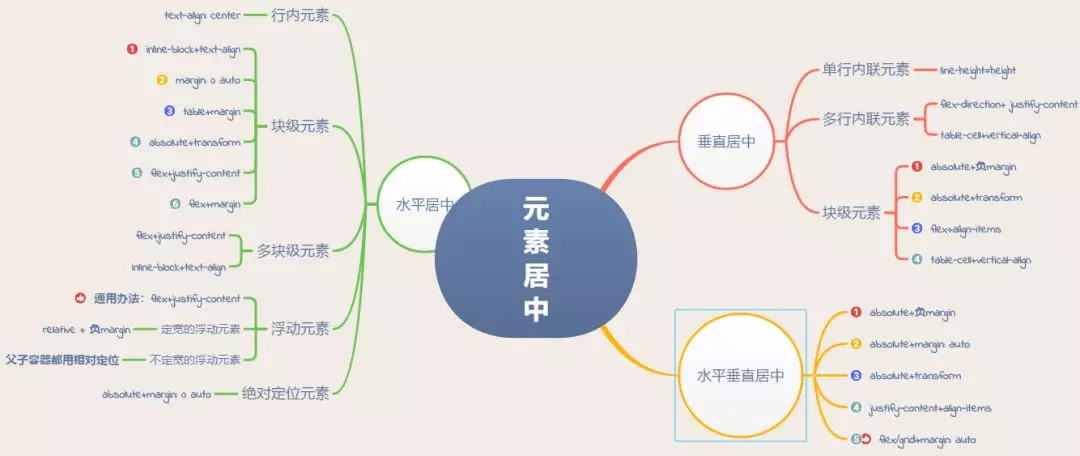
原文档地址：https://mp.weixin.qq.com/s?\_\_biz=MzAxODE2MjM1MA==&mid=2651556051&idx=1&sn=271aedfa33ecf221f29098c0d3ba91b1&chksm=80255f12b752d6043577e730c5b41b7581b65740ebf54291e78e92e0602d3ac5c048aa74cad5&mpshare=1&scene=23&srcid=#rd

前言

本文主要介绍水平居中，垂直居中，还有水平垂直居中各种办法，思维导图如下：



一、水平居中

**1.行内元素水平居中**

**利用 text-align: center 可以实现在块级元素内部的行内元素水平居中**。此方法对inline、inline-block、inline-table和inline-flex元素水平居中都有效。

.parent{//在父容器设置

    text-align:center;

}

此外，如果块级元素内部包着也是一个块级元素，**我们可以先将其由块级元素改变为行内块元素，再通过设置行内块元素居中以达到水平居中**。

<div class="parent">

  <div class="child">Demo</div>

</div>

<style>

  .parent{

    text-align:center;

  }

  .child {

    display: inline-block;

  }

</style>

**2.块级元素的水平居中**

这种情形可以有多种实现方式，下面我们详细介绍:

**①将该块级元素左右外边距margin-left和margin-right设置为auto**

.child{

    width: 100px;//确保该块级元素定宽

    margin:0 auto;

}

**②使用table+margin**

**先将子元素设置为块级表格来显示（类似），再将其设置水平居中**

**display:table在表现上类似block元素，但是宽度为内容宽。**

<div class="parent">

  <div class="child">Demo</div>

</div>

<style>

  .child {

    display: table;

    margin: 0 auto;

  }

</style>

**③使用absolute+transform**

**先将父元素设置为相对定位，再将子元素设置为绝对定位，向右移动子元素，移动距离为父容器的一半，最后通过向左移动子元素的一半宽度以达到水平居中。**

<div class="parent">

  <div class="child">Demo</div>

</div>

<style>

  .child {

    position:absolute;

    left:50%;

    transform:translateX(-50%);

  }

  .parent {

    position:relative;

  }

</style>

**不过transform属于css3内容，兼容性存在一定问题，高版本浏览器需要添加一些前缀**。

**④使用flex+justify-content**

**通过CSS3中的布局利器flex中的justify-content属性来达到水平居中。**

<div class="parent">

  <div class="child">Demo</div>

</div>

<style>

  .parent {

    display: flex;

    justify-content:center;

  }

</style>

**⑤使用flex+margin**

**通过flex将父容器设置为为Flex布局，再设置子元素居中。**

<div class="parent">

  <div class="child">Demo</div>

</div>

<style>

  .parent {

    display: flex;

  }

  .child {

    margin:0 auto;

  }

</style>

**3.多块级元素水平居中**

**①利用flex布局**

**利用弹性布局(flex)，实现水平居中，其中justify-content 用于设置弹性盒子元素在主轴（默认横轴）方向上的对齐方式**，本例中设置子元素水平居中显示。

#container {

    display: flex;

    justify-content: center;

}

**源代码演示 http://www.42du.cn/run/60**

**②利用inline-block**

**将要水平排列的块状元素设为display:inline-block，然后在父级元素上设置text-align:center，达到与上面的行内元素的水平居中一样的效果**。

.container {

   text-align: center;

}

.inline-block {

   display: inline-block;

}

**源代码演示 http://www.42du.cn/run/59**

**4.浮动元素水平居中**

* 对于定宽的浮动元素，通过子元素设置relative + 负margin
* 对于不定宽的浮动元素，父子容器都用相对定位
* 通用方法(不管是定宽还是不定宽)：flex布局

**①定宽的非浮动元素**

**通过子元素设置relative + 负margin,原理见下图：**

**注意：样式设置在浮动元素本身**

.child {

   position:relative;

   left:50%;

   margin-left:-250px;

}

<div class="parent">

  <span class="child" style="float: left;width: 500px;">我是要居中的浮动元素</span>

</div>

**②不定宽的浮动元素**

通过父子容器都相对定位，偏移位移见下图：  
  
**注意：要清除浮动，给外部元素加上float。这里的父元素就是外部元素**

<div class="box">

    <p>我是浮动的</p>

    <p>我也是居中的</p>

</div>

.box{

    float:left;

    position:relative;

    left:50%;

}

p{

    float:left;

    position:relative;

    right:50%;

}

**③通用办法flex布局(不管是定宽还是不定宽)**

**利用弹性布局(flex)的justify-content属性，实现水平居中**。

.parent {

    display:flex;

    justify-content:center;

}

.chlid{

    float: left;

    width: 200px;//有无宽度不影响居中

}

<div class="parent">

  <span class="chlid">我是要居中的浮动元素</span>

</div>

**5.绝对定位元素水平居中**

这种方式非常独特，**通过子元素绝对定位，外加margin: 0 auto来实现**。

<div class="parent">

    <div class="child">让绝对定位的元素水平居中对齐。</div>

</div>

  .parent{

        position:relative;

    }

   .child{

         position: absolute; /\*绝对定位\*/

         width: 200px;

         height:100px;

         background: yellow;

         margin: 0 auto; /\*水平居中\*/

         left: 0; /\*此处不能省略，且为0\*/

         right: 0;/\*此处不能省略，且为0\*/

    }

二、垂直居中

**1.单行内联元素垂直居中**

<div id="box">

     <span>单行内联元素垂直居中。</span>。

</div>

<style>

 #box {

    height: 120px;

    line-height: 120px;

    border: 2px dashed #f69c55;

    }

</style>

**2.多行内联元素垂直居中**

**①利用flex布局（flex）**

**利用flex布局实现垂直居中，其中flex-direction: column定义主轴方向为纵向**。这种方式在较老的浏览器存在兼容性问题。

<div class="parent">

    <p>Dance like nobody is watching, code like everybody is.

    Dance like nobody is watching, code like everybody is.

    Dance like nobody is watching, code like everybody is.</p>

</div>

<style>

    .parent {

        height: 140px;

        display: flex;

        flex-direction: column;

        justify-content: center;

        border: 2px dashed #f69c55;

    }

</style>

**②利用表布局（table）**

**利用表布局的vertical-align: middle可以实现子元素的垂直居中**

<div class="parent">

    <p class="child">The more technology you learn, the more you realize how little you know.

    The more technology you learn, the more you realize how little you know.

    The more technology you learn, the more you realize how little you know.</p>

</div>

 <style>

    .parent {

        display: table;

        height: 140px;

        border: 2px dashed #f69c55;

    }

    .child {

        display: table-cell;

        vertical-align: middle;

    }

</style>

**3 块级元素垂直居中**

**①使用absolute+负margin(已知高度宽度)**

**通过绝对定位元素距离顶部50%，并设置margin-top向上偏移元素高度的一半，就可以实现了**。

<div class="parent">

    <div class="child">固定高度的块级元素垂直居中。</div>

</div>

.parent {

   position: relative;

}

.child {

   position: absolute;

   top: 50%;

   height: 100px;

   margin-top: -50px;

}

**②使用absolute+transform**

**当垂直居中的元素的高度和宽度未知时，可以借助CSS3中的transform属性向Y轴反向偏移50%的方法实现垂直居中**。但是部分浏览器存在兼容性的问题。

<div class="parent">

    <div class="child">未知高度的块级元素垂直居中。</div>

</div>

.parent {

   position: relative;

}

.child {

   position: absolute;

   top: 50%;

   transform: translateY(-50%);

}

**③使用flex+align-items**

**通过设置flex布局中的属性align-items，使子元素垂直居中**。

<div class="parent">

    <div class="child">未知高度的块级元素垂直居中。</div>

</div>

.parent {

    display:flex;

    align-items:center;

}

**④使用table-cell+vertical-align**

**通过将父元素转化为一个表格单元格显示（类似 <td> 和 <th>），再通过设置 vertical-align属性，使表格单元格内容垂直居中。**

<div class="parent">

  <div class="child">Demo</div>

</div>

<style>

  .parent {

    display: table-cell;

    vertical-align: middle;

  }

</style>

三、水平垂直居中

这种情形也是有多种实现方式，接下去我们娓娓道来：

**方法1：绝对定位与负边距实现（已知高度宽度）**

**这种方式需要知道被垂直居中元素的高和宽，才能计算出margin值，兼容所有浏览器**。

// css部分

#container {

   position: relative;

}

#center {

   position: absolute;

   top: 50%;

   left: 50%;

   margin: -50px 0 0 -50px;

}

// html部分(这部分不做变化,下面例子直接共用)

<body>

  <div id='container'>

    <div id='center' style="width: 100px;height: 100px;background-color: #666">center</div>

  </div>

</body>

**方法2：绝对定位与margin:auto（已知高度宽度）**

**这种方式无需知道被垂直居中元素的高和宽，但不能兼容低版本的IE浏览器。**

#container {

   position: relative;

   height:100px;//必须有个高度

}

#center {

   position: absolute;

   top: 0;

   left: 0;

   right: 0;

   bottom: 0;

   margin: auto;//注意此处的写法

}

**方法3：绝对定位+CSS3(未知元素的高宽)**

**利用Css3的transform，可以轻松的在未知元素的高宽的情况下实现元素的垂直居中**。  
CSS3的transform固然好用，但在项目的实际运用中必须考虑兼容问题，大量的hack代码可能会导致得不偿失。

#container {

  position: relative;

}

#center {

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

}

**方法4：flex布局**

**利用flex布局，其中justify-content 用于设置或检索弹性盒子元素在主轴（横轴）方向上的对齐方式；而align-items属性定义flex子项在flex容器的当前行的侧轴（纵轴）方向上的对齐方式。不能兼容低版本的IE浏览器。**

#container {//直接在父容器设置即可

   height: 100vh;//必须有高度

   display: flex;

   justify-content: center;

   align-items: center;

 }

**方法5：flex/grid与margin:auto(最简单写法)**

**容器元素设为 flex 布局或是grid布局，子元素只要写 margin: auto 即可,不能兼容低版本的IE浏览器。**

#container {

  height: 100vh;//必须有高度

  display: grid;

}

#center {

  margin: auto;

}