Práctica códigos manual jQuery. Parte I, II y III

Objetivos

Responder las siguientes preguntas.

1. En una línea, define qué es jQuery.
2. Identifica la última versión jQuery
3. Indica las diferencias entre la versión DEVELOPMENT, PRODUCTION y SLIM.
4. Indica la línea donde introduces todo el código de la librería jQuery.
5. Indica qué es el jQuery CDN.
6. Indica brevemente y con tus palabras las ventajas del CDN de jQuery.
7. Indica la línea donde introduces las últimas versiones de al menos dos jQuery CDN.
8. Indica cómo jquery ejecuta un código cuando el árbol DOM está totalmente cargado. Indica el equivalente en javaScript.
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)
10. Indica cómo puedes reemplazar el clásico $(document).ready(){...} con jQuery
11. En una línea, explica qué hace el método each() de jQuery. Explica qué es la iteración implícita.
12. Indica el argumento que ha de enviarse al método each().
13. Englobado en el contexto del each:
    a. Explica la utilidad de la palabra reservada this.
    b. Indica cómo se utiliza el índice de la iteración.
    c. Explica la utilidad de return false.
    d. Indica la diferencia entre return true y no ponerlo. Explícalo mediante un trozo de código.
14. Indica las diferencias y semejanzas entre el método size() y la propiedad length. Indica las ventajas e inconvenientes de utilizar uno u otra.
15. Indica qué hace el método data().
16. Tipos de datos admitidos por data()
17. Indica qué significa que data() almacena valores por referencia.
18. Cuántos objetos se crean si data() opera sobre un conjunto de elementos.
19. Indica qué hace el método removeData().
20. Identifica con tu propio código (en una línea a ser posible) los distintos tipos de selectores. Indica cómo se recogen en una variable.

Mi equipo:

La práctica ha sido realizada bajo un equipo con sistema operativo MacOs, todo el software usado a sido actualizado a la última versión disponible el día de la práctica.

Preguntas

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

   Es una librería de JavaScript de código abierto que permite agregar interactividad y efectos visuales en un sitio web.

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

   La última versión 3.3.1

3. **Indica las diferencias entre la versión DEVELOPMENT, PRODUCTION y SLIM.**

   La versión **DEVELOPMENT** está pensada para su uso en la etapa de desarrollo, por lo cuál viene sin minificar y de manera legible para el desarrollador. La versión **PRODUCTION** viene de forma minificada para que la carga de la misma sea más eficiente, no es legible para el desarrollador, y por último la versión **SLIM** viene sin algunas funcionalidades, como ajax etc, para que ocupe menos espacio, está pensada para proyectos rápidos y poco complicados.

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

```
// si hacemos uso de CDN

<script
 src="https://code.jquery.com/jquery-3.3.1.js"
 integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60="
 crossorigin="anonymous">
</script>

// bajando la librería e importándola

<script
type=”text/javascript”
 src="rutaDelArchivoJquery>
</script>
```

5. **Indica qué es el jQuery CDN.**

   Una CDN (Red de Distribución de Contenido o Content Delivery Network, por sus siglas en inglés) es un conjunto de servidores ubicados en diferentes zonas geográficas que contienen copias locales de los contenidos de los clientes.

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

   Fundamentalmente, es conveniente utilizar una CDN cuando tu sitio web aspira a ser de carácter internacional o cuando quieres distribuir audio y vídeo a múltiples usuarios. Una CDN es importante si tienes una tienda de comercio electrónico y quieres vender más allá de tu país, al igual que si deseas crear una página de cine en *streaming*.

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

   https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.js

   https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js

8. **Indica cómo jquery ejecuta un código cuando el árbol DOM está totalmente cargado. Indica el equivalente en javaScript.**

   //jquery

   $(handler)

   // javascript vanilla

   window.addEventListener(handler)

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).**

   $("#id") // selector, selecciona los elementos del árbol DOM que coincida con el selector.

   $("<h1>Prueba HTML</h1>") // crear un elemento html.

   $("div", $(this)) // cambiar el contexto del elemento.

10. **Indica cómo puedes reemplazar el clásico $(document).ready(){...} con jQuery.**

    $(handler)

11. **En una línea, explica qué hace el método each() de jQuery. Explica qué es la iteración implícita.**

    El método .each() se utiliza para realizar un recorrido de un objeto jquery, su uso es similar al clásico forEach() de javascript vanilla.

    La iteración implícita, es una característica de jquery, mediante la cual se realiza un recorrido implícito para añadirle una característica o funcionalidad a todos los elementos de ese objeto jquery.

12. Indica el argumento que ha de enviarse al método each().

index -> each(index)

element-> each(element), en el caso de querer realizar una modificación de algún elemento de la collection.

13. Englobado en el contexto del each:

1. Explica la utilidad de la palabra reservada this.

Su utilidad es referenciar el elemento que lo lanza.

2. Indica cómo se utiliza el índice de la iteración.

.each(index)

3. Explica la utilidad de return false.

Corta la iteración de un bucle.

4. Indica la diferencia entre return true y no ponerlo. Explícalo mediante un trozo de código.

return true, rompe la ejecución de una función devolviendo el valor true.

```
let nombre = jesús;
if (nombre === “jesús”)
        return true;
```

14. Indica las diferencias y semejanzas entre el método size() y la propiedad length. Indica las ventajas e inconvenientes de utilizar uno u otra.

Funcionalmente son equivalentes y devuelven el mismo valor, pero en cuanto a rendimiento .size() implica una llamada a una función que en su interior accede a la propiedad .length.

Por duplicidad funcional, y por razones de rendimiento, se consideró obsoleto .size() en jQuery 1.8 y se eliminó definitivamente en jQuery 3.0, por lo que aunque son equivalentes habría que evitar el uso de .size() para que el código funcione en versiones modernas y futuras de jQuery.

15. Indica qué hace el método data().

Nos permite asociar cualquier tipo de datos a un elemento del DOM o colección de ellos, para reutilizarlos en cualquier punto de un script evitando agujeros de memoria.

su sintaxis es :

```
.data( key, value );
```

También podemos asignar varios valores al mismo tiempo pasando al método *data* un objeto compuesto de pares clave-valor:

```
.data( obj );
```

Ejemplo:

```
$( '#user_mail' ).data( 'status', 'send' );

console.log( $( '#user_mail' ).data( 'status' ) ); // send
```

16. Tipos de datos admitidos por data().

Los tipos de datos admitidos son tipo string y objeto.

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

Almacena valores por referencia, para hacer un uso más efectivo de la memoria.

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

Solamente uno.

19. Indica qué hace el método removeData().

Elimina los valores asignados mediante 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.

let selectorId= $(“#id”)

let selectorEtiqueta= $(“div”)

let selectorClase= $(“.clase”)