**字体设置**

font-family：一次性可以设置多个字体，中间使用，隔开，每个字体尽量使用引号包含

font-size： 字体大小

font-style：italic和normal

font-weight：bold和normal

color：三种颜色值设置方式

还有一种简写方式：font

**font-style font-weight font-size font-family**

* 前面两种可以省略，它们的默认值都是normal，顺序也可以交换
* 后面两种不能省略，也不能交换顺序。

**文本设置**

* text-decoration：修饰线，underline、overline、line-through、none
* text-indent：文本缩进，通常使用em来实现段落缩进
* text-align：文本水平对齐方式，left、center、right、justify（只对英文有效）
* line-height：行高，设置段落或列表之间的行距，还可以实现垂直居中

# CSS布局初步

## 1.CSS布局思路及核心知识点

**布局思路**

思考网页的信息语义和结构。根据这些信息把一个网页分成不同的内容块，以及每块内容的目的，然后再根据这些内容的目的用不同的语义元素建立相应的HTML结构。

**核心知识点：**

* 盒模型 （流动布局）
* 浮动 （浮动布局）
* 定位（层布局）

## 盒模型概念

从日常生活中出发，

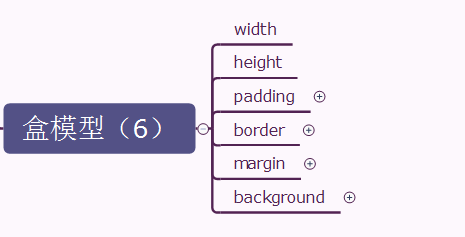
在我们的HTML中，所有的标签都是**矩形**的。围绕着整个矩形，我们可以使用一系列的属性来控制它的显示外观。

具体来说，有如下一些属性：

* width: 宽度
* height：高度
* border：边框
* padding：内边距 补白
* margin：外边距 边界
* background：背景

### (1).什么是盒模型？

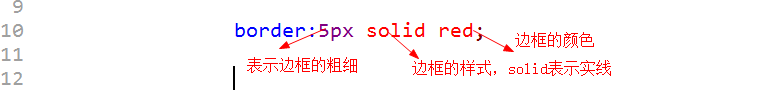
CSS定义所有的元素都可以拥有像盒子一样的外形和平面空间，即都包含**边界**、**边框**、**补白**、**内容区域**、**背景**（包括背景颜色和背景图片），这就是盒模型。



**border设置**

border右三部分构成：

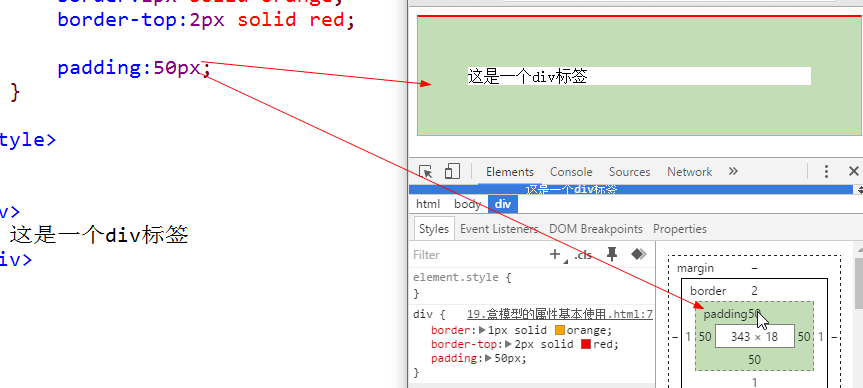
* 边框的粗细
* 边框的样式
* 边框的颜色



**padding**

表示的内边距，或者补白。

所处的区域是指 **border和内容之间的那个空间**。



如果我们只写一个属性值的时候，意味着四个方向上的内边距都是一样的。

但是在实际开发时，需要设置不同的内边距。怎么办？

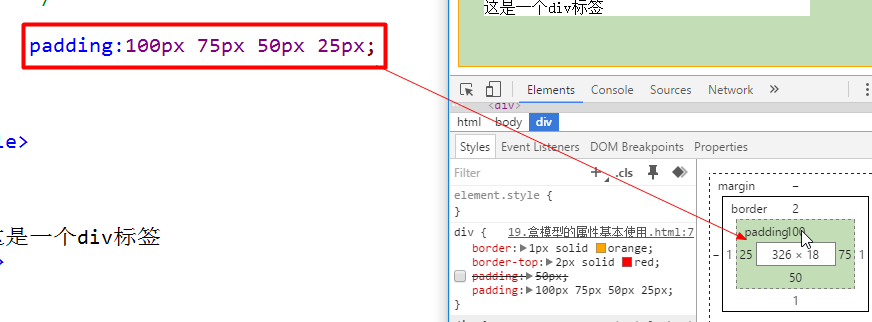
可以单独的设置四个方向上的padding值

* padding-top:
* padding-right:
* padding-bottom:
* padding-left:



实际上，还有简写的方式，

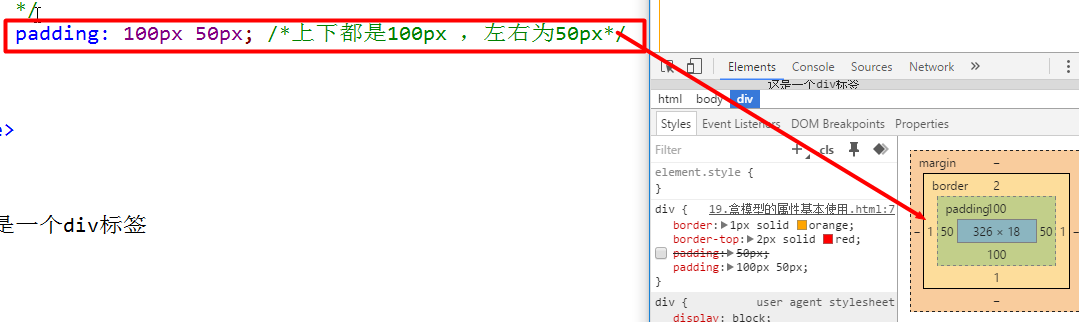
在设置padding属性值的，可以按照**上、右、下、左**的顺序来设置四边的padding值。



怎么记这个顺序：

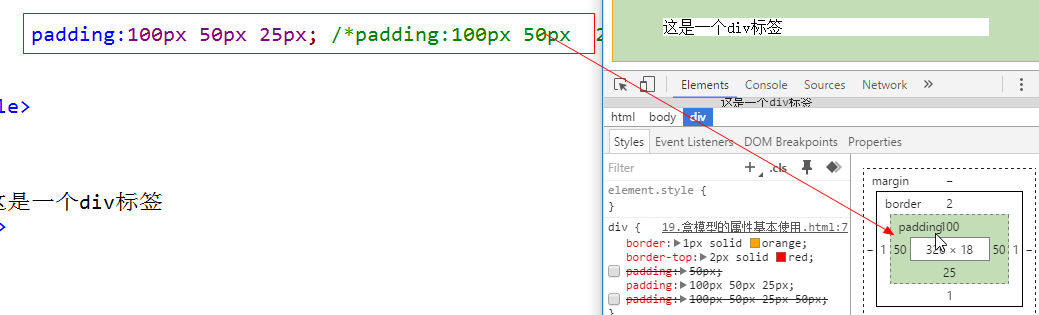
* 顺时针，从12点开始
* **TR**ou**BL**e，T-top R-right B-bottom L-left

如果padding-top和padding-bottom相同，并且padding-right和padding-left相同的话，我们可以简写：padding: tb rl



还有一种情况：

上下padding不同，左右padding相同



小结：

关于padding的属性值，有四种写法：

* 只有一个值 上、右、下、左都相同
* 有两个值：上下是相同的，左右是相同的
* 有三个值：上下不同，第一个值对应的上，第二个值表示的左右，它们是相同的，第三个值表示下，（**这个容易搞错**）
* 有四个值：分别对应于上、右、下、左。

它们都是空格隔开

**d.margin**

用来设置外边距的。

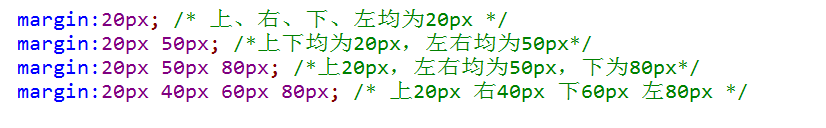
看似对当前的盒子没什么大的影响，实际上不是的。

我们在计算当前盒子在整个页面中所占空间的时候，是必须要考虑margin 的。

和padding的使用差不多。

margin的写法也有如下几种：

* 一个值
* 两个值
* 三个值
* 四个值



当然，和padding一样，也可以单独的设置四个方向上的margin值

* margin-top
* margin-right
* margin-bottom
* margin-left

关于margin还有一个技巧，可以实现水平居中效果。



**e.background**

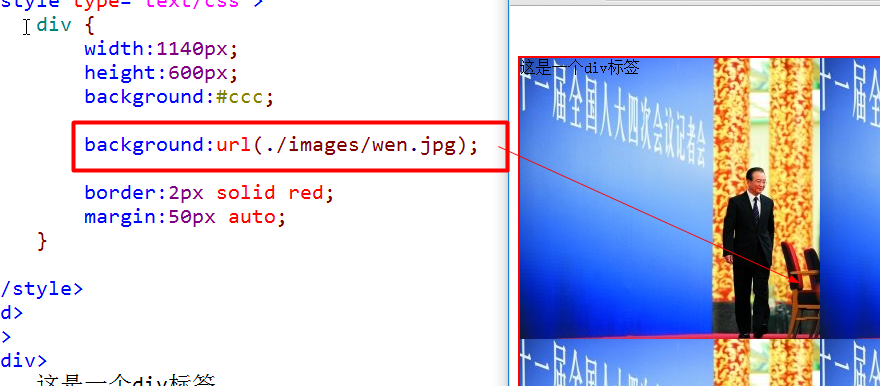
设置盒子的背景。

包括如下两种：

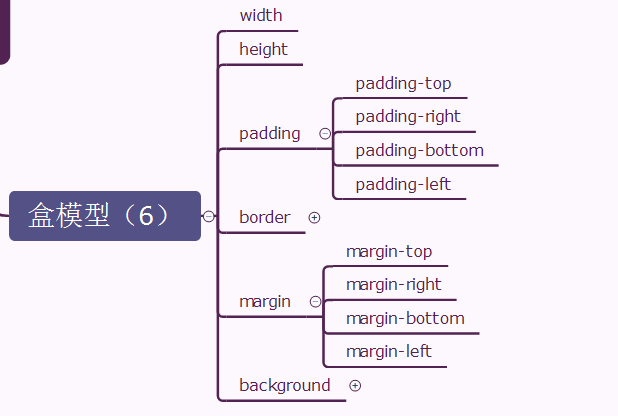
* 背景颜色
* 背景图片

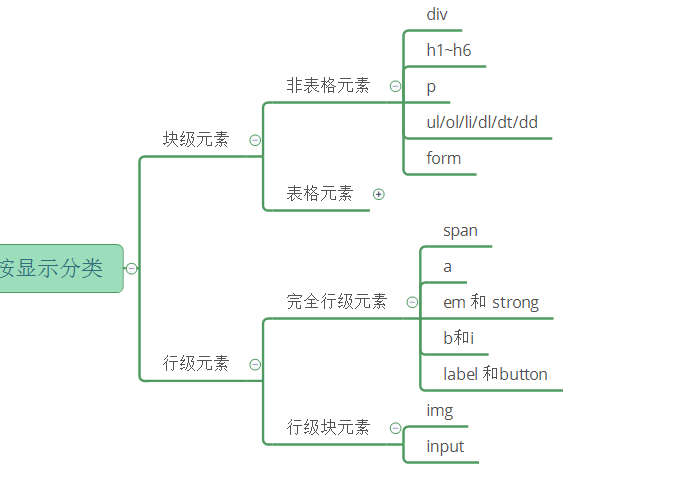


背景图片的使用



小结：





以后我们每写一个标签，首先问题就是弄清它的性别关系。

实际上，在盒模型中，

* 对于块级元素，所有的这6大特性都能生效。
* 对于行级元素，width和height不起作用，设置了也是白设，border有效、background有效、margin和padding在水平方向上有效，在垂直方向上无效。

**块级和行内元素各有什么特点，请列举常见的块级元素和行内元素**

在所有的元素中，根据其表现形式，又分为两种：

* 块级：占据一整行的，盒模型的6大属性表现均ok，
* 行内（内联）：有多少内容就占多少空间，width和height无效，margin和padding只在水平方向上有效。

常见的块级：div，h1~h6，p，ul，ol，li，dl，dt，dd，form，table

常见的行内：a，span，em，strong

常见的行内块：img，input

## 浮动初步

浮动是css中的难点，也是重点。

**作用**：用来实现布局的，使得块级元素能够在同一行上和平共处。

**属性**：float

**属性值**：left、right

**浮动的作用，以及如何使用的？**

作用：实现布局，使得块级元素能够在同一行中排列。

float：left/right

使用的时候，需要注意清除浮动。

在css中，有三种布局模型：

* 流动布局，默认的
* 浮动布局，使用float来实现
* 层布局，使用position来实现的

**什么时候会用到层布局？**

以京东、淘宝、hao123为例进行分离，有很多具体的层布局应用

**层布局有哪些好处？**

能够实现更为丰富的布局方式，为用户提供更好的用户体验。

在css中，有如下三种定位方式：

* 相对定位
* 绝对定位
* 固定定位

它们是通过position属性来实现的

有如下几个值：

* relative，相对定位
* absolute，绝对定位
* fixed，固定定位
* static，不定位，默认的流动布局，一般不会写