

Eventos.

1. Eventos de ratón

1.1. Activados con botones.

- ✓ **onClick**: activado cuando se hace clic en un objeto;
- ✓ **onDbClick**: activado con un doble clic;
- ✓ **onMouseDown**: activado al pulsar el botón izquierdo del ratón;
- ✓ **onMouseUp**: activado cuando deja de pulsarse el botón izquierdo del ratón;
- ✓ **onContextMenu**: activado cuando se pulsa el botón derecho del ratón al abrir el ContextMenu.

1.2. Acciones con movimiento.

- ✓ **onMouseOver**: activado cuando el ratón se mueve por encima de un objeto;
 - ✓ **onMouseOut**: activado cuando el ratón sale de un objeto;
 - ✓ **onMouseMove**: se mueve el cursor del ratón
- Activados al arrastrar el ratón**
- ✓ **onDragDrop**: evento que se activa cuando un usuario arrastra un objeto por la ventana del navegador;
 - ✓ **onMove**: se activa cuando un objeto mueve una ventana o un marco;
 - ✓ **onDragStart**: evento que se activa en cuanto el usuario inicia a arrastrar un objeto;
 - ✓ **onDrag**: se activa cuando el ratón arrastra un objeto o una selección de texto a la ventana del mismo navegador o incluso de otro o al Desktop;
 - ✓ **onDragEnter**: se activa en cuanto el usuario arrastra un objeto a un objetivo válido de ese mismo navegador o de otro;
 - ✓ **onDragOver**: se activa cuando el usuario arrastra un objeto a un objetivo válido para alojarlo; es similar al evento precedente pero es activado después de aquél;
 - ✓ **onDragLeave**: se activa cuando el usuario arrastra un objeto a un objetivo válido para alojarlo, pero no lo suelta;
 - ✓ **onDragEnd**: se activa cuando el usuario suelta el objeto al término del arrastre.
 - ✓ **onDrop**: se activa cuando se deja de pulsar el botón del ratón tras una operación de arrastre;

2. Eventos de teclado

- ✓ **onKeyPress**: evento que se activa cuando un usuario pulsa y deja de pulsar una tecla o también cuando la mantiene pulsada;
- ✓ **onKeyDown**: activado cuando se pulsa la tecla;
- ✓ **onKeyUp**: activado cuando una tecla, que se había pulsado, deja de pulsarse;
- ✓ **onHelp**: activado cuando se pulsa la tecla F1;

3. Eventos de cambio

- ✓ **onChange**: activado cuando el contenido de un campo de un formulario sufre una modificación o no está ya seleccionado;

- ✓ **onCellChange**: activado cuando se modifica un elemento en un Database; por esta característica suya tiene un uso no exactamente vinculado a Javascript;
- ✓ **onPropertyChange**: evento activado cuando cambia la propiedad de un elemento;
- ✓ **onReadyStateChange**: evento activado cuando el estado de carga de un elemento cambia; el evento es útil, por ejemplo, para verificar que un elemento se haya cargado

4. Eventos vinculados al foco

- ✓ **onFocus**: este handler es el contrario de **onBlur** por lo que se activa cuando el usuario entra en un campo;
- ✓ **onBlur**: activado cuando el ratón o el cursor sale de la ventana actual utilizando el ratón o el carácter TAB. Aplicado a los formularios, sin embargo, este handler se activa si se sale del campo cuya marca contiene el control;
- ✓ **onSelect**: activable cuando se selecciona texto dentro de una casilla de texto, tanto con el ratón, como manteniendo pulsado SHIFT y seleccionando con las teclas flecha;
- ✓ **onSelectStart**: se activa cuando se comienza a seleccionar un evento;
- ✓ **onbeforeEditFocus**: se activa con un doble clic o con un clic en un objeto que tiene ya la selección, cuando éste está en DesignMode;
- ✓ **onLoseCapture**: se activa cuando un objeto pierde la captura del ratón.

5. Eventos de carga de objetos

- ✓ **onLoad**: este handler funciona en la subida de objetos, en particular ventanas e imágenes;
- ✓ **onUnload**: es el opuesto del anterior y funciona cuando se deja una ventana para cargar otra o también para volver a cargar la misma (con el botón refresh);
- ✓ **onAbort**: funciona cuando el usuario para la subida de un objeto, bien haciendo clic en otro enlace, bien pulsando la tecla stop del navegador;
- ✓ **onError**: se activa cuando la subida de un objeto causa un error, pero sólo si éste se debe a un error de sintaxis del código y no del navegador, de manera que funciona en un enlace equivocado de una imagen de la página, pero no en un enlace equivocado de subida de una página entera;
- ✓ **onBeforeUnload**: este handler funciona de la misma manera que onUnload pero se carga un momento antes;
- ✓ **onStop**: este handler funciona cuando se para la subida de la página con la tecla stop del navegador y debería funcionar también de la misma manera que onUnload, cargándose antes que éste, pero después de onBeforeUnload.

6. Eventos de movimiento de ventanas

- ✓ **onResize**: este handler se activa cuando el usuario reduce o agranda una ventana o un marco y, en el caso particular de Explorer, un objeto donde estén fijadas la altura y la anchura, así como la posición.

- ✓ **onScroll:** activado cuando se efectúa el scrolling de la página tanto con el ratón, como con las teclas PGUP y PGDOWN o con el método doScroll.

7. Eventos de formulario

- ✓ **onSubmit:** este handler se activa al pulsar el botón de envío de un formulario.
- ✓ **onReset:** este handler se activa al pulsa.