

CSS居中布局问题

窦连军 @八月虎baidu

北京乐美无限科技有限公司

水平居中

水平居中 `text-align:center` 与 `margin: 0 auto`

- 行内元素水平居中，父盒子上用`text-align:center`;
- 块级元素水平居中： 自身设置`width`为定值、`margin: 0 auto`;



在浏览器窗口中实现水平居中

我们已经知道如何在窗口中水平居中了：

1. 设置 *body* 为 `text-align: center` 属性，令 *行内* 元素水平居中；可以使用 `display: inline-block` 将块级元素转变为行内元素，从而实现居中。
2. 创建一个 *div* 容器，将左右外边距 *margins* 设为 *auto*，再给该 *div* 指定一个 *width* 属性，可令块级元素水平居中。可使用 `display: block` 将行内元素转变为块级元素，从而实现居中。

水平居中的注意事项

有两种情况需要说明：

1. 要混淆操作对象。*text-align*属性作用对象为容器元素中的文本\图片之类的行内元素，能够使该容器内的行内元素内容实现水平对齐。不应将该属性应用于行内元素上。*text-align*不会使元素自身进行水平对齐。
2. 我们设置图片标签 {margin:0 auto;}，我们就犯了一个小错误，*img*属于行内元素，不可以设置图片标签的 *margin*属性，如果你一定想要设置，那么首先要将它的属性转变为块元素，如下面的代码：
img {display:block; margin:0 auto;}

垂直居中


单行文本的垂直居中

- 设置line-height大于文本字体尺寸、并设置height与line-height相等。

居中的文字

```
<div id= “child” >居中的文字</div>
```

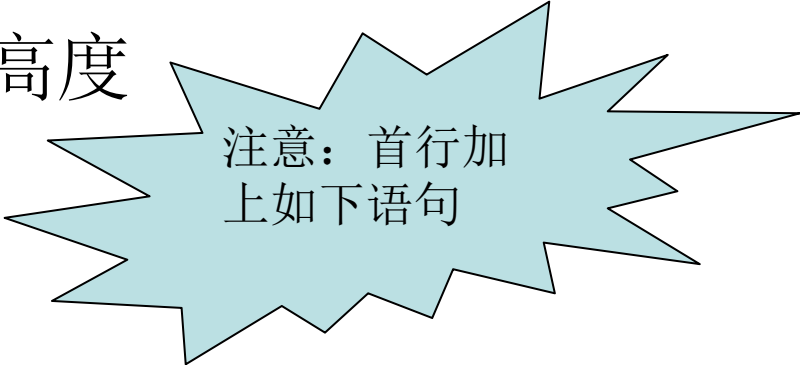
```
#child {  
    line-height: 200px;  
    height: 200px;  
}
```



不适用
多行情
况!

简单图片的垂直居中

- 设置height与line-height相等。
- 设置line-height大于图片的高度
- 设置vertical-align: middle;



注意：首行加上如下语句

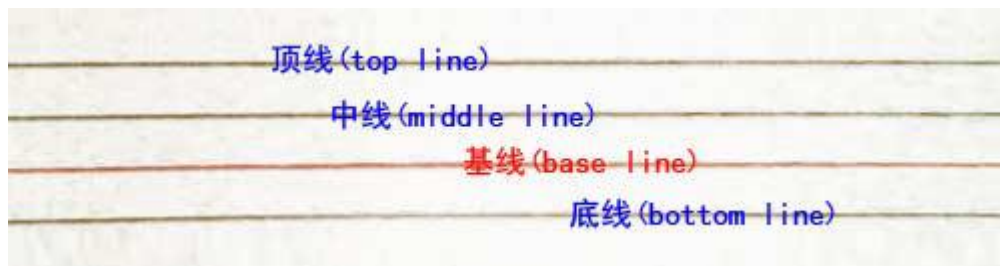
<!DOCTYPE html>

```
<div id="parent">  
     </div>
```

```
#parent { line-height: 200px; height: 200px}  
#parent img { vertical-align: middle; }
```


垂直居中: vertical-align

- vertical-align 只能应用于行内元素和 table-cell 的元素。可以使用 display: table-cell 将任何元素表现形式转变为类似 td 的元素性质。
- 为让同行的 inline 元素按中间线对齐，可以给各个 inline 元素都加上 vertical-align: middle;




刚学英语的时使用的那个英语本子每行有四条线

多行文本居中

```
<div class="wrapper"> Drupal ipsum  
dolor sit modula webmaster hook  
panels block panels html5 field  
alter overlay!! </div>
```

```
<style type="text/css">  
.wrapper {  
  display: table-cell;  
  width: 200px;  
  height: 150px;  
  text-align: center;  
  vertical-align: middle;  
  border: 1px dotted #656565;  
}  
</style>
```

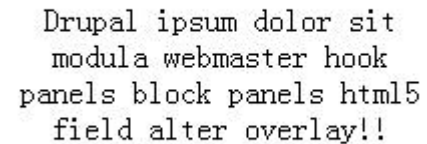


Drupal ipsum dolor sit modula
webmaster hook panels block
panels html5 field alter
overlay!!

多行文本居中（table-cell帶padding）

```
<div class="wrapper"> Drupal ipsum  
dolor sit modula webmaster hook  
panels block panels html5 field  
alter overlay!! </div>
```

```
<style type="text/css">  
.wrapper {  
  display: table-cell;  
  width: 200px;  
  padding: 20px;  
  text-align: center;  
  vertical-align: middle;  
  border: 1px dotted #656565;  
}  
</style>
```



Drupal ipsum dolor sit
modula webmaster hook
panels block panels html5
field alter overlay!!

帶padding修饰,
起到“留白”作用

对任意元素居中

- 方法1: table+tr+td元素方式

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title></title>
</head>
<body>

<table style="width: 100%; height: 200px">
  <tr>
    <td style="text-align: center; vertical-align: middle;">
      任意元素，这里都可以居中显示
    </td>
  </tr>
</table>

</body>
</html>
```

对任意元素居中

•方法2: css—— table display方式

```
<div class="class-a">
  <div class="class-b">
    | 任意元素，这里都可以居中显示
  </div>
</div>
<style type="text/css">
  .class-a {
    display: table;
    width: 100%;
    height: 200px;
  }
  .class-b {
    display: table-cell;
    text-align: center;
    vertical-align: middle;
  }
</style>
```

```
.class-b {
  .....
  display: inline-
block; } /*for old
IE*/
```

对任意元素居中

•方法3： 伪元素方式

```
<style type="text/css">
  /* 该元素的父元素可以是任意高度和宽度 */
  .block {
    text-align: center;
  }

  /* 伪元素：幽灵元素，只是为了帮助居中 */
  .block:before {
    content: '';
    display: inline-block;
    height: 100%;
    vertical-align: middle;
    margin-right: -0.25em; /* 可用于调整空白 */
  }


  /* 需要居中的元素，可以是任意高度和宽度 */
  .centered {
    display: inline-block;
    vertical-align: middle;
    width: 300px;
  }
</style>
<div class="block" style="height: 200px;">
  <div class="centered">
    | 要居中了!
  </div>
</div>
```

对任意元素居中

•方法4：绝对定位和负外边距的使用

```
<div id="parent">  
  <div id="child">Content here</div>  
</div>
```

```
#parent {position: relative;}  
#child {  
  position: absolute;  
  top: 50%; left: 50%;  
  height: 30%; width: 50%;  
  margin: -15% 0 0 -25%; /* 负偏移值为宽高的一半 */  
}
```




适用于知道子元素的宽高情况。

对任意元素居中

•方法5：绝对定位和margin的使用

```
<div id="parent">  
  <div id="child">Content here</div>  
</div>
```

```
#parent {position: relative;}  
#child {  
  position: absolute;  
  top: 0; left: 0; bottom: 0; right: 0;  
  height: 30%; width: 50%;  
  margin: auto;  
}
```



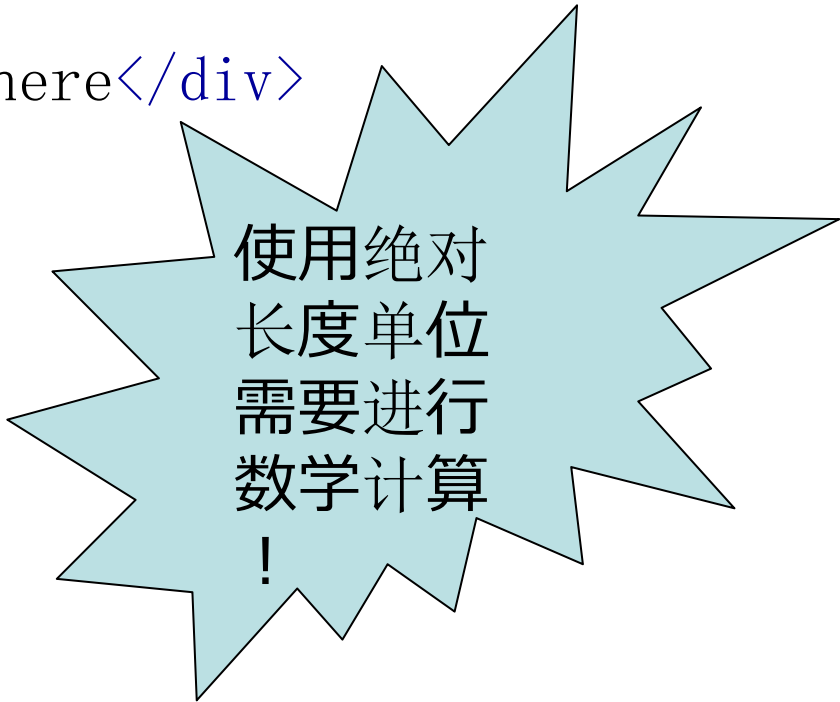
IE7不适用!

对任意元素居中

•方法6：上下内边距的使用

```
<div id="parent">  
  <div id="child">Content here</div>  
</div>
```

```
#parent {  
  padding: 5% 0;  
}  
#child {  
  padding: 10% 0;  
}
```



使用绝对
长度单位
需要进行
数学计算
!

对任意元素居中

- 方法7：借助前置的占位元素

```
<div id="parent">  
  <div id="floater"></div>  
  <div id="child">Content here</div>
```

```
</div>
```

```
#parent { height: 250px; }
```

```
#floater {
```

```
  float: left;
```

```
  height: 50%;
```

```
  width: 100%;
```

```
  margin-bottom: -50px; /* 为子元素child的高度一半，负值 */
```

```
}
```

```
#child { clear: both; height: 100px; }
```

对任意元素居中

- 方法8： 借助transform

```
<div id="parent">  
  <div id="child">Content here</div>  
</div>
```

```
#parent {  
  position: relative;  
}  
#child {  
  position: absolute;  
  top: 50%;  
  transform: translateY(-50%);  
}
```

若干注意事项

Body元素的高度问题

注意：body元素的缺省高度不是100%！而是等于内容区实际大小。

可使用如下CSS代码设置body的显示区域为全屏高度：

```
html, body {  
    height: 100%;  
    margin: 0px;  
    padding: 0px;  
}
```

善于使用div

善用div来定义CSS盒子

我们已经看到CSS盒模型的属性可以同样应用于简单的HTML块元素 (p, h1, ...)

然而，我们更常常使用<div> 元素，而且同时连带使用id和class属性。

然后，我们在CSS中，将宽高（width、height）、漂移（float）、外边距（margin）和内边距（padding）及边框属性定义到这些div上。

这种方式给了我们更多的选择来创建和控制元素的布局，而不用再关心其具体的内容。

举例：用 div 来定义CSS 盒子

id example: in the XHTML:

```
<div id="sidebar">  
    <p> blah, blah, blah,  
    blah, blah, blah. </p>  
</div>
```

In the CSS:

```
#sidebar {  
    float: right;  
    width: 300px;  
    margin-left: 25px;  
}
```

class example: in the XHTML:

```
<div class="sidebar">  
    <p> blah, blah, blah,  
    blah.</p>  
  
</div>
```

In the CSS:

```
.sidebar {  
    float: right;  
    width: 300px;  
    margin-left: 25px;  
}
```