Códigos manual jQuery. Parte I, II, III

Fecha entrega: 14/01/2019
Autores: José Rafael Álvarez Espino

1. **En una línea, define qué es jQuery.**

   Es una librería para JavaScript que simplifica la manera de interactuar con el dom.

2. **Identifica la última versión jQuery.**

   La última versión es la 3.3.1

3. **Indica las diferencias entre la versión DEVELOPMENT, PRODUCTION y SLIM.**
   - **Development:** Está con el código sin comprimir, con lo que ocupa más espacio, pero se podrá leer la implementación de las funciones del framework, que puede ser interesante en etapa de desarrollo.
   - **Production:** Es la adecuada para páginas web en producción, puesto que está minimizada y ocupa menos espacio, con lo que la carga de nuestro sitio será más rápida.
   - **Slim:** Es una versión reducida que no contiene algunas funciones como ajax o efectos.

4. **Indica la línea donde introduces todo el código de la librería jQuery.**

   Se introduce dentro del head, en una etiqueta script.

5. **Indica qué es el jQuery CDN.**

   CDN es el acrónimo de red de distribución de contenidos**,** en el caso de jQuery, nos permite incluir las librerías de código del framework desde los servidores de internet.

6. **Indica brevemente y con tus palabras las ventajas del CDN de jQuery.**

   Mayor velocidad de carga, ya que está alojado en servidores potentes. Además, el usuario no deberá de descargar siempre el archivo al entrar a tu web ya que estará almacenado en cache. También jQuery estará actualizado siempre a la última versión.

7. **Indica la línea donde introduces las últimas versiones de al menos dos jQuery CDN.**

   <script src="https://code.jquery.com/jquery-3.3.1.js"></script>

   <script src="https://code.jquery.com/jquery-3.3.1.slim.js"></script>

8. **Indica cómo jQuery ejecuta un código cuando el árbol DOM está totalmente cargado. Indica el equivalente en javaScript.**
   - **jQuery:** $(function(){

     });
   - **JavaScript:** window.addEventListener("load",función);

9. **Función $ o función jQuery. Indica brevemente los argumentos que puedes enviarle. Añádele a la explicación un breve código de ejemplo (distinto al del manual)**
   - jQuery(selector [, context])
   - jQuery(element)
   - jQuery(elementArray)
   - jQuery(object)
   - jQuery(selection)

- jQuery()
- jQuery(html [, ownerDocument])
- jQuery(html, attributes)
- jQuery(callback)

10. **Indica cómo puedes reemplazar el clásico $(document).ready(){...} con**
    $(función(){
    });
11. **En una línea, explica qué hace el método each() de jQuery. Explica qué es la iteración implícita.**

    Es una función de iterador genérico, que se puede utilizar para iterar sin problemas en objetos y matrices.

    Iteración implícita: Nos evita tener que estar programando bucles de código para buscar todos elementos en el DOM que cumplen el criterio dado.
12. **Indica el argumento que ha de enviársele al método each().**

    Una función que puede tener como parámetros índice y elemento.
13. **Englobado en el contexto del each:**
    - **Explica la utilidad de la palabra reservada this.**

      Sirve para acceder al elemento seleccionado.
    - **Indica cómo se utiliza el índice de la iteración.**

      Debe recibir un parámetro la función asignada al each.
    - **Explica la utilidad de return false.**

      El return false para por completo la iteración del each.
    - **Indica la diferencia entre return true y no ponerlo. Explícalo mediante un trozo de código.**

      Si ponemos return true pasará a la siguiente iteración al momento, sin ejecutar las líneas de código que haya después del return.

      ```
      $("p").each(function(i){
      elemento = $(this);
      if(elemento.css(“color”) == "red")
              return true;
      elemento.css(“color”,”red”);
      }
      ```

      Si llega al return true pasará a la siguiente iteración sin ejecutar la línea de abajo.
14. **Indica las diferencias y semejanzas entre el método size() y la propiedad length. Indica las ventajas e inconvenientes de utilizar uno u otra.**

    Tanto size() como length devuelve el número de elementos que hemos seleccionado. La diferencia es que length, al ser una propiedad, es más rápida que el método size().

15. **Indica qué hace el método data().**

    Sirve para almacenar o consultar un dato en un elemento seleccionado. Es decir, podemos guardar en algún elemento del DOM un dato que podemos leer desde cualquier parte.

16. **Tipos de datos admitidos por data()**

    La clave tiene que ser de tipo String, pero el valor puede ser cualquier tipo de JavaScript, incluyendo arrays y objetos.

17. **Indica qué significa que data() almacena valores por referencia.**

    Significa que si nosotros almacenamos un objeto, lo que se está guardado es una referencia a ese objeto y no una copia, por lo que si modificamos ese objeto también estará modificado en el objeto almacenado con data() ya que es el mismo.

18. **Cuántos objetos se crean si data() opera sobre un conjunto de elementos.**

    Una vez por cada elemento que haya en el conjunto.

19. **Indica qué hace el método removeData().**

    Elimina un dato de un elemento guardado con data().

20. **Identifica con tu propio código (en una línea a ser posible) los distintos tipos de selectores. Indica cómo se recogen en una variable.**

    $("h1") -> Selecciona todos los h1

    $("#miParrafo") -> Selecciona una etiqueta que tiene como id miParrafo

    $(".clase1.clase2") -> Selecciona los elementos que tienen class="clase1 clase2"

    $("*") -> Selecciona todos los elementos que tiene la página

    $("div,p") -> Selecciona todos los elementos división y párrafo
    $(".clase1,.clase2") -> Selecciona los elementos que tienen la clase "clase1" o "clase2"

    $("#miid,.miclase,ul) ->Selecciona el elemento con id="miid", los elementos con class="miclase" y todas las listas ul