# 复习

标准文档流中标签分类：

块级元素：可以设置宽高；不设置宽度，自动盛满父盒子内容宽度；独占一行

行内元素：不能设置宽高；大小是内容撑开；多个行内元素显示在一行

display: block;(转为块级元素)

inline(转为行内元素)

none(没有)

脱标：浮动，绝对定位，固定定位

浮动性质：

1浮动脱标，不在区分块级和行内，既可以设置宽高，又可以显示在一行。

2 浮动元素依次贴边显示

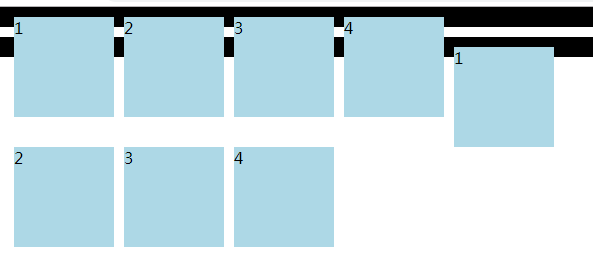
3 浮动元素让出标准流位置

4 文字不会压盖住

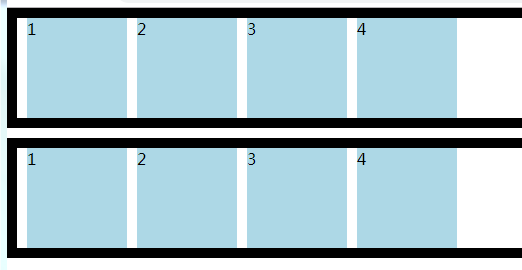
# 浮动存在的问题

浮动的元素不能撑高父盒子的高度（高度自适应）

浮动的元素会影响后面浮动的元素



## 直接给父盒子添加height



解决：浮动的元素会影响后面浮动的元素

没解决：高度自适应

## 给后面浮动元素父盒子添加clear

clear:清除浮动影响

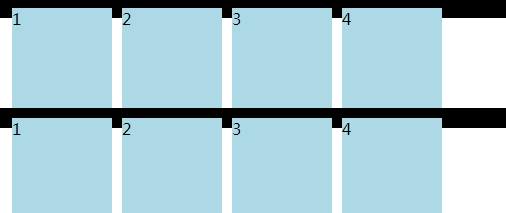
left:清除左浮动影响

right:清除右浮动影响

both:清除左右浮动影响

解决：浮动的元素会影响后面浮动的元素

没解决：高度自适应

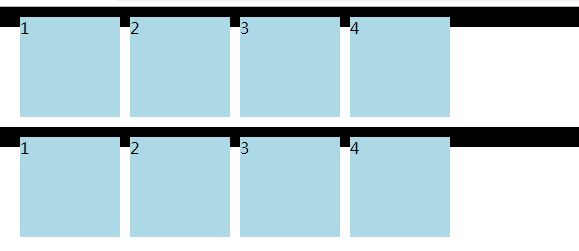


## 外墙法

在两个父盒子之间添加一堵“有高度”，“清除浮动”墙

解决：浮动的元素会影响后面浮动的元素，在视觉上有了间隔

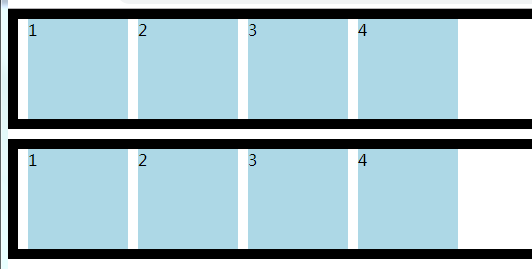
没解决：高度自适应



## 内墙法

在所有浮动元素最后添加一堵“清除浮动”墙

解决：浮动的元素会影响后面浮动的元素，高度自适应



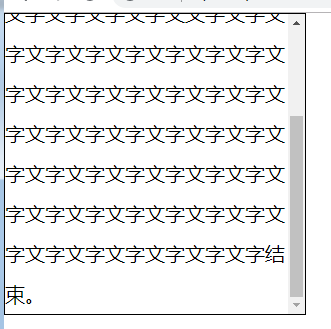
总结：虽然内墙法解决了所有的问题，但是并不会使用。因为给页面增加了很多无意义的标签。

## overflow

overflow:溢出处理方法

auto：溢出滚动

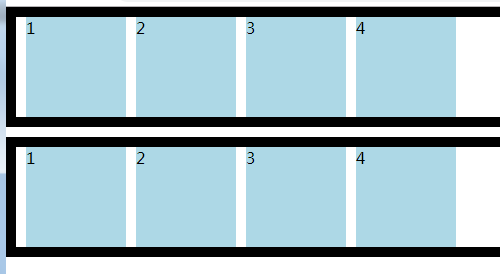
当内容超过盒子设置的高度，自动出现滚动条，拉动滚动条可以查看全部内容



hidden:溢出隐藏

当内容超过盒子设置的高度，将超过部分内容隐藏。尤其设置padding，padding-bottom内容不会隐藏，将边框之外的内容隐藏。

使用overflow:hidden给父盒子添加。可以解决所有的问题。



总结：如果子盒子浮动，父盒子没有设置height,希望父盒子高度自适应，给父盒子添加overflow: hidden.

overflow: hidden;两个引申含义

强制检测子盒子高度，将检测的高度自动撑高父盒子。（高度自适应）

强制检测浮动子盒子，管理浮动元素不让浮动影响其他浮动的元素（浮动元素对后面元素影响）

# a标签

## 2.1 a标签4个伪类

a标签特殊的属性和状态

根据用户不同行为，a标签对应四个不同状态

a:link a标签未点击之前的状态

a:visited a标签点击之后的状态

a:hover a标签悬停状态

a:active a标签鼠标激活状态

|  |
| --- |
| 1. a:link { 2. color: red; 3. } 4. a:visited { 5. color: green; 6. } 7. a:hover { 8. color: blue; 9. } 10. a:active { 11. color: orange; 12. } |

伪类和类类似，比如类名.box书写的样式属性打开网页可以立即渲染。伪类中书写样式属性不能立即加载，

只有用户操作相应的行为，才可以加载相应的伪类。

伪类也是权重和类相同。

## 2.2 伪类顺序

伪类按照顺序书写才可以正常加载。

a:link a:visited a:hover a:active

## 2.3 应用

a标签是行内元素

文字属性可以从祖先继承，但是a标签下划线和文字颜色不能从祖先继承，必须单独给a标签设置。

a标签设置的样式属性都在加载在四个伪类样式中，

特殊的样式属性在单独给伪类设置

|  |
| --- |
| 1. .nav { 2. width: 960px; 3. height: 40px; 4. margin: 50px auto; 5. } 6. .nav ul li { 7. float: left; 8. width: 120px; 9. height: 40px; 10. } 11. .nav ul li a { 12. /\*转块设置\*/ 13. display: block; 14. height: 40px; 15. line-height: 40px; 16. background-color: orange; 17. text-align: center; 18. } 19. /\*伪类有特殊样式属性在后面单独层叠\*/ 20. .nav ul li a:hover { 21. background-color: lightblue; 22. color: #fff; 23. } 24. .nav ul li.cur a { 25. background-color: lightblue; 26. color: #fff; 27. } |



# 背景

## 3.1 background-color

背景色：渲染在边框以内

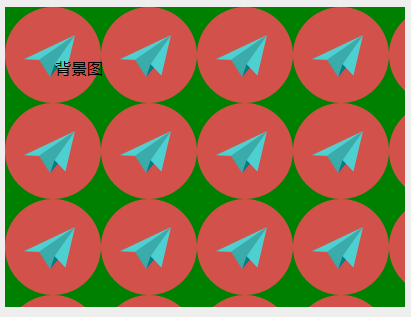
属性值：三种色值都可以使用rgba(255,255,255，.5);

|  |
| --- |
| 1. background-color: rgba(255,0,0,.5); |

## 3.2 background-image

背景图：背景图渲染在背景色之上

url():统一资源定位符，书写图片路径



## 3.3 background-repeat

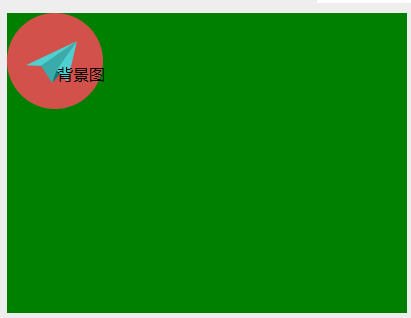
背景是否重复

默认值：repeat

在水平方向平铺：repeat-x

在垂直方向平铺：repeat-y

不重复：no-repeat



利用背景图实现背景渐变。切1px背景图，然后在水平方向进行重复。

## 3.4 background-position

背景定位：背景图在整个大背景区域中的位置。

像素表示法：水平位置 垂直位置；



第一个参数：50px 背景图左上角相对于大背景区域左上角水平方向偏移量。

第二个参数：100px 背景图左上角相对于大背景区域左上角垂直方向偏移量。

## 3.5 精灵图

css 精灵图技术：css sprite。

网页上有很多小的碎图片，每一个图片加载都会发起一次http请求。这些小图片会发起很多http请求，降低网页的加载速度。把很多小碎图片放在一张图片上，只会发起一次http请求，提高网页的加载速度。这种技术就是css 精灵图技术。

1限制盒子宽高

2通过background-position的到想要的小精灵图

|  |
| --- |
| 1. div { 2. /\*限制盒子宽高\*/ 3. width: 103px; 4. height: 134px; 5. background-image: url(images/zimu.jpg); 6. background-repeat: no-repeat; 7. float: left; 8. margin-right: 10px; 9. /\*通过background-position的到想要的任意小图片\*/ 10. background-position: -404px -17px; 11. } 12. .i { 13. width: 79px; 14. height: 128px; 15. background-position: -155px -189px; 16. /\*padding: 50px; 17. border: 20px solid #000;\*/ 18. } 19. .y { 20. width: 101px; 21. height: 126px; 22. background-position: -477px -529px; 23. } |

## 3.6 精灵图制作

1图片格式一般设置为透明，png

2 精灵图大小不要大，小图片从上下依次摆放。

3 从左到右摆放

## 3.7 单词表示法

水平方向：left right center

垂直方向：top bottom center

单词表示法使用：大背景图居中。

通栏大banner居中

|  |
| --- |
| 1. .banner { 2. /\*通栏 省略宽度\*/ 3. height: 500px; 4. background-color: lightblue; 5. background-image: url(images/banner.jpg); 6. background-position: center top; 7. } |

## 3.8 百分数表示法

水平100%对应的含义：大背景区域宽度-背景图宽度

垂直100%对应的含义：大背景区域高度-背景图高度

|  |
| --- |
| 1. background-position: 50% 50%; //百分数设置背景图水平垂直居中 |



|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |