  1. 盒模型

       2. 外边距合并

       3. 定位机制

       4. Float

       5. Position：属性有哪些取值，它们的行为是什么？ 无依赖绝对定位？

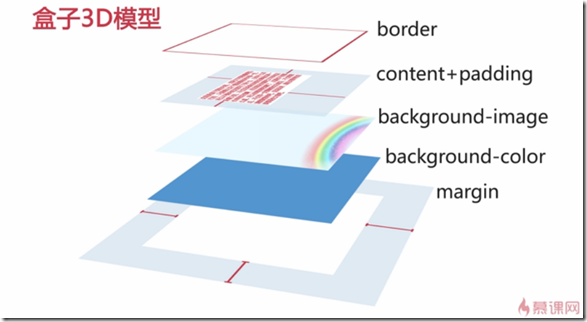
       6. Display：常见属性值有哪些取值？

       7. 对BFC规范的理解？

**1. 盒模型**

盒模型从内向外包括content,padding,border和margin。

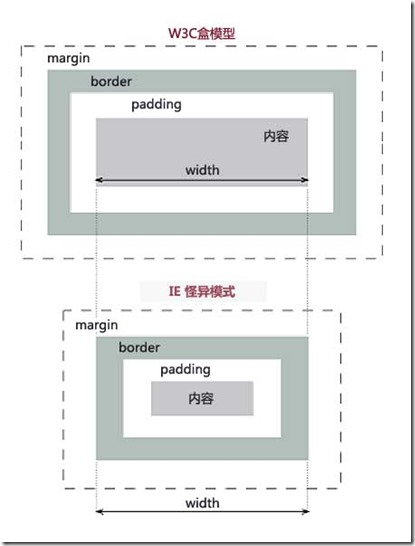
从前往后分别是：border，content+padding,background-image,background-color,margin

[](http://images0.cnblogs.com/blog/644616/201508/051002374399578.png)

盒模型有两种模式：W3C标准模式和IE怪异模式，

W3C标准模式content不包括padding和border.

IE怪异模式中content包括padding和border。

[](http://images0.cnblogs.com/blog/644616/201508/051002416427361.jpg)

IE下只要完整定义DOCTYPE，都会触发**标准模式**，而如果DOCTYPE缺失则在**ie6，ie7，ie8**下将会触发**怪异模式（quirks 模式）**。

所以：如果发现页面元素width或height有问题，则主要观察页面的DOCTPEY声明是否正确。

怪异模式并不是一点使用价值都没有，在CSS3中可以通过属性box-sizing设置盒模型：

box-sizing: content-box; // 默认情况,标准模式box-sizing: border-box; //采用怪异模式

比如：在页面进行流体布局时，将宽度定义为百分数，最后就需要将盒模型定义为怪异模型，增加padding而不会影响整体width,否则会超过整体宽度的100%。

**2. 外边距合并**

（注释：只有普通文档流中块框的垂直外边距才会发生外边距合并。行内框、浮动框或绝对定位之间的外边距不会合并。）

外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。

**合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者**。

#### **外边距合并**

外边距合并（叠加）是一个相当简单的概念。但是，在实践中对网页进行布局时，它会造成许多混淆。

简单地说，外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

当一个元素出现在另一个元素上面时，第一个元素的下外边距与第二个元素的上外边距会发生合并。请看下图：

当一个元素包含在另一个元素中时（假设没有内边距或边框把外边距分隔开），它们的上和/或下外边距也会发生合并。请看下图：

尽管看上去有些奇怪，但是外边距甚至可以与自身发生合并。

假设有一个空元素，它有外边距，但是没有边框或填充。在这种情况下，上外边距与下外边距就碰到了一起，它们会发生合并：

如果这个外边距遇到另一个元素的外边距，它还会发生合并：

这就是一系列的段落元素占用空间非常小的原因，因为它们的所有外边距都合并到一起，形成了一个小的外边距。

外边距合并初看上去可能有点奇怪，但是实际上，它是有意义的。以由几个段落组成的典型文本页面为例。第一个段落上面的空间等于段落的上外边距。如果没有外边距合并，后续所有段落之间的外边距都将是相邻上外边距和下外边距的和。这意味着段落之间的空间是页面顶部的两倍。如果发生外边距合并，段落之间的上外边距和下外边距就合并在一起，这样各处的距离就一致了。

**3. 定位机制**

      CSS中存在3中定位机制： 标准文档流，浮动和绝对定位；

**标准文档流**：从上到下，从左到右输出文档内容，由块级元素和行级元素组成。

**浮动**：使块级元素横向布局，多列布局关键。

**绝对定位**：设置浮动的元素不会处于标准文档流中,目的：横向多列布局和一些复杂效果，如遮罩层，全屏显示等。

**4. Float**

       元素会向左或向右移动，直到触碰到容器（父元素）为止。值有left，right和none；

       浮动最开始诞生的原因只是为了让文字环绕图片。

**Float主要特性**：

                              坍塌性：被设置了float的元素会脱离文档流，**父级包裹不住子集**

                              包裹性：浮动元素宽度不起作用，**内容撑开宽度**；

                              清空格：会导致节点脱离文档流结构，空格，换行都不起作用；

        具有坍塌性的样式有：

                                      float,  position:absolute/fixed/sticky

        体现包裹性的css样式有：

                                      float

                                      display:table/table-cell、inline-block

                                      position:absolute/fixed/sticky

                                      Overflow:hidden/scroll

**清浮动**：由于浮动有坍塌性，父级包裹不住子级（只能用在浮动元素的父级上。不必过多使用）

[IMG_258](http://www.cnblogs.com/Peng2014/p/javascript:void(0);" \o "复制代码)

.clear:after{

content: "";

display: table;

clear: both;

}

.clear{

\*zoom: 1; /\*兼容IE低版本\*/

}

[IMG_259](http://www.cnblogs.com/Peng2014/p/javascript:void(0);" \o "复制代码)

**5. Position:属性有哪些取值，它们的行为是什么？** （是否脱离文档流？是否改变元素特性？怎么定位？）

           取值有：static / relative / absolute / fixed / inherit

**static**: 默认值。没有定位，元素出现在正常的流中 （忽略 top, bottom, left, right z-index 声明）。

**relative**: 1.不使元素脱离文档流；

                         2. 不影响元素本身的特性； 块元素还是块元素，内嵌还是内嵌;

                         3. 如果没有定位偏移量，对元素本身没有任何影响；

                         4. 如果有定位偏移量，则相对于其正常位置进行定位；

                         5. 还拥有z-index声明;

**absolute**:1. 使元素完全脱离文档流；（和float一样）

                         2. 使内嵌支持宽高；（和float一样）

                         3. 块属性标签内容撑开宽度(如果为设定宽高）；（和float一样，包裹性)

                         4. 如果没有定位偏移量，位置不变，具有跟随性；

                         5. 如果有定位偏移量，如果有定位父级相对于定位父级发生偏移，没有定位父级相对于整个文档<html>发生偏移；

                             (注意是**定位父级**，一般在需要绝对定位的父级上添加相对定位position:relative;)

                         6. 还拥有z-index声明;

**fixed**: 与绝对定位的特性基本一致，差别是始终相对屏幕窗口尺寸进行定位，（兼容性，IE6不支持，利用JS解决）

**inherit**:规定应该从父元素继承 position 属性的值。

            设置absolute会使得元素已有的float失效。不过float和absolute同时使用的情况不多；

           上文提到了absolute会使元素悬浮在页面之上，如果有多个悬浮元素，层级如何确定？答案是“后来者居上”

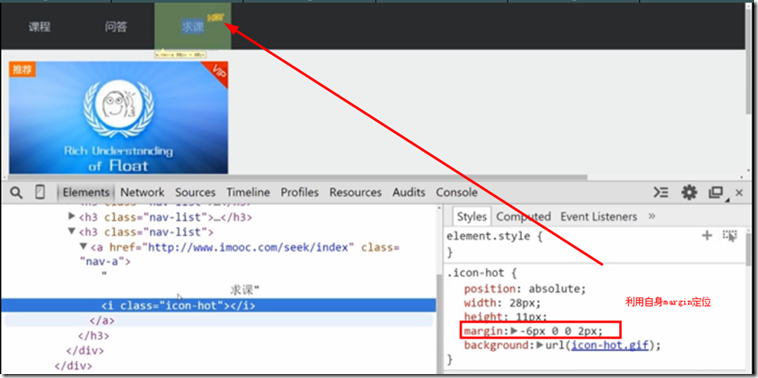
           Absolute与float 兄弟关系，都具有包裹性、破坏性。

**5.2 无依赖绝对定位的原理？**

             无依赖绝对定位，不用在父级添加relative，利用绝对定位的**跟随特性**，如果没有偏移量则元素位置不变，

             从而利用margin值进行定位（不过在移动端使用负margin会出现问题）

             无依赖绝对定位为页面布局与重构提供了更加广阔的选型新思路，详情查看张鑫旭的[CSS深入理解之absolute](http://www.imooc.com/learn/192)。

[](http://images0.cnblogs.com/blog/644616/201508/051002472674143.png)

**6. Display：常见属性值有哪些取值？**

|  |  |
| --- | --- |
| 值 | 描述 |
| none | 此元素不会被显示。 |
| block | 此元素将显示为块级元素，此元素前后会带有换行符。 |
| inline | 默认。此元素会被显示为内联元素，元素前后没有换行符。 |
| inline-block | 行内块元素。（CSS2.1 新增的值） |
| list-item | 此元素会作为列表显示。 |
| table | 此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。 |

**7. 对BFC规范的理解？**

W3C CSS 2.1 规范中的一个概念,它决定了元素如何对其内容进行定位,以及与其他元素的关 系和相互作用。