

补充知识点：去掉标签前面的小圆点
语法：

```
list-style:none;
```

1.圆角边框（重点）

在css3中，新增了圆角边框样式

border-radius属性用于设置元素的外边框圆角

语法：

```
border-radius:length;  
例如：border-radius:10px;
```

- length参数值可以为数值或者百分比的形式
- 如果是正方形，想要设置一个圆，把数值改为高度或者宽度的一半即可，或者直接写50%
- 如果是一个矩形，想要在左右两端设置一个半圆的圆角，把数值设置为高度的一半即可
- 该属性事一个简写属性，可以跟4个值，分别代表左上角，右上角，右下角，左下角
- 也可以分开写：
border-top-left-radius,border-top-right-radius,border-bottom-left-radius,border-bottom-right-radius

2.盒子阴影（重点）

css3中新增了盒子阴影

box-shadow属性为盒子添加阴影

语法：

```
box-shadow:h-shadow v-shadow blur spread color insert;
```

值	描述
h-shadow	必需。水平阴影的位置。允许负值
v-shadow	必需。垂直阴影的位置。允许负值
blur	可选。 模糊距离
spread	可选。阴影的尺寸

值	描述
color	可选。阴影的颜色
insert	可选。将外部阴影（outset）改为内部阴影

注意：

- 默认为外阴影（outset）但是不可以写这个单词，否则导致阴影无效
- 盒子阴影不占用空间，不会影响其他盒子的排列

3.文字阴影

css3中，可以使用text-shadow属性将阴影应用于文本

语法：

```
text-shadow:h-shadow v-shadow blur color;
```

值	描述
h-shadow	必需。水平阴影的位置。允许负值
v-shadow	必需。垂直阴影的位置。允许负值
blur	可选。模糊的距离
color	可选。阴影的颜色

4.浮动

4.1. 传统网页布局的三种方式

网页布局的本质——用css来拜访盒子。把盒子放到相应的位置

css提供了三种传统的布局方式（简单说，就是盒子如何排列）：

- 普通流（标准流）
- 浮动
- 定位

4.2. 标准流（普通流、文档流）

所谓标准流,就是标签按照默认规定好的方式排列

- 块级元素会独占一行，从上向下顺序排列
常用元素：div,hr,p,h1~h6,ul,ol,dl,form,table
- 行内元素会按照顺序，从左到右顺序排列，碰到父元素边缘则自动换行
常用元素：span,a,i,em等

标准流是最基本的布局方式
实际开发中，一个页面基本都包含了三种布局方式（后面移动端学习新的布局方式）

4.3. 为什么需要浮动

很多的布局效果，标准流没有办法完成，此时就可以利用浮动完成布局。因为浮动可以改变元素标签默认的排列方式

- 浮动最典型的应用：可以让多个块级元素一行内排列显示
- 网页布局第一原则：多个块级元素纵向排列找标准流，多个块级元素横向排列找浮动

4.4. 什么是浮动

float属性用于创建浮动框，将其移动到一边，直到左边缘或者右边缘触及包含块或另一个浮动框的边缘

语法：

```
选择器{float:属性值; }
```

属性值	描述
none	元素不浮动（默认值）
left	元素向左浮动
right	元素向右浮动

4.5. 浮动特性（重难点）

- 浮动元素会脱离标准流（脱标）
- 浮动的元素会一行内显示并且元素顶部对齐
- 浮动的元素会具有行内块元素的一些特性

浮动 (float) 元素的最重要特性:

脱离标准普通流的控制移动到指定位置 (俗称脱标)

浮动的盒子不再保留原先的位置

如果多个盒子都设置了浮动, 则它们会按照属性值一行内显示并且顶端对齐排列

浮动元素会具有行内块元素特性

任何元素都可以浮动, 不管原先是什么模式的元素, 添加浮动后具有行内块元素相似的特性

如果块级元素没有设置宽度, 默认和父元素一样宽, 但是添加浮动后, 它的大小根据内容来决定

浮动的盒子之间是没有缝隙的, 是紧挨在一起的

行内元素同理

4.6. 浮动元素经常和标准流父级搭配使用

为了约束元素位置, 我们网页布局一般采用的策略是:

先用标准流的父元素排列上下位置, 之后内部子元素采取浮动排列左右位置, 符合网页布局第一准则

网页布局的第二准则: **先设置盒子大小, 再设置盒子位置**

5. 常见网络布局

5.1. 常见网页布局

5.2. 浮动布局注意点

(1) 浮动和标准流父盒子搭配使用

先用标准流的父元素排列上下位置, 之后内部子元素采取浮动排列左右位置

(2) 一个元素浮动了, 理论上其余的兄弟元素也要浮动

一个盒子中有多个子盒子, 如果其中一个盒子浮动了, 那么其他盒子也应该浮动, 以防引起问题

浮动的盒子只会影响浮动盒子后面的标准流, 不会影响前面的标准流

6. 清除浮动

6.1. 为什么要清除浮动

由于父级盒子很多情况下，不方便给高度，但是子盒子浮动又不占用空间，最后父级盒子高度为0时，就会影响下面的标准流盒子

6.2. 清除浮动的本质

清除浮动的本质就是清除浮动元素造成的影响

如果父盒子本身有高度，则不需要清除浮动

清除浮动后父级就会根据浮动的盒子自动检测高度。父级有了高度，就不会影响下面的标准流了

语法：

```
选择器{clear:属性值;}
```

属性值	描述
left	不允许左侧有浮动元素（清除左侧浮动的影响）
right	不允许右侧有浮动元素（清除右侧浮动的影响）
both	同时清除左右两侧的浮动的影响

- 实际工作中，几乎只用clear:both;
- 清除浮动的策略是： 闭合浮动

6.3. 清除浮动的方法

- 额外标记法,也称为隔墙法，是w3c推荐的做法
- 父级添加overflow属性
- 父级添加after伪元素
- 父级添加双伪元素

额外标签法

额外标签法会在浮动元素末尾加一个空的标签。例如<div style="clear:both"></div>,或者其他标签，(如</br>等)

要求新添加的空标签必须是块级元素

给父级添加overflow

可以给父级添加overflow属性，将其属性值设置为hidden、auto、或scroll

缺点：无法显示溢出部分

:after伪元素法

:after方式是额外标签法的升级版，也是给父级添加元素

语法：

```
.clearfix:after{
  content:"";
  display:block;
  height:0;
  clear:both;
  visibility:hidden;
}
```

以下为IE6,7专有：

```
.clearfix{
  *zoom:1;
}
```

双伪元素清除浮动

也是给父元素添加

语法：

```
.clearfix:before,.clearfix:after{
  content:"";
  display:table;
}
.clearfix:after{
  clear:both;
}
.clearfix{
  *zoom:1;
}
```