

Alumno/a: Mario Navarro Madrid

Cuestionario JQuery

1. En una línea define que es jQuery.

Se trata de un framework para Javascript para añadir funcionalidades sin tener que programarlas.

2. Identifica la última versión de jQuery

La última versión de jQuery es la 3.3.1.

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

La versión development contiene el código correctamente formateado, la production está minificada en una sola línea y la versión slim no contiene ajax ni efectos.

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

Dentro del head, antes de los archivos js que vayan utilizar jQuery.

5. Indica que es el jQuery CDN.

Se trata de cargar jQuery a través de una url en vez de cargarlo localmente.

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

Ofrece una mayor velocidad de carga y si el usuario ya ha visitado otra web con ese CDN, se cacheará y no tendrá que cargarlo en tu web.

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

En la etiqueta head.

8. Indica cómo [jquery](#) ejecuta un código cuando el árbol DOM está totalmente cargado. Indica el equivalente en JavaScript.

Mediante \$(document).ready(). En javascript sería mediante evento, tal que así document.addEventListener("DOMContentLoaded", function()).

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 `$(document).ready(){...}` con [jQuery](#).
`$(function()).`

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

Sirve para iterar un conjunto de elementos seleccionados con la función jQuery.

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

Se le pasa la función que se debe aplicar a los elementos.

13. Englobado en el contexto del `each`:

1. Explica la utilidad de la palabra reservada `this`.

Hace referencia al elemento que se está iterando.

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

Itera los elementos dados por el selector usado.

3. Explica la utilidad de `return false`.

Se sale de la iteración.

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

Al llegar a `true`, pasa a la siguiente iteración. Si no lo ponemos, continúa el código.

```
{
  $(document).ready(function () {
    $("div").each(function (index) {
      if($(this).html() == "white")
        return true; salta a la siguiente iteración
      if($(this).html() == "nada")
        return false; detiene la iteración
      $(this).css("color", $(this).html()); aplica el 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.

Ambos sirven para obtener el número de elementos seleccionados. Length además de obtener el número, lo almacena.

15. Indica qué hace el método data().

Sirve para guardar o cargar información de un elemento.

16. Tipos de datos admitidos por data()

Cualquier tipo, ya se objeto o dato primitivo como cadenas, enteros o flotantes.

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

Los valores no se almacenan de forma independiente, sino que son referencias al objeto o dato almacenado.

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

Se crea un objeto por cada elemento.

19. Indica qué hace el método removeData().

Elimina un dato u objeto 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.

Como vemos en la imagen, podemos utilizar cualquier tipo de selector y su almacenamiento es tan sencillo como almacenarlo.

```
$(document).ready(function () {  
    $("p i").css("color", "#66F");  
    $("table.mitable td").css("background-color", "#55ff00");  
    //selectores parent > child  
    $("p.parrafo-rojo > b").css("color", "red");  
    $(".mitd > *").css("border", "1px solid #ff9900");  
    //selectores prev + next  
    $("i + b").css("font-size", "40px");  
    $(".elem-lista + li").css("opacity", 0.2);  
    //selectores prev ~ siblings  
    $("#a2 ~ div.clase").css("font-size", "180%");  
    $("#miparrafo ~ table").css("border", "3px dotted #dd6600");  
    let p = $("#miparrafo");
```