CSS之垂直居中问题探讨

在前端开发过程中，盒子居中是常常用到的。其中 ，居中又可以分为水平居中和垂直居中。水平居中是比较容易的，直接设置元素的margin: 0 auto就可以实现。但是垂直居中相对来说是比较复杂一些的。下面我们一起来讨论一下实现垂直居中的方法。

1. 文本居中问题
2. 如果一个容器中只有一行文字，对它实现居中相对比较简单，我们只需要设置它的实际高度height和所在行的高度line-height相等即可。如：

Div{height:25px;line-height:25px}

1. 多行文本垂直居中设置上下的padding值一样即可，如：

Div{padding:25px}

1. 多行文本固定高度居中

Div{display:table-cell;vertical-align:middle;height:200px;}

注意：外部div不能使用浮动，不能使用绝对定位或固定定位，总之不能脱离文档流。

1. 多行文本外部容器高度未知的情况下：

Div{

Position:relative;

Height:200px;

Div{

Position:absolute;

Top:0;

Left:0;

Right:0;

Bottom:0;

Width:100%;

Margin:auto;

Display:table;

}

}

1. 块级元素垂直水平居中
2. 对于已知高度的元素，采用绝对定位。（兼容性最好）

Div{

Position:relative;

Width:400px;

Height:200px;

Div{

Position:absolute;

Height:100px;

Top:50%;

Margin-top:-50px;

}

}

1. 对于不能使用绝对定位的元素，可以采用如下方式：

<div id="wrapper">

    <div id="null"></div>

    <div id="content">不能使用绝对定位</div>

</div>

#wrapper{

Width:400px;

Height:200px;

Overflow:hidden;

Text-align:center;

#null{

Width:100%;

Height:50%;

}

#content{

Height:100px;

Margin-top:-50px;

}

}

思路是：用一个块级元素，设置已知大小，在让其高度达到父级容器的一半大小，再把要居中的元素往上提半个高度。和上一个方法同理。

1. **垂直居中一张图片（行内元素）**

<div id="wrapper">

    <img src="http://img0.bdstatic.com/img/image/2016ss1.jpg" alt="">

    <span id="block"></span>

</div>

#wrapper{

Width:400px;

Height:200px;

Text-align:center;

Img{

Vertical-align:middle;

}

#block{

Line-height:200px;

}

}

注意：里面多了一个span标签，是因为vertical-align这个属性是相对于兄弟级行高来定位的，并且只对行内元素有效，包括行内块元素，所以加一个span来撑开父级的行高，以此来居中。

1. 子级元素未知高度的情况：

<div id="wrapper">

    <div id="content">内容</div>

</div>

#wrapper{

Display:table;

Width:400px;

Height:200px;

Text-align:center;

#content{

Display:table-cell;

Vertical-align:middle;

}

}

1. 绝对定位居中法：

<div id="wrapper">

<div id="content"></div>

</div>

#wrapper{

Position:relative;

Width:400px;

Height:200px;

#content{

Position:absolute;

Top:0;

Left:0;

Right:0;

Bottom:0;

Margin:auto;

}

}

1. 绝对定位加上transform反向偏移

<div id="wrapper">

<div id="content"></div>

</div>

#wrapper{

Position:relative;

Width:400px;

Height:200px;

#content{

Position:absolute;

Margin:auto;

Top:50%;

Left:50%;

Width:200px;

Height:100px;

Tranform:translate(-50%,-50%);

}

}

1. 弹性布局

<div id="wrapper">

<div id="content"></div>

</div>

#wrapper{

Position:relative;

Width:400px;

Height:200px;

Display:flex;

Align-items:center;

Justify-content:center;

#content{

Width:200px;

Height:100px;

}

}

flexbox布局。此乃布局终极大法，专治各种布局定位难题！优点：能解决各种排列布局问题，实现方式符合人类认知。缺点：PC端某些旧浏览器支持度不高。

纯CSS实现居中的各种方案就总结到这里了。