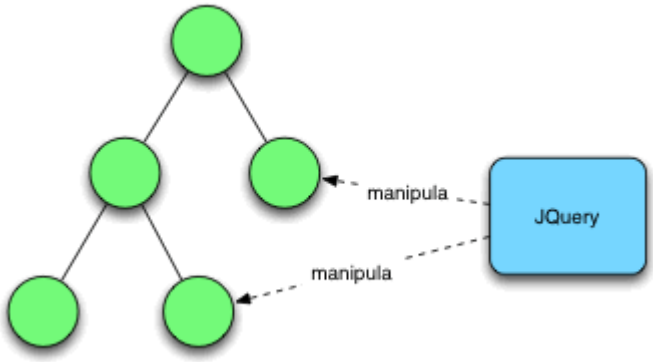
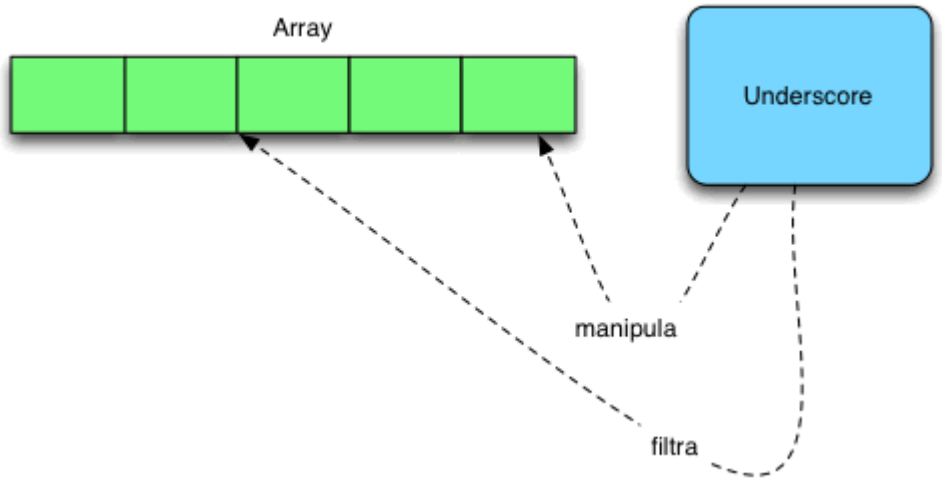
Hoy por hoy la mayoría de los desarrolladores utilizamos JQuery como framework de capa de presentación que nos ayuda a simplificar y reducir las tareas que tenemos que realizar en entornos Web. Ahora bien también hay que recordar que el framework se centra en el manejo y manipulación de la capa de presentación (Arbol DOM). En este artículo vamos a hablar de Underscore.js que es un gran complemento para JQuery.



JQuery y Underscore

Cuando mas manejamos JQuery mas sencillo es darse cuenta que nos faltan algunas cosas fundamentales a nivel de Javascript .Quiza la mas importante es su capacidad para manipular datos . JQuery no dispone de funciones potentes para manejo de arrays y de estructuras JSON .¿Es esto un punto en su contra? .La realidad es justamente lo contrario . JQuery es un framework de presentación NO de negocio. Por lo tanto han decidido centrarse en una cosa y ser el mejor en ella . ¿Como podemos solventar nuestros problemas de manipulación de datos?. En este caso es relativamente sencillo , podemos utilizar la libreria Underscore.js (bajar de aqui) que esta centrada justamente en la gestión del negocio.



Vamos a ver un ejemplo en el que definimos una tabla y rellenamos la tabla utilizando una estructura JSON y Underscore.js

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script type="text/javascript" src="jquery-1.10.2.min.js">
</script>
<script type="text/javascript" src="underscore.js">
</script>

<script type="text/javascript">
$(document).ready(function(evento) {

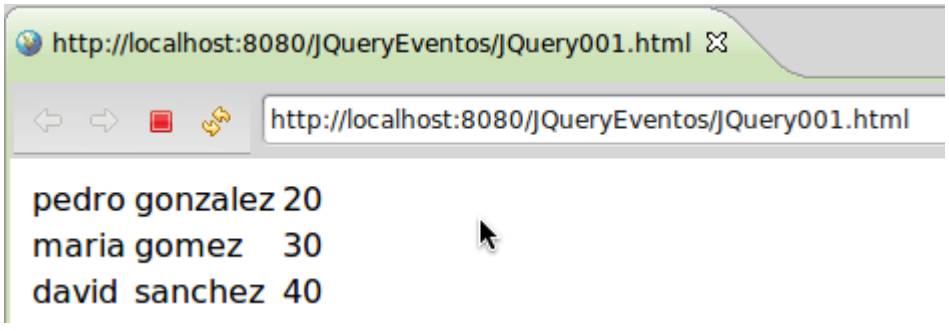
var listaPersonas=
[{"nombre":"pedro","apellidos":"gonzalez","edad":20},
 {"nombre":"maria","apellidos":"gomez","edad":30},
 {"nombre":"david","apellidos":"sanchez","edad":40},];
```

```
//uso de underscore
_.each(listaPersonas,function(elemento) {

 $("#tabla").append("<tr><td>"+elemento.nombre+"</td><td>"
 +elemento.apellidos+
 "</td><td>"+elemento.edad+"</td></tr>");

})
});
</script>
</head>
<body>
<table id="tabla">
</table>
</body>
</html>
```

Hemos usado el método each de underscore (_.each) para generar la tabla que se muestra a continuación:



Underscore y Filtrados

Supongamos que queremos añadir ahora una caja de texto con el campo nombre y un botón

de filtrar. De tal forma que cuando introducimos un nombre y pulsamos se realice un filtrado por ese nombre .Si los datos JSON nos hubieran venido via AJAX podríamos hacer otra petición AJAX que nos filtre . Sin embargo todos podemos intuir que si ya tenemos los datos en el cliente deberiamos filtrar de forma directa y ahorrarnos peticiones . Esta operación la podemos realizar utilizando otro de los métodos de la librería underscore concretamente el método where .Vamos a ver como queda el código.

```
</pre>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<script type="text/javascript" src="jquery-1.10.2.min.js">
</script>

<script type="text/javascript" src="underscore.js">
</script>

<script type="text/javascript">
$(document).ready(function(evento) {

var listaPersonas=
[{"nombre":"pedro","apellidos":"gonzalez","edad":20},
 {"nombre":"maria","apellidos":"gomez","edad":30},
 {"nombre":"david","apellidos":"sanchez","edad":40},

 ];

_.each(listaPersonas,function(elemento) {

 $("#tabla").append("<tr><td>"+elemento.nombre+"</td><td>"+
```

```
 elemento.apellidos+"</td><td>"+elemento.edad+"</td></tr>");

})

$("#boton").click(function() {

 $("#tabla tr").remove();

 var listaFiltro=_.where(listaPersonas, {nombre:$("#nombre").val()});
 _.each(listaFiltro,function(elemento) {

 $("#tabla").append("<tr><td>"+
 elemento.nombre+"</td><td>"+
 elemento.apellidos+"</td><td>"+
 elemento.edad+"</td></tr>");

})
});

});
</script>
</head>
<body>
<table id="tabla">

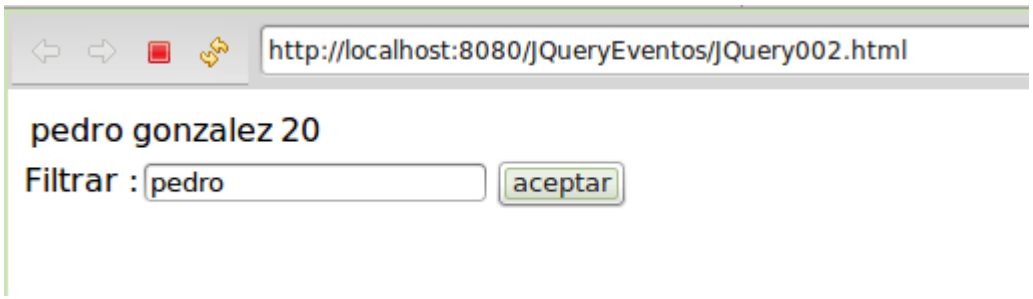
 

</table>

Filtrar :<input type="text" name="nombre" id="nombre"><input
type="button" id="boton" value="aceptar" />
```

```
</body>
</html>
<pre>
```

El resultado de usar Underscore será el siguiente.



Hemos realizado un filtrado sin tener que recurrir a AJAX y manipulando unicamente el cliente.