# 

[自适应网页设计（Responsive Web Design）](#_n7lscxhame5w)

[一、"自适应网页设计"的概念](#_4f0clcnf67ov)

[二、允许网页宽度自动调整](#_sut7lulo1421)

[三、不使用绝对宽度 xx%](#_316er98pfjv4)

[四、相对大小的字体 em/rem](#_acyhf532ff1c)

[五、流动布局（fluid grid）](#_c0cm141v50bo)

[六、选择加载CSS](#_u116zo1k24a)

[七、CSS的@media规则](#_clwwdux8vrx2)

[八、图片的自适应（fluid image）](#_z91xmmmaamet)

[响应式图像教程](#_qlmxt7ji9nih)

[一、问题的由来](#_p2km90csdetc)

[二、像素密度的选择：srcset属性](#_ktdbb51kl4bt)

[三、图像大小的选择：srcset属性 + sizes属性](#_qpvr2y2z3d3f)

[三、图像大小的选择：srcset属性 + sizes属性](#_1ren2ccwix20)

[四、<picture>标签，<source>标签](#_s5xniu5i8ck5)

[五、<source>标签的type属性](#_m9ms4v6neal)

# 自适应网页设计（Responsive Web Design）

## 一、"自适应网页设计"的概念

## 二、允许网页宽度自动调整

在网页代码的头部，加入一行[viewport元标签](https://developer.mozilla.org/en/mobile/viewport_meta_tag)。

　　<meta name="viewport" content="width=device-width, initial-scale=1" />

网页宽度默认等于屏幕宽度（width=device-width），原始缩放比例（initial-scale=1）为1.0，即网页初始大小占屏幕面积的100%。

所有主流浏览器都支持这个设置，包括IE9。对于那些老式浏览器（主要是IE6、7、8），需要使用[css3-mediaqueries.js](http://code.google.com/p/css3-mediaqueries-js/)。

　　<!--[if lt IE 9]>

　<scriptsrc="http://css3-mediaqueries-js.googlecode.com/svn/trunk/css3-mediaqueries.js"></script>

　　<![endif]-->

## 三、不使用绝对宽度 xx%

只能指定百分比宽度：　**width: xx%;**

或者　**width:auto;**

## 四、相对大小的字体 em/rem

字体也不能使用绝对大小（px），而只能使用相对大小（em）。

　　body {

　　　　font: normal 100% Helvetica, Arial, sans-serif;

　　}

上面的代码指定，字体大小是页面默认大小的100%，即16像素。

　　h1 {

　　　　font-size: 1.5em;

　　}

## 五、流动布局（fluid grid）

[**"流动布局"**](http://www.alistapart.com/articles/fluidgrids/)**的含义是，各个区块的位置都是浮动的，不是固定不变的。**

　　.main {

　　　　float: right;

　　　　width: 70%; }

　　.leftBar {

　　　　float: left;

　　　　width: 25%;}

[float](http://designshack.net/articles/css/everything-you-never-knew-about-css-floats/)的好处是，如果宽度太小，放不下两个元素，后面的元素会自动滚动到前面元素的下方，不会在水平方向overflow（溢出），避免了水平滚动条的出现。

另外，绝对定位（position: absolute）的使用，也要非常小心。

## 六、选择加载CSS

"自适应网页设计"的核心，就是CSS3引入的[Media Query](http://www.w3.org/TR/CSS21/media.html)模块。

它的意思就是，自动探测屏幕宽度，然后加载相应的CSS文件。

　　<link rel="stylesheet" type="text/css"

　　　　media="screen and (max-device-width: 400px)"

　　　　href="tinyScreen.css" />

上面的代码意思是，如果屏幕宽度小于400像素（max-device-width: 400px），就加载tinyScreen.css文件。

　　<link rel="stylesheet" type="text/css"

　　　　media="screen and (min-width: 400px) and (max-device-width: 600px)"

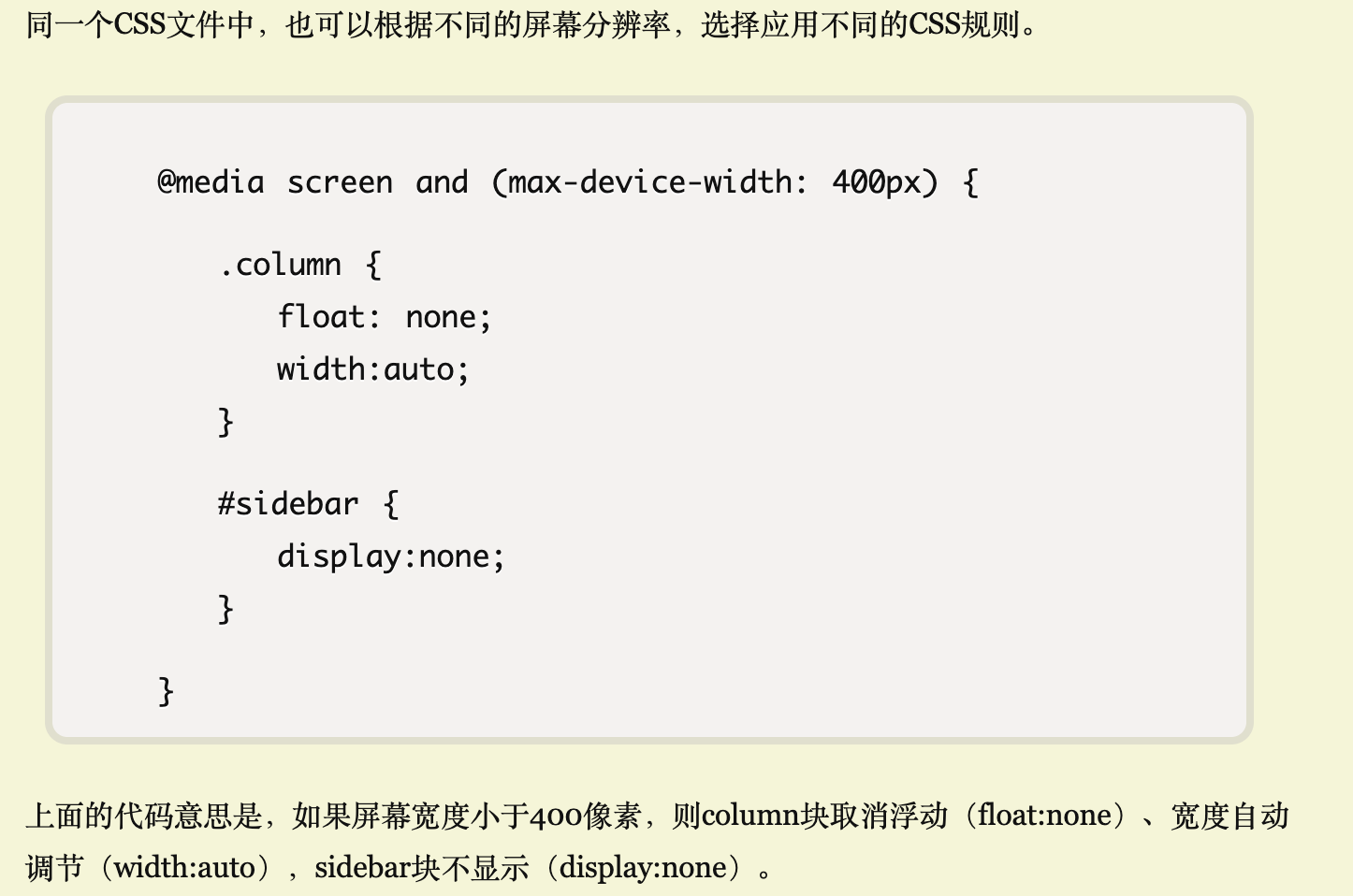
　　　　href="smallScreen.css" />

如果屏幕宽度在400像素到600像素之间，则加载smallScreen.css文件。

除了用html标签加载CSS文件，还可以在现有CSS文件中加载。

　　@import url("tinyScreen.css") screen and (max-device-width: 400px);

## 七、CSS的@media规则



## 八、图片的自适应（fluid image）

除了布局和文本，"自适应网页设计"还必须实现图片的自动缩放。



# 响应式图像教程

响应式图像的解决方案有很多，JavaScript 和 CSS 都可以实现。本文介绍最简单的、语义性最好的 HTML 方法，浏览器原生支持。

## 一、问题的由来

三大弊端。

（1）体积

一般来说，桌面端显示的是大尺寸的图像，文件体积较大。手机的屏幕较小，只需要小尺寸的图像，可以节省带宽，加速网页渲染。

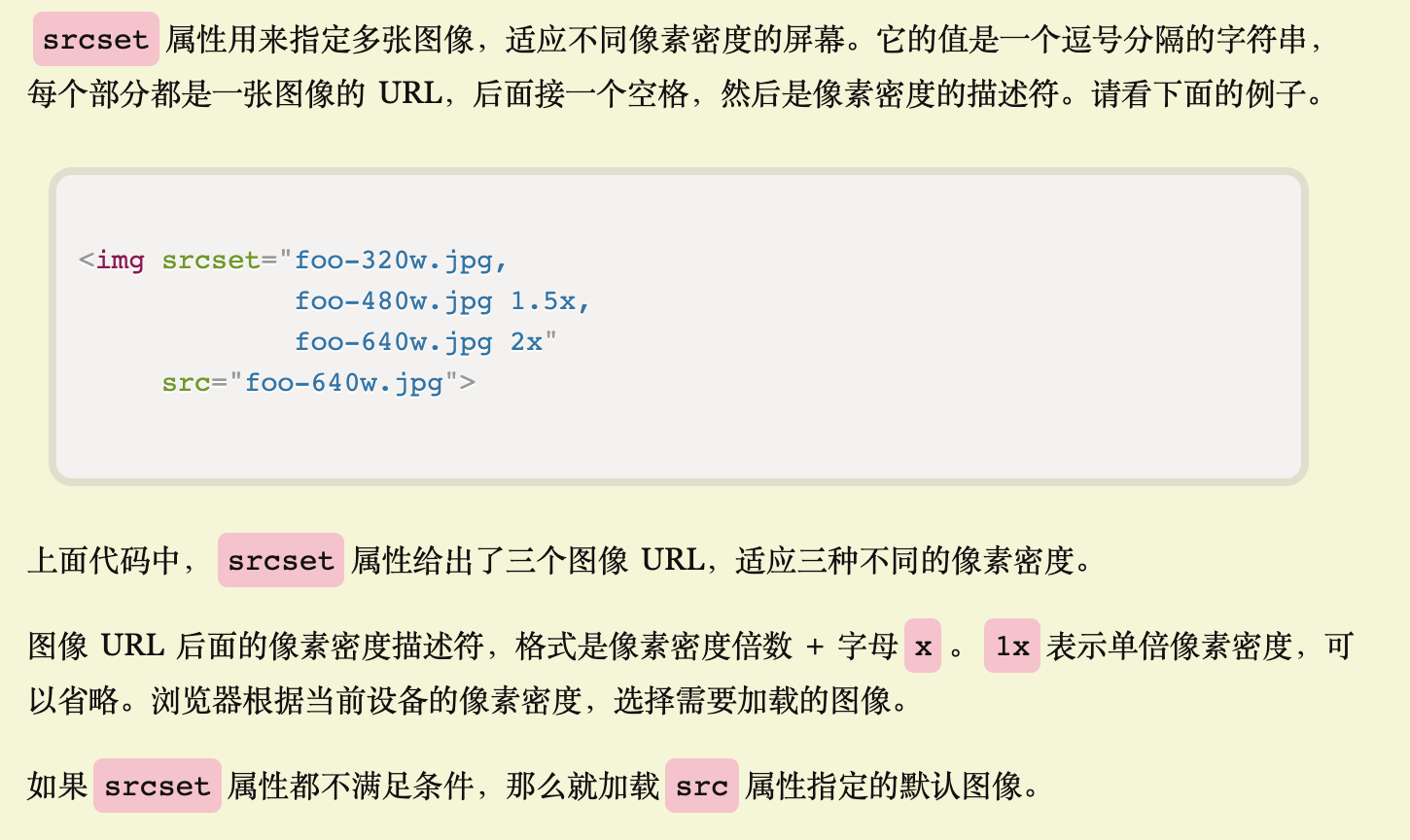
（2）像素密度

桌面显示器一般是单倍像素密度，而手机的显示屏往往是多倍像素密度，即多个像素合成为一个像素，称为 Retina 屏幕。图像文件很可能在桌面端很清晰，放到手机上会有点模糊，因为像素扩充了。

（3）视觉风格

桌面显示器的面积较大，图像可以容纳更多细节。手机的屏幕较小，许多细节是看不清的，需要突出重点。

## 二、像素密度的选择：srcset属性

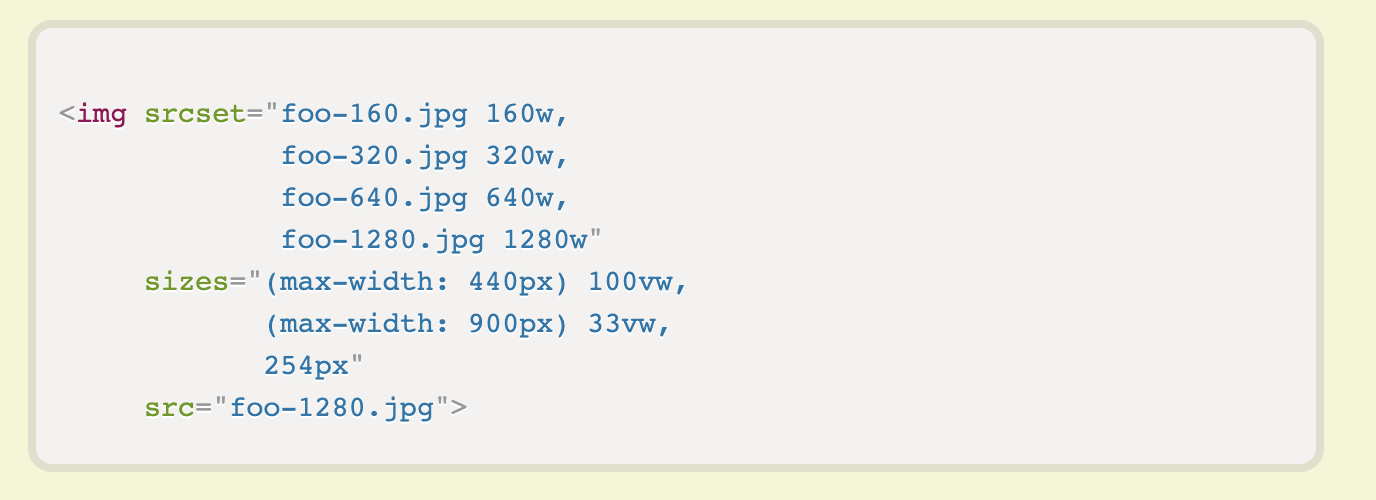


## 

## 三、图像大小的选择：srcset属性 + sizes属性

## 

## 三、图像大小的选择：srcset属性 + sizes属性



上面代码中，sizes属性给出了三种屏幕条件，以及对应的图像显示宽度。宽度**不超过440**像素的设备，图像显示**宽度为100%**；宽度441像素到900像素的设备，图像显示**宽度为33%**；宽度900像素以上的设备，图像显示宽度为254px。

## 

## 

## 四、<picture>标签，<source>标签

上面两节分别解决了像素密度和屏幕大小的适配，但是如果要同时适配不同像素密度、不同大小的屏幕，应该怎么办呢？





## 五、<source>标签的type属性

