

湖南石化职院教案

日 期	2019年 11月 2日	周次	第 十一周	时数	2课时
教学单元	CSS+DIV布局定位方法1				
教学目标	知识目标：CSS布局概念 固定宽度布局 圆角框				
	技能目标：CSS布局概念 固定宽度布局 圆角框				
	素质目标：熟悉CSS布局概念，会设置固定宽度布局				
教学重点	CSS布局概念 圆角框				
教学难点	CSS布局概念				
教学方法	上课之前通过先讲解知识点目标让学生先预热,然后再进行代码讲解讲解完代码之后让学生自己练习,有不懂的学生再次进行讲解确保每一位学生都可以听懂				
教学手段	再每次讲解完代码之后让学生进行自己练习的过程中,在下面转督促学生去写代码并且看到学生又不会敲或者报错时,了解情况进行讲解最后总结学生所犯的错误				
教学内容及过程					

一. 定位：position: relative/absolute/fixed/static/inherit/sticky

static(默认)：元素出现在正常的流中（但是，若设置会导致 top, bottom, left, right 或者 z-index 声明）。

relative：元素相对于之前正常文档流中的位置发生偏移，并且原先的位置仍然被占据。因此，"left:20" 会向元素的 left 位置添加 20 像素。

absolute：元素不再占有文档流位置，并且相对于包含块进行偏移(所谓的包含块就是最近一级外层元素 position 不为 static 的元素)。

fixed：生成绝对定位的元素（因此该元素也不再占有文档流位置），但是元素相对于浏览器窗口进行定位。

sticky：(这是 css3 新增的属性值)粘性定位，其实，它就相当于 relative 和 fixed 混合。最初会被当作是 relative，相对于原来的位置进行偏移；一旦超过一定阈值之后，会被当成 fixed 定位，相对于视口进行定位。测试代码

偏移量属性：top, left, right, bottom; 我们知道这些属性不会对 static 的元素起到作用。这也就是其最大区别于 margin 属性的地方，margin 主要针对盒模型设置外边距。

二. 尺寸：px, %, rem, em

百分比：百分比的参照物是父元素，50% 相当于父元素 width 的 50%

rem：这个对于复杂的设计图相当有用，它是 html 的 font-size 的大小

em：它虽然也是一个相对的单位，相对于父元素的 font-size，但是，并不常用，主要是计算太麻烦了。

盒模型：(1)标准盒子模型 box-sizing: content-box, width 的长度等于 content 的宽度；(2)box-sizing: border-box，盒子模型的 width=border+padding+content 的总和。

三. 浮动：float: left/right

浮动最开始不是用来布局，主要是用来做文字环绕的。

清楚浮动的两种方法：

overflow: 将父元素的 overflow，设置成 hidden。

after 伪类：对子元素的 after 伪类进行设置。

flex 布局的重要概念如下：

main axis: 水平主轴

cross axis: 垂直的交叉轴

Flex 容器(parent)

```
.parent {  
    display: flex | inline-flex;    //容器被设置为 flex 布局后，其子元素  
    //的 float、clear、vertical-align 的属性将会失效。  
}
```

flex 容器上的 6 个属性

1.1 三种基本方案

多栏布局有三种基本的实现方案：固定宽度、流动、弹性。

固定宽度。布局的大小不会随用户调整浏览器窗口大小而变化，一般是 900 到 1100 像素宽（最常见的是 960 像素）。

流动。布局的大小会随用户调整浏览器窗口大小而变化。（结合 CSS 媒体查询，能够适应最大和最小的屏幕，业界称之为 响应式设计。）

弹性。在浏览器窗口变宽时，不仅布局变宽，而且所有内容元素的大小也会变化。（实现太过复杂，不多介绍。）

1.2 布局高度

多数情况下，布局中结构化元素（乃至任何元素）的高度是 不必或者不应该设定的。

因为保持元素 height 属性的默认值 auto 不变，才能使元素根据自己包含内容的增加而在垂直方向上扩展。这样扩展的元素会把下方的元素向下推，而布局也能随内容数量的增减而垂直伸缩。

1.3 布局宽度

为了使浏览器窗口宽度合理变化，布局能作出适当的调整，我们需要精细地控制 布局宽度。

通过给整个外包装（#wrapper）设定宽度值（960px），并将其水平外边距设定为 auto，就能实现居中。随着向里面添加内容，相关的栏的高度会增加。

设置宽度并浮动中间三栏（nav、article 和 aside），让它们并排显示。使用属性：width 和 float。

三栏的总宽度加起来要等于外包装的宽度（ $150 + 600 + 210 = 960$ ）。同样使用该方法就可以加任意多栏，只要它们的总宽度等于外包装的宽度即可。

页脚（**footer**）位于浮动元素后面，所以就会尽量往上移动。解决这个问题的方法就是使用清除浮动（`clear:both`；或者使用 `clear:left`；也可以，因为这里只有左浮动元素）。

3.为栏设定内边距和边框

在上面的布局中，只要一调整各栏中的内容，布局就可能超过容器宽度，而右边的栏就可能滑到左边的栏下方。例如为了让内容与栏边界空开距离，为栏添加水平外边距和内边距，或者为了增加爱栏间距，为栏添加外边距，导致布局宽度增大，进而浮动栏下滑；又或者在栏中添加大图片，或者没有空格的长字符串（如长 URL），也会导致栏宽大超过布局宽度。

三栏的总宽度加起来要等于外包装的宽度（ $150 + 600 + 210 = 960$ ）。同样使用该方法就可以加任意多栏，只要它们的总宽度等于外包装的宽度即可。

页脚（**footer**）位于浮动元素后面，所以就会尽量往上移动。解决这个问题的方法就是使用清除浮动（`clear:both`；或者使用 `clear:left`；也可以，因为这里只有左浮动元素）。

3.为栏设定内边距和边框

在上面的布局中，只要一调整各栏中的内容，布局就可能超过容器宽度，而右边的栏就可能滑到左边的栏下方。例如为了让内容与栏边界空开距离，为栏添加水平外边距和内边距，或者为了增加爱栏间距，为栏添加外边距，导致布局宽度增大，进而浮动栏下滑；又或者在栏中添加大图片，或者没有空格的长字符串（如长 URL），也会导致栏宽大超过布局宽度。

参考资料及 应用资源	HTML+DIV+CSS网页设计与布局实用教程
作 业	课堂代码
教学反思	总结上课的问题,再做出改善。