Tarea 7.1 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 un framework o librería para JavaScriptque contine procesos y rutinas listos para usar que facilitaran la programación al desarrollador.

2. **Identifica la última versión jQuery**

   Existen dos versiones jQuery 1.x y jQuery 2.x. A fecha de creación de este documento 12/01/2015 las últimas versión de la librería son:

   - La última versión de jQuery 1.x se recogen en jQuery 1.11.2
   - La última versión de jQuery 2.x se recogen en jQuery 2.1.3

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

   La versión para producción está minimizada y ocupa menos espacio con lo que la carga del sitio será más rápida.

   La versión DEVELOPMENT no está comprimida con lo que ocupa más espacio, pero en la que se puede leer claramente la implementación de las funciones de la librería.

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

   ```
   <script src="js/jquery-2.1.3.js" type="text/javascript"></script>
   ```

5. **Indica qué es el jQuery CDN.**

   Es un servicio que nos ofrecen las grandes empresas de internet que nos permite incluir las librerías de código jQuery desde sus servidores, sin necesidad de tenerlo en local proporcionándonos mayor velocidad de carga de nuestro sitio.

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

   a. **Ventajas**

      - Mayor velocidad de carga de nuestro sitio.
      - Alta probabilidad de que el cliente de nuestra página tenga una copia en su caché tras haber visitado otra página web que también utilice el CDN.

b. Inconvenientes

- Necesidad de estar conectados a internet para acceder al CDN
- Menor control sobre el script ya que no se podrá modificar si lo necesitas

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

jQuery CDN

Versión 1.11.2

```
<script src="//code.jquery.com/jquery-1.11.2.min.js"></script>

<script src="//code.jquery.com/jquery-migrate-1.2.1.min.js"></script>
```

Versión 2.1.3

```
<script src="//code.jquery.com/jquery-2.1.3.min.js"></script>

<script src="//code.jquery.com/jquery-migrate-2.1.3.min.js"></script>
```

Google CDN

1.x snippet:

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
```

2.x snippet:

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
```

Microsoft CDN

jQuery version 2.1.3

- http://ajax.aspnetcdn.com/ajax/jQuery/jquery-2.1.3.js
- http://ajax.aspnetcdn.com/ajax/jQuery/jquery-2.1.3.min.js
- http://ajax.aspnetcdn.com/ajax/jQuery/jquery-2.1.3.min.map

jQuery version 1.11.2

- http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.11.2.js
- http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.11.2.min.js

- http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.11.2.min.map

jsDelivr CDN

Versión 2.1.3

https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.js

https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js

https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.map

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

```
$(document).ready();
```

El equivalente en JavaScript es `window.onload`

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. Enviando un selector y un contexto

```
//Cambia el color del body a negro

$( document.body ).css( "background", "black" );
```

b. Pasando un HTML

```
//Crea un elemento div y todo su contenido y lo añade al body

$( "<div><p>¡Hola Mundo!</p></div>" ).appendTo( "body" );
```

c. Pasando elementos

```
//Colocará un borde gris a loa párrafos que se encuentren
//dentro de un elemento div

$( "div > p" ).css( "border", "1px solid gray" );
```

d. Pasando una función (callback)

```
//La función será ejecutada después de que el efecto hide se
//haya completado

$(document).ready(function(){

  $("button").click(function(){
```

```
        $("p").hide("slow",function(){
          alert("The paragraph is now hidden");
      });
     });
    });
```

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

```
$(function(){
      //El documento está preparado
});
```

11. En una línea, explica qué hace el método each() de jQuery.

Realiza una iteración sobre el conjunto de elementos que contenga el objeto jQuery.

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

El método each() recibe una función que es la que se tiene que ejecutar para cada elemento, adicionalmente, la función que se envía como parámetro, puede recibir un parámetro que es el índice actual sobre el que se está iterando.

13. Englobado en el contexto del each:

a. Explica la utilidad de la palabra reservada this.

La palabra reservada "this" hace referencia al elemento del DOM que hemos seleccionado con la función que pasamos como argumento.

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

A la función que pasamos como argumento al método each() puede recibir a su vez como parámetro el índice de los elementos que estamos iterando.

c. Explica la utilidad de return false.

La función que enviamos como parámetro al método each() puede devolver valores.

Si la función devuelve "false" se consigue detener por completo el proceso de iteración de each().

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

Cuando la función que pasamos como parámetro devuelve “true” se continúa la iteración del bucle.

```
<!doctype html>
<html lang="en">
<head>
 <meta charset="utf-8">
 <title>each demo</title>
 <style>
  div {
   width: 40px;
   height: 40px;
   margin: 5px;
   float: left;
   border: 2px blue solid;
   text-align: center;
  }
  span {
   color: red;
  }
 </style>
 <script src="js/jquery-2.1.3.js" type="text/javascript"></script>
</head>
<body>

<button>Change colors</button>
<span id="span1"></span>
<span id="span2"></span>
<div></div>
<div ></div>
<div></div>
<div id="continue">Continue</div>
<div id="stop">Stop here</div>
<div></div>
<div></div>
<div></div>

<script>
$( "button" ).click(function() {
 $( "div" ).each(function( index, element ) {
  // element == this
  $( element ).css( "backgroundColor", "yellow" );

  if ( $ ( this ).is( "#continue" ) ) {
   $( element ).css( "backgroundColor", "red" );
   $("#span1").text("Continua la iteración en el div con indice #" + index + ".");
```

```
      return true;
    };
    if ( $( this ).is( "#stop" ) ) {
      $( "#span2" ).text( "Stopped at div index #" + index );
      return false;
    };
  });
});
</script>

</body>
</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.

El método size() devuelve el número de elementos que hay en el objeto jQuery (entero).

La propiedad length nos sirve también para obtener el número de elementos del objeto jQuery seleccionado. La propiedad almacena directamente este valor con lo que es más rápido que hacerlo con el método size().

15. Indica qué hace el método data().

El método data() sirve tanto para guardar un dato en un elemento como para consultarlo. Esto lo hace dependiendo del número de parámetros que reciba:

- Si recibe un parámetro data(nombre): devuelve el valor que haya en el dato cuyo nombre se pasa como argumento.
- Si recibe dos parámetros, data(nombre, valor): almacena un dato, cuyo nombre recibe en el primer parámetro, con el valor que recibe en el segundo parámetro.

16. Tipos de datos admitidos por data().

Al método data() se lepueden pasar dos argumentos

data(clave, valor)

El argumento “clave” es de tipo “string”

El argumento “valor” puede ser de cualquier tipo JavaScript excepto “undefined”

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

En el caso que estemos almacenando un objeto Javascript con data() sobre uno o varios elementos, no se copia el objeto, sino que se asigna 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.**

    No se crea ninguno, si no que el objeto es asignado por refrencia

19. **Indica qué hace el método removeData(). ¿Está sobreescrito?**

    Elimina un dato de un elemento

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, #idelemento, .miclase, .clase1.clase2, p a, p > span, p.parrafo1+p,#parrafo2 ~ a”);

    h1 → Selecciona todos los h1,

    #idelemento → Seleciona una etiqueta que tiene el atributo id=”idelemento”

    .miclase → Selecciona los elementos que tienen clase=”miclase”

    .clase1.clase2 → Selecciona los elementos que tienen clase=”clase1 clase2”

    p a → Selecciona todas las etiquetas a que hay dentro de las etiquetas p.

    p > span → selecciona todas las etiquetas span que son hijas de los párrafos.

    p.parrafo1+p → Selecciona los párrafos que están después de cualquier párrafo que tenga la clase “parrafo1”.

    ,#parrafo2 ~ a → Selecciona los elementos “a” que son hermanos del elemento con identificador “parrafo2”.