Javascript es infinito y es muy habitual encontrarse con problemas o retos en el día a día. Uno de los problemas más habituales se denomina Javascript Framing. ¿Qué es esto? . Esta situación ocurre de una manera muy habitual cuando tenemos eventos como por ejemplo “mouseover” en una página. Vamos a diseñar un ejemplo que nos permita entender su funcionamiento.

```
<html>
<head>
<style type="text/css">

#zona {

width:200px;
height: 200px;
background-color: blue;
}

</style>
<script src="jquery-1.11.1.js" type="text/javascript"></script>
<script type="text/javascript">

$(document).ready(function() {
$("#zona").mousemove(function(evento) {

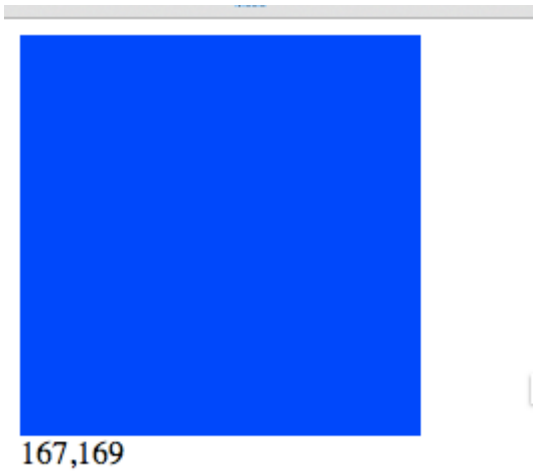
$("#mensaje").html(evento.clientX +","+ evento.clientY);
})

});
</script>
```

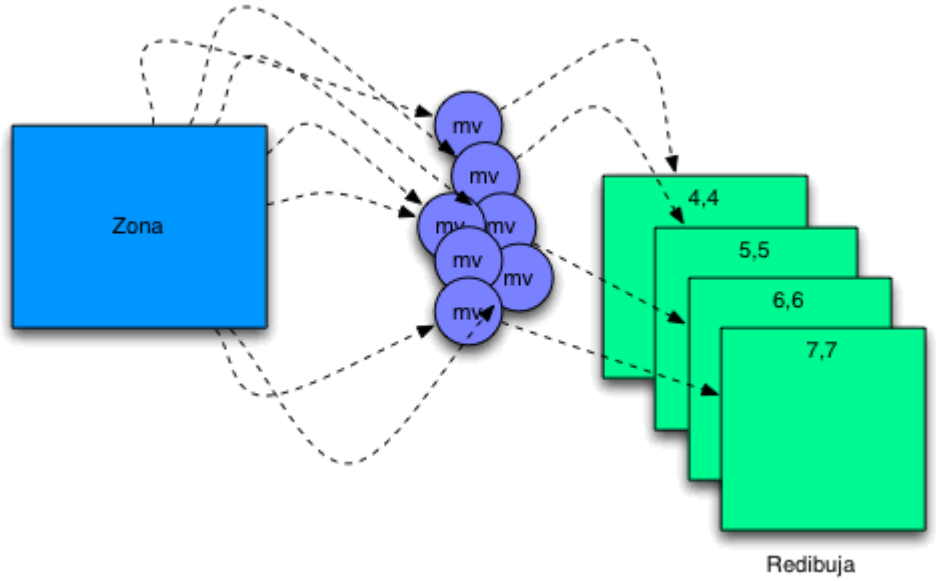
```
</head>
<body>
<div id="zona">

</div>
<div id="mensaje">
</div>
</body>
</html>
```

En este ejemplo tenemos una página HTML que dispone de una “zona”  de color azul que cuando pasamos el ratón por ella nos imprime las coordenadas X,Y en un div.




Todo parece correcto sin embargo estamos obligando al navegador a redibujar la página continuamente cada vez que el evento de mousemouse se produce.




Esto supone un problema de rendimiento ya que forzamos al navegador a redibujar la página constantemente. Para solventar esto podemos apoyarnos en Javascript Framing una técnica que utiliza el método setInterval para reducir las veces que se redibuja mejorando el rendimiento vamos a verlo.

```
<html>
<head>
<style type="text/css">

#zona {

width:200px;
height: 200px;
background-color: blue;
}
```

```
</style>
<script src="jquery-1.11.1.js" type="text/javascript"></script>
<script type="text/javascript">

$(document).ready(function() {

var x;
var y;
$("#zona").mousemove(function(evento) {

x=evento.clientX;
y=evento.clientY;
})

function imprimir() {

$("#mensaje").html(x +","+ y);
}

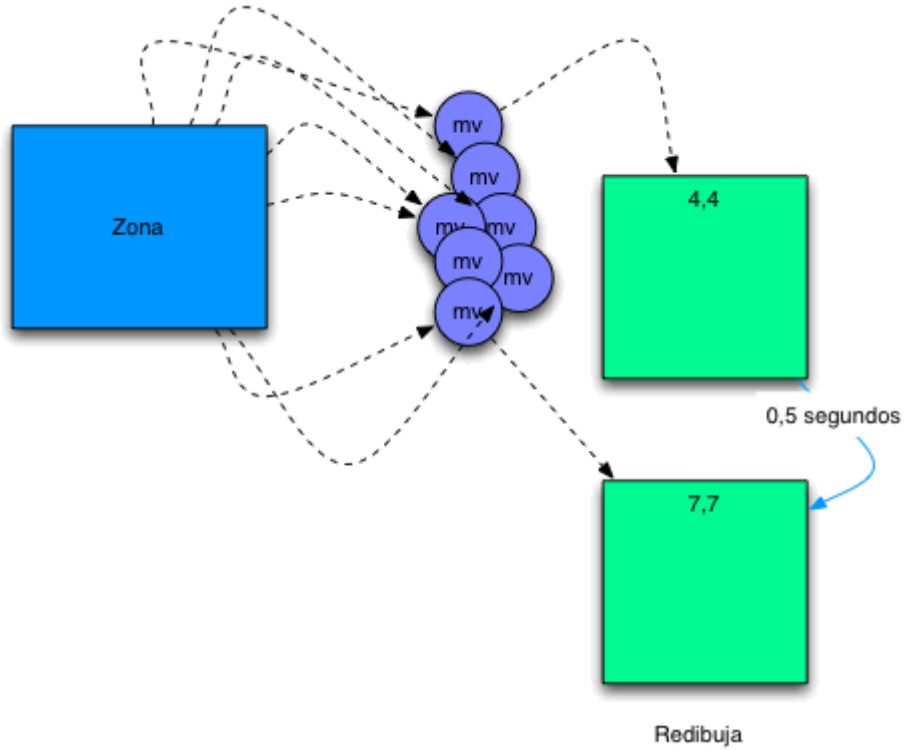
setInterval(imprimir,500);

});
</script>
</head>
<body>
<div id="zona">

</div>
<div id="mensaje">
</div>
</body>
```

```
</html>
```

Como podemos ver en este caso hemos definido una función de imprimir que se encarga de redibujar las coordenadas cada 500 milisegundos en vez de cada vez que el evento se produzca. Esto no nos afectará en cuanto a la funcionalidad pero si que mejorará de forma significativa el rendimiento ya que hay que redibujar la página muchas menos veces.




Pocos lenguajes hay con mas trucos que Javascript a la hora de trabajar con él

Otros artículos relacionados: Usando JQueryContexto , JavaScript Console