

第 27 章 CSS 传统布局[下]

学习要点：

- 1.定位布局
- 2.box-sizing
- 3.resize

主讲教师：李炎恢

本课程由 北风网 和 瓢城 Web 俱乐部 提供

本章主要探讨 HTML5 中 CSS 早期所使用的传统布局，很多情况下，这些布局方式还是非常有用的。

一、定位布局

在使用定位布局前，我们先了解一下定位属性的用法。CSS2 提供了 position 属性来实现元素的绝对定位和相对定位。

属性	说明
static	默认值，无定位。
absolute	绝对定位，使用 top、right、bottom、left 进行位移。
relative	相对定位，使用 top、right、bottom、left 进行位移。
fixed	以窗口参考定位，使用 top、right、bottom、left 进行位移。

```
//绝对定位，脱离文档流，以窗口文档左上角 0,0 为起点
header {
    position: absolute;
    top: 100px;
    left: 100px;
}
```

所谓脱离文档流的意思，就是本身这个元素在文档流是占位的。如果脱离了，就不占有文档的位置，好像浮在了空中一般，有了层次感。

由于绝对定位脱离了文档流，出现层次概念。那么每个元素到底在那一层，会不会冲突覆盖。这时通过 z-index 属性来判定它们的层次关系。

属性	说明
auto	默认层次
数字	设置层次，数字越大，层次越高

//设置在 100 层上

```
header {  
    z-index: 100;  
}
```

//以窗口参考定位，脱离文档流，会随着滚动条滚动而滚动

```
header {  
    position: fixed;  
    top: 100px;  
    left: 100px;  
}
```

//相对定位，不脱离文档流，占位偏移

```
header {  
    position: relative;  
    top: 100px;  
    left: 100px;  
}
```

这三种分别都在各自的情况下使用，均比较常用。但还有一种情况，就是：**1.**既要脱离文档流（这样元素之间不会相互冲突）；**2.**以父元素，比如 **body** 或其他父元素为参考点（这样可以实现区域性绝对定位）；**3.**还必须是绝对定位。

//第一步，将需要设置参考点的父元素设置为相对，且不设置坐标

```
body {  
    position: relative;  
}
```

//第二步，如果父元素设置了参考点，子元素的绝对定位将以它为基准

```
header {  
    position: absolute;  
    top: 0px;  
    left: 0px;  
}
```

1. 固定布局

//CSS 部分

```
body {  
    width: 960px;  
    margin: 0 auto;  
    position: relative;  
}
```

```
header {
```

```
width: 960px;
height: 120px;
background-color: olive;
position: absolute;
top: 0;
left: 0;
}

aside {
width: 200px;
height: 500px;
background-color: purple;
position: absolute;
top: 120px;
left: 0;
}

section {
width: 760px;
height: 500px;
background-color: maroon;
position: absolute;
top: 120px;
/*left: 200px;*/
right: 0;
}

footer {
width: 960px;
height: 120px;
background-color: gray;
position: absolute;
top: 620px;
}
```

在上面，基本都用了定位来进行固定布局。但细心的可以发现，其实只有右侧需要实行绝对定位，其他就按照普通的摆放即可。对于设计成流体布局，只要将长度设置成百分比即可。

二. box-sizing

在盒模型那个章节，我们了解到元素盒子如果加入了内边距 **padding** 和边框 **border** 后，它的总长度会增加。那么如果这个元素用于非常精确的布局时，我们就需要进行计算增减。这其实是比较烦人的操作，尤其是动态设置页面布局的时候。

CSS3 提供了一个属性 **box-sizing**，这个属性可以定义元素盒子的解析方式，从而可

以选择避免掉布局元素盒子增加内边距和边框的长度增减问题。

属性	说明
<code>content-box</code>	默认值， <code>border</code> 和 <code>padding</code> 设置后用于元素的总长度。
<code>border-box</code>	<code>border</code> 和 <code>padding</code> 设置后不用于元素的总长度。

```
//设置 border-box 让 border 和 padding 不在额外增加元素大小
aside {
  width: 200px;
  height: 500px;
  background-color: purple;
  padding: 10px;
  border: 5px solid red;
  box-sizing: border-box;
  float: left;
}
```

`box-sizing` 是 CSS3 推出的，各个厂商在实现时设置了私有前缀。

	Opera	Firefox	Chrome	Safari	IE
支持需带前缀	无	2 ~ 28	4 ~ 9	3.1 ~ 5	8.0+
支持不带前缀	10.1+	29+	10+	6+	9.0+

```
//完整形式
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
-ms-box-sizing: border-box;
box-sizing: border-box;
```

三. `resize`

CSS3 提供了一个 `resize` 属性，来更改元素尺寸大小。

属性	说明
<code>none</code>	默认值，不允许用户调整元素大小。
<code>both</code>	用户可以调节元素的宽度和高度。
<code>horizontal</code>	用户可以调节元素的宽度。
<code>vertical</code>	用户可以调节元素的高度。

一般普通元素，默认值是不允许的。但如果是表单类的 `textarea` 元素，默认是允许的。而普通元素需要设置 `overflow:auto`，配合 `resize` 才会出现可拖拽的图形。

```
//允许修改
aside {
    resize: both;
    overflow:auto;
}
```

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 Web 俱乐部(ycku.com)联合提供：

本次主讲老师：李炎恢

谢谢大家，再见！