# 响应式网页设计

网络的起步快到超乎所有人的想象，而且仍在疯狂的成长。现在，经过了几年，移动端以爆炸式的发展登上历史舞台。移动互联网使用的增长和通常意义上互联网使用的增长有着很大的区别。

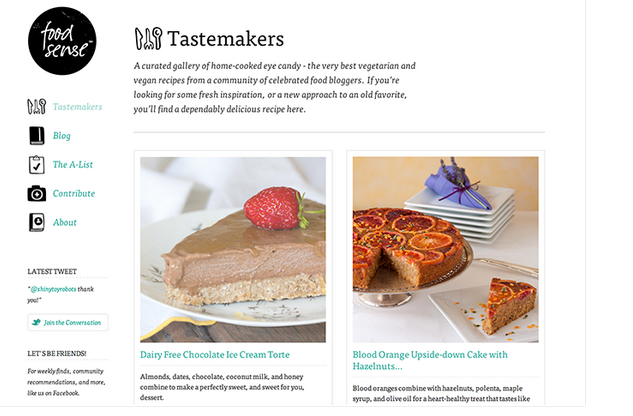
现如今很难找到谁拥有一个或多个手机设备却没有联网的。在英国，移动电话数量比人口还多，并且这一趋势使得近年移动网络的使用量多过笔记本网络使用量。

电话网络使用的增长给出了怎么做出能服务于所有用户的网站。产业回应给出的答案是响应式网页设计，也就是所谓的RWD(Responsive Web Design)

## --响应式一览—

响应式网页设计是建立能够在所有设备和任何尺寸屏幕上都适用的网站，不管屏幕大小或是手机端还是台式机。响应式网页设计专注为每个人提供一种直观舒适的体验。台式电脑用户和手机用户均可从响应式网页获益。

响应式网站设计这个术语本身是杜撰的，但之后经过了Ethan Marcotte极大的研究和发展。这课里的绝大多数内容都是Ethan在网上和它的书《[*Responsive Web Design*](http://www.abookapart.com/products/responsive-web-design/)》里最早提到的，这书很值得一看。



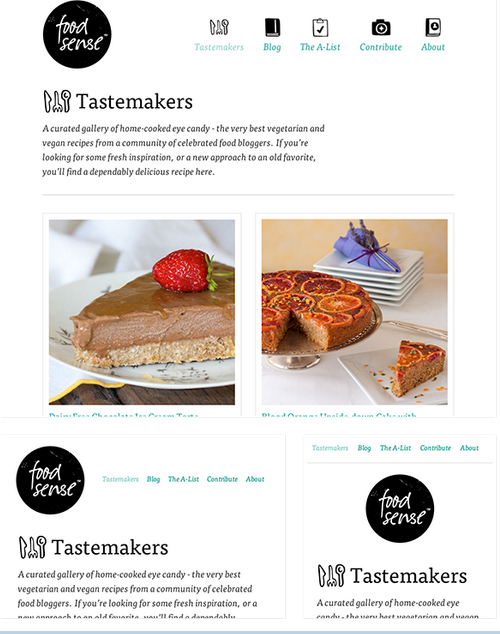


Fig. 4.01

[Food Sense](http://foodsense.is/) 有漂亮的网站外观，可以响应不同的视图尺寸。不管视图窗口过大或过小food sense网站都能调整并给用户创造出很自然的视觉体验。

### 响应 VS. 适配 VS. 移动端

“响应“的术语可能不是新的，相比可能术语“适配”或“移动端”更为人熟知。可能会让你想知道这三者到底有什么区别。

响应式和适配网页设计有很近的关系，很多时候把它们看做是一样的。响应式通常指的是对任何的改变做出快速和准确的反映，而适配是指根据新的目的或情景例如改变可以很容易就实现。基于响应式设计的网站可以在不同的因素作用下持续的并流动性的做出改变，诸如窗口视图宽度；而适配网站是基于当前知道的一些原因而构建的。把两者结合是最理想的，可以为功能性网站提供最完美的模板。具体使用哪个术语没有很大区别。

另一方面，移动端(手机)，通常意义上是在一个新的专业领域单独为手机用户构建的专门的网站。这个有时候也会用到，但却不是一个好的想法。手机网站是可以很轻但存在新的代码库和浏览器解析的依赖关系，这些都会给开发者和用户带来阻碍。

现在最流行的技术就是响应式网页设计，并侧重于能根据不同的浏览器和设备窗口视图动态响应的设计，并据此可以响应的改变布局和内容。这个方法把三个好处都带来了：响应式，适配和移动端(手机).

## --弹性布局—

响应式网页设计分为三个主要部分:**弹性布局**，**媒介查询**和**弹性媒体**。第一部分，弹性布局，是使用弹性网格(flexible grid)构建网站布局的操作，可以在宽度上动态的任意改变尺寸。弹性网格使用相对宽度的单位构建，大多数是百分比或em单位的单元。这些相对长度之后使用width, margin或者padding来定义网格属性值。

#### 🡪相对窗口长度

CSS3引进了一些新的相对长度单位，特别是针对浏览器窗口或者设备尺寸的。这些新的单位包括vw, vh, vmin和 vmax。这些新单位的全面支持还不是很好，但还在改进。在构建响应式网站时它们起到很大的作用。

Vw: 窗口视图宽度 vh: 窗口视图高度

Vmin: 窗口视图的最低宽度和高度

Vmax: 窗口视图的最高宽度和高度

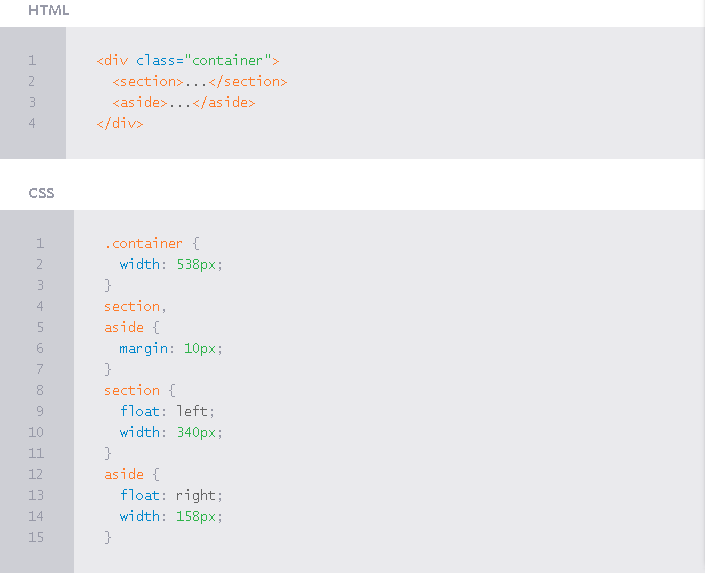
弹性布局不主张使用固定测量单位，诸如像素和英寸。因为不同的设备窗口视图的宽高是一直在变的。网站布局需要适应这些变化而定值会带来很多局限。幸运的是，Ethan指出使用相对值这一公式可以轻松定义出弹性布局的比例。

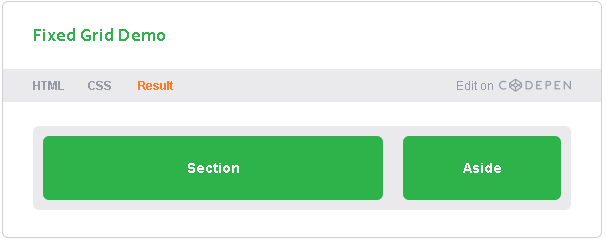
该公式指出把目标元素的宽度用其父元素宽度分隔，得出的结果就是目标元素的相对宽度。

target ÷ context = result

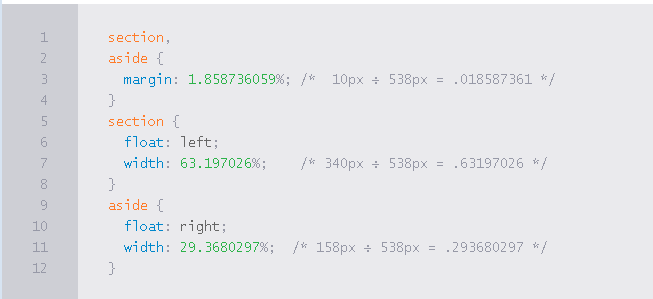
### 弹性网格

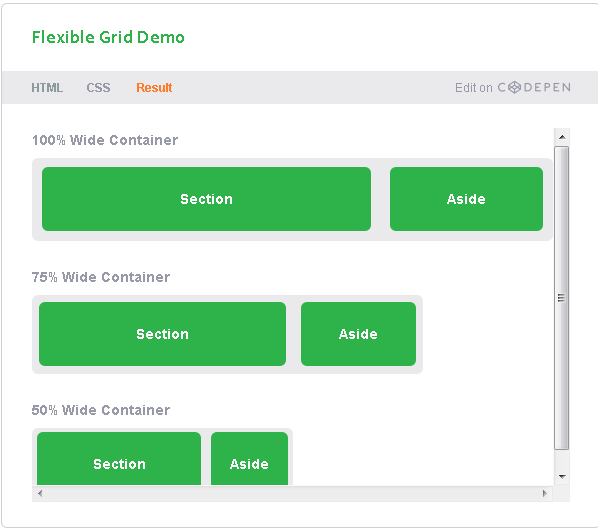
让我们看看这个公式在两列布局中是怎么起作用的。下面我们有一个类名为container的父级div并且包含着section和aside元素。目的是让section在左边而aside在右边，并且两边有相同的margin值。通常对于这种布局所用的标记和样式看起来有点像下面这样的：





使用弹性网格公式我们可以把这些固定单位长度变成相对单位的。在这个例子里我们使用百分比当然em单位也是可以的。注意，不管父元素container有多宽，section和aside的margin值以及宽度都会成比例变化。





使用弹性布局概念和公式，把它应用到网格每一个部分可以做出完完全全的动态网站，可以据比例响应任何窗口尺寸。想使用弹性布局做更多的控制，你可以使用min-width, max-width, min-height和max-height属性。

仅仅使用弹性布局是不够的。有时候浏览器窗口太小以至于即便布局根据尺寸变化做出来相应的列也不能很好的显示里面的内容。特别是当布局变得太小或者太大时，里面的文字会变得模糊不清以至于布局遭到破坏。这种情况下，媒介查询可以用来帮助建立更好的体验。

## --媒介查询--

媒介查询用来作为当媒体类型被当做目标和包含在样式中查询的一个扩展。媒介查询给不同的浏览器和设备环境提供识别不同样式的能力，诸如试图窗口宽度或设备的方位。能够针对性的使用样式给响应式页面设计世界的应用带来了更多可能性。

### 初始化媒介查询

目前有很多使用媒介查询的方法：在已存在的样式列表里使用@media规则；使用@import规则导入新的样式列表；或者在HTML页面内部链接一个外部的样式。通常建议是在已有的样式列表中使用@media规则以减少多余的http请求。



每个媒介查询可能会使用带有一个或多个关于媒体类型的表达式。常用的媒体类型包括all, screen, print, tv 和 braille。在HTML5规则中包含了新的媒体类型，甚至包含了3d-glasses. 如果媒体类型没有使用媒介查询指定默认媒体类型是screen.

带有媒体类型的媒介查询表达式可能会带有不同的媒介属性和值，之后用来指定是true或false。如果一个媒介属性值被指定为true，则会应用上相应的样式。如果一个媒介属性值被指定为false则与之对应的样式会被忽略。

### 媒介查询里的逻辑运算符

媒介查询里的逻辑运算符有很大的作用。在使用媒介查询时有三种不同的逻辑运算符，其中包括and, not 和 only.

在媒介查询中使用and逻辑运算符允许添加额外的条件，用以确保浏览器或者设备可以同时做事件a, b, c或者其他。多个相互独立的媒介查询要用逗号分隔，相当于无意的或者运算符。下面的例子是选中了所有宽度在800到1024像素里的媒体类型。



逻辑运算符not用于否定查询，特别是查找在所有中唯一的一个。在下面的例子中表达式指向的是所有没有彩色屏幕的设备。也就是说黑白或者单色的屏幕在这个例子中会被应用上样式。



逻辑运算符only是一个新的运算符并且使用HTML4规则的用户代理不能识别该运算符，那些不支持该媒介查询的浏览器或设备就不能看到对应的样式了。下面的表达式只会选择有portrait orientation屏幕的用户代理渲染媒介查询。



#### 🡪省略媒体类型

如果使用not和only逻辑运算符时媒体类型是空的。这种情况下默认是all。

### 媒介查询中媒体的特性

媒介查询中知道媒介查询的语法和知道逻辑运算符是怎么工作的是很重要的入门但最主要的是使用媒体属性工作。媒体属性定义了在媒介查询中该使用什么样的属性和特性。

### 媒介特性height和width

Height和width是用来设置浏览器或设备窗口高度和宽度最常使用的媒体属性。可以使用height, width, device-height和 device-width 媒体属性值定义高和宽。这些媒体属性值也可以加上min 或max 作为前缀修饰符，用来创建像min-width或者 max-device-width的属性。

Height和width属性在window浏览器中是以视图渲染窗口的height和width为基准的。而device-height和device-width属性是基于设备外边的宽度和高度的，因此可能看起来比实际渲染的区域要大。这些媒体属性的长宽单位可以是任何长度单位，相对的绝对的都行。



响应式设计最常用的属性包括min-width和max-width。可以用来在台式或移动设备上创建响应式网站而不用考虑设备间的差异。

#### 🡪使用max和min做前缀

min和max前缀可以用在很多媒体属性中。其中min前缀指的是大于或等于该值而max前缀表明的是小于或等于该值。在使用min和max前缀时要避免和HTML产生语法冲突，特别是不要使用>和<号。

### 媒体属性orientation

定向这一媒介属性决定了设备是用于风景方向的还是人像方向的。当显示的宽长于高时会触发风景模式而当显示的高宽于长时会触发人像模式。这一媒介属性在手机设备里起到很大作用。



### 媒介属性aspect ratio

属性aspect-ratio和device-aspect-ratio是给指定渲染区域或者输出设备指定宽高像素比例的。其中min和max前缀可用于不同的因素比例属性，用于给指定的内容定义或高或低的比例。

因素比例值由两个正值通过正斜杠分隔组成。第一个整值标识像素的宽第二个值标识像素的宽。



#### 🡪媒介属性的pixel ratio

除了因素比例这一媒介属性还存在pixel-ratio这一媒介属性。这些属性包括可以使用min和max 前缀的device-pixel-ratio属性。特别地，像素比例属性可以给出更好的设备定义，包括retina(视网膜)显示。使用该方式进行媒介查询看起来是这样的：

@media only screen and (-webkit-min-device-pixel-ratio: 1.3), only screen and (min-device-pixel-ratio: 1.3) {...}

### 媒介属性 resolution

媒介属性resolution指的是外部设备在像素级别上的分辨率，也就是每英寸的点数或DPI。这一媒介属性也接受min和max做前缀。另外，resolution属性也有每像素的点数(1.3dppx),每厘米的点数(118dpcm),还有一些基于其他单位的像素值。



### 其他一些媒介属性

其他的媒介属性包括定义输出颜色的color, color-index和monochrome，还有定义位图设备用的grid属性和扫描电视机进程的scan属性。这些属性不常用但在需要的时候同样能起到作用。

#### 🡪浏览器支持的媒介查询

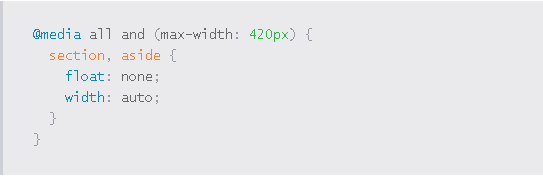
很不幸的是媒介查询不支持IE8以下的，和一些其他的老式浏览器。但是，也有一些针对这些情况用JavaScript编写的polyfills 。

[Respond.js](https://github.com/scottjehl/Respond/)是只用于查找最大/最小宽媒体类型的轻量级polyfill，对那些只使用一种媒介查询类型的来说是最好的。

[CSS3-MediaQueries.js](http://code.google.com/p/css3-mediaqueries-js/)更高级也更大些，是用于支持更大数组更复杂媒体查询的polyfill. 另外，任何polyfill都是有性能隐患并潜在拉低网站速度。使用的时候要确保使用的polyfill得到的效果值得性能的付出。

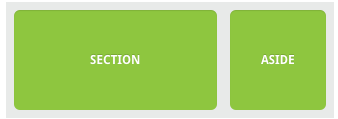
### 媒介查询实例

使用媒介查询现在我们重写之前创建的弹性布局。现在例子里的一个问题是当视图窗口过小时aside模块的宽度变得太短相当于没用。当视图窗口宽度低于420像素时我们可以关掉section和aside的float变化布局来改变这个宽度。



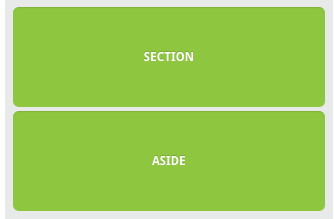
**Fig. 4.02**

没使用媒介查询时section和aside宽度变得很小。小到不能包含什么实质性的内容。



**Fig. 4** .03

使用媒介查询移除float并改变它们的宽度，现在section和aside可以充满整个视图窗口，可以给已存在的文本内容提供足够的空间。



#### 🡪定义断点

你在使用媒介查询时可能根据直觉根据不同的设备分辨率定义出不同的视图窗口断点，诸如320px, 480px, 768px, 1024px,1224px 或者其它。这样的做法是很不好的。

当你创建响应式网站是它必须能响应一组不同的视图窗口尺寸，在不考虑设备的情况下。只有当网站开始乱了，看起来奇怪或者影响到用户体验了才考虑使用断点。

另外，新的设备和分辨率一直都在更新。如果想要适应所有这些尺寸那将是永无止境的。

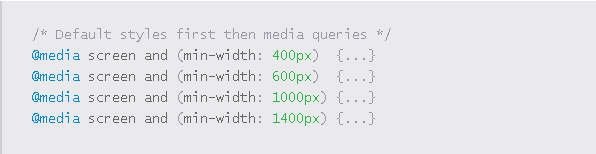
## --移动为要 [手机为先]—

关于使用媒介查询一项很流行的技术是手机为先(mobile first). [mobile first](http://www.lukew.com/presos/preso.asp?26)方法是指在给网站使用样式时默认样式设置成给较小尺寸的视图窗口的，然后使用媒介查询给变大的视图窗口添加样式。

Mobile为先的操作理念是给移动设备上的用户，通常使用的是较小尺寸视图窗口，不应该是先为桌面电脑的用户编写样式然后再后期为mobile重写样式。这样做会浪费带宽。对于要浏览时髦网站的用户来说带宽简直太重要啦。

Mobile为先同样主张在以移动用户为主的情况下做设计。不久之前，网络大部分消耗量是在移动设备上。相应的为之做出计划并开发内在的移动体验。

Mobile为先的媒介查询最初的突破表现如下：

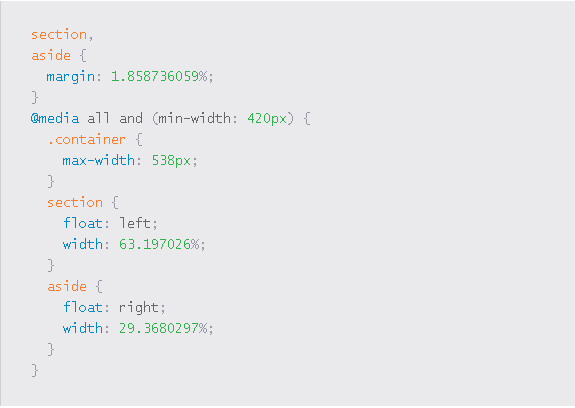


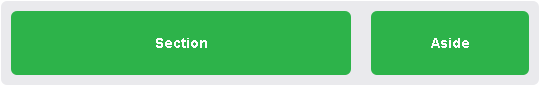
另外，使用媒体查询可以阻止下载不必要的媒体设备。通常说来，建议在mobile样式中避免使用CSS3的阴影，渐变，透明和动画。当过度使用这些时，将会导致加载变慢甚至会剪断手机使用寿命。



### 移动为先实例

给我们先前的例子加上媒介查询，我们重写一些样式以在420像素以下的窗口视图有更好的展示。下面我们使用移动样式为先重写代码然后再使用媒介查询给超过420像素的窗口添加样式。





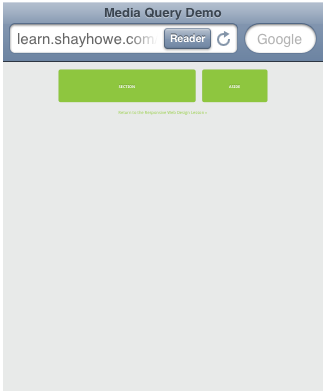
要注意，现在和之前的代码量一样。唯一的区别是移动设备只要渲染加载一个CSS声明就可以。除非是在不更改任何原样式时在更大的视图窗口加载，不然其他的样式都会被阻止。

## --Viewport--

近来移动设备显示网站方面大体做的不错。即便有时会使用一些辅助，特别是定义网站的窗口尺寸，单位或者分辨率。为了补救这些，Apple发明了viewport这个动态标签[meta 标签]

**Fig. 4**.04

即便该例子使用了媒介查询，很多移动设备仍然不知道这个网站最初的宽度或者规模是多少。所以，它们可能不会中断媒介查询。[意思在特定尺寸下不能根据设备用媒介查询得到良好的展示]



### Viewport 的 宽和高

在viewport的meta标签里使用height或者width可以分别给窗口设置宽高。每个值既可以是正整值也可以是关键字。比如height属性值关键字device-height是可以的，而对于width属性关键字device-width也是可以的。使用这些关键字会继承设备默认的宽高值。

最好的情况要得到最好的网页展示，我们建议是应用上设备默认的device-height和device-width值。



**Fig. 4** .05

使用device-width可以让设备知道网站该有的宽度，让网站有合适的尺寸并且可应用上任何有效的媒介查询。



### Viewport 规模

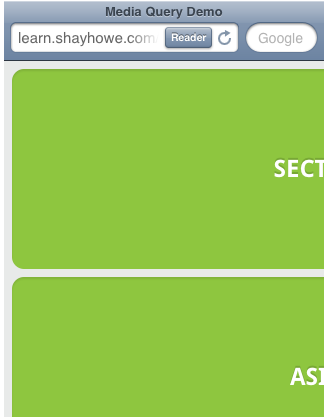
可以使用minimum-scale, maximum-scale, initial-scale 和 user-scalable属性控制网站在移动设备上的规模以及用户可以怎么继续改变网站规模。

网站的initial-scale应该设置为1因为这是一个基于设备高度，图像定位和窗口尺寸的比例。如果该设备是风景模式的这个比例应该是基于设备宽度和窗口尺寸的。Initial-scale的值应该是0到10之间的正整数值。



**Fig. 4** .06

使用大于1的正数会使网站比默认的规模要大很多。一般说来，这个值通常设置为1.



Minimum-scale和maximum-scale决定了窗口规模大小。当使用minimum-scale 正值时必须小于等于initial-scale的值。同样道理，使用maximum-scale正值是应该大于等于initial-scale值。这两个值都必须设置在0到10之间。



通常说来，这些值不能设置成和initial-scale一样的。这会阻止任何zoom ，而阻止zoom可以使用user-scalable完成这一功能。把user-scalable设置为no可以禁用zoom。相反的，设置user-scalable为yes可以启用zoom。

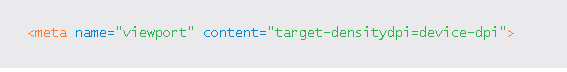
关闭一个网站变化规模(scale)的功能是不明智的。这就没有了易用性和可用性，就不能让网站展示出该有的样子了。So



### Viewport 分辨率

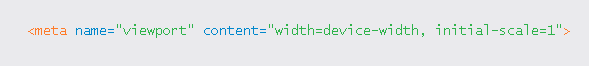
让浏览器自己根据视图窗口规模决定网页规模大小通常是不可取的。当需要给网页添加更多控制，特别是设备的分辨率，可以使用target-densitydpi值。这一属性接受device-dpi, high-dpi, medium-dpi, low-dpi或者确切的DPI数值。

使用target-densitydpi值的机会是很少的，但是当要控制像素时会很有用。



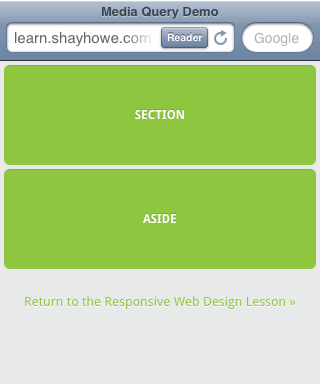
### 合并viewport值

Viewport 的meta标签即接受单个的值也接受多个值，即可一次性设置多个属性。设置多个值需要在content里面用逗号分隔每个属性。下面是一个建议使用的viewport 值，其中使用了width和initial-scale属性。



**Fig. 4**.07

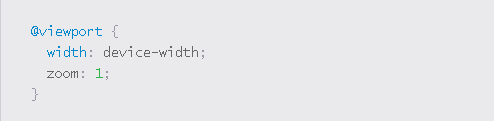
Width-device 和 initial-scale=1的结合提供了初始的尺寸和通常所需的zoom效果。



#### 🡪viewport 的CSS规则

由于viewport这个meta标签在设置网站样式渲染中起了很重要的作用所以建议把viewport这个在HTML中的meta标签转到CSS中用@规则使用。这样可以帮助样式与内容分离，是的网页更具语义性。

目前一些浏览器已经支持使用@viewport规则，但也并不是全面的支持。可以在CSS中使用@viewport规则把之前提到的viewport的meta标签变成这样：

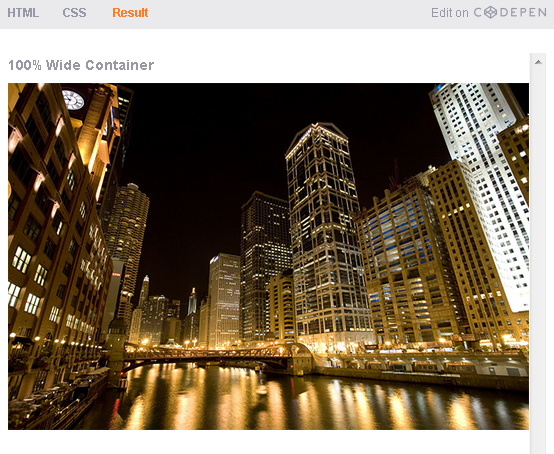


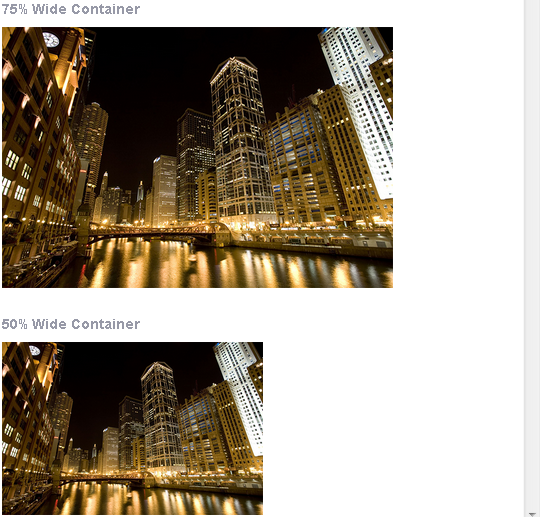
## --弹性媒体 [可变介质]—

最后，在响应式网页设计中同样重要的因素是可变媒介。当视图窗口开始变化尺寸时图片，视频和其他一些需要变化的媒体类型不能随着窗口尺寸的变化而变化。

让媒介变得可变的一个快捷方法是使用max-width属性将其设置为100%。用以确保当视图窗口变小时其他的包含在其中的媒介都能相应变化。







### 弹性的嵌套媒介

遗憾的是max-width这以属性并不适用所有媒体类型，特别是iframe和嵌套的媒体。对一些第三方网站，像是YouTube，它们使用iframe嵌套媒介时就很麻烦了。幸好，这里也有一些[办法](http://www.alistapart.com/articles/creating-intrinsic-ratios-for-video/)。

要想使嵌套的媒体也变得完全响应式，嵌套的媒介需要被包含在一个父元素中并且是absolute定位的。父元素要设置为width为100%用来适应试图窗口的宽度。同时父元素需要设置height为0用来触发IE浏览器的haslayout机制。

然后给父元素设置底部的padding值，并且要设置为和视频一样的长宽比。这就使得父元素宽度和高度是成比例的。还记得之前提到的响应式设计公式么？如果一个视频的长宽比是16:9,9被均分为16后每份为0.5625，意为要设置底部的padding为56.25%.之所以要在底部而不是顶部设置padding是为了避免在IE5.5中发生崩塌，然后给父元素添加上absolute的定位。

