

7.1. ¿Qué método se asocia a un elemento para capturar un evento?

- a) eventCreator().
- b) appendEvent().
- c) addEventListener().
- d) addEvent().

7.2. ¿Dónde indicas qué evento concreto quieres capturar?

- a) En el primer parámetro del gestor de eventos.
- b) En el segundo parámetro del gestor de eventos.
- c) En el tercer parámetro del gestor de eventos.
- d) En el cuarto parámetro del gestor de eventos.

7.3. ¿A qué te refieres al hablar del objeto del evento?

- a) Al elemento sobre el que aplicas un evento.
- b) Al elemento del DOM del que quieres capturar el evento.
- c) Al nombre del evento que quieres capturar.
- d) A un objeto que está asociado a cualquier evento y contiene todas las propiedades y métodos relacionados con el evento.

7.4. ¿Qué quiere decir que un evento se ha propagado?

- a) Que un evento lanzado por un elemento contagia a su hermano.
- b) Que puedes capturarlo desde el nivel más profundo al más superficial en su jerarquía del DOM.
- c) Que puedes capturar eventos de cualquier elemento desde cualquier otro elemento.
- d) Que se ha extendido a otros elementos cualesquiera del DOM.

7.5. ¿Cuántos eventos puedes capturar sobre el mismo elemento?

- a) Aquellos cuyo comportamiento no esté ya predefinido.
- b) Todos los que pueda lanzar ese elemento.
- c) Un máximo de diez.
- d) Solo uno.

7.6. ¿Cuántas veces se puede capturar un mismo evento para el mismo elemento?

- a) Todas las veces que se produzca el evento.
- b) Una única vez; tras esto hay que volver a definir el gestor del evento.
- c) Un máximo de diez veces.
- d) Depende del elemento.

7.7. `preventDefault()` es un método que se utiliza para:

- a) Evitar que se lance un evento que elijas asociado a un elemento.
- b) Cancelar el comportamiento predeterminado de un elemento.
- c) Redefinir un gestor de eventos que ya habías definido.
- d) Definir un gestor de eventos común para un conjunto de elementos.

7.8. Puedes lanzar manualmente eventos sobre:

- a) El elemento que tienes seleccionado.
- b) Cualquier elemento que desees, si es aplicable ese evento.
- c) El último elemento al que se ha accedido.
- d) El primer elemento disponible en el DOM.

7.9. ¿Cómo accederías al tercer formulario que aparece en una página?

- a) `document.forms[1]`.
- b) `document.forms[2]`.
- c) `document.forms[3]`.
- d) `document.forms[4]`.

7.10. ¿Con qué método devuelves el formulario a su estado inicial?

- a) `undo()`.
- b) `rollback()`.
- c) `restart()`.
- d) `reset()`.

7.11. ¿Qué método invocarías para enviar un formulario de manera programática?

- a) `send()`.
- b) `submit()`.
- c) `go()`.
- d) `launch()`.

7.12. ¿Qué contiene `document.forms(0).elements["telefono"].value`?

- a) El nombre del primer formulario, llamado telefono, que aparece en el DOM.
- b) El valor predeterminado que has dado al elemento HTML telefono.
- c) El tamaño máximo del control telefono en un formulario oculto del DOM.
- d) El valor introducido por el usuario en el campo telefono del primer formulario del documento.

7.13. ¿Dónde es recomendable realizar la validación de datos de un formulario?

- a) En el servidor.
- b) En el cliente.
- c) Es indiferente.
- d) No se pueden realizar validaciones de datos de formularios, debido a su carácter dinámico.

7.14. ¿Cuál de los siguientes eventos se lanza sobre un cuadro de entrada de texto (input) cuando lo abandonas para interactuar con otro control del formulario?

- a) keyup.
- b) blur.
- c) dragstop.
- d) mousenter.

7.15. ¿Cómo puedes saber si el usuario ha pulsado la tecla [Ctrl] de la derecha o de la izquierda del teclado?

- a) No puedes saberlo.
- b) Consultando el valor de la propiedad location.
- c) Consultando el valor de la propiedad keyboardZone.
- d) Invocando al método getTheKey().

7.16. Para indicar que un elemento se puede arrastrar debes:

- a) Modificar su propiedad style para bajar la opacidad.
- b) Establecer su atributo class="draggable".
- c) Incluir el elemento draggable="true" en el elemento.
- d) No es necesario indicarlo, solo debes aplicar el gestor del evento sobre ese elemento y automáticamente cualquier elemento se puede arrastrar.

7.17. ¿Qué eventos del portapapeles tienes a tu disposición?

- a) Ninguno, es una utilidad del sistema operativo y JavaScript no tiene acceso.
- b) copy, cut y paste.
- c) flush, import y export.
- d) grab, commit y grant.

7.18. ¿Qué ocurre si añades un elemento a una lista en la que todos los elementos tienen asociado un evento?

a) Automáticamente el nuevo elemento hereda los gestores de eventos de sus hermanos.

b) La respuesta a) sería correcta si al nuevo elemento le añades el atributo `propagation="true"`.

c) El nuevo elemento no tendrá asociado ningún gestor de eventos que no estuviera predefinido por su propia naturaleza.

d) La operación no puede realizarse si previamente no has asociado los mismos eventos al nuevo elemento antes de añadirlo.