Cascade

# Cascada

Algunos métodos no tienen valor de retorno. Por ejemplo, es típico que los métodos que establecen o cambian el estado de un objeto no devuelvan nada. Si hacemos que esos métodos devuelvan this en lugar de undefined, podemos activar *cascades*. En una cascada, podemos llamar a muchos métodos del mismo objeto en secuencia en una única sentencia. Una biblioteca Ajax que habilite las cascadas nos permitiría escribir con un estilo como éste:

getElement('myBoxDiv')  
 .move(350, 150)  
 .width(100)  
 .height(100)  
 .color('red')  
 .border('10px outset')  
 .padding('4px')  
 .appendText("Please stand by")  
 .on('mousedown', function (m) {  
 this.startDrag(m, this.getNinth(m));  
 }).  
 .on('mousemove', 'drag')  
 .on('mouseup', 'stopDrag')  
 .later(2000, function ( ) {  
 this  
 .color('yellow')  
 .setHTML("What hath God wraught?")  
 .slide(400, 40, 200, 200);  
 })  
 .tip("This box is resizeable");

En este ejemplo, la función getElement produce un objeto que da funcionalidad al elemento DOM con id="myBoxDiv". Los métodos nos permiten mover el elemento, cambiar sus dimensiones y estilo, y añadir comportamiento. Cada uno de esos métodos devuelve el objeto, de modo que el resultado de la invocación puede utilizarse para la siguiente invocación.

La conexión en cascada puede producir interfaces muy expresivas. Puede ayudar a controlar la tendencia a crear interfaces que intentan hacer demasiadas cosas a la vez.