

CSS 布局 - 水平 & 垂直对齐

水平 & 垂直居中对齐

元素居中对齐

要水平居中对齐一个元素(如 <div>), 可以使用 `margin: auto;`。

设置到元素的宽度将防止它溢出到容器的边缘。

元素通过指定宽度, 并将两边的空外边距平均分配:

div 元素是居中的

实例

```
.center {  
margin: auto;  
width: 50%;  
border: 3px solid green;  
padding: 10px;  
}
```

[尝试一下 »](#)

注意: 如果没有设置 `width` 属性(或者设置 100%), 居中对齐将不起作用。

文本居中对齐

如果仅仅是为了文本在元素内居中对齐, 可以使用 `text-align: center;`

文本居中对齐

实例

```
.center {  
text-align: center;  
border: 3px solid green;  
}
```

[尝试一下 »](#)

提示: 更多文本对齐实例, 请参阅 [CSS 文本](#) 章节。

图片居中对齐

要让图片居中对齐, 可以使用 `margin: auto;` 并将它放到 **块** 元素中:



实例

```
img {  
  display: block;  
  margin: auto;  
  width: 40%;  
}
```

尝试一下 »

左右对齐 - 使用定位方式

我们可以使用 `position: absolute;` 属性来对齐元素:

菜鸟教程 -- 学的不仅是技术，更是梦想！！

实例

```
.right {  
  position: absolute;  
  right: 0px;  
  width: 300px;  
  border: 3px solid #73AD21;  
  padding: 10px;  
}
```

尝试一下 »

注释：绝对定位元素会被从正常流中删除，并且能够交叠元素。

提示：当使用 `position` 来对齐元素时，通常 `<body>` 元素会设置 `margin` 和 `padding`。这样可以避免在不同的浏览器中出现可见的差异。

当使用 `position` 属性时，IE8 以及更早的版本存在一个问题。如果容器元素（在我们的案例中是 `<div class="container">`）设置了指定的宽度，并且省略了 `!DOCTYPE` 声明，那么 IE8 以及更早的版本会在右侧增加 17px 的外边距。这似乎是为滚动条预留的空间。当使用 `position` 属性时，请始终设置 `!DOCTYPE` 声明：

实例

```
body {
margin: 0;
padding: 0;
}
.container {
position: relative;
width: 100%;
}
.right {
position: absolute;
right: 0px;
width: 300px;
background-color: #b0e0e6;
}
```

尝试一下 »

左右对齐 - 使用 float 方式

我们也可以使用 `float` 属性来对齐元素：

实例

```
.right {
float: right;
width: 300px;
border: 3px solid #73AD21;
padding: 10px;
}
```

尝试一下 »

当像这样对齐元素时，对 `<body>` 元素的外边距和内边距进行预定义是一个好主意。这样可以避免在不同的浏览器中出现可见的差异。

注意：如果子元素的高度大于父元素，且子元素设置了浮动，那么子元素将溢出，这时候你可以使用 "`clear` `fix` (清除浮动)" 来解决该问题。

我们可以在父元素上添加 `overflow: auto;` 来解决子元素溢出的问题：

实例

```
.clearfix {  
  overflow: auto;  
}
```

[尝试一下 »](#)

当使用 float 属性时，IE8 以及更早的版本存在一个问题。如果省略 !DOCTYPE 声明，那么 IE8 以及更早的版本会在右侧增加 17px 的外边距。这似乎是为滚动条预留的空间。当使用 float 属性时，请始终设置 !DOCTYPE 声明：

实例

```
body {  
  margin: 0;  
  padding: 0;  
}  
.right {  
  float: right;  
  width: 300px;  
  background-color: #b0e0e6;  
}
```

[尝试一下 »](#)

垂直居中对齐 - 使用 padding

CSS 中有很多方式可以实现垂直居中对齐。一个简单的方式就是头部顶部使用 padding：

我是垂直居中。

实例

```
.center {  
  padding: 70px 0;  
  border: 3px solid green;  
}
```

[尝试一下 »](#)

如果要水平和垂直都居中，可以使用 padding 和 text-align: center：

我是水平和垂直都居中的。

实例

```
.center {  
padding: 70px 0;  
border: 3px solid green;  
text-align: center;  
}
```

尝试一下 »

垂直居中 - 使用 line-height

我是垂直居中的。

实例

```
.center {  
line-height: 200px;  
height: 200px;  
border: 3px solid green;  
text-align: center;  
}  
/* 如果文本有多行, 添加以下代码: */  
.center p {  
line-height: 1.5;  
display: inline-block;  
vertical-align: middle;  
}
```

尝试一下 »

垂直居中 - 使用 position 和 transform

除了使用 `padding` 和 `line-height` 属性外,我们还可以使用 `transform` 属性来设置垂直居中:

实例

```
.center {  
height: 200px;  
position: relative;  
border: 3px solid green;  
}  
.center p {  
margin: 0;  
position: absolute;  
top: 50%;  
left: 50%;  
transform: translate(-50%, -50%);  
}
```

[尝试一下 »](#)

提示: 更多 transform 属性内容可以参阅 [2D 翻转章节](#)。

更多实例

[CSS 使用 margin 让 div 居中对齐](#)

[CSS 使用绝对定位 让 div 右对齐](#)

[← CSS Float \(浮动 \)](#)[CSS 伪类 →](#)

[点我分享笔记](#)