# CSS继承性+层叠性+盒子+浮动

## CSS继承性

<style>

div{

color: pink;

font-size: 30px;

width: 500px;

background: green;

padding: 100px;

}

p{

color: red;

}

</style>

<div>

<p>我们都是好孩子</p>

<ul>

<li>

<p>我们都是栋梁之才</p>

</li>

</ul>

</div>

通过以上的代码。，我们可以看到p标签继承了div设置的样式。

继承不仅仅是子代，可以是子代的子代。可以一直延续。

继承是有原则的：一般是color、text—、font-、line- 这些属性。

## 层叠性

### 2.1

在css中有许多的重叠样式，在众多的样式中选择元素需要的那一个。那么该如何去选择呢？ 在浏览器的内核中有着严谨的权重计算公式，通过权重计算公式去选择需要的样式。

权重公式：id选择器，类选择器，标签选择器 优先级依次减小。

1. 如果元素被选中了，那么就可以使用权重公式来计算大小，大的将被使用。如果一样大小，则谁写在了后面就使用谁。
2. 如果元素没有被选中，则通过继承的方式传递样式，这时权重可以看做为0. 这种情况下我们使用就近原则，就近原则：谁靠着该元素近那么就选谁。
3. 如果元素没有被选中，并且距离元素都一样近，这时我们又可以使用权重计算公式。 如果权重一样大小，则谁写在了后面就使用谁。

### 2.2

.two{

color: green;

}

.one{

color: red;

}

<p class="one two" id="haha">我们都在学习H5</p>

以上这种情况，是按照样式的先后去选择。与html中元素设置的class属性名字前后顺序无关。

### 2.3

！Important ：提升属性的优先级。

#haha{

color: orange;

font-size: 20px;

}

p{

color: blue !important; 提升属性的优先级

color: blue; !important;

font-size: 14px;

}

/\*不影响就近原则\*/

.heiheihei{

color:yellow !important;

}

.xixi{

color: skyblue;

}

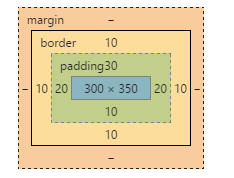
需要注意三点：

1. 它只针对于属性有效，需要注意正确的书写方式，写在分号内。
2. 它无法提升继承性的选择器权重（也就是未选中该属性的选择器），该是0还是0
3. 它不影响就近原则。

## 盒子模型

### 3.1盒子

盒子包含5个元素：width、height、padding、border、margin



Width：一定要注意width是内容的宽度。

Height：注意height是内容的高度。

Padding：内边距，内容与边框之间的间隙

Border：边框

Margin：外边距，相对于内边距记忆。边框外部的间隙

### 3.2padding内边距

Padding有四种书写方式：

1. padding：20px； 代表上右下左四个方向都是20像素的边距
2. Padding：20px 30px；第一个数字是上下方向的大小，第二个数字是左右方向的大小
3. Padding：20px 30px 40px；第一个数字是上方向的大小，第二个数字是左右方向上的大小，第三个是下反向的边距大小
4. Padding：10px 20px 30px 40px；这分别是四个方向上的边距大小。

Padding边距的另一种写法形式

padding-top: 10px;

padding-right: 70px;

padding-bottom: 80px;

padding-left: 90px;

分别写四个方向上的边距大小。

### 3.3border边框

Border的写法：

border: 10px solid #567;

三个属性分别代表了：宽度、样式、颜色； 颜色是可以省略的默认为黑色，但是其它两个属性是不可以省略的。

线条样式：有很多种，举例：solid实线 dashed虚线 outset突出感 inset凹

border-width: 10px;

border-style: dashed;

border-color: #567;

单独写边的宽度、样式以及颜色。这样写还是四条边共用。

border-top: 10px solid red;

border-right: 5px solid yellow;

border-bottom: 15px solid blue;

border-left: 20px solid #789;

这是分别给每一条边设置宽度、样式和颜色。

border-top-style: dashed;

这是顶边设置样式属性，也就是说每一条边每一种属性都可以单独去写。

## 标准文档流

### 4.1

标准文档流中的几种现象：

1. 排布的顺序为从左到右，从上到下。
2. 空白折叠现象，如果html语言中有换行或者空格、缩进都会替换为空格在网页中显示。
3. 在同一行中，底部基线对齐。
4. 文本自动换行。

根据标准文档流将标签分为两种类型：

块级标签：独占行，可以设置宽高以及边距，一般可以放任意内容（P标签除外）。

行内标签：可以在一行内显示，不可以设置宽高以及某些间距，一般是文本级标签（p除外）。

### 4.2

在标准文档流中，也可以实现块级元素与行级元素的身份互换。只需要通过display属性进行互换

Display：block；设置为块级元素

Display：inline；设置为行内级元素

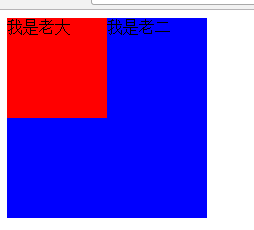
Display：inline\_block; 设置为行内块级元素。行内块级元素：在行内显示，不独占行。并且可以设置宽高和边距。

## 浮动

浮动就是让我们的元素脱离标准文档流，目的是为了布局好看！

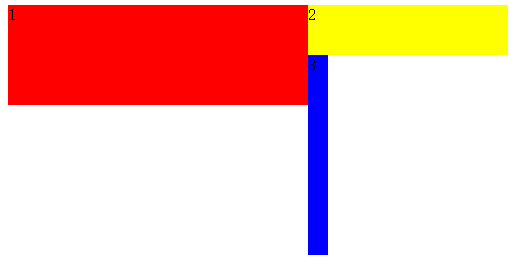
浮动的现象：

1. 脱离标准文档流被叫做脱流，同时会出现字围现象。我是老二没有被盖住！！



1. 浮动的元素会相互贴靠，而且如果父容器空间足够大，则浮动的元素会正常紧靠也就是后一个元素会紧靠前一个元素。如果父容器空间不够大，那么元素会找上级的上级去紧靠，如果还找不到则继续找上级紧靠，没有了上级可靠那就靠边。







1. 浮动的元素不分块级和行内类别，也就是说都可以设置宽高可边距。
2. 浮动以后，同一行内的元素以顶边作为基线对齐。