CSS中px、rem和em的区别：

**一、**px像素（Pixel）。相对长度单位。像素px是相对于显示器屏幕分辨率而言的。(引自CSS2.0手册)

**二、**em是相对长度单位。相对于当前对象内文本的字体尺寸。如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。(引自CSS2.0手册)

1. em的值并不是固定的；

2. em会继承父级元素的字体大小。

**注意：**

1. body选择器中声明Font-size=62.5%；

2. 将你的原来的px数值除以10，然后换上em作为单位；

3. 重新计算那些被放大的字体的em数值。避免字体大小的重复声明。

**三、**rem是[CSS3](http://www.html5cn.org/portal.php?mod=list&catid=16)新增的一个相对单位（root em，根em），这个单位引起了广泛关注。这个单位与em有什么区别呢？区别在于使用rem为元素设定字体大小时，仍然是相对大小，但相对的只是HTML根元素。这个单位可谓集相对大小和绝对大小的优点于一身，通过它既可以做到只修改根元素就成比例地调整所有字体大小，又可以避免字体大小逐层复合的连锁反应。目前，除了IE8及更早版本外，所有浏览器均已支持rem。对于不支持它的浏览器，应对方法也很简单，就是多写一个绝对单位的声明。这些浏览器会忽略用rem设定的字体大小。下面就是一个例子：

p {font-size:14px; font-size:.875rem;}

px比em更加容易使用,em指字体高，任意浏览器的默认字体高都是16px。所以未经调整的浏览器都符合: 1em=16px，所以10px=0.625em。为了简化font-size的换算，需要在css中的body选择器中声明Font-size=62.5%，这就使em值变为16px\*62.5%=10px, 这样12px=1.2em, 10px=1em, 也就是说只需要将你的原来的px数值除以10，然后换上em作为单位就行了。

# CSS3的calc()使用

 calc()最大的好处就是用在流体布局上，可以通过calc()计算得到元素的宽度。

#### calc()能做什么？

calc()能让你给元素的做计算，你可以给一个div元素，使用百分比、em、px和rem单位值计算出其宽度或者高度，比如说“width:calc(50% + 2em)”，这样一来你就不用考虑元素DIV的宽度值到底是多少，而把这个烦人的任务交由浏览器去计算。

#### calc()语法

calc()语法非常简单，就像我们小时候学加 （+）、减（-）、乘（\*）、除（/）一样，使用数学表达式来表示：

.elm{

width: calc(expression);

}

其中"expression"是一个表达式，用来计算长度的表达式。

#### calc()的运算规则

calc()使用通用的数学运算规则，但是也提供更智能的功能：

1. 使用“+”、“-”、“\*” 和 “/”四则运算；
2. 可以使用百分比、px、em、rem等单位；
3. 可以混合使用各种单位进行计算；
4. 表达式中有“+”和“-”时，其前后必须要有空格，如"widht: calc(12%+5em)"这种没有空格的写法是错误的；
5. 表达式中有“\*”和“/”时，其前后可以没有空格，但建议留有空格。

#### 浏览器的兼容性

浏览器对calc()的兼容性还算不错，在IE9+、FF4.0+、Chrome19+、Safari6+都得到较好支持，同样需要在其前面加上各浏览器厂商的识别符，不过可惜的是，移动端的浏览器还没仅有“firefox for android 14.0”支持，其他的全军覆没。大家在实际使用时，同样需要添加浏览器的前缀。

elm{

*/\*Firefox\*/*

-moz-calc(expression);

*/\*chrome safari\*/*

-webkit-calc(expression);

*/\*Standard \*/*

calc();

}

CSS3中视区相关单位：VW，VH。

VW：相对于视窗的宽度，视窗宽度是100vw；

VH：相对于视窗的高度，视窗高度是100vh。

VM(vmin)：相对于视窗的宽度或高度，取决于哪个更小。