**1：想问自适应页面和响应式页面展现形式到底有什么区别？**

自适应就是网站上的页面，在其他设备上浏览，只是更为友好的展示（原网页的布局在任何设备上都可以保持一致，没有因为屏幕窄宽大小而会影响视觉效果（不会被遮挡）和响应效果）  
响应式，是指在网站上的页面，由于采用响应式的布局，也可以在任何设备上无障碍显示，但是网页的样式早已根据响应式的布局该变成专为其他设备所准备的样式。  
总结：  
自适应，在手机登设备上浏览，网页与PC端无任何差别，所有内容均在浏览器窗口，不会显示“原网页”那种需要用手放大，拖拽去浏览的情况。  
响应式，在手机等设备上浏览，网页与PC端内容相似，但布局是自动改变为专为手机等其他设备所准备的。

2：**网格布局**

* 网格布局的作用在于更有效地控制元素在网页中所占比例的大小。比如，博客中有一个留言板模块，在比较大的屏幕上，我们希望它占了右边 25% 的宽度，在手机等比较小的屏幕上，我们希望它占 100% 的宽度，出现在博客文章下方。网格布局是一种实现这一需求的办法，它的好处是，把所有的宽度分为固定栏数（常用 12 栏），从而更高效的控制元素宽度。而这功能，我们使用 HTML 和 CSS 就能实现了。
* 以 BootStrap 的网格系统为例，DOM 元素类名形如 col-md-4；其中 col 是“列” column 的缩写；md 是 medium 的缩写，适用于应屏幕宽度大于 768px 的场景；4 是占四栏的意思。因此，col-md-4 的意思是，在屏幕宽度大于 768px 时，该元素占四栏

# 3：如何理解[class\*="col-"] ??

**这个是正则的css**

**比如class\*="col-"，表示所有css以col开始的都使用这个规则，**

**由于bootstrap里面很多cssname=col-xxx-xxx**

**所以这个css是为了定义所有col-xxx-xxx的规则的**

4：｛box-sizing: border-box;｝？

-webkit-box-sizing: border-box;/\*-webkit代表chrome、safari私有属性\*/  
-moz-box-sizing: border-box;/\*-moz代表firefox浏览器私有属性\*/  
-ms-box-sizing: border-box;/\*-ms代表IE浏览器私有属性\*/

box-sizing: border-box把所有元素的内边距和边框**不再会增加它的宽度**

其实这是css3中对盒模型定义高度和宽度的最佳方式

**我们经常会遇到一个父级的div有1px的边框而且还需要100%的宽度,这个时候在火狐和chrome与一些新的ie浏览器会出现超出的现象！  
 我们也遇到一个ul里边4个li,每个li需要给个边框和padding,这时候我们最快的方法是每个li的width=‘25%’，但是实际出来的效果确实把ul撑开了  
 如上两个简单的实际情况我想大家都遇到过，我们都希望我们给的宽包含border和padding这两个属性，今天我们能够实现了,box-sizing这个属性能让我们心想事成,随心所欲！下面我和大家分享下box-sizing这个属性：  
 说到 IE 的 bug，在 IE6以前的版本中，IE对盒模型的解析出现一些问题，跟其它浏览器不同，将 border 与 padding 都包含在 width 之内。而另外一些浏览器则与它相反，是不包括border和padding的。  
 在我们开发的过程中会发现，有时候，如果对页面中的大区域进行设置时，将border、padding计算到width和height之内，反而更灵活。但W3C的 CSS2.1规范却规定了他们并不能被包含其中。考虑到这个问题，css3中引入了一个新的属性：box-sizing，它具有“content-box”和”border-box“两个值。**

**box-sizing:content-box  
 当我们设置 box-sizing: content-box; 时，浏览器对盒模型的解释遵从我们之前认识到的 W3C 标准，当它定义width和height时，它的宽度不包括border和padding。  
box-sizing:border-box  
 当我们设置box-sizing: border-box; 时，浏览器对盒模型的解释与 IE6之前的版本相同，当它定义width和height时，border和padding则是被包含在宽高之内的。内容的宽和高可以通过定义的“width”和 “height”减去相应方向的“padding”和“border”的宽度得到。内容的宽和高必须保证不能为负，必要时将自动增大该元素border box的尺寸以使其内容的宽或高最小为0。**

5：**为什么「box-sizing:border-box」没有被广泛采用？**

Result： <https://www.zhihu.com/question/20691294>

6：@media screen 媒体查询代码

这里的媒体查询是指在不同的屏幕大小上显示不同的样式，响应式的代码一般都把宽度写成百分比，用@media要多写几个段。

运用@media实现网页自适应中的几个关键分辨率

从上面我们可以看出有几个临界点的分辨率，那么我们就可以轻松的来写自己的自适应代码了

**@media (min-width: 768px){ //>=768的设备 }**

**@media (min-width: 992px){ //>=992的设备 }**

**@media (min-width: 1200){ //>=1200的设备 }**

注意下顺序，如果你把@media (min-width: 768px)写在了下面那么很悲剧，

**@media (min-width: 1200){ //>=1200的设备 }**

**@media (min-width: 992px){ //>=992的设备 }**

**@media (min-width: 768px){ //>=768的设备 }**

因为如果是1440,由于1440>768那么你的1200就会失效。

所以我们用min-width时，**小的放上面大的在下面**，同理如果是用max-width那么就是**大的在上面，小的在下面**

@media (max-width: 1199){ //<=1199的设备 }

@media (max-width: 991px){ //<=991的设备 }

@media (max-width: 767px){ //<=768的设备 }

# 7：CSS3的calc()使用

http://www.w3cplus.com/css3/how-to-use-css3-calc-function.html