Hola Ignacio, te hago un resumen de las funciones de JQuery que más utilizamos.

Nos interesa sobre todo que hagas ejemplos de Ajax, autocomplete, dialog por ejemplo y también de las validaciones y sobre todo del manejo de eventos a través de triggers, on, off que creo que le da mucho potencial al JQuery para manejo de pantallas. Las otras funciones puedes hacer ejemplos sencillos sin complicarse la vida.

Intro

1. Actualizar jquery

https://github.com/jquery/jquery-migrate/#readme

In your web page, load this plugin after the script tag for jQuery, for example:

<script src="https://code.jquery.com/jquery-3.6.3.js"></script>

<script src="https://code.jquery.com/jquery-migrate-3.4.1.js"></script>

1. Deprecated

Revisar los métodos en desuso o no existentes en mi versión jQuery.

<https://api.jquery.com/category/deprecated/>

1. Documentación de librería.

Oficial:

<https://jquery.com/>

<https://jqueryui.com/>

<https://api.jquery.com/>

https://learn.jquery.com/

Tutoriales con ejemplos para cambiar y probar online:

<https://www.tutorialsteacher.com/jquery/jquery-post-method>

<https://www.w3schools.com/jquery/>

1. Seguridad

https://security.snyk.io/package/npm/jquery

https://nvd.nist.gov/products/cpe/search/results?namingFormat=2.3&keyword=cpe%3A2.3%3Aa%3Ajquery%3Ajquery%3A3.3.1

[https://nvd.nist.gov/vuln/search/results?adv\_search=true&isCpeNameSearch=true&query=cpe%3A2.3%3Aa%3Ajquery%3Ajquery%3A3.3.1%3A\*%3A\*%3A\*%3A\*%3Anode.js%3A\*%3A\*](https://nvd.nist.gov/vuln/search/results?adv_search=true&isCpeNameSearch=true&query=cpe%3A2.3%3Aa%3Ajquery%3Ajquery%3A3.3.1%3A*%3A*%3A*%3A*%3Anode.js%3A*%3A*)

https://github.com/orgs/SonarSource/repositories?type=all

NGINX

https://security.snyk.io/package/npm/jquery

https://www.ccn-cert.cni.es/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/522-ccn-stic-812-seguridad-en-entornos-y-app-web/file.html

https://www.ccn-cert.cni.es/component/search/?searchword=javascript&searchphrase=all&Itemid=784

https://www.cvedetails.com/product/11031/Jquery-Jquery.html?vendor\_id=6538

https://blog.elhacker.net/2021/09/xss-o-crosssite-scripting-tipos-de-ataque-y-prevencion.html

https://www.virtuesecurity.com/jquery-security-model-explained/

1. Librería desarrollo

https://github.com/jquery/jquery

1. Otros

<https://github.com/davidam/jquery-examples>

<https://www.postman.com/>

<https://jqueryui.com/autocomplete/>

|  |
| --- |
| **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>jQuery UI Autocomplete - Default functionality</title>  <link rel="stylesheet" href="//code.jquery.com/ui/1.13.2/themes/base/jquery-ui.css">  <link rel="stylesheet" href="/resources/demos/style.css">  <script src="https://code.jquery.com/jquery-3.6.0.js"></script>  <script src="https://code.jquery.com/ui/1.13.2/jquery-ui.js"></script>  <script>  $( **function**() {  **var** availableTags = [  "ActionScript",  "AppleScript",  "Asp",  "BASIC",  "C",  "C++",  "Clojure",  "COBOL",  "ColdFusion",  "Erlang",  "Fortran",  "Groovy",  "Haskell",  "Java",  "JavaScript",  "Lisp",  "Perl",  "PHP",  "Python",  "Ruby",  "Scala",  "Scheme"  ];  $( "#tags" ).autocomplete({  source: availableTags  });  } );  </script>  </head>  <body>    <div class="ui-widget">  <label for="tags">Tags: </label>  <input id="tags">  </div>      </body>  </html> |
|  |

<https://jqueryui.com/dialog/>

|  |
| --- |
| **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>jQuery UI Dialog - Default functionality</title>  <link rel="stylesheet" href="//code.jquery.com/ui/1.13.2/themes/base/jquery-ui.css">  <link rel="stylesheet" href="/resources/demos/style.css">  <script src="https://code.jquery.com/jquery-3.6.0.js"></script>  <script src="https://code.jquery.com/ui/1.13.2/jquery-ui.js"></script>  <script>  $( **function**() {  $( "#dialog" ).dialog();  } );  </script>  </head>  <body>    <div id="dialog" title="Basic dialog">  <p>This is the default dialog which is useful for displaying information. The dialog window can be moved, resized and closed with the &apos;x&apos; icon.</p>  </div>      </body>  </html> |

triggers, on, off

https://api.jquery.com/trigger/

|  |
| --- |
| **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <title>trigger demo</title>  <style>  button {  margin: 10px;  }  div {  color: blue;  font-weight: bold;  }  span {  color: red;  }  </style>  <script src="https://code.jquery.com/jquery-3.6.3.js"></script>  </head>  <body>    <button>Button #1</button>  <button>Button #2</button>  <div><span>0</span> button #1 clicks.</div>  <div><span>0</span> button #2 clicks.</div>    <script>  $( "button" ).first().click(**function**() {  update( $( "span" ).first() );  });    $( "button" ).last().click(**function**() {  $( "button" ).first().trigger( "click" );  update( $( "span" ).last() );  });    **function** update( j ) {  **var** n = parseInt( j.text(), 10 );  j.text( n + 1 );  }  </script>    </body>  </html> |

Te he pasado en este documento también algunos ejemplos que hemos utilizados en una aplicación MVC para que veas como lo utilizamos. Los ejemplos que tú nos proporciones deberían ser para ASP.NET que es la tecnología que ahora mismo conocemos todos los programadores.

Se

**MAS UTILIZADAS:**

$.ajax

INTRO GET POST DELETE PUT

<https://editor.swagger.io/>

|  |
| --- |
|  |

$.autocomplete

https://jqueryui.com/autocomplete/

|  |
| --- |
|  |

.dialog

**Dialog Widget JQueryUI This widget requires some functional CSS, otherwise it won't work. If you build a custom theme, use the widget's specific CSS file as a starting point.**

Se ve un ejemplo completo de un objeto JQueryUI con todas sus opciones

|  |
| --- |
| // ventana con mensaje  **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>jQuery UI Dialog - Default functionality</title>  <link rel="stylesheet" href="//code.jquery.com/ui/1.13.2/themes/base/jquery-ui.css">  <link rel="stylesheet" href="/resources/demos/style.css">  <script src="https://code.jquery.com/jquery-3.6.0.js"></script>  <script src="https://code.jquery.com/ui/1.13.2/jquery-ui.js"></script>  <script>  $( **function**() {  $( "#dialog" ).dialog();  } );  </script>  </head>  <body>    <div id="dialog" title="Titulo ventana">  <p>Texto de ventana movible y redimensionable. Ciérrela con el botón &apos;X&apos; </p>  </div>  </body>  </html> |
| //modificadores  dialogClass: "no-close", //ocultar btn de cerrado  appendTo: "#someElem" // anexado a , por defecto el body  autoOpen: false //oculto hasta llamar open  buttons: [{matriz}] //  classes: {"ui-dialog": "highlight" //aplica clases  closeOnEscape: false//no cerrar con escape  closeText: "hide"//texto en btn cerrado  dialogClass: "alert"//= classes deprecated  //otros  **height //altura**  $( ".selector" ).dialog({ height: 400});  **hide //oculto**  $( ".selector" ).dialog({ hide: { effect: "explode", duration: 1000 }});  **maxHeight minHeight  //máxima mínima altura**  $( ".selector" ).dialog({ maxHeight: 600});  **maxWidth minWidth //máxima mínima anchura**  $( ".selector" ).dialog({ maxWidth: 600});  **position //posición**  $( ".selector" ).dialog({ position: { my: "left top", at: "left bottom", of: button } });  **resizable // redimensionable**  $( ".selector" ).dialog({ resizable: false });  **show //mostrar y animacíón**  $( ".selector" ).dialog({ show: { effect: "blind", duration: 800 } });  **title //título ventana**  $( ".selector" ).dialog({ title: "Dialog Title" });  **width //anchura**  $( ".selector" ).dialog({ width: 500}); |
| Métodos:  $( ".selector" ).dialog( "close" );//cierra  $( ".selector" ).dialog( "destroy" );//elimina objeto  $( ".selector" ).dialog( "instance" );//recupera objeto  **var** isOpen = $( ".selector" ).dialog( "isOpen" ); //devuelve estado  $( ".selector" ).dialog( "moveToTop" );//posición arriba  $( ".selector" ).dialog( "open" );//abre ventana diálogo  **var** options = $( ".selector" ).dialog( "option" );//devuelve las opciones del objeto |
| Extensores  El widget de diálogo se crea con la fábrica de widgets y se puede ampliar. Al extender los widgets, tiene la capacidad de anular o agregar al comportamiento de los métodos existentes.:  <http://learn.jquery.com/jquery-ui/widget-factory/extending-widgets/> |
| Eventos . Se s  $( ".selector" ).dialog({  beforeClose: **function**( event, ui ) {}  });  $( ".selector" ).dialog({  close: **function**( event, ui ) {}  });  $( ".selector" ).dialog({  create: **function**( event, ui ) {}  });  $( ".selector" ).dialog({  drag: **function**( event, ui ) {}  });  $( ".selector" ).dialog({  dragStart: **function**( event, ui ) {}  });  $( ".selector" ).dialog({  dragStop: **function**( event, ui ) {}  });  $( ".selector" ).dialog({  focus: **function**( event, ui ) {}  });  $( ".selector" ).dialog({  open: **function**( event, ui ) {}  });  $( ".selector" ).dialog({  resize: **function**( event, ui ) {}  });  $( ".selector" ).dialog({  resizeStart: **function**( event, ui ) {}  });  $( ".selector" ).dialog({  resizeStop: **function**( event, ui ) {}  }); |

.datepicker

Widgets para fechas

https://jqueryui.com/datepicker/

|  |
| --- |
| **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>jQuery UI Datepicker - Default functionality</title>  <link rel="stylesheet" href="//code.jquery.com/ui/1.13.2/themes/base/jquery-ui.css">  <link rel="stylesheet" href="/resources/demos/style.css">  <script src="https://code.jquery.com/jquery-3.6.0.js"></script>  <script src="https://code.jquery.com/ui/1.13.2/jquery-ui.js"></script>  <script>  $( **function**() {  $( "#datepicker" ).datepicker();  } );  </script>  </head>  <body>    <p>Date: <input type="text" id="datepicker"></p>      </body>  </html> |
| **//opciones**  $( "#datepicker" ).datepicker( "option", "showAnim", $( this ).val() ); |
| showOtherMonths: true,  selectOtherMonths: true |
| showButtonPanel: true |
| changeMonth: true,  changeYear: true |
| //num meses  numberOfMonths: 3,  showButtonPanel: true |
| //formatos de fechas  $( "#datepicker" ).datepicker( "option", "dateFormat", $( this ).val() );  <option value="mm/dd/yy">Default - mm/dd/yy</option>  … |
| //btn con icono  showOn: "button",  buttonImage: "images/calendar.gif",  buttonImageOnly: true,  buttonText: "Select date" |
| //idioma  $.datepicker.regional[ $( this ).val() ] );  <option value="zh-TW">Chinese Traditional (&#x7E41;&#x9AD4;&#x4E2D;&#x6587;)</option> |
| //formatos 2  altField: "#alternate",  altFormat: "DD, d MM, yy"  //DD día de la semana en literal MM mes literal … |
| //limitar la selección  $( "#datepicker" ).datepicker({ minDate: -20, maxDate: "+1M +10D" }); |
| //desde hasta  //https://jqueryui.com/datepicker/#date-range  //Ver ejemplo completo  **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>jQuery UI Datepicker - Select a Date Range</title>  <link rel="stylesheet" href="//code.jquery.com/ui/1.13.2/themes/base/jquery-ui.css">  <link rel="stylesheet" href="/resources/demos/style.css">  <script src="https://code.jquery.com/jquery-3.6.0.js"></script>  <script src="https://code.jquery.com/ui/1.13.2/jquery-ui.js"></script>  <script>  $( **function**() {  **var** dateFormat = "mm/dd/yy",  **from** = $( "#from" )  .datepicker({  defaultDate: "+1w",  changeMonth: true,  numberOfMonths: 3  })  .on( "change", **function**() {  to.datepicker( "option", "minDate", getDate( this ) );  }),  to = $( "#to" ).datepicker({  defaultDate: "+1w",  changeMonth: true,  numberOfMonths: 3  })  .on( "change", **function**() {  **from**.datepicker( "option", "maxDate", getDate( this ) );  });    **function** getDate( element ) {  **var** date;  **try** {  date = $.datepicker.parseDate( dateFormat, element.value );  } **catch**( error ) {  date = null;  }    **return** date;  }  } );  </script>  </head>  <body>    <label for="from">From</label>  <input type="text" id="from" name="from">  <label for="to">to</label>  <input type="text" id="to" name="to">      </body>  </html> |
| showWeek: true,  firstDay: 1 |

$.blockUI

|  |
| --- |
| <script type="text/javascript">      $(document).ready(function() {            $('#blockButton').click(function() {              $('div.test').block({ message: null });          });            $('#blockButton2').click(function() {              $('div.test').block({                  message: '<h1>Processing</h1>',                  css: { border: '3px solid #a00' }              });          });            $('#unblockButton').click(function() {              $('div.test').unblock();          });            $('a.test').click(function() {              alert('link clicked');              return false;          });      });  </script>  <div id="domMessage" style="display:none;">      <h1>Estamos en ello.</h1>  </div> |

$.unblockUI;

<https://jquery.malsup.com/block/#element>

**VALIDACIONES:**

**https://github.com/jquery-validation/jquery-validation**

.rules

<https://jqueryvalidation.org/rules/>

|  |
| --- |
| $( "#myinput" ).rules( "add", { minlength: 2}); |
| //  $( "#myinput" ).rules( "add", {  required: true,  minlength: 2,  messages: {  required: "Required input",  minlength: jQuery.validator.format("Please, at least {0} characters are necessary")  }  }); |
| $( "#myinput" ).rules( "remove", "min max" ); |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| //otros ejemplos   |  |  | | --- | --- | | correo | email: function( value, element ) {  return this.optional( element ) || /^[a-zA-Z0-9.!#$%&'\*+\/=?^\_`{|}~-]+@[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?)\*$/.test( value );  }, | | URL | // <http://jqueryvalidation.org/url-method/>  url: function( value, element ) {  return this.optional( element ) || /^(https?|s?ftp):\/\/(((([a-z]|\d|-|\.|\_|~|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(%[\da-f]{2})|[!\$&'\(\)\\*\+,;=]|:)\*@)?(((\d|[1-9]\d|1\d\d|2[0-4]\d|25[0-5])\.(\d|[1-9]\d|1\d\d|2[0-4]\d|25[0-5])\.(\d|[1-9]\d|1\d\d|2[0-4]\d|25[0-5])\.(\d|[1-9]\d|1\d\d|2[0-4]\d|25[0-5]))|((([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])([a-z]|\d|-|\.|\_|~|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])\*([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])))\.)+(([a-z]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(([a-z]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])([a-z]|\d|-|\.|\_|~|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])\*([a-z]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])))\.?)(:\d\*)?)(\/((([a-z]|\d|-|\.|\_|~|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(%[\da-f]{2})|[!\$&'\(\)\\*\+,;=]|:|@)+(\/(([a-z]|\d|-|\.|\_|~|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(%[\da-f]{2})|[!\$&'\(\)\\*\+,;=]|:|@)\*)\*)?)?(\?((([a-z]|\d|-|\.|\_|~|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(%[\da-f]{2})|[!\$&'\(\)\\*\+,;=]|:|@)|[\uE000-\uF8FF]|\/|\?)\*)?(#((([a-z]|\d|-|\.|\_|~|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(%[\da-f]{2})|[!\$&'\(\)\\*\+,;=]|:|@)|\/|\?)\*)?$/i.test( value );  }, | | fecha | date: function( value, element ) {  return this.optional( element ) || !/Invalid|NaN/.test( new Date( value ).toString() );  }, | | num | number: function( value, element ) {  return this.optional( element ) || /^-?(?:\d+|\d{1,3}(?:,\d{3})+)?(?:\.\d+)?$/.test( value );  }, | | Longitud mínima | minlength: function( value, element, param ) {  var length = $.isArray( value ) ? value.length : this.getLength( value, element );  return this.optional( element ) || length >= param;  }, | | Longitud máxima | maxlength: function( value, element, param ) {  var length = $.isArray( value ) ? value.length : this.getLength( value, element );  return this.optional( element ) || length <= param;  }, | | Longitud con rango | rangelength: function( value, element, param ) {  var length = $.isArray( value ) ? value.length : this.getLength( value, element );  return this.optional( element ) || ( length >= param[ 0 ] && length <= param[ 1 ] );  }, | | Valor mayor que | min: function( value, element, param ) {  return this.optional( element ) || value >= param;  }, | |

.validate

.valid

.serialize

.serializeObject()

**EVENTOS:**

$(document).trigger

.off

.on

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **<!-Quita o pone manejadores de eventos- 🡪**  **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <title>off demo</title>  <style>  button {  margin: 5px;  }  button**#theone** {  color: red;  background: yellow;  }  </style>  <script src="https://code.jquery.com/jquery-3.6.3.js"></script>  </head>  <body>    <button id="theone">Botón inactivo...</button>  <button id="bind">Añade Click</button>  <button id="unbind">Borra Click</button>  <div style="display:none;">Click!</div>    <script>  **function** flash() {  $( "div" ).show().fadeOut( "slow" );  }  $( "#bind" ).click(**function**() {  $( "body" )  .on( "click", "#theone", flash )  .find( "#theone" )  .text( "Btn Activado" );  });  $( "#unbind" ).click(**function**() {  $( "body" )  .off( "click", "#theone", flash )  .find( "#theone" )  .text( " Botón inactivo..." );  });  </script>    </body>  </html>  <https://api.jquery.com/off/> | | **--off**  Remove all event handlers from all paragraphs:  $( "p" ).off();  Remove all delegated click handlers from all paragraphs:  $( "p" ).off( "click", "\*\*" );  Remove just one previously bound handler by passing it as the third argument:   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9 | **var** foo = **function**() {  *// Code to handle some kind of event*  };    *// ... Now foo will be called when paragraphs are clicked ...*  $( "body" ).on( "click", "p", foo );    *// ... Foo will no longer be called.*  $( "body" ).off( "click", "p", foo ); |   Unbind all delegated event handlers by their namespace:   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11 | **var** validate = **function**() {  *// Code to validate form entries*  };    *// Delegate events under the ".validator" namespace*  $( "form" ).on( "click.validator", "button", validate );    $( "form" ).on( "keypress.validator", "input[type='text']", validate );    *// Remove event handlers in the ".validator" namespace*  $( "form" ).off( ".validator" ); | | | **--on .on( events [, selector ] [, data ], handler ) --añade un evento a uno o mas elementos**   |  |  | | --- | --- | |  | $( "#dataTable tbody tr" ).on( "click", **function**() {  console.log( $( this ).text() );  });  $( "p" ).on( "click", **function**() {  alert( $( this ).text() );  }); | | | **//añade párrafo después del clickeado con activación de click sobre el elemento**  **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <title>on demo</title>  <style>  p {  background: yellow;  font-weight: bold;  cursor: pointer;  padding: 5px;  }  p**.over** {  background: #ccc;  }  span {  color: red;  }  </style>  <script src="https://code.jquery.com/jquery-3.6.3.js"></script>  </head>  <body>    <p>Haga click aquí</p>  <span></span>    <script>  **var** count = 0;  $( "body" ).on( "click", "p", **function**() {  $( this ).after( "<p>Nuevo párrafo con evento click justo después del pinchado! " + (++count) + "</p>" );  });  </script>    </body>  </html> | | **//disparador**  //añade evento click para todo div  $( "div" ).on( "click", **function**( event, salutation, name ) {  alert( salutation + ", " + name );  });  //dispara función “click” pasando un JSON de 2 elementos  $( "div" ).trigger( "click", [ "Goodbye", "Jim" ] ); | |  | |

event.preventDefault();

|  |
| --- |
| **//Si es llamado la acción por defecto es cancelada**  **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <title>event.preventDefault demo</title>  <script src="https://code.jquery.com/jquery-3.6.3.js"></script>  </head>  <body>    <a href="https://jquery.com">default click action is prevented</a>  <div id="log"></div>    <script>  $( "a" ).click(**function**( event ) {  event.preventDefault();  $( "<div>" )  .append( "default " + event.type + " prevented" )  .appendTo( "#log" );  });  </script>    </body>  </html> |

event.stopImmediatePropagation();

|  |
| --- |
| **//** **Impide que el resto de los controladores se ejecuten y evita que el evento se expanda en el árbol DOM.**  **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <title>event.stopImmediatePropagation demo</title>  <style>  p {  height: 30px;  width: 150px;  background-color: #ccf;  }  div {  height: 30px;  width: 150px;  background-color: #cfc;  }  </style>  <script src="https://code.jquery.com/jquery-3.6.3.js"></script>  </head>  <body>    <p>paragraph</p>  <div>division</div>    <script>  //  $( "p" ).click(**function**( event ) {  event.stopImmediatePropagation();  });  $( "p" ).click(**function**( event ) {  *// nunca será ejecutada*  $( this ).css( "background-color", "#f00" );  });  $( "div" ).click(**function**( event ) {  *// Será ejecutada div a rojo*  $( this ).css( "background-color", "#f00" );  });  </script>    </body>  </html>  Nota:  stopPropagation will prevent any **parent** handlers from being executed (*Prevents the event from bubbling up the DOM tree, preventing* ***any parent handlers*** *from being notified of the event.*)  stopImmediatePropagation will prevent any **parent** handlers **and also any other** handlers from executing  https://www.carlrippon.com/stoppropagation-v-stopimmediatepropagation/ |

event.isPropagationStopped()

|  |
| --- |
| **//Devuelve si la propagación de eventos fue detenida o no**  **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <title>event.isPropagationStopped demo</title>  <script src="https://code.jquery.com/jquery-3.6.3.js"></script>  </head>  <body>    <button>click me</button>  <div id="stop-log"></div>    <script>  **function** propStopped( event ) {  **var** msg = "";  **if** ( event.isPropagationStopped() ) {  msg = "called";  } **else** {  msg = "not called";  }  $( "#stop-log" ).append( "<div>" + msg + "</div>" );  }    $( "button" ).click(**function**(event) {  propStopped( event );  event.stopPropagation();  propStopped( event );  });  </script>    </body>  </html> |

**OTRAS:**

.select

|  |
| --- |
| //captura la selección de elementos para poder ejecutar código  $( "#target" ).select(**function**() {  alert( "Handler for .select() called." );  }); |
| **<!doctype html>**  <html lang="en">  <head>  <meta charset="utf-8">  <title>select demo</title>  <style>  p {  color: blue;  }  div {  color: red;  }  </style>  <script src="https://code.jquery.com/jquery-3.6.3.js"></script>  </head>  <body>    <p>Click and drag the mouse to select text in the inputs.</p>  <input type="text" value="Con cien cañones por banda,">  <input type="text" value="viento en popa, a toda vela,">  <div></div>    <script>  //: = animada selecciona todos los elementos input capturando la animación seleccionar  $( ":input" ).select(**function**() {  //muestra texto en el div durante un segundo y lo oculta  $( "div" ).text( "Has seleccionado texto de alguna de las cajas Input" ).show().fadeOut( 1000 );  });  </script>    </body>  </html> |

.blur

|  |  |
| --- | --- |
| //evento cuando el objeto pierde el foco  $("input").blur(function(){  alert("This input field has lost its focus.");  });  **Nota**: Short answer: focusin    bubbles, focus does not.          focusout bubbles, blur   does not.  Focus puede ser usado en el padre, además tiene focusin  Ejemplo   |  | | --- | | //HTML:  <div class="parent">  <input type="text" />  </div>  <div class="log"></div>  //Javascript:  $('.parent')  .focusin(function(){log('div focusin');})  .focusout(function(){log('div focusout');})  .focus(function(){log('div focus');})  .blur(function(){log('div blur');});  $('input')  .focusin(function(){log('input focusin');})  .focusout(function(){log('input focusout');})  .focus(function(){log('input focus');})  .blur(function(){log('input blur');});  function log(str){  $('.log').append($('<div/>').text(str));  } | |

.change

Captura los cambios de objeto. Al perder el foco en objetos text y text áreas después del cambio

|  |
| --- |
| Definition and Usage  The change event occurs when the value of an element has been changed (**only works on <input>, <textarea> and <select> elements**).  The change() method triggers the change event, or attaches a function to run when a change event occurs.  Note: For select menus, the change event occurs when an option is selected. For text fields or text areas, the change event occurs when the field loses focus, after the content has been changed.  $("input").change(function(){   alert("The text has been changed."); }); |

.show

.hide

Mostrar u ocultar objeto

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("#hide").click(function(){  $("p").hide();  });  $("#show").click(function(){  $("p").show();  });  });  </script>  </head>  <body>  <p>If you click on the "Hide" button, I will disappear.</p>  <button id="hide">Hide</button>  <button id="show">Show</button>  </body>  </html> |

.bind

.unbind

El método on() adjunta uno o más controladores de eventos para los elementos seleccionados y elementos secundarios.

|  |
| --- |
| The bind() method was deprecated in version 3.0. Use the on() method instead.  The unbind() method was deprecated in version 3.0. Use the off() method instead.  The bind() method attaches one or more event handlers for selected elements, and specifies a function to run when the event occurs. |
| <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("p").on("click", function(){  alert("Se hizo click en el párrafo.");  });  });  </script>  </head>  <body>  <p>Párrafo para hacer click.</p>  </body>  </html> |
| <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("p").on("click", function(){  $(this).css("background-color", "pink");  });  $("button").click(function(){  $("p").off("click");  });  });  </script>  </head>  <body>  <p>Clickea este párrafo para cambiar el color de fondo.</p>  <p>Clickea el botón siguiente y despues clickea el párrafo (el evento de click es borrado).</p>  <button>Elimina el evento click los párrafos</button>  </body>  </html> |

.attr

.removeAttr

Cambia, añade o elimina attributos de uno o varios objetos

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("#btn1").click(function(){  $("img").attr("width", "500" );  $("img").attr("height", "100");  });    $("#btn2").click(function(){  $("p").removeAttr("style");  });  });  </script>  </head>  <body>  <img src="https://www.google.es/images/branding/googlelogo/1x/googlelogo\_color\_272x92dp.png" alt="Pulpit Rock" width="284" height="213"><br>  <button id= "btn1">Establece alto y ancho de la imagen</button>  <button id= "btn2">Elimina estilos de párrafos</button>  <p style="font-size:120%;color:red">Párrafo 1º.</p>  <p style="font-weight:bold;color:blue">Párrafo 2º.</p>  </body>  </html> |
|  |

.prop

Modifica las propiedades de los objetos

“El método prop() establece o devuelve propiedades y valores de los elementos seleccionados.

Cuando este método se utiliza para devolver el valor de la propiedad, devuelve el valor de el PRIMER elemento coincidente.

Cuando este método se utiliza para establecer valores de propiedad, establece uno o más Pares propiedad/valor para el conjunto de elementos coincidentes.

Nota: El método prop() debe utilizarse para recuperar los valores de las propiedades, por ejemplo, propiedades DOM (como tagName, nodeName, defaultChecked) o sus propias propiedades personalizadas.

**Truco**: Para recuperar atributos HTML, utilice el método attr() en su lugar.

**Truco**: Para quitar una propiedad, utilice el método removeProp().”

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("button").click(function(){  var $x = $("div");  $x.prop("color", "FF0000");  $x.append("El color tiene el valor: " + $x.prop("color"));  $x.removeProp("color");  $x.append("<br>El color AHORA tiene el valor: " + $x.prop("color"));  });  });  </script>  </head>  <body>  <button>Añadir o borrar propiedad</button><br><br>  <div></div>  </body>  </html> |

.is

Devuelve boleano si se cumple condición

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("p").click(function(){  if ($("p").parent().is("div")) {  alert("Parent of p is div");  }  });  });  </script>  </head>  <body>  <div>  <p>Click me to find out if I my parent is a div element.</p>  </div>  </body>  </html> |
| <ul>  <li>list <strong>item 1</strong></li>  <li><span>list item 2</span></li>  <li>list item 3</li>  </ul>  $( "ul" ).click(**function**( event ) {  **var** target = $( event.target );  console.log(target);  **if** ( target.is( "li" ) ) {  target.css( "background-color", "red" );  }  }); |
|  |

.css

|  |
| --- |
| //Métodos:  addClass() Agrega uno o más nombres de clase a los elementos seleccionados  after() Inserta contenido después de los elementos seleccionados  append() Inserta contenido al final de los elementos seleccionados  appendTo() Inserta elementos HTML al final de los elementos seleccionados  attr() Establece o devuelve atributos/valores de elementos seleccionados  before() Inserta contenido antes de los elementos seleccionados  clone() Hace una copia de los elementos seleccionados  css() Establece o devuelve una o más propiedades de estilo para elementos seleccionados  detach() Elimina elementos seleccionados (mantiene datos y eventos)  vacío () Elimina todos los nodos secundarios y el contenido de los elementos seleccionados  hasClass() Comprueba si alguno de los elementos seleccionados tiene un nombre de clase específico  height() Establece o devuelve la altura de los elementos seleccionados  html() Establece o devuelve el contenido de los elementos seleccionados  innerHeight() Devuelve la altura de un elemento (incluye relleno, pero no borde)  innerWidth() Devuelve el ancho de un elemento (incluye relleno, pero no borde)  insertAfter() Inserta elementos HTML después de los elementos seleccionados  insertBefore() Inserta elementos HTML antes de los elementos seleccionados  offset() Establece o devuelve las coordenadas de desplazamiento para los elementos seleccionados (en relación con el documento)  offsetParent() Devuelve el primer elemento padre posicionado  outsideHeight() Devuelve la altura de un elemento (incluye relleno y borde)  outsideWidth() Devuelve el ancho de un elemento (incluye relleno y borde)  position() Devuelve la posición (en relación con el elemento principal) de un elemento  prepend() Inserta contenido al principio de los elementos seleccionados  prependTo() Inserta elementos HTML al principio de los elementos seleccionados  prop() Establece o devuelve propiedades/valores de elementos seleccionados  remove() Elimina los elementos seleccionados (incluidos datos y eventos)  removeAttr() Elimina uno o más atributos de los elementos seleccionados  removeClass() Elimina una o más clases de los elementos seleccionados  removeProp() Elimina una propiedad establecida por el método prop()  replaceAll() Reemplaza los elementos seleccionados con nuevos elementos HTML  replaceWith() Reemplaza los elementos seleccionados con contenido nuevo  scrollLeft() Establece o devuelve la posición de la barra de desplazamiento horizontal de los elementos seleccionados  scrollTop() Establece o devuelve la posición de la barra de desplazamiento vertical de los elementos seleccionados  text() Establece o devuelve el contenido de texto de los elementos seleccionados  toggleClass() Alterna entre agregar/eliminar una o más clases de elementos seleccionados  unwrap() Elimina el elemento padre de los elementos seleccionados  val() Establece o devuelve el atributo de valor de los elementos seleccionados (para elementos de formulario)  width() Establece o devuelve el ancho de los elementos seleccionados  wrap() Envuelve elemento(s) HTML alrededor de cada elemento seleccionado  wrapAll() Envuelve elemento(s) HTML alrededor de todos los elementos seleccionados  wrapInner() Envuelve elemento(s) HTML alrededor del contenido de cada elemento seleccionado |
| //Ejemplos |
| <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("#btn1").click(function(){  $("p").append(" <b>Texto añadido</b>.");  });  $("#btn2").click(function(){  $("ol").append("<li>Elemento añadido</li>");  });  });  </script>  </head>  <body>  <p>Párrafo 1.</p>  <p>Párrafo 2.</p>  <ol>  <li>Elemento de lista 1</li>  <li>Elemento de lista 2</li>  <li>Elemento de lista 3</li>  </ol>  <button id="btn1">Añade texto</button>  <button id="btn2">Añade elemento a la lista list item</button>  </body>  </html> |

.toggleClass

.each

.extend

.find

.closest

.html

.empty

.append

$.validator.unobtrusive.parse(document);

|  |
| --- |
|  |

EJEMPLOS DE CODIGO:

$("#btnBuscarCatastro").click(function (event, ui) {

event.preventDefault();

var page = 1;

$.**ajax**({

beforeSend: $.blockUI({ message: 'Buscando ...' }),

type: "POST",

url: '/MVC\_SITRA/Comun/Catastro/BuscarCatastro',

data: {

manzana: $('#FILTRO\_MANZANA').val(),

finca: $('#FILTRO\_FINCA').val(),

cargo: $('#FILTRO\_CARGO').val(),

cod\_calle: $("#FILTRO\_COD\_CALLE").val(),

numero: $("#FILTRO\_NUMERO").val(),

dni: $("#FILTRO\_DNI").val(),

ref\_catastral: $("#FILTRO\_REF\_CATASTRAL").val(),

page: page

},

cache: false,

success: function (response) {

$("#gridCatastro").html(response);

},

complete: function () { $.unblockUI(); }

});

return false;

});

|  |
| --- |
|  |

Contar

|  |
| --- |
| <https://jquery.malsup.com/block/>  <https://jquery.malsup.com/block/#element> |

//autocompleted calles alcobendas

$("#FILTRO\_NOMBRE\_CALLE").**autocomplete**({

source: '/MVC\_SITRA/Comun/Territorio/jsonObtenerCalles',

minLength: 1,

open: function (event, ui) {

$("#FILTRO\_COD\_CALLE").val('');

},

select: function (event, ui) {

$("#FILTRO\_COD\_CALLE").val(ui.item.IDCALLE);

},

search: function () { $(this).addClass('loading'); },

open: function () { $(this).removeClass('loading'); },

change: function (event, ui) {

if (!ui.item) {

$(this).removeClass('loading');

this.value = '';

$("#FILTRO\_NOMBRE\_CALLE").val('');

}

}

});

//autocomplete calles de Alcobendas

$("#@(nombreControl)\_nombreCalle").**autocomplete**({

source: function (request, response) {

$.ajax({

url: '@Url.Action(actionName: "jsonObtenerCalles", controllerName: "Territorio", routeValues: new { area = "Comun" } )',

type: "POST",

data: { term: request.term },

success: function (data) {

response(data);

}

})

},

autofocus: true,

context: this,

delay: 500,

minLength: 1,

select: function (event, ui) {

$("#@(nombreControl)\_nombreCalle").val(ui.item.NOMBRE\_CALLE);

$("#@(nombreControl)\_idCalle").val(ui.item.IDCALLE);

$("#@(nombreControl)\_idTipoCalle").val(ui.item.IDTIPO\_CALLE);

$("#@(nombreControl)\_idTipoCalleCmb").val(ui.item.IDTIPO\_CALLE);

$("#@(nombreControl)\_codigoPostal").val(ui.item.CODIGO\_POSTAL\_CALLE);

if ("@(nombreControl)" != "DireccionNotificacion")

{

ObtenerRefCatastral\_@(nombreControl)();

}

ObtenerCategoriasCalle\_@(nombreControl)();

EnviarEventoCambioCalleNumero\_@(nombreControl)();

return false;

},

open: function (event, ui) {

$(this).removeClass('loading');

$("#@(nombreControl)\_idCalle").val('');

$("#@(nombreControl)\_manzana").val('');

$("#@(nombreControl)\_finca").val('');

},

search: function () {

$(this).addClass('loading');

},

change: function (event, ui) {

if (!ui.item) {

$(this).removeClass('loading');

this.value = '';

$("#@(nombreControl)\_codigoCalle").val('');

$("#@(nombreControl)\_nombreCalle").val('');

$("#@(nombreControl)\_idTipoCalle").val('');

$("#@(nombreControl)\_idTipoCalleCmb").val('');

$("#@(nombreControl)\_codigoPostal").val('');

$("#@(nombreControl)\_manzana").val('');

$("#@(nombreControl)\_finca").val('');

}

}

});

//autocompleted Provincias (Este autocomplete busca por las letras de inicio no por el contenido...).

$("#@(nombreControl)\_nombreProvincia").**autocomplete**({

source: function (request, response) {

var matcher = new RegExp("^" + $.ui.autocomplete.escapeRegex(request.term), "i");

$.ajax({

url: '@Url.Action(actionName: "jsonObtenerProvincias", controllerName: "Territorio", routeValues: new { area = "Comun" } )',

type: "POST",

data: { term: request.term },

success: function (data) {

dataFiltrado = $.grep(data, function (n) {

return matcher.test(n.value);

});

response(dataFiltrado);

}

})

},

autofocus: true,

context: this,

delay: 500,

minLength: 1,

select: function (event, ui) {

$("#@(nombreControl)\_nombreProvincia").val(ui.item.value);

$("#@(nombreControl)\_idProvincia").val(ui.item.IDPROVINCIA);

$("#@(nombreControl)\_idMunicipio").val('');

$("#@(nombreControl)\_nombreMunicipio").val('');

return false;

},

open: function (event, ui) {

$(this).removeClass('loading');

$('#idProvinciaVia').val('');

},

search: function () {

$(this).addClass('loading');

},

change: function (event, ui) {

if (!ui.item) {

$(this).removeClass('loading');

this.value = '';

$("#@(nombreControl)\_idProvincia").val('');

$("#@(nombreControl)\_nombreProvincia").val('');

$(this).closest("table").find('#idMunicipio').val('');

$(this).closest("table").find('#nombreMunicipio').val('');

}

}

$("#errorServidor").**dialog**({

autoOpen: false,

width: "650px",

show: {

effect: "clip",

duration: 700

},

hide: {

effect: "clip",

duration: 700

}

});

function cancelarPago() {

$(document).**trigger**("eventPagoCancelado");

}

$(document).**off**("eventPagoCancelado");

$(document).**on**("eventPagoCancelado", function (event) {

$("#btnImprimirAutoliquidacion").removeAttr('disabled');

$("#dialogPasarelaDePago").empty();

$("#dialogPasarelaDePago").dialog("close");

}

);

if ($("#frmPasarelaDePago").**valid**()) {

$.**ajax**({

beforeSend: $.blockUI({ message: 'Un momento, por favor' }),

type: "POST",

url: '@Url.Action(actionName: "AceptarPago", controllerName: "PasarelaDePago")',



data: $("form").serialize(),



cache: false,

success: function (response) {

$.unblockUI();

var pedido;

pedido = {

HayError: response.HayError,

MsgError: response.MsgError,

NumPedido: response.NumPedido,

ClaveFirma: response.ClaveFirma,

Comercio: response.Comercio,

Terminal: response.Terminal,

Hash: response.Hash,

Autorizacion: response.Autorizacion,

Operacion: response.Operacion,

Respuesta: response.Respuesta,

Importe: response.Importe,

Datos: response.Datos,

DatosAdicionales: response.DatosAdicionales,

IdAplicacion: response.IdAplicacion,

ClaveAplicacion: response.ClaveAplicacion,

Fecha: '@ViewBag.FechaPago'

}

try {

if (response.HayError) {

var event = jQuery.Event("eventPagoFinalizadoConError");

$(document).**trigger**(event, pedido);

if (!event.**isPropagationStopped**()) {

$("#lblErrorPago").text(response.MsgError);

$("#errorServidor").dialog("open");

}

}

else {

$(document).**trigger**("eventPagoFinalizado", pedido);

}

}

catch (ex) { }

},

error: function (request, status, error) { $.unblockUI(); },

});

}

$(document).**off**("eventPagoFinalizado");

$(document).**on**("eventPagoFinalizado",

function (event, pedido) {

var result;

result = "Núm. pedido: " + pedido.NumPedido + "\n";

result += "Clave firma: " + pedido.ClaveFirma + "\n";

result += "Comercio: " + pedido.Comercio + "\n";

result += "Terminal: " + pedido.Terminal + "\n";

result += "Hash: " + pedido.Hash + "\n";

result += "Autorizacion: " + pedido.Autorizacion + "\n";

result += "Importe: " + pedido.Importe + "\n";

result += "Operacion: " + pedido.Operacion + "\n";

result += "Respuesta: " + pedido.Respuesta + "\n";

alert(result);

}

);

// Busca la persona

function ajaxBuscarPersonaSitra(strTipoDocumento, strDocumento, imagenProcesando) {

var PersonaSitra;

$.**ajax**({

type: "POST",

url: '@Url.Action("ObtenerPersonaPorDocumentoYTipo", "PersonaSitra")',

data: { TipoDocumento: strTipoDocumento, Documento: strDocumento },

cache: false,

success: function (response) {

imagenProcesando.hide();

PintarPersonaSitra(response);

PersonaSitra = JSON.stringify(response);

$(document).trigger("eventBotonBuscarPersonaSitra", [PersonaSitra]);

},

beforeSend: function () { imagenProcesando.show(); LimpiarPersonaSitraSeleccionado(); },

error: function (request, status, error) { imagenProcesando.hide(); LimpiarPantalla(); },

complete: function () { imagenProcesando.hide(); }

});

};

$("#btnPagar").click(function (event) {

event.preventDefault();

$.**ajax**({

type: "POST",

url: '@Url.Action(actionName: "PasarelaDePagoAjax", controllerName: "PasarelaDePago", routeValues: new { area = "Comun" })',

data: $("form").serialize(),

cache: false,

success: function (response) {

$("#\_PasarelaDePagoAjax").empty();

$("#\_PasarelaDePagoAjax").html(response);

$("#\_PasarelaDePagoAjax").dialog('open');

},

});

return false;

});

$("#\_PasarelaDePagoAjax").**dialog**({

autoOpen: false,

modal: true,

draggable: true,

closeText: "Cerrar ventana",

closeOnEscape: 'true',

height: 'auto',

width: 'auto',

position: 'center',

resizable: false,

title: 'Pasarela de pago',

open: function (event, ui) {

$(this).dialog('option', 'position', 'center');

},

});

$("#\_GridPersonas").empty();

$.**ajax**({

type: "POST",

beforeSend: $.**blockUI**({ message: 'Buscando ...' }),

url: '@Url.Action(actionName: "ObtenerPersonas", controllerName: "CambioDomiciliacion")',

data: $("#frmBusqueda").serialize(),

cache: false,

success: function (response) {

if (!response.HayError) {

$("#\_GridPersonas").dialog('open');

$("#\_GridPersonas").append(response);

}

else {

MostrarMsjMensaje({ MSJ\_TIPO\_MENSAJE: '3', MSJ\_MENSAJE: response.MensajeError });

}

},

complete: function () {

$.**unblockUI**();

}

});

// CONFIGURACION DE LOS CALENDARIOS DE OCUPACIONES

calendarioOcupaciones = {

duration: "fast",

showAnim: "drop",

showOptions: {direction: "up"},

showButtonPanel: true,

changeMonth: true,

changeYear: true,

onSelect: function () {$(this).valid();$("#frmOcupacion").valid();}

};

$("#FECHA\_INICIO").**datepicker**(calendarioOcupaciones);

$("#FECHA\_FIN").**datepicker**(calendarioOcupaciones);

$( '#F\_Fin\_Autoliquid' ).datepicker( 'option', 'minDate', selectedDate );

$( '#F\_Inicio\_Autoliquid' ).datepicker( 'option', 'maxDate', selectedDate );

// Se fuerza a que entre en el evento de change de la situación..

$("#ID\_ENUM\_SITUACION").**trigger**("change");

// Evento lanzado cuando se ha grabado, modificado o eliminado una ocupacion

$(document).**off**("eventGridOcupacionEliminada eventGridOcupacionGuardada eventGridOcupacionAgregada");

$(document).**on**("eventGridOcupacionEliminada eventGridOcupacionGuardada eventGridOcupacionAgregada",

function (event, idOcupacion) {

event.preventDefault();

BuscarOcupaciones();

return false;

}

);

$("#UNIDADES").**rules**("add", { required: true, messages: { required: "Las unidades son obligatorias" } });

$("#ImporteNegativo").**rules**("add", {

notEqual: 0,

messages: {

notEqual: "El importe no puede ser 0."

}

});

$("#METROS3").**rules**("add", { required: true, regex: /^\d+(\,\d{1,2})?$/, range:[0.01,99999],

messages: { required: "Los metros cúbicos son obligatorios",

regex: "Los metros cúbicos tienen que tener el formato nnnnn,nn",

range: "Los metros cúbicos tienen que estar comprendidos entre 0,01 y 99999" } });

$("#frmPasarelaDePago").**validate**({

messages: {

txtNumeroTarjeta: {

required: "El número de tarjeta es obligatorio",

digits: "El número de tarjeta no es valido",

minlength: "El número de tarjeta debe tener dieciseis dígitos",

maxlength: "El número de tarjeta debe tener dieciseis dígitos"

},

txtMesCaducidadTarjeta: {

required: "El mes de caducidad de la tarjeta es obligatorio",

digits: "El mes de caducidad de la tarjeta no es valido",

minlength: "El mes de caducidad de la tarjeta debe tener dos dígitos",

maxlength: "El mes de caducidad de la tarjeta debe tener dos dígitos"

},

txtAnioCaducidadTarjeta: {

required: "El año de caducidad de la tarjeta es obligatorio",

digits: "El año de caducidad de la tarjeta no es valido",

minlength: "El año de caducidad de la tarjeta debe tener cuatro dígitos",

maxlength: "El año de caducidad de la tarjeta debe tener cuatro dígitos"

},

txtCodigoTarjeta: {

required: "El código de seguridad de la tarjeta es obligatorio",

digits: "El código de seguridad de la tarjeta no es valido",

minlength: "El código de seguridad de la tarjeta debe tener tres dígitos",

maxlength: "El código de seguridad de la tarjeta debe tener tres dígitos"

}

},

errorContainer: $('#errorContainer'),

errorLabelContainer: $('#errorContainer ul'),

wrapper: 'li'

});

var datos = { Residuo: form.**serializeObject**() };

$.**extend**(datos, { Ejercicios: ejercicios });

var datos = form.**serializeObject**();

$.**extend**(datos, { seleccionados: $("#gridRemesaOcupaciones")[0].obtenerSeleccionados() }, { TipoRemesa: TipoRemesa });

Depuración javascript

console.log(objeto/mensage);

// si queremos ver el valor de la variable c

console.log(c);

//

a=1;

b=2;

console.log(a+b);

if (a>0) { console.log(a)};

console.log(jQuery().jquery);

Navegador

|  |
| --- |
| Chrome  Abre el navegador.  En el menú, selecciona "Más herramientas".  En herramientas, elije "Herramientas de desarrollador".  Finalmente, seleccione Consola.  ¿Cómo modificamos la página web en Chrome?  Para ello lo que tienes que hacer es ir a Filesystem, darle al "+" y añadir una nueva carpeta donde esté el código fuente. Seleccionamos la carpeta, y el navegador pedirá permisos para poder acceder al disco duro (lógicamente) porque los navegadores siempre te piden este permiso.  Firefox  Abre el navegador.  En el menú, selecciona "Desarrollador web".  Finalmente, selecciona "Consola web".  Edge  Abre el navegador.  En el menú, selecciona "Herramientas para desarrolladores".  Finalmente, selecciona "Consola".  Ópera  Abre el navegador.  En el menú, selecciona "Desarrollador".  Desde "Desarrollador", selecciona "Herramientas de desarrollador".  Finalmente, selecciona "Consola".  Safari  Abre el Safari, Preferencias, Avanzado en el menú principal.  Marcar la opción "Mostrar menú de desarrollo en la barra de menú".  Cuando aparece la nueva opción "Desarrollar" en el menú:  Elije "Mostrar consola de JavaScript". |
|  |

F12 Fuentes /Fichero definir punto F8 F11 …..

Lista completa

<https://developer.chrome.com/docs/devtools/shortcuts/?utm_source=devtools>

<https://www.campusmvp.es/recursos/post/como-usar-chrome-como-editor-web-profesional-gratuito.aspx>

Eventos Ajax

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ajax Events**  Ajax requests produce a number of different events that you can subscribe to. Here's a full list of the events and in what order they are triggered.  There are two types of events:  **Local Events**  These are callbacks that you can subscribe to within the Ajax request object, like so:   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9 | $.ajax({  beforeSend: **function**(){  *// Handle the beforeSend event*  },  complete: **function**(){  *// Handle the complete event*  }  *// ......*  }); |   **Global Events**  These events are triggered on the document, calling any handlers which may be listening. You can listen for these events like so:  $(document).on("ajaxSend", function(){  $("#loading").show();  }).on("ajaxComplete", function(){  $("#loading").hide();  });  Global events can be disabled for a particular Ajax request by passing in the global option, like so:   |  |  | | --- | --- | | 1  2  3  4  5 | $.ajax({  url: "test.html",  global: false,  *// ...*  }); |   **Events**  This is the full list of Ajax events, and in the order in which they are triggered. The indented events are triggered for each and every Ajax request (unless a global option has been set). The ajaxStart and ajaxStop events are events that relate to all Ajax requests together.   * **ajaxStart** (Global Event) This event is triggered if an Ajax request is started and no other Ajax requests are currently running.   + **beforeSend** (Local Event) This event, which is triggered before an Ajax request is started, allows you to modify the XMLHttpRequest object (setting additional headers, if need be.)   + **ajaxSend** (Global Event) This global event is also triggered before the request is run.   + **success** (Local Event) This event is only called if the request was successful (no errors from the server, no errors with the data).   + **ajaxSuccess** (Global Event) This event is also only called if the request was successful.   + **error** (Local Event) This event is only called if an error occurred with the request (you can never have both an error and a success callback with a request).   + **ajaxError** (Global Event) This global event behaves the same as the local error event.   + **complete** (Local Event) This event is called regardless of if the request was successful, or not. You will always receive a complete callback, even for synchronous requests.   + **ajaxComplete** (Global Event) This event behaves the same as the complete event and will be triggered every time an Ajax request finishes. * **ajaxStop** (Global Event) This global event is triggered if there are no more Ajax requests being processed. |

Selectores :

|  |  |
| --- | --- |
| clase | <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("button").click(function(){  $(".test").hide();  });  });  </script>  </head>  <body>  <h2 class="test">This is a heading</h2>  <p class="test">This is a paragraph.</p>  <p>This is another paragraph.</p>  <button>Click me</button>  </body>  </html> |
| Tipo objeto | $(document).ready(function(){   $("button").click(function(){     $("p").hide();   }); }); |
| Id de objeto | $(document).ready(function(){   $("button").click(function(){     $("#test").hide();   }); }); |
| todos | $("\*") |
| Todas las ref | $("[href]") |
| Todas los registros de las tablas | $("tr") |
| Registros impares | $("tr:odd”) |
|  |  |

[Selectors | jQuery API Documentation](https://api.jquery.com/category/selectors/)

Repaso (Camel con 1ª minúscula)

|  |
| --- |
| Click |
| Dblclick |
| Mouseenter |
| Keypress |
| $("p").on("click", function(){ |
| fadeOut |
| fadeTo(“slow”,0,2) |
| fadeToggle |
| slideToggle |
| $("div").animate({ opacity: '0.4', height: '500px', width: '500px'}); |
| $("div").animate({fontSize: '100px'}, "slow"); |
| $("div").text |
| $("div").html |
| $("input").val(); |
| $("a").attr("href"); |
| $("img").attr("src", "myimage.jpg"); |
| $("#myLink").text("Demo");$("#myLink").attr("href", "https://www.demo.com"); |
| $("p").append("YES!"); |
| $("p").prepend("YES!"); |
| $("div").remove(); |
| $("div").empty(); //borra hijos |
| $("div").remove(".test, .demo"); //borra div de clases |
| $("p").addClass("important"); |
| $("p").addClass("important test"); |
| $("p").removeClass("important"); |
| $("p").click(function(){ $(this).toggleClass("important");}); |
| $("p").css("background-color", "pink"); |
| $("p").css("border", "5px dotted red"); |
| $("p").css("background-color"); |
| $("p").css({ "background-color": "green", "color": "white", "font-size": "25px", "padding": "15px"}); |
| $("div").height() |
| $("div").innerHeight() //con padding |
| $("div").outerHeight() //con padding y bordes |
| //Envolver un elemento <div> para todo <p> existente:  $("button").click(function(){  $("p").wrap("<div></div>");  }); |

$.get('http://localhost:55725/Default', function(response) { console.log(response); });

<https://uniwebsidad.com/libros/fundamentos-jquery/capitulo-8/escribir-extensiones>

[**https://uniwebsidad.com/libros/ajax**](https://uniwebsidad.com/libros/ajax)

lote de ejemplos

C:\Cursos\Ayuntamiento de Alcobendas\Proyectos\jquery-examples-master

<https://api.jquery.com/hover/>

<file:///C:/Cursos/Ayuntamiento%20de%20Alcobendas/jquery-validation-master/demo/index.html?name=aa&email=aa%40aa&url=http%3A%2F%2Faa&comment=a>

docu <https://jqueryui.com/>

desofuscador online

<http://jsnice.org/>

https://editor.swagger.io/

<https://jqueryui.com/autocomplete/#folding>

<http://127.0.0.1:5500/0DAY/index.html>

<http://127.0.0.1:5500/ejemplos3/CURSOPENDIENTE/EJEMPLOS/Animaciones.html>

<http://127.0.0.1:5500/ejemplos3/CURSOPENDIENTE/EJEMPLOS/1Emploconvarios.html>

C:\Users\i.lorente.DANYLOCAL\source\repos111\WebSite1WebformsAsp

<https://jqueryvalidation.org/>

<https://raygun.com/blog/debug-javascript-firefox/>

<https://raygun.com/blog/debug-javascript-visual-studio/>

<https://code.visualstudio.com/docs/editor/debugging>

Selectores

1. Básicos

/\* Seleccionar etiquetas del DOM \*/

$("elemento")

/\* Seleccionar id del DOM \*/

$("#idelemento")

/\* Seleccionar clase del DOM \*/

$(".claseelemento")

/\* seleccionar las etiquetas "p" \*/

$("p")

/\* seleccionar los elementos con id="inicio" \*/

$("#inicio")

/\* seleccionar los elementos con clase="principal" \*/

$(".principal")

/\* seleccionar los elementos con id="inicio" con clase="principal" y que son etiqueta "p" \*/

$("p#inicio.principal")

/\* seleccionar las etiquetas "a" con clase="principal" dentro de una etiqueta "p" \*/

$("p a.principal")

/\* seleccionar los elementos con id="inicio" o clase ="principal" \*/

$("#inicio,.principal")

/\* seleccionar todos los elementos \*/

$("\*")

/\* seleccionar todas las etiquetas "p", "a" y "div" \*/

$("p,a,div")

1. Hijos

|  |  |
| --- | --- |
| 1 | $("p > a") |

En este ejemplo controlamos los enlaces (etiqueta «a») que están dentro de un párrafo (etiqueta «p»). Es decir, que los enlaces (elemento ‘a’) sean hijos directos de ‘p’. Si hubiese otro elemento/etiqueta como un «div» o «span» que contiene al elemento ‘a’ no se controlaría.

|  |  |
| --- | --- |
|  | $("ul.miclase li > p > a") |

En este otro ejemplo estaríamos controlando los enlaces que están dentro de un párrafo que a su vez están en un elemento ‘li’ de una lista con clase = «miclase».

|  |  |
| --- | --- |
| 1 | $("div + a") |

En el código superior controlamos los enlaces (elemento ‘a’) que son precedidos inmediatamente por el elemento ‘div’, es decir que ‘a’ debe ir justo después de ‘div’ sin que ningún otro elemento se interponga entre ellos.

|  |  |
| --- | --- |
| 1 | $("div ~ a") |

Con el código anterior seleccionamos los enlaces (elemento ‘a’) prececidos por cualquier hermano suyo en un ‘div’.

|  |  |
| --- | --- |
| 1 | $("\*") |

Todos los hijos de la página

1. Por atributos

/\* seleccionar los elementos con el atributo href \*/

$("[href]")

/\* seleccionar los elementos con el atributo src \*/

$("[src]")

/\* seleccionar los elementos con el atributo class \*/

$("[class]")

/\* seleccionar los elementos con el atributo id \*/

$("[id]")

1. Atributos y valores

/\* seleccionar elementos con atributo href=index.html \*/

$("[href='index.html']")

/\* seleccionar elementos con atributo href distinto a index.html \*/

$("[href!='index.html']")

/\* seleccionar elementos con atributo src que finalicen con .png \*/

$("[src$='.png']")

/\* seleccionar elementos con atributo src que comiencen con 'imagen.jpg' o con atributo src ='imagen.jpg' \*/

$("[src|='imagen.jpg']")

/\* seleccionar elementos con atributo title que comiencen con 'alergia' \*/

$("[title^='alergia']")

/\* seleccionar elementos con atributo title que contengan la palabra específica 'alergia' \*/

$("[title~='alergia']")

/\* seleccionar elementos con atributo title que contengan la palabra 'alergia' \*/

$("[title\*='alergia']")

1. Posición

/\* seleccionar el primer enlace en la web, elemento 'a' \*/

$("a:first")

/\* seleccionar el último enlace en la web, elemento 'a' \*/

$("a:last")

/\* seleccionar la fila par de una tabla, elemento tr \*/

$("tr:even")

/\* seleccionar la fila impar de una tabla, elemento tr \*/

$("tr:odd")

1. Posición y tipo

/\* seleccionar (elementos 'a') que sean el primer elemento hijo de su elemento padre \*/

$("a:first-child")

/\* seleccionar (elementos 'a') que sean el último elemento hijo de su elemento padre \*/

$("a:last-child")

/\* seleccionar (elementos 'a') que sean el primer elemento hijo de tipo 'a' de su elemento padre \*/

$("a:first-of-type")

/\* seleccionar (elementos 'a') que sean el último elemento hijo de tipo 'a' de su elemento padre \*/

$("a:first-of-type")

/\* seleccionar (elementos 'a') que sean el quinto elemento hijo de su elemento padre \*/

$("a:nth-child(5)")

/\* seleccionar (elementos 'a') que sean el segundo elemento hijo de su elemento padre comenzando por el final \*/

$("a:nth-last-child(2)")

/\* seleccionar (elementos 'a') que sean el segundo elemento hijo de tipo 'a' de su elemento padre \*/

$("a:nth-of-type(2)")

/\* seleccionar (elementos 'a') que sean el cuarto elemento hijo de tipo 'a' de su elemento padre comenzando por el final \*/

$("a:nth-last-of-type(4)")

/\* seleccionar (elementos 'a') que sean el único elemento hijo de su elemento padre \*/

$("a:only-child")

/\* seleccionar (elementos 'a') que sean el único elemento hijo de tipo 'a' de su elemento padre \*/

$("a:only-of-type")

1. Posición dentro de padre

/\* seleccionar el quinto elemento de una lista, el indice comienza en 0 \*/

$("ul li:eq(4)")

/\* seleccionar todos los elementos de una lista a partir de elemento 3, el indice comienza en 0 \*/

$("ul li:gt(2)")

/\* seleccionar todos los elementos de una lista anteriores del elemento 4, el indice comienza en 0 \*/

$("ul li:lt(3)")

1. Múltiples

/\* seleccionar varios elementos, parrafos, divs, span... \*/

$("div, p, span")

/\* seleccionar el primer elemento que sea un enlace con clase igual a 'texto' y con id igual a 'enlace' ... \*/

$("a[class='texto'][id='enlace']:first")

1. Inputs

/\* seleccionar todos los inputs \*/

$(":input")

/\* seleccionar todos los input de tipo text \*/

$(":text")

/\* seleccionar todos los input de tipo password \*/

$(":password")

/\* seleccionar todos los input de tipo radio \*/

$(":radio")

/\* seleccionar todos los input de tipo checkbox \*/

$(":checkbox")

/\* seleccionar todos los input de tipo submit \*/

$(":submit")

/\* seleccionar todos los input de tipo reset \*/

$(":reset")

/\* seleccionar todos los input de tipo button \*/

$(":button")

/\* seleccionar todos los elementos de tipo image \*/

$(":image")

/\* seleccionar todos los elementos de tipo file \*/

$(":file")

/\* seleccionar todos los input habilitados \*/

$(":enabled")

/\* seleccionar todos los input deshabilitados \*/

$(":disabled")

/\* seleccionar todos los input seleccionados (elementos option) \*/

$(":selected")

/\* seleccionar todos los input checkeados (radio button) \*/

$(":checked")

1. Otros

/\* seleccionar todos los input que no sean de tipo button \*/

$("input:not(:button)")

/\* seleccionar todas las etiquetas de encabezados <h1>, <h2>, <h3>... \*/

$(":header")

/\* seleccionar todos los elmentos animados \*/

$(":animated")

/\* seleccionar el elemento en el que está actualmente el foco \*/

$(":focus")

/\* seleccionar el elemento en el que está actualmente el foco \*/

$(":focus")

/\* seleccionar el elemento que contiene el texto 'Hola' \*/

$(":contains('Hola')")

/\* seleccionar todos los párrafos (elemento 'p')  que contienen enlaces (elemento 'a') \*/

$("p:has(a)")

/\* seleccionar todos los elementos vacíos \*/

$(":empty")

/\* seleccionar todos los elementos visibles \*/

$(":visible")

/\* seleccionar todos los elementos párrafos (elemento 'p') visibles \*/

$("p:visible")

/\* seleccionar todos los elementos no visibles \*/

$(":hidden")

/\* seleccionar todos los elementos párrafos (elemento 'p') no visibles \*/

$("p:hidden")

/\* seleccionar el elemento padre o raíz \*/

$(":root")

/\* seleccionar los elementos con hijos no vacíos \*/

$(":parent")

## **Métodos y funciones para efectos en JQuery**

En la siguiente tabla se listan todos los métodos de JQuery, disponibles para crear efectos y animaciones.

|  |  |
| --- | --- |
| Método | Función que realiza |
| animate() | Ejecuta una animación personalizada en los elementos seleccionados |
| clearQueue() | Elimina todas las funciones restantes en cola de los elementos seleccionados |
| delay() | Establece un retardo para todas las funciones en cola en los elementos seleccionados |
| dequeue() | Elimina la siguiente función de la cola y luego ejecuta la función |
| fadeIn() | Se desvanece en los elementos seleccionados |
| fadeOut() | Desvanece los elementos seleccionados |
| fadeTo() | Desvanece los elementos seleccionados a una opacidad dada |
| fadeToggle() | Alterna entre los métodos fadeIn () y fadeOut () |
| finish() | Detiene, elimina y completa todas las animaciones en cola para los elementos seleccionados |
| hide() | Oculta los elementos seleccionados |
| queue() | Muestra las funciones en cola en los elementos seleccionados |
| show() | Muestra los elementos seleccionados |
| slideDown() | Desliza (muestra) los elementos seleccionados |
| slideToggle() | Alterna entre los métodos slideUp () y slideDown () |
| slideUp() | Desliza (oculta) los elementos seleccionados |
| stop() | Detiene la animación en ejecución para los elementos seleccionados |
| toggle() | Alterna entre los métodos hide () y show () |

### **Métodos de JQuery para manipular HTML y CSS**

Lista de todos los métodos y funciones para agregar, eliminar o modificar estilos CSS.  
Además para modificar directamente el contenido HTML.

|  |  |
| --- | --- |
| Método | Función que realiza |
| addClass() | Agrega uno o más nombres de clase a los elementos seleccionados |
| after() | Inserta contenido después de elementos seleccionados |
| append() | Inserta contenido al final de los elementos seleccionados |
| appendTo() | Inserta elementos HTML al final de los elementos seleccionados |
| attr() | Establece o devuelve atributos / valores de elementos seleccionados |
| before() | Inserta contenido antes de elementos seleccionados |
| clone() | Hace una copia de los elementos seleccionados |
| css() | Establece o devuelve una o más propiedades de estilo para los elementos seleccionados |
| detach() | Elimina elementos seleccionados (mantiene datos y eventos) |
| empty() | Elimina todos los nodos secundarios y el contenido de elementos seleccionados |
| hasClass() | Comprueba si alguno de los elementos seleccionados tiene un nombre de clase especificado |
| height() | Establece o devuelve la altura de los elementos seleccionados |
| html() | Establece o devuelve el contenido de los elementos seleccionados |
| innerHeight() | Devuelve la altura de un elemento (incluye el relleno, pero no el borde) |
| innerWidth() | Devuelve el ancho de un elemento (incluye el relleno, pero no el borde) |
| insertAfter() | Inserta elementos HTML después de los elementos seleccionados |
| insertBefore() | Inserta elementos HTML antes de elementos seleccionados |
| offset() | Establece o devuelve las coordenadas de desplazamiento de los elementos seleccionados (en relación con el documento) |
| offsetParent() | Devuelve el primer elemento padre |
| outerHeight() | Devuelve la altura de un elemento (incluye relleno y borde) |
| outerWidth() | Devuelve el ancho de un elemento (incluye relleno y borde) |
| position() | Devuelve la posición (relativa al elemento padre) de un elemento |
| prepend() | Inserta contenido al principio de los elementos seleccionados |
| prependTo() | Inserta elementos HTML al principio de los elementos seleccionados |
| prop() | Establece o devuelve propiedades / valores de elementos seleccionados |
| remove() | Elimina los elementos seleccionados (incluidos datos y eventos) |
| removeAttr() | Elimina uno o más atributos de elementos seleccionados |
| removeClass() | Elimina una o más clases de elementos seleccionados |
| removeProp() | Elimina una propiedad establecida por el método prop () |
| replaceAll() | Reemplaza elementos seleccionados con nuevos elementos HTML |
| replaceWith() | Sustituye elementos seleccionados por nuevos contenidos. |
| scrollLeft() | Establece o devuelve la posición de la barra de desplazamiento horizontal de los elementos seleccionados |
| scrollTop() | Establece o devuelve la posición de la barra de desplazamiento vertical de los elementos seleccionados |
| text() | Establece o devuelve el contenido de texto de los elementos seleccionados |
| toggleClass() | Alterna entre agregar / eliminar una o más clases de elementos seleccionados |
| unwrap() | Elimina el elemento padre de los elementos seleccionados |
| val() | Establece o devuelve el atributo de valor de los elementos seleccionados (para elementos de formulario) |
| width() | Establece o devuelve el ancho de los elementos seleccionados |
| wrap() | Envuelve elementos HTML alrededor de cada elemento seleccionado |
| wrapAll() | Envuelve elementos HTML alrededor de todos los elementos seleccionados |
| wrapInner() | Envuelve elementos HTML alrededor del contenido de cada elemento seleccionado |

### **Métodos de JQuery para AJAX**

Lista de todos los métodos y funciones para AJAX.  
AJAX es el arte de intercambiar datos con el servidor y actualizar el contenido de cualquier página, sin tener que recargarla.

|  |  |
| --- | --- |
| Método | Función que realiza |
| $.ajax() | Realiza una solicitud asíncrona AJAX |
| $.ajaxPrefilter() | Maneje las opciones personalizadas de Ajax o modifique las opciones existentes antes de enviar cada solicitud y antes de que sean procesadas por $ .ajax () |
| $.ajaxSetup() | Establece los valores predeterminados para futuras solicitudes AJAX |
| $.ajaxTransport() | Crea un objeto que maneja la transmisión real de datos Ajax |
| $.get() | Carga datos desde un servidor utilizando una solicitud AJAX HTTP GET |
| $.getJSON() | Carga los datos codificados en JSON de un servidor que utiliza una solicitud HTTP GET |
| $.getScript() | Carga (y ejecuta) un JavaScript desde un servidor utilizando una solicitud AJAX HTTP GET |
| $.param() | Crea una representación serializada de una matriz o un objeto (se puede utilizar como cadena de consulta URL para solicitudes AJAX) |
| $.post() | Carga datos desde un servidor mediante una solicitud AJAX HTTP POST |
| ajaxComplete() | Especifica una función que se ejecutará cuando se complete la solicitud AJAX |
| ajaxError() | Especifica una función que se ejecutará cuando la solicitud AJAX finaliza con un error |
| ajaxSend() | Especifica una función que se ejecutará antes de enviar la solicitud AJAX |
| ajaxStart() | Especifica una función que se ejecutará cuando comience la primera solicitud AJAX |
| ajaxStop() | Especifica una función que se ejecutará cuando todas las solicitudes AJAX hayan finalizado |
| ajaxSuccess() | Especifica una función que se ejecutará cuando se complete una solicitud AJAX correctamente |
| load() | Carga los datos de un servidor y coloca los datos devueltos en el elemento seleccionado |
| serialize() | Codifica un conjunto de elementos de formulario como una cadena para su envío |
| serializeArray() | Codifica un conjunto de elementos de formulario como una matriz de nombres y valores |

### **Otros métodos de JQuery**

Lista métodos y funciones misceláneas, es decir las que no están comprendidas en las anteriores clasificaciones.

|  |  |
| --- | --- |
| Método | Función que realiza |
| data() | Adjunta datos a, o obtiene datos de elementos seleccionados |
| each() | Ejecutar una función para cada elemento coincidente |
| get() | Obtener los elementos DOM acompañados por el selector |
| index() | Buscar un elemento dado entre los elementos emparejados |
| $.noConflict() | Libere el control de jQuery de la variable $ |
| $.param() | Crear una representación serializada de una matriz o un objeto (se puede utilizar como cadena de consulta de URL para solicitudes AJAX) |
| removeData() | Elimina una pieza de datos previamente almacenada |
| size() | Eliminado en la versión 3.0. Utilice la propiedad length en su lugar. |
| toArray() | Recuperar todos los elementos DOM contenidos en el conjunto jQuery, como una matriz |

### **Eventos de JQuery**

Lista de todos los eventos que se pueden usar en JQuery, para ejecutar métodos y funciones.  
Los eventos son los que desencadenan y disparan las acciones necesarias para ejecutar cualquier función.

|  |  |
| --- | --- |
| Evento | Como funciona |
| blur() | Conecta / desencadena el evento de desenfoque |
| change() | Conecta / activa el evento de cambio |
| click() | Conecta / activa el evento de clic |
| dblclick() | Conecta / activa el evento de doble clic |
| event.currentTarget | El elemento DOM actual dentro de la fase de burbujeo del evento |
| event.data | Contiene los datos opcionales transferidos a un método de evento cuando el controlador de ejecución actual esta enlazado |
| event.delegateTarget | Devuelve el elemento donde se ha conectado el controlador de eventos jQuery actualmente llamado |
| event.isDefaultPrevented() | Devuelve si event.preventDefault () fue llamado para el objeto de evento |
| event.isImmediatePropagationStopped() | Devuelve si event.stopImmediatePropagation () se llama para el objeto de evento |
| event.isPropagationStopped() | Devuelve si event.stopPropagation () fue llamado para el objeto de evento |
| event.namespace | Devuelve el espacio de nombres especificado cuando se desencadena el evento |
| event.pageX | Devuelve la posición del ratón en relación con el borde izquierdo del documento |
| event.pageY | Devuelve la posición del mouse en relación con el borde superior del documento |
| event.preventDefault() | Evita la acción predeterminada del evento |
| event.relatedTarget | Devuelve el elemento entrado o salido en el movimiento del ratón. |
| event.result | Contiene el ultimo / anterior valor devuelto por un manejador de eventos disparado por el evento especificado |
| event.stopImmediatePropagation() | Evita que se llamen a otros controladores de eventos |
| event.stopPropagation() | Evita que el evento burbujee el árbol DOM, evitando que los manejadores de origen sean notificados del evento. |
| event.target | Devuelve el elemento DOM que ha desencadenado el evento. |
| event.timeStamp | Devuelve el número de milisegundos desde el 1 de enero de 1970, cuando se desencadena el evento |
| event.type | Devuelve que tipo de evento se ha activado |
| event.which | Devuelve la tecla del teclado o el botón del ratón para el evento |
| focus() | Conecta / activa el evento de enfoque |
| focusin() | Conecta un controlador de eventos al evento focus |
| focusout() | Conecta un controlador de eventos al evento de enfoque |
| hover() | Conecta dos manejadores de eventos al evento hover |
| keydown() | Conecta / activa el evento keydown |
| keypress() | Conecta / activa el evento de pulsaciones de teclas |
| keyup() | Conecta / activa el evento keyup |
| mousedown() | Conecta / activa el evento mousedown |
| mouseenter() | Conecta / activa el evento mouseenter |
| mouseleave() | Adjunta / activa el evento mouseleave |
| mousemove() | Adjunta / activa el evento mousemove |
| mouseout() | Conecta / activa el evento mouseout |
| mouseover() | Conecta / activa el evento mouseover |
| mouseup() | Adjunta / desencadena el evento mouseup |
| off() | Elimina los manejadores de eventos conectados con el método on () |
| on() | Agrega los controladores de eventos a elementos |
| one() | Agrega uno o más controladores de eventos a elementos seleccionados. Este controlador se lo se puede activar una vez por elemento |
| $.proxy() | Toma una función existente y devuelve una nueva con un contexto particular |
| ready() | Especifica una función a ejecutar cuando el DOM está completamente cargado |
| resize() | Adjunta / activa el evento de cambio de tamaño |
| scroll() | Conecta / activa el evento de desplazamiento |
| select() | Conecta / activa el evento seleccionado |
| submit() | Adjunta / activa el evento de envío |
| trigger() | Activa todos los eventos vinculados a los elementos seleccionados |
| triggerHandler() | Dispara todas las funciones enlazadas a un evento especificado para los elementos seleccionados |