# 十一 知识点

## 1 盒子居中

### 1.1 内容水平居中text-align:center

可以让盒子内的内容居中，包括：文字、行内元素、行内块元素。如下span被居中了：

#div {  
 background-color: aquamarine;  
 text-align: center;  
 width: 500px;  
}

<div id="div">  
 <span>天安门</span>  
</div>

注意：如果盒子有了高度，text-align:center只能让内容水平居中，不能垂直居中。

### 1.2 块级元素盒子居中margin:0 auto;

margin:0 auto; 能让盒子本身水平居中，0是上下外边距

margin:auto; 上下左右都是居中

原理：auto是自动充满的意思，当书写margin-left:auto; 左侧自动充满，盒子会在右边出现，当书写margin-right:auto; 右侧自动充满，盒子会在左侧出现。如果左右都自动充满，盒子自然就居中了。

当然，在一些公司盒子水平居中会直接书写：

margin-left:atuo;

margin-right:auto;

这样让左右都自动填充，与题目中的写法道理一样，但是可以省略解析上下 0 的步骤。

总结：

让盒子本身居中：margin:0 auto；

让盒子内容居中：text-align:center;

### 1.3 非标准流居中

left:50%;

margin-left:-100px;

原因：脱标后margin:0 auto不起作用，直接书写left数值会影响适配。

解析：首先子盒子走父盒子宽度的一半 ，然后子盒子往回走自己宽度的一半（-100px代表的是方向）

## 定位盒子剧中的对其

加了定位和浮动的盒子，margin:0 auto就完全失效了。

居中原理：定位的盒子 left:50% ，然后再往左走自己宽度的一半。

left:50%;

margin-left:-宽度一半

## 2 内外边距误区

### 2.1 平级盒子外边距合并

有这样两个在一起的div：

<body>  
 <div id="div1"></div>  
 <div id="div2"></div>  
</body>

指定外边距：

<style>  
 #div1 {  
 width: 200px;  
 height: 200px;  
 background: aqua;  
 margin-bottom: 100px;  
 }  
 #div2 {  
 width: 200px;  
 height: 200px;  
 background: yellowgreen;  
 margin-top: 100px;  
 }  
</style>

此时div1和div2的边距并不是200px，而时100px，这是浏览器本身的问题，一般开发中直接避免同时使用外边距即可。

### 2.2 父子盒子外边距塌陷

## 6 外边距塌陷

嵌套的盒子，直接给子盒子设置外边距

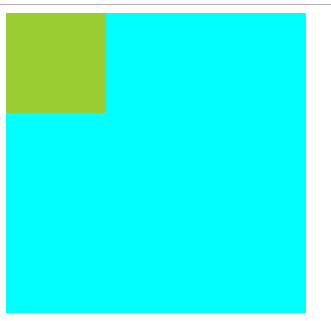
|  |
| --- |
| <**style**>  .**father**{  **width**: 500**px**;  **height**:500**px**;  **background-color**: **red**;  }  .**son**{  **width**: 300**px**;  **height**: 300**px**;  **background-color**: **green**;  **margin-top**: 100**px**; *<!--* **子盒子理应向下移动100px，父子却一起塌陷** *-->* }  </**style**>   <**div class="father"**>  <**div class="son"**></**div**>  </**div**> |

解决方案：

父盒子设置边框 或者 父盒子设置overflow:hidden（会触发bfc）

<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <style>  
 #div1 {  
 width: 500px;  
 height: 500px;  
 background: aqua;  
 }  
 #div2 {  
 width: 200px;  
 height: 200px;  
 background: yellowgreen;  
 }  
 </style>  
</head>  
<body>  
 <div id="div1">  
 <div id="div2"></div>  
 </div>  
</body>

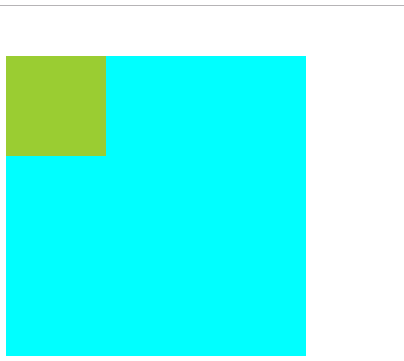
如图所示：



开发中我们经常需要将子盒子div2往下移动，可以修改父盒子的内边距或者修改子盒子的外边距。如果采用子盒子的外边距来做：

#div2 {  
 width: 100px;  
 height: 100px;  
 margin-top: 50px;  
 background: yellowgreen;  
}

此时，并不会出现预料中的子盒子往下移动，而是父子盒子一起往下移动：



注意：外边距塌陷只会存在于垂直外边距上。

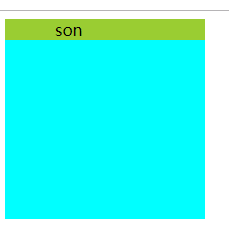
解决方案：

方案一：给父盒子指定1像素的上边框或者内边距

方案二：给父盒子添加overflow:hidden

### 2.3 子盒子内边距不撑开父盒子

很多时候，我们需要如图所示的场景：子盒子son内的内容往右移动一点。



<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <style>  
 #father {  
 width: 200px;  
 height: 200px;  
 background: aqua;  
 }  
 #son {  
 padding-left: 50px;  
 background: yellowgreen;  
 }  
 </style>  
</head>  
<body>  
 <div id="father">  
 <div id="son">son</div>  
 </div>  
</body>

上述问题中，子盒子没有宽度，默认等于父盒子宽度，但是又想染子盒子的文字往右移动一点。有两种方案：

方案一：给父盒子设置内边距，但是这时候需要额外修改父盒子大小；（因为给了宽度）

方案二：给子盒子设置内边距，此时内边距不会撑开父子盒子（因为没有给定宽度，如果给定了子盒子宽度，会撑开子盒子）

## 3 布局

### 3.1 版心



### 3.2 通栏

不设宽度的div很容易制作通栏，但是很多场景下，通栏时，内部的文字图片内容仍然是和版心位置一致的，如下导航是通栏的，但是导航内的内容仍然和版心一致：



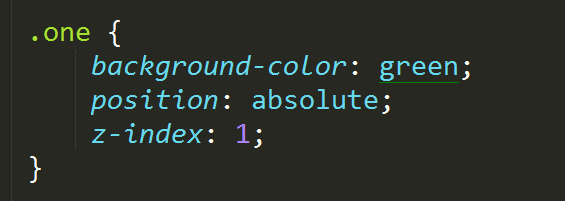
实现方案：导航通栏，导航内部设置一个盒子和版心一致。

## 4 定位盒子的层级关系z-index

后定位的盒子的层级要高于前面定位的盒子的层级，使用z-index设置定位盒子之间的层级关系，z-index可以取负数。

只有定位的盒子（除了static）才有z-index；

如果都是绝对定位，他们默认的z-index都是0；



## 3规避脱标流

浮动、绝对定位、固定定位都会造成脱标，但是  
 1 浮动可以清除浮动影响；  
 2 固定定位在浏览器上不会随着屏幕滚动而动  
 3 绝对定位会以有相对定位的父级左上角为原点

在网页布局中，布局优先标准流，然后考虑浮动和定位。

如果要让一个元素实现模式转化： 优先使用display

如果想让一个块级元素移动到另一侧： margin-left: auto;

## 4 vertical-align

图片默认和文字以底线对其，如图所示：



使用vertical-align:bottom可以底线对其。

vertical-align用来设置元素的垂直对其方式，默认值是baseline，middle可以实现文字图片垂直居中显示，使用vertical-align起作用的元素只有：行内块元素和table。

|  |
| --- |
| <style type="text/css">  img.top {vertical-align:text-top}  img.bottom {vertical-align:text-bottom}  </style>  </head>  <body>  <p>  这是一幅<img class="top" border="0" src="/i/eg\_cute.gif" />位于段落中的图像。  </p>  <p>  这是一幅<img class="bottom" border="0" src="/i/eg\_cute.gif" />位于段落中的图像。  </p> |

## 5 CSS可见性

overflow:hidden 将容器中超出部分隐藏

display: none; 将页面中的元素进行隐藏，不占位置（block为显示）

visibility:hidden 隐藏盒子，占据位置

opacity:0 隐藏盒子，占据位置

position/top/left/...-999px 隐藏盒子，而且占位置。

使用text-indent（文本缩进）可以实现内容移除效果，

## 6 鼠标样式

cursor: pointer 鼠标变成小手

cursor: default; 小白

cursor : move; 移动

cursor : text ; 文本输入

## 7背景半透明

CSS3中的背景半透明：background:rgba(0,0,0,0.5)

opacity: 0.5; 也可以让盒子半透明，但是里面的内容也半透明

## 8 文字居中与大小

文字居中的办法：

1. 给文字设置标签，然后用定位。（比较精准，但是麻烦）
2. Text-align: center; (方便，但是不能精准控制)
3. 给定两边padding或者margin。（比较方便，精准控制）

## 9 JS关闭a链接跳转

<a href="javascript:;">**1111111111**</a>

<a href="javascript:void(0)">**1111**</a>

## 10 权重问题

left比right，权重高。有left又有right的时候，执行left的值。

top比bottom，权重高。有top又有bottom的时候，执行top的值。

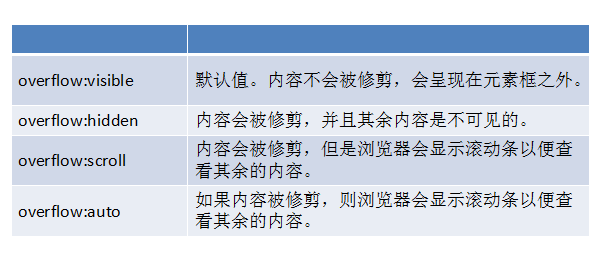
## 11 列表样式取消

**ul**{ **list-style**: **none**;}

## 12 表格单元格合并

border-collapse: collapse;

## 13 overfllow



## 14 精灵图

精灵图是一种处理网页背景图像的方式（背景图片），可以给一个元素将精灵图设置为背景图片，使用坐标找到指定的背景图片，减少请求压力

浏览器坐标系：圆点往右为正，圆点往上移动为负

## 15 普通图片与背景图片

<img src=””> 我们经常用来做如 产品展示 这样的图片。而背景图通常用于小图标背景或者超大背景图。



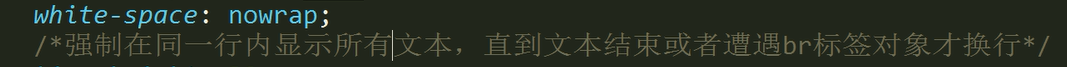
## 17 盒子宽高计算



## 18 inline-block与float

inline-block也可以实现类似float的效果，但是同一行的元素间有间隔。

## 19 其他



滑动门：

案例：导航中两边的背景随着文字的增多而不变



