

Códigos manual jQuery. Parte I, II y III

Entrega de los códigos del manual jQuery. Parte 1, 2 y 3.
En un documento pdf responde a las siguientes preguntas:

1. En una línea, define qué es jQuery.

Es una biblioteca de JavaScript que simplifica bastante el uso de este lenguaje.

2. Identifica la última versión jQuery

La última versión disponible es jQuery 3.1.1 (<https://code.jquery.com/jquery-3.1.1.js>)

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

La versión DEVELOPMENT corresponde a la versión de Desarrollo. Esta versión es la más clara de ver, ya que el código no se encuentra ofuscado para así poder entender de forma correcta los métodos o clases que JQuery posee. Al no estar el código ofuscado ocupa más que las otras versiones.

La versión PRODUCTION corresponde a la versión de Producción. Se ofusca el código para que así su tamaño sea inferior con respecto a la versión de Desarrollo y ofrecer una mayor velocidad en la carga de datos (de esta forma se consigue que la web sea más rápida al cargar los scripts de JQuery).

La versión SLIM es una versión en la que no encontremos algunos métodos/módulos. La mayor ausencia en esta versión es la de Ajax, además de algunos efectos y métodos obsoletos (deprecated). La versión DEVELOPMENT y PRODUCTION poseen una versión SLIM con las características que se han mencionado.

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

Para esto, tan solo deberíamos de importar nuestra librería JQuery (que previamente hemos descargado he introducido en la carpeta js):

```
<script type="text/javascript" src="js/jquery-3.1.1.min.js"></script>
```

Además, tenemos otra posibilidad para tener nuestra librería JQuery siempre actualizada gracias al uso de un CDN. En este caso, podríamos importar JQuery de la siguiente forma:

```
<script src="https://code.jquery.com/jquery-3.1.1.min.js" integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8=" crossorigin="anonymous"></script>
```

Óscar Heredia Tartajo

O incluso, usando otros CDN como el de Google:

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
```

5. Indica qué es el jQuery CDN.

Del inglés Content Delivery Network, es una Red de Entrega de Contenidos que entrega en caché el contenido de una web a sus visitantes. Si en nuestra web usamos JQuery CDN (u otro CDN) siempre tendremos JQuery actualizado ya que, esta red de entrega nos distribuirá los ficheros que forman la biblioteca de JQuery (siempre actualizados).

Estos CDN (servidores) se encuentran distribuidos por todo el mundo. De esta forma, cuándo la web necesite usar estas librerías, se realizará una petición al nodo que se encuentra más cerca. Además, si este nodo no está disponible por alguna razón, se usará otro nodo para poder suministrar a la web que lo necesite la librería en cuestión.

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

Gracias a los CDN, la carga de la librería JQuery será mucho más rápida. Además, cómo se guarda en la caché, es probable que el cliente que visite la web ya haya cacheado el script y no tendrá que esperar a la descarga del mismo.

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

Google CDN:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
```

Microsoft CDN:

```
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.1.1.min.js"></script>
```

CDNJS CDN:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
```

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

JQuery:

```
$(function() { /*Código de la función*/ });
```

Óscar Heredia Tartajo

JavaScript:

```
window.onload = function() { /*Código de la función*/ };
```

o

```
window.onload = init;
```

o

```
window.addEventListener("DOMContentLoaded", init);
```

En ambos casos, init sería la función que ejecutamos cuándo se cargue el DOM.

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)

A la función \$ podemos pasarle:

- selectorCss: en este caso, podemos pasarle una clase o una id para editar su estilo con propiedades css.
- String: creará un nodo HTML en el DOM según el string que reciba.
- Función (callback): podemos pasarle una función que ejecutará una vez el DOM esté totalmente cargado.

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

Se reemplaza por:

```
$(function() { /*Código de la función*/ });
```

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

Realiza iteraciones sobre un conjunto de elementos previamente seleccionados con \$.

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

El argumento que se le envía es la función que ejecutará en cada iteración del conjunto de elementos, y opcionalmente el índice para tratarlo.

13. Englobado en el contexto del each:

1. Explica la utilidad de la palabra reservada this.

Referencia al objeto que está siendo tratado en esa iteración.

Óscar Heredia Tartajo

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

El índice referencia al objeto que está siendo iterado.

3. Explica la utilidad de return false.

Salte del método each()

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

Cuándo colocamos el return true; estamos diciendo que en esa condición, no hará absolutamente nada.

```
$(function(){
    $("div").each(function(i){
        elemento = $(this);
        if(elemento.html() == "white")
            return true;
        if(elemento.html() == "nada")
            return false;
        elemento.css("color", elemento.html());
    });
});
```

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

Ambos devuelven el número de elementos del conjunto de objetos al que se le aplica el método.

15. Indica qué hace el método data().

El método data() actúa de getter y setter en JQuery, todo depende del número de parámetros que reciba:

- Si recibe un solo parámetro, funcionará como un getter y devolverá el valor del parámetro que le estamos pasando.
- Si recibe dos parámetros, funcionará como un setter y asignará a la variable (primer argumento/parámetro) el valor enviado (segundo argumento/parámetro). De esta forma, estaríamos usando un setter que del estilo clave valor.

16. Tipos de datos admitidos por data()

Cualquiera de los elementos que seleccionemos con la función jQuery().

Óscar Heredia Tartajo

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

Esto quiere decir que no se harían copias independientes del objeto a guardar, sino que permanecería tal cual y lo que se asignaría como dato es una referencia a ese único objeto.

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

Se guarda un dato por cada elemento del objeto jQuery seleccionado.

19. Indica qué hace el método `removeData()`. ¿Está sobrescrito?

Con este método conseguimos borrar el valor del parámetro que le pasemos.

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

```
$(function(){
    $("a").click(function(evento){
        selectorClase = $(".miClase").prop("value");
        selectorVariasClases = $(".miClase.miClase2").prop("value");
        selectorId = $("#miId").prop("value");
        selectorVariasIds = $("#miId,#miId2").prop("value");
        selectorElemento = $("p").prop("value");
        selectorVariosElementos = $("p,div").prop("value");
        selectorDescendente = $("div p").prop("value");
        selectorDescendenteDirecto = $("div > p").prop("value");
        selectorHermanos = $("#miId ~ div").prop("value");
        selectorHermanosContiguos = $("#miId + div").prop("value");
        selectorGeneral = $("*").prop("value");
    });
});
```