Documentación de widgets en JqueryUI usados en la práctica.

DATEPICKER

Se ha usado el widget datepicker en 2 ocasiones; tanto en el input para la fecha de inicio del alquiler como para la de su fin.

Aquí el código:

```
/* DATEPICKER */
$(function () {
    $("#datepicker").datepicker({
        numberOfMonths: 2,
        showButtonPanel: true

    });
    $("#datepicker1").datepicker({
        numberOfMonths: 2,
        showButtonPanel: true

    });
    $("#datepicker").datepicker("setDate", new Date());
    $("#datepicker1").datepicker("setDate", new Date());
});
$(function ($) {
    $.datepicker.regional['es'] = {
        closeText: 'Cerrar',
        prevText: '< Ant',
        nextText: 'Sig >',
        currentText: 'Hoy',
        monthNames: ['Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo', 'Junio', 'Julio', 'Agosto', 'Septiembre', 'Octubre', 'Noviembre', 'Diciembre'],
        monthNamesShort: ['Ene', 'Feb', 'Mar', 'Abr', 'May', 'Jun', 'Jul', 'Ago', 'Sep', 'Oct', 'Nov', 'Dic'],
        dayNames: ['Domingo', 'Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado'],
        dayNamesShort: ['Dom', 'Lun', 'Mar', 'Mié', 'Juv', 'Vie', 'Sáb'],
        dayNamesMin: ['Do', 'Lu', 'Ma', 'Mi', 'Ju', 'Vi', 'Sá'],
        weekHeader: 'Sm',
        dateFormat: 'dd/mm/yy',
        firstDay: 1,
        isRTL: false,
        showMonthAfterYear: false,
        yearSuffix: ''
    };
    $.datepicker.setDefaults($.datepicker.regional['es']);
```

Ya que el datepicker que viene por defecto se ajusta al formato que estaba buscando, tanto en los inputs que contienen el id “datepicker” y “datepicker1”, sólo he editado un par de parámetros como lo son “numberOfMonths” para que me muestre 2 meses a la vez, y la opción de “showButtonPanel” está en true para que tengamos acceso a un panel de control dentro del widget.

Además, para que por defecto al iniciar nuestro datepicker ponga una fecha, la cual es la del día actual, creo un nuevo objeto Date y lo añado a la opción “setDate” del widget.

Más adelante, he hecho la traducción del datepicker para que su formato esté en español. Las opciones como “prevText” y “nextText” serán unos botones para pasar a los siguientes/anteriores meses. La única opción que pienso que no habla por sí sola, es la de “isRTL”, que significa “Is Right To Left”, que para nosotros estará en false, pero para otros idiomas que se leen de derecha a izquierda como el árabe, tendremos que dejarla en true.

AUTOCOMPLETE

En mi caso, quiero insertar el widget “Autocomplete” en el input donde el usuario escribirá la ciudad desde donde hace la reserva.

Aquí el código:

```
/* AUTOCOMPLETE */
$(function () {
    var getData = function (request, response) {
        $.getJSON(
            "http://gd.geobytes.com/AutoCompleteCity?callback=?&q=" + request.term,
            function (data) {
                response(data);
            });
    };
    var selectItem = function (event, ui) {
        $("#inputCity").val(ui.item.value);
        return false;
    }
    $("#inputCity").autocomplete({
        source: getData,
        select: selectItem,
        minLength: 4,
        change: function () {
            $("#inputCity").val("").css("display", 2);
        }
    });
});
```

Creo una variable que contiene una función, la cuál hace una petición a la API de Geobytes. Esta nos devuelve un objeto JSON, el cual tratamos más adelante como podemos ver en $(“#inputCity”).val(ui.item.value).

Después añadimos las opciones de origen y el ítem a seleccionar, un “minLength” de 4 para que el autocomplete empiece a realizar peticiones una vez se hayan escrito 4 letras como mínimo.

TABS

En mi caso he decidido estructurar el formulario de reserva en varias partes, y para aprovechar el uso de JqueryUI, lo he hecho así con los Tabs, o pestañas.

```
$(function () {
    $("#tabs").tabs();
});
```

```
<div id="tabs">
    <ul>
        <li><a href="#tabs-1">Datos de cliente</a></li>
        <li><a href="#tabs-2">Datos del producto</a></li>
        <li><a href="#tabs-3">Fecha</a></li>
        <li><a href="#tabs-4">Estilazos</a></li>
    </ul>
```

Llamamos a los Tabs y creamos la estructura de un menú, que contendrá tantos tabs como queramos. Esto nos creará el menú, y las pestañas se crearán en html con un DIV con id, en nuestro caso, “tabs-1/2/3/4”. Al hacer click en algún elemento del menú nos mostrará el contenido que tiene dentro y ocultará los del resto.

TOOLTIP

Para hacer más orientativa la página también he hecho el uso del widget “Tooltip”.

Se ha usado con este código:

```
$(document).tooltip({
    track: true,
    show: {
        effect: "slideDown",
        delay: 250
    },
    hide: {
        effect: "drop",
        delay: 0,
        duration: 100
    }
});
});
```

Las opciones que le he dado han sido por ejemplo “track”, que hace que mientras el ratón esté encima del elemento donde quiero mostrar el tooltip, persiga al cursor hasta que se salga del elemento. Para su animación, he usado el efecto “slideDown” en la aparición, y el efecto “drop” para la desaparición, con una duración de 250 y 100 milisegundos respectivamente.

A la hora de insertarlos, lo único que hay que hacer es añadir la etiqueta “title=”Mensaje a mostrar en el tooltip”” para meter contenido dentro del tooltip. Así:

```
<div class="form-group col-md-6" title="Tus apellidos, porfavor">
```

ACCORDION

He insertado el widget “Accordion” en la selección del producto que se va a alquilar en mi página.

```
$("#accordion").accordion({
    heightStyle: "content",
    collapsible: true
});
```

```
<div id="accordion">
    <h3 title="Elige una furgoneta de las listadas">Furgoneta</h3>
    <div class="selectorFurgoneta">
        <select id="car-type">
            <option disabled selected>Elige una furgoneta...</option>
            <option>Renault Transit</option>
            <option>Citroen Berlingo</option>
            <option>Fiat Ducato</option>
            <option>For Courier</option>
            <option>Hyundai H1</option>
            <option>Kia Besta</option>
            <option>Mercedes Vito</option>
            <option>Mitsubishi L300</option>
            <option>Nissan Cabstar E</option>
        </select>
    </div>
    <h3 title="Selecciona el tipo de transmisión para tu fabulosa furgoneta; manual o automática.">
        Transmisión</h3>
    <div class="selectorTransmision">
        <label for="transmission-standard">Manual</label>
        <input type="radio" name="transmission" id="transmission-standard">
        <label for="transmission-automatic">Automático</label>
        <input type="radio" name="transmission" id="transmission-automatic">
    </div>
    <h3 title="Número de plazas que deseas que tenga tu furgoneta">Plazas</h3>
    <div class="selectorPlazas">
        <select id="car-plazas">
            <option selected disabled>Número de plazas...</option>
            <option>5 plazas</option>
            <option>7 plazas</option>
            <option>+7 plazas</option>
        </select>
    </div>
</div>
```

Como podemos ver, inserto el contenido dentro de un DIV con etiqueta “accordion”. Las opciones que le he asignado al accordion han sido “heightStyle: Content”, para que la altura se adecue al contenido que haya dentro, y “collapsible: true”, lo que nos permite cerrar todos los elementos del accordion si queremos; si no ponemos esta opción, o la ponemos en false, siempre habrá un elemento abierto del accordion. La forma de añadir “hijos” al accordion, es con etiquetas de texto y después de ellas etiquetas DIV con el contenido. Las etiquetas de texto serán el título del elemento.

También he usado el widget **Select Menu** Para la selección de las furgonetas disponibles.

```
$(".furgoneta").selectmenu();
```

Este se aplica al elemento que contenta la clase “furgoneta”

CONTROL GROUP

Para la selección de los extras de la reserva, entre otros, he usado el widget “Control group”.

```
$(function () {
    $(".controlgroup").controlgroup()
});
```

```
<div class="controlgroup col-md-6 text-left">
    <fieldset>
        <h3>Extras</h3>
        <label for="insurance-v1">Aire acondicionado</label>
        <input type="checkbox" name="aireAcondicionado" id="insurance-v1">
        <label for="insurance-v2">Asistencia de freno</label>
        <input type="checkbox" name="insurance" id="insurance-v2">
        <label for="insurance-v3">Asientos tapizados</label>
        <input type="checkbox" name="insurance" id="insurance-v3">
        <label for="insurance-v4">Rueda de recambio</label>
        <input type="checkbox" name="insurance" id="insurance-v4">
        <label for="insurance-v5">Extintor</label>
        <input type="checkbox" name="insurance" id="insurance-v5">
        <label for="insurance-v6">Disco del Canelita</label>
        <input type="checkbox" name="insurance" id="insurance-v6">
    </fieldset>
</div>
```

Mi grupo de controles está formado por inputs de tipo “checkbox”. En el apartado anterior, se ve cómo también he añadido inputs de type radio en el accordion, el cuál es otro controlgroup.

THEMES

La introducción de distintos temas en una página ha sido implementada de la siguiente forma:

```
/* TEMAS */
$(function () {
    $("#dialogoTemas").dialog({
        autoOpen: false,
        width: 300,
        height: 200,
        show: {
            effect: "slide",
            duration: 500
        },
        hide: {
            effect: "explode",
            duration: 250
        }
    });

    $("#temas").on("click", function () {
        $("#dialogoTemas").dialog("open");
    });
});
function cambiarEstilo(nuevoEstilo) {
    console.log('Tema cambiado')
    var body = $('body');
    body.removeClass("oscuro");
    body.removeClass("verde");
    body.removeClass("amarillo");
    body.addClass(nuevoEstilo);
};
```

```
<div id="tabs-4">
    <button id="temas" class="btn btn-primary w-100">Abrir Temas</button>
    <div id="dialogoTemas">
        <div class="row">
            <div class="col-md-12 col-12 mt-1 w-100">
                <button id="oscuro" class="btn btn-secondary w-100" onclick="cambiarEstilo('oscuro')">Tema Oscuro</button>
            </div>
            <br>
            <div class="col-md-12 col-12 mt-1 w-100">
                <button id="verde" class="btn btn-success w-100" onclick="cambiarEstilo('verde')">Tema Verde</button>
            </div>
            <br>
            <div class="col-md-12 col-12 mt-1 w-100">
                <button id="amarillo" class="btn btn-warning w-100" onclick="cambiarEstilo('amarillo')">Tema Amarillo</button>
            </div>
            <br>
        </div>
    </div>
</div>
```

Para ello en mi pestaña de Estilazos, he creado un botón que abrirá un diálogo, el cuál mostrará el contenido que hay dentro de la etiqueta DIV con ID “dialogoTemas”. Este tendrá una altura y anchura, efectos para aparición como “slide” y “explode” para su desaparición, con una duración determinada. Este se abre al hacer click en la pestaña, que su id es “temas”. Al hacer click se muestra el diálogo con 3 botones, los cuales al clickarlos ejecutan la función cambiarEstilo, a la que se le pasa como parámetro el estilo, y dentro de ella desactiva todos los temas posibles que haya, y añade al cuerpo el tema que se va a poner. Al pasarle al body la clase que queramos, cargará el tema que hayamos descargado, con el **scope** que será el mismo que la clase que pasamos.

DIALOG

Como ya he nombrado antes, he usado el widget “Dialog”. Lo he usado en 2 ocasiones, en la citada anteriormente y en esta:

```
    $(function () {
        $("#dialog").dialog({
            autoOpen: false,
            show: {
                effect: "slide",
                duration: 500
            },
            hide: {
                effect: "explode",
                duration: 250
            }
        });

        $("#opener").on("click", function () {
            $("#dialog").dialog("open");
        });
    });
```

```
<button id="opener" style="width: 100%;" type="submit" class="btn btn-primary">Enviar</button>
<div id="dialog" title="¡Formulario enviado!">
    <p>Formulario enviado con éxito</p>
    <button style="width: 100%" class="btn btn-secondary" onclick="window.location.reload()">De
        acuerdo
    </button>
</div>
```

Como podemos ver, tenemos un botón con id “opener”, el cuál será el que ejecute la llamada a la aparición del diálogo. Al hacerle click, nos mostrará un diálogo con una animación de 500ms con el efecto slide, y al cerrarse hará una animación con efecto “explode” y que tarda 250ms en realizarse. Cuando hacemos click en el “opener”, nos muestra el contenido de “dialog”. En este caso simplemente muestra un mensaje que nos dice que el formulario ha sido rellenado con éxito, y un botón que recargará la ventana.

EFECTOS

Para hacer una demostración de los efectos que se pueden hacer con JqueryUI, he creado 3 botones con 3 efectos distintos que afectan al contenido de un DIV.

```
/* EFECTOS */

$(function () {
    $("#efecto1").on("click", function () {
        $(".primeraEfecto1").switchClass("primeraEfecto1", "segundaEfecto1", 1000);
        $(".segundaEfecto1").switchClass("segundaEfecto1", "primeraEfecto1", 1000);
    });
});
$(function () {
    var state = true;
    $("#efecto2").on("click", function () {
        if (state) {
            $("#afectada").animate({
                backgroundColor: "#aa0000",
                color: "#fff",
                width: 500
            }, 1500);
        } else {
            $("#afectada").animate({
                backgroundColor: "#fff",
                color: "#000",
                width: 240
            }, 500);
        }
        state = !state;
    });
});

function callback() {
    setTimeout(function() {
        $( "#afectada" ).removeAttr( "style" ).hide().fadeIn();
    }, 500 );
};
```

Estos botones tienen las IDs “efecto1/2/3/”, y el div que está afectado por estas animaciones es un DIV con ID “afectada”. En el efecto 1, hago un cambio de clases. Por defecto tiene la clase “primeraEfecto1”, y al hacer click, cambia esta clase por “segundaEfecto1”, que simplemente cambia el tamaño de la letra con un delay de 1 segundo. En el segundo efecto, cambio directamente el color de fondo, anchura y color de la letra al hacer click, y al hacerlo de nuevo, cambio de nuevo el color de fondo a blanco, color de la letra a negro, y la anchura a 240px. En el tercer efecto, tengo una función llamada “callback()”, y esta se ejecuta al hacer click en el botón. Simplemente oculta el DIV con ID “afectada” con una duración de medio segundo, con un efecto fadeIn.