像素之间或大于 1100 像素时 - 更改 <div> 的外观 \*/

@media screen and (max-width: 900px) and (min-width: 600px), (min-width: 1100px) {

div.example {

font-size: 50px;

padding: 50px;

border: 8px solid black;

background: yellow;

}

}

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>  @media screen and (max-width: 900px) and (min-width: 600px), (min-width: 1100px) {    div.example{      font-size: 50px;      padding: 50px;      border: 8px solid black;      background: yellow;    }  }  </style>  </head>  <body>  <h1>在不同的屏幕尺寸上更改 DIV 的外观</h1>  <div class="example">Example DIV.</div>  <p>当浏览器的宽度介于 600 和 900 像素之间或大于 1100 像素时，请更改 DIV 的外观。<b>请调整浏览器窗口大小以查看效果。</b></p>  </body>  </html> |

# 例子：下面的设置只对屏幕有效

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>      .box{          width: 300px;          height: 400px;          border: 1px solid green;      }      /\* 媒体查询，下面的设置只对显示器有效 \*/  @media screen {        .box{            background-color: gold;            text-align: center;            line-height: 400px;        }  }  </style>  </head>  <body>    <div class="box">Example DIV.</div>  </body>  </html> |
|  |
|  |

# 如果我们把screen改为print可以发现效果反过来

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>      .box{          width: 300px;          height: 400px;          border: 1px solid green;      }      /\* 媒体查询，下面的设置只对显示器有效 \*/  @media print {        .box{            background-color: gold;            text-align: center;            line-height: 400px;        }  }  </style>  </head>  <body>    <div class="box">Example DIV.</div>  </body>  </html> |
|  |
|  |