

FRANCISCO RAMÍREZ RUIZ

JQuery Parte I II III

14 DE ENERO DEL 2019

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

1.- Define qué es JQuery

Es una librería de JavaScript, de código abierto y permite agregar interactividad a un sitio web.

2.- Indica la última versión JQuery



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

- DEVELOPMENT → Código bien estructurado ideal para aprendizaje.
- PRODUCTION → Está comprimida para ocupar menos espacio.
- SLIM → Solo posee algunas funcionalidades.

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

La línea se introduce dentro de la etiqueta <head>

Si tuviésemos la librería descarga se haría como lo hemos ido implementando a lo largo de las anteriores prácticas

```
<head>
  <title>Francisco Ramírez</title>
  <meta name="author" content="Francisco Ramírez Ruiz">
  <script src="https://code.jquery.com/jquery-3.3.1.js"
    integrity="sha256-2Kok7Mb0yxpgUVvAk/HJ2jigOSYS2auK4Pfbzm7uH60="
    crossorigin="anonymous"></script>
  <link rel="stylesheet" type="text/css" media="screen" href="main.css" />
</head>
```

5.- Indica qué es el JQuery CDN

CDN (Content Delivery Network) es un servicio de servidores externos los cuales permiten obtener los códigos sin necesidad de descargarlos.

6.- Indica las ventajas del CDN de JQuery

La ventaja principal es la mejoría en la velocidad de carga, ya que estos servidores son de grandes empresas y el script se enviará más rápido. Otra ventaja muy importante es la caché, si un cliente o usuario a visitado una página con script JQuery posiblemente tenga cacheado el script y su velocidad en nuestra página será más rápida aún.

7.- Indica la línea de al menos las dos últimas versiones de 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>

<https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.map>

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

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

<https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.slim.min.map>

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

En JavaScript sin librerías sería:

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

En JQuery:

```
$(init);
```

Init es una función, también se podría hacer uso de una función anónima.

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)

Los argumentos que les podemos pasar son elementos del árbol DOM \$("div"), el contexto del selector \$("div", this), un objeto \$ (obj)

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

\$(init)

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

Se utiliza recorrer un objeto JQuery, parecido a JS.

La iteración implícita es usada en los selectos de los elementos del DOM, con ello evitamos bucles para obtener todos los elementos con ese criterio

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

El argumento debe de ser una función, por ejemplo:

```
$( "div" ).each(function() {  
    $( this ).addClass( "elemento" );  
});
```

13.- Englobado en el contexto del each:

1. Explica la utilidad de la palabra reservada this.

La variable this permite referencia al elemento lanzado del DOM

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

El índice recorre los elementos del selector que le hemos indicado

```
$( "div > ul > li > a" ).addClass( "enlace" );
```

3. Explica la utilidad de return false.

Detiene la iteración que se está recorriendo.

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

Con return true se dirige a la siguiente iteración del bucle que se está recorriendo.

Sino se coloca sigue comprobando el resto de código.

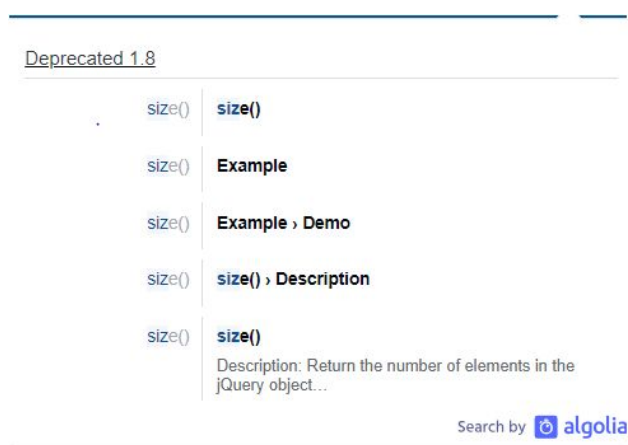
```
let kk = 5;
```

```
if(kk == 5) 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.

Son semejantes, sin embargo `size` devuelve un número de elementos seleccionados mientras que `length` además se encargan de almacenarlos.

El método `size`, hacía uso del método `length` por lo que actualmente está deprecated.



15.- Indica qué hace el método `data()`.

El método `data` permite asociar cualquier tipo de dato a un elemento del DOM.

Dispone de dos sintaxis:

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

```
.data (obj); → objeto javascript
```

16.- Tipos de datos admitidos por data()

Cualquier tipo de Javascript, incluyendo arrays y objetos.

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

Significa que no harían copias sino que, se asignaría a un único objeto.

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

Uno por cada elemento del JQuery objeto seleccionado.

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

Elimina todos 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 elemento = $("elemento")
```

```
let idElemento = $("#idElemento")
```

```
let claseElemento = $(".claseElemento")
```

```
//Cualquier selectores CSS3
```

```
let todo = $("*") → Todos los elementos
```