**一、引入媒体查询**

在页面开发中，我们常用以下两种方法引入媒体查询：

**1. link引入**

<**link** rel="stylesheet" media="screen and (max-width: 720px)" href="small.css" />

可以用link方式将样式文件按断点或者是媒体类型分割。比如我们可以把专用于打印的样式单独放到一个css文件里面，如此一来只需要将上面代码做media=”print”的更改就可以在页面被打印的时候应用此样式文件里的样式。但是，值得注意的一点是：**link标签中的样式文件始终会被加载**。对，它不会根据当前media中的判断条件做动态加载，而是会无视这些条件而直接把css文件加载进来，然后在满足条件的情况下应用相应的已经加载好的样式。

**2. @media引入**

@**media** screen and (max-width: 720px) {

.someclass{

display: none;

}

}

上面的代码是直接写在某个css文件中的，当引入这个css文件的页面在满足@media后面的条件时，就会应用@media花括号中的样式。

此外我们还可以使用@import语句来引入，如下：

@import url(print.css) print;

**二、媒体特性**

虽然我们可以查询多个不同的媒体特性（你可以在[这里](http://www.w3.org/html/ig/zh/wiki/CSS3%E5%AA%92%E4%BD%93%E6%9F%A5%E8%AF%A2" \l ".E5.AA.92.E4.BD.93.E7.89.B9.E6.80.A7" \o "CSS3媒体查询" \t "_blank)查看到较为完整的媒体特性列表），但响应式网页设计最常使用的特性有以下几个：

* **min-width** : 当浏览器宽度大于查询中定义的值时适用的规则。
* **max-width** : 当浏览器宽度小于查询中定义的值时适用的规则。
* **min-height** : 当浏览器高度大于查询中定义的值时适用的规则。
* **max-height** : 当浏览器高度小于查询中定义的值时适用的规则。
* **orientation=portrait**:高度大于或等于宽度的浏览器适用的规则，即手机竖屏时适用的规则。
* **orientation=landscape** : 宽度大于高度的浏览器适用的规则，即手机横屏时适用的规则。

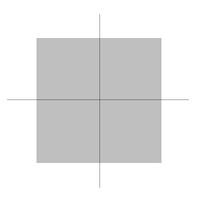
此外，我们还要说下device-pixel-radio这个媒体特性，它常被用来处理高清屏幕导致的图像模糊的问题。从字面意思可以理解出它指的是“设备像素比例”，这个比例就是指一个CSS像素在渲染到物理设备上显示的时候会乘以的那个比例系数，典型的比例系数有1x、2x和3x。就拿iphone6来说：

Screen Width = 375px /\*CSS 中的像素值\*/

Screen Height = 667px /\*CSS 中的像素值\*/

Screen Width = 750px /\*实际设备的像素值\*/

Screen Height = 1334px /\*实际设备的像素值\*/

因此我们不难得出device-pixel-ratio为2，也就是说在这样的retina屏幕上**1**个CSS像素是用**4**个物理像素来渲染的。就如同一个CSS像素被放大了4倍（每边乘以系数device-pixel-radio:2）来显示：  


## 三、逻辑操作符

操作符 not，and 和 only等可以用来构建复杂的媒体查询。

**1、and** 操作符用来把多个媒体属性合并到同一条媒体查询中。只有当每个属性都为真时，这条查询的结果才为真，相当于“且”。

@media only screen

**and** (min-width : 320px)

**and** (max-width : 480px){ /\* 样式代码 \*/}

and操作符不仅可以组合媒体特性也可以组合媒体类型，如上面代码中的only screen。

**2、only** 操作符表示仅在媒体查询匹配成功的情况下应用指定样式，但必须为其指定媒体类型。

<**link** rel="stylesheet" media="only screen and (color)" href="example.css" />

最常用的可能就是对screen或print媒体类型的检测了。此外，我们可以用上面的语句作为一种hack的方式防止样式应用给不支持媒体特性查询的古董浏览器。

**3、逗号(,)**操作符可以将多个媒体查询以逗号分隔放在一起；只要其中任何一个为真，整个媒体语句就返回真。相当于 “或”。

@media only screen

**and** (-webkit-min-device-pixel-ratio: 2),

only screen **and** (min--moz-device-pixel-ratio: 2),

only screen **and** (-o-min-device-pixel-ratio: 2/1),

only screen **and** (min-device-pixel-ratio: 2) {/\* 样式代码 \*/}

其实我们在上面检测高清屏的时候就已经用过逗号操作符了，这条查询是为了兼容不同的浏览器的。

**4、not** 操作符用来对一条媒体查询的结果进行取反，相当于“非”，但必须为其指定媒体类型。

@media **not** **print** **and** (color){/\* 样式代码 \*/}

用来排除某种特定的媒体类型，注意上面的查询not是作用于整个查询语句的，也就是说上面的代码等价于：

@media **not** (**print** **and** (color)){/\* 样式代码 \*/}