

Contenido

- JQuery
- Manipulación del DOM con JQuery
- Plugins de JQuery

JQuery

- Biblioteca que simplifica enormemente la programación con JavaScript
- Se considera de fácil aprendizaje y proporcionando un acceso simple a una serie de funciones y métodos de JavaScript ya prediseñados.
- JQuery es una biblioteca de JavaScript rápida, pequeña y con muchas funciones - “Doc. Oficial”
- JQuery **no es un lenguaje de programación si no, una biblioteca de JavaScript.**

jQuery toma muchas tareas comunes que requieren muchas líneas de código JavaScript para llevarlas a cabo y las envuelve en métodos que se pueden llamar con una sola línea de código.

La biblioteca jQuery contiene las siguientes características:

- Manipulación HTML / DOM
- Manipulación de CSS
- Métodos de eventos HTML
- Efectos y animaciones
- **AJAX** (Asynchronous JavaScript and XML)
- Utilidades



JQuery

La propia librería se encargará de ejecutar el código que sea compatible con el browser del cliente que está accediendo a nuestra web.

Sigue **siendo la biblioteca de JavaScript más utilizada**, estando presente en un 77,2% de los sitios web



JQuery

SINTAXIS DE JQUERY

JQuery tiene una sintaxis un poco diferente a JavaScript puro, esta consta de un signo de dólar, que indica que se está trabajando con JQuery y, entre paréntesis, encontraremos indicado el selector al cual nos dirigimos. Esto puede ir seguido de una acción a realizar con este selector.

```
$(selector).acción();
```



jQuery

```
$(selector).evento(function(){  
  // La acción ocurre aquí  
});
```



```
$("p").dblclick(function(){  
  $(this).hide();  
});
```



PLUGINS DE BOOTSTRAP JQUERY

Bootstrap trae consigo plugins de JQuery que amplían su funcionalidad, otorgando un sitio más interactivo. Hablar en detalle de cada uno es complejo, ya que la cantidad de plugins disponibles para uso son muchos, pero podemos nombrar algunos:

- Fuel UX
- Bootstrap Lightbox
- Bootbox.js
- Bootstrap Switch
- JQuery File Upload
- Pick a Color



QUÉ MÁS PODEMOS REALIZAR CON JQUERY

Con la biblioteca JQuery podemos realizar la captura de datos y manipulación de datos del DOM, de la misma forma que lo hacemos con Vanilla JavaScript (JavaScript puro, sin librería ni framework), pero utilizando los beneficios del uso de JQuery.

Podemos crear animaciones, eventos, etc.

Al igual que antes, podemos descargar JQuery o bien **incluirla como un CDN**.



Instalación de JQuery

jQuery se distribuye en diferentes versiones, cada una de ellas adaptada a distintas necesidades de desarrollo y producción.

<https://releases.jquery.com/>

Versión	Descripción	Tamaño Aproximado	Uso
Uncompressed	Completa y sin comprimir	~250KB	Desarrollo y depuración
Minified	Comprimida y minimizada	~90KB	Producción
Slim	Ligera, sin características adicionales	~70KB	Cuando no se necesitan Ajax/efectos
Slim Minified	Ligera y minimizada	~30KB	Producción, sin características adicionales



Métodos de JQuery

Métodos de efectos

Proporcionan varias técnicas para agregar animación a una página web, incluyendo efectos simples y personalizados.

```
animate();  
finish();  
delay();  
hide();  
fadeIn();  
show();  
fadeOut();  
stop();  
toggle();  
fadeTo();  
fadeToggle();
```

Métodos de manipulación de HTML y CSS

Manipulan los elementos, cambiando sus atributos, estilos y/o propiedades.

```
addClass();  
clone();  
after();  
css();  
append();  
detach();  
appendTo();  
empty();  
attr();  
hasClass();  
before();
```



Plugins de JQuery

Existe una gran cantidad de plugins que extienden su funcionalidad y facilitan la implementación de características avanzadas en aplicaciones web.

Plugins Populares:

jQuery UI: añade una amplia gama de widgets interactivos, efectos y temas contruidos sobre el núcleo de jQuery. Componentes como acordeones, autocompletado, botones, diálogos, barras de progreso, deslizadores, etc

```
<link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">  
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
```

<https://jqueryui.com/>



Métodos de JQuery

Métodos útiles y comunes

.ready(): Ejecutar código JavaScript tan pronto como el documento HTML esté completamente cargado y el DOM (Document Object Model). ***“una vez estes listo...”***

.click(): Ejecutar código JavaScript cuando se haga click.

.hover(): Ejecutar el código JavaScript cuando se pase el cursor por encima.



jQuery UI

Es una biblioteca construida sobre jQuery que proporciona componentes de interfaz de usuario (UI) preconstruidos, como widgets, efectos y temas personalizables.

Funcionalidades Principales:

Widgets: Incluye una variedad de widgets interactivos y reutilizables como:

Datepicker: Un calendario para seleccionar fechas.

Accordion: Un panel de contenido que se expande y colapsa.

Dialog: Una ventana modal para mostrar información o capturar datos del usuario.

Slider: Un control deslizante para seleccionar un valor dentro de un rango.

Autocomplete: Un campo de entrada con sugerencias automáticas basadas en la entrada del usuario.

Efectos: Añade efectos visuales avanzados como animaciones de rebote, deslizamiento, explosión, etc., que van más allá de las animaciones básicas de jQuery.

Interacciones: Proporciona comportamientos interactivos como arrastrar y soltar (drag and drop), redimensionar (resizable), seleccionar (selectable), y ordenar (sortable).

¿Vale la pena JQuery?

Aprender jQuery hoy en día es menos esencial que en el pasado, ya que muchas de las funcionalidades que jQuery facilitaba han sido incorporadas directamente en JavaScript (ES5 y versiones posteriores). Sin embargo, hay algunas razones por las que puede ser útil aprenderlo:

Mantenimiento de Proyectos Existentes: Muchos sitios web y aplicaciones más antiguas todavía utilizan jQuery. Si te encuentras trabajando en un proyecto con historia, es probable que necesites entender y trabajar con jQuery.

Simplicidad y Syntactic Sugar: jQuery ofrece una sintaxis simplificada que puede hacer algunas tareas más rápidas y fáciles de implementar, especialmente para manipulaciones del DOM, animaciones y eventos.



¿Vale la pena JQuery?

Aprender jQuery hoy en día es menos esencial que en el pasado, ya que muchas de las funcionalidades que jQuery facilitaba han sido incorporadas directamente en JavaScript (ES5 y versiones posteriores). Sin embargo, hay algunas razones por las que puede ser útil aprenderlo:

Ecosistema de Plugins: jQuery tiene un gran número de plugins disponibles para extender su funcionalidad, aunque muchos de estos plugins también se pueden encontrar en versiones puras de JavaScript o en otras bibliotecas modernas.

Aprendizaje y Comparación: Aprender jQuery puede ser útil para entender cómo han evolucionado las mejores prácticas en desarrollo web y cómo el JavaScript moderno ha simplificado muchas tareas que antes requerían jQuery.



¿Vale la pena JQuery?

Si tu objetivo es aprender habilidades relevantes y modernas, es más recomendable enfocarse en JavaScript nativo (ES6+), junto con frameworks y librerías modernas como **React, Vue.js, o Angular**.





AJAX

Asíncrono

jQuery Ajax Methods

jQuery.get(url, [data], [callback],
[type])

jQuery.getJSON(url, [data],
[callback])

jQuery.getScript(url, [callback])

jQuery.post(url, [data], [callback],
[type])

load(url, data, callback):

serialize()



serializeArray()

ajaxComplete(callback)

ajaxStart(callback)

ajaxError(callback)

ajaxSend(callback)

ajaxStop(callback)

Capacidad para enviar y recibir datos del servidor de manera asíncrona. Esto significa **que la página web puede continuar funcionando y respondiendo a la interacción del usuario mientras se realizan operaciones de red en segundo plano.**

Interactividad: Mejora la experiencia del usuario permitiendo actualizar partes de la página sin necesidad de recargarla por completo.

Rendimiento: Reduce el tráfico de red y el tiempo de carga, ya que solo se intercambian los datos necesarios.

Modularidad: Permite una arquitectura más modular y escalable en las aplicaciones web

¿Qué es una Función de Callback?

Una **función de callback** es simplemente una función que se pasa como argumento a otra función, y que luego se ejecuta (o "llama") después de que esa función haya terminado su trabajo. El término "callback" viene del hecho de que la función original "llama de vuelta" a la función que se le pasó como argumento.

```
function procesarDato(dato, callback) {  
    console.log('Procesando dato:', dato);  
    callback(); // Llamamos a la función de callback }  

```

```
function mostrarResultado() {  
    console.log('El dato ha sido procesado');  
}  
procesarDato('miDato', mostrarResultado);
```

Ejercicio

Resumen de la clase

