

水平垂直居中

水平居中-行内元素

如果被设置元素为文本、图片等行内元素时，通过给父元素设置 `text-align:center` 来实现。

html代码：

```
<body>
  <div class="txtCenter">我想要在父容器中水平居中显示。</div>
</body>
```

css代码：

```
<style>
  .txtCenter{
    text-align:center;
  }
</style>
```

上面代码中，div是文本的父元素。

水平居中-定宽块状元素 (通过css样式设置了宽度，块状元素的宽度不随浏览器窗口大小而改变)

满足定宽和块状两个条件的元素是可以通过设置 “**左右margin**” 值为 “**auto**” 来实现居中的。

html代码：

```
<body>
  <div>我是定宽块状元素，哈哈，我要水平居中显示。</div>
</body>
```

css代码：

```
<style>
div{
  border:1px solid red;/*为了显示居中效果明显为 div 设置了边框*/
  width:200px;/*定宽*/
  margin:20px auto;/* margin-left 与 margin-right 设置为 auto */
}
</style>
```

注意：元素的“上下 margin” 可以随意设置。

水平居中-不定宽块状元素（没有设置其宽度或者宽度表示为百分数，块状元素的宽度会随着该元素的父元素的大小而发生改变）

方法（一）

在元素外面加入 `table`（包括**tbody**、**tr**、**td**）标签。

为这个 table 设置 “**左右 margin auto**”（和定宽块状元素的方法一样）。

加入table标签是利用table标签的长度自适应性——即不定义其长度也不默认父元素body的长度（table其长度根据其内文本长度决定），因此可以看做一个定宽度块元素，然后再利用定宽度块状居中的margin的方法，使其水平居中。

html代码：

```
<div>
  <table>
    <tbody>
      <tr><td>
        <ul>
          <li>我是第一行文本</li>
          <li>我是第二行文本</li>
          <li>我是第三行文本</li>
        </ul>
      </td></tr>
    </tbody>
  </table>
</div>
```

css代码:

```
<style>
table{
  border:1px solid;
  margin:0 auto;
}
</style>
```

方法（二）

设置 **display: inline**（设置为 行内元素 显示）。

给父元素设置 **text-align:center** 来实现居中效果。

优势：不用增加无语义标签，

劣势：将块状元素变成了行内元素，少了一些功能，比如设定长度值。

方法（三）

给父元素设置 **float:left**，然后给父元素设置 **position:relative** 和 **left:50%**。

子元素设置 **position:relative** 和 **left: -50%** 来实现水平居中。

可以这样理解：

把父元素浮动到左边，同时向右偏移50%，相当于把父元素的左边缘对齐到整行的中间。

再把子元素向左偏移50%，这样刚好就居中了。

html代码:

```
<body>
<div class="container">
  <ul>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
  </ul>
</div>
</body>
```

css代码:

```

<style>
.container{
    float:left;
    position:relative;
    left:50%
}

.container ul{
    list-style:none;
    margin:0;
    padding:0;

    position:relative;
    left:-50%;
}

.container li{
    float:left;
    display:inline;
    margin-right:8px;
}
</style>

```

垂直居中-父元素高度确定的单行文本

通过设置父元素的 **height**（高度）和 line-height（行高）一致来实现。

line-height 与 font-size 的计算值之差，在 CSS 中成为“行间距”。分为两半，分别加到一个文本行内容的顶部和底部。

弊端：当文字内容的长度大于块的宽时，就有内容脱离了块。

html代码：

```

<div class="container">
    hi, imooc!
</div>

```

css代码：

```

<style>
.container{
    height:100px;
    line-height:100px;
    background:#999;
}
</style>

```

垂直居中-父元素高度确定的多行文本、图片等

方法（一）

使用插入 **table**（包括**tbody**、**tr**、**td**）标签，同时设置 **vertical-align: middle**。

父元素设置垂直居中属性 **vertical-align**，对 **inline-block** 类型的子元素都有用。

因为 td 标签默认情况下就设置了 vertical-align 为 middle，所以就不需要显式地设置了。

html代码:

```
<body>
<table><tbody><tr><td class="wrap">
<div>
    <p>看我是否可以居中。</p>
</div>
</td></tr></tbody></table>
</body>
```

css代码:

```
table td{
    height:500px;
    background:#ccc
}
```

方法（二）

在 chrome、firefox 及 IE8 以上的浏览器下可以设置块级元素 **display:table-cell**（设置为表格单元显示），激活 vertical-align 属性。

html代码:

```
<div class="container">
    <div>
        <p>看我是否可以居中。</p>
        <p>看我是否可以居中。</p>
        <p>看我是否可以居中。</p>
    </div>
</div>
```

css代码:

```
<style>
.container{
    height:300px;
    background:#ccc;
    display:table-cell;/*IE8以上及Chrome、Firefox*/
    vertical-align:middle;/*IE8以上及Chrome、Firefox*/
}
</style>
```

好处: 不用添加多余的无意义标签。

缺点: 兼容性不好, 不兼容 IE6、7。而且这样修改display的block变成了table-cell, 破坏了原有的块状元素的性质。

水平垂直居中-绝对定位元素

先将绝对定位元素设置 **top:50%; left:50%;** 使其左上角在父元素中居中。

再将 **margin-top, margin-left** 设置为负的高度和宽度的一半。

```
<div class="absolute"></div>
.absolute{
    position: absolute;
    top:50%;
    left:50%;
    height: 200px;
```

```
width: 200px;
background-color: #ccc;
margin-top: -100px;
margin-left: -100px;
}
```

缺点：必须知道这个觉得定位元素宽度和高度才能并设置负margin值。