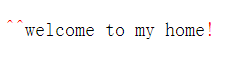
**:before和:after简例介绍**

　　:before和:after的作用就是在指定的元素**内容**（而不是元素本身）之前或者之后插入一个包含content属性指定内容的行内元素，最基本的用法如下：

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width,initial-scale=1">  
 <title></title>  
 <style>  
 #pid:before {  
 content: "^^";  
 color: red;  
 }  
 #pid:after {  
 content: "!";  
 color: red;  
 }  
 </style>  
</head>  
<body>  
 <p id="pid">welcome to my home</p>  
</body>  
</html>

这段代码会在#pid元素内容之前插入一个'^^'，以及在内容之后添加一个'!'，插入的行内元素是作为#pid的子元素，效果如下:



**2、:before和:after惊人用法**

在这里展示一个常用的场景，很多人写过如下清除浮动的代码：

<div id="container">  
 <div class="content-left">this is left content.</div>  
 <div class="content-right">this is right content.</div>  
 <div class="clear"></div>  
</div>  
  
对应的css代码如下：

.content-left{  
 float: left;  
 width: 150px;  
 font-weight:bold;">red;  
}  
.content-right{  
 float: right;  
 width: 450px;  
 font-weight:bold;">yellow;  
}  
.clear{  
 clear: both;  
}  
  
为了清除上面的浮动，多添加了一个<div>元素，并给此<div>元素添加了clear样式，这种做法破坏了HTML5的语义化原则，因此应对css样式进行修改，添加如下代码：

.clearfix{  
 zoom: 1;  
 }  
 .clearfix:before,  
 .clearfix:after{  
 display: table;  
 content: "";  
 }  
.clearfix:after{  
 clear: both;  
 }  
只要在父容器上应用clearfix这个类即可实现清除浮动。。。