# 方向检测

**$("#box").bind("mouseenter mouseleave",function(e) {**

**var w = $(this).width();**

**var h = $(this).height();**

**var x = (e.pageX - this.offsetLeft - (w / 2)) \* (w > h ? (h / w) : 1);**

**var y = (e.pageY - this.offsetTop - (h / 2)) \* (h > w ? (w / h) : 1);**

**var direction = Math.round((((Math.atan2(y, x) \* (180 / Math.PI)) + 180) / 90) + 3) % 4; //direction的值为“0,1,2,3”分别对应着“上，右，下，左”**

**var eventType = e.type;**

**var dirName = new Array('上方','右侧','下方','左侧');**

**if(e.type == 'mouseenter'){**

**$("#result").html(dirName[direction]+'进入');**

**}else{**

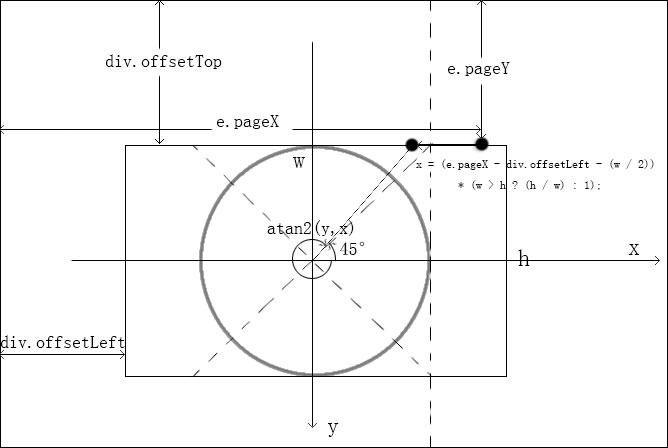
**$('#result').html(dirName[direction]+'离开');**

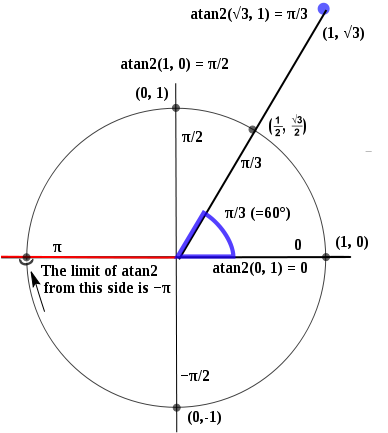
**}**

**});**

原理分析：

以div容器的中心点作为圆心，以高和宽的最小值作为直径画圆，将圆以[π/4，3π/4)，[3π/4，5π/4），[5π/4，7π/4)，[-π /4，π/4)划分为四个象限，鼠标进入容器时的点的atan2(y,x)值在这四个象限里分别对应容器边框的下，右，上，左。如下图所示：





var x = (e.pageX - this.offsetLeft - (w / 2)) \* (w > h ? (h / w) : 1);

计算x坐标值时，如果点原来的x坐标的绝对值大于圆的半径值，则按 h/w 这个比例进行缩小，使得到的点的位置在容器的边界位置所对应的象限区间里。 y 坐标的计算也是一样。

var direction = Math.round((((Math.atan2(y, x) \* (180 / Math.PI)) + 180) / 90) + 3) % 4;

((Math.atan2(y, x) \* (180 / Math.PI)将点的坐标对应的弧度值换算成角度度数值，这里加上180不是并非必要的，而是在这个算法里必须的，因为atan2(y,x)返回值的范围是[-π,π],而不是我们习惯的[0,2π],负值会影响结果的正确性(比如右上和右下算出来的结果会不同)，而且确实也使得得到的结果0,1,2,3的顺序符合了习惯。 除以90，再取四舍五入值，是一个很精妙的用法，使得可以以45°为分界线。

Math.atan2这个函数算出与x轴的夹角，按理说应该给坐标轴旋转45度才对啊？ 但是并不需要，因为用了Math.round最后，四舍五入。