

## Preguntas jQuery Parte I, II y III

Pablo León Alcaide

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

Biblioteca de JS que permite simplificar la manera de interactuar con los documentos HTML

- **Identifica la última versión jQuery**

Es la version 3.2.1

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

La versión slim tiene funciones eliminadas, la versión de desarrollo está documentada para facilitar su manejo para los desarrolladores, mientras que la versión production suprime dicha documentación.

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

```
<script src="jquery-1.3.2.min.js" type="text/javascript"></script>
```

- **Indica qué es el jQuery CDN.**

Es un servicio que nos permite incluir las librerías de código de jQuery desde los servidores de algunas importantes empresas.

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

Nos permite disponer de una mayor velocidad de entrega de nuestro sitio web. Además permite el cacheado.

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

```
<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
```

- **Indica cómo jquery ejecuta un código cuando el árbol DOM está totalmente cargado.**

**Indica el equivalente en JavaScript.**

jQuery → \$(document).ready

JS → window.onload

- **Función \$ o función jQuery. Indica brevemente los argumentos que puedes enviarle.**

**Añádele a la explicación un breve código de ejemplo (distinto al del manual)**

se pueden pasar elementos, html y funciones

```
let elemento = $(document.body);
```

```
$(function(){
```

...

```
});
```

```
let miLista = $("<ul><li>elemento1</li><li>elemento2</li></ul>");
```

- **Indica cómo puedes reemplazar el clásico `$(document).ready(){...}` con [jQuery `\$\(window\).load\(\)`](#).**
- **En una línea, explica qué hace el método [each\(\)](#) de jQuery. Explica qué es la [iteración implícita](#).**

Itera sobre todos los elementos del DOM que se hayan seleccionado

La iteración implícita se refiere al recorrido que hacen la mayoría de los métodos que devuelven un objeto jQuery recorriendo todos los elementos de la colección jQuery.

Estos casos, no siempre es necesario iterar explícitamente con la función `each()`

Ej.: `($("p")).addClass("misParrafos");`

- **Indica el argumento que ha de enviársele al método `each()`.**

Recibe una función

- **Englobado en el contexto del `each`:**

1. **Explica la utilidad de la palabra reservada `this`.** → Se refiere al elemento que estamos iterando
2. **Indica cómo se utiliza el índice de la iteración.** → El índice es recibido como parámetro en la función que es a su vez parámetro de `each()`
3. **Explica la utilidad de `return false`.** → Nos permite salir de la iteración por completo
4. **Indica la diferencia entre `return true` y no ponerlo. Explícalo mediante un trozo de código.**

`Return true` a diferencia de no ponerlo, termina con la iteración actual y sigue iterando el resto de elementos.

```
$(“p”).each(function(i){
    if (i === 2)
        return true;
    //código.... ← no se ejecutará si i===2, pero seguirá iterando el resto de párrafos
});
```

- **Indica las diferencias y semejanzas entre el método [size\(\)](#) y la propiedad `length`. Indica las ventajas e inconvenientes de utilizar uno u otra.**

**Ambos sirven para obtener el número de elementos seleccionados con la función jQuery.**

`Size` devuelve el número de elementos que hay en el objeto, mientras que `length` almacena directamente dicho valor, por eso es más rápido y aconsejable que utilizar el método `size()`. Además, `size()` está deprecated, lo cual es un importante inconveniente.

- **Indica qué hace el método [data\(\)](#).**

Sirve para mostrar (1 parámetro) y almacenar (2 parámetros) datos, que pueden ser valores numéricos, texto, arrays, objetos...

- **Tipos de datos admitidos por `data()`**

`Data` admite valores numéricos, cadenas de texto, arrays y objetos JS o jQuery

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

Significa que no se copia el objeto, sino que se asigna por referencia, por lo que no se harían copias independientes del objeto al guardarlo, sino distintas referencias al mismo objeto.

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

Se crea un sólo objeto.

- **Indica qué hace el método `removeData()`.**

Elimina el dato almacenado en el elemento

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

Por etiquetas `$("button");`

Por identificador `$("#idElemento");`

Por clase `$(".misParrafos");`

Por varias clases `$(".misParrafos .misListas");`

Todos los elementos `$("*");`

Concatenando varios selectores `$(ul, #menu, .listas);`