



Programación Javascript Avanzado

Document Object Model (DOM).

Manipulación de Páginas HTML

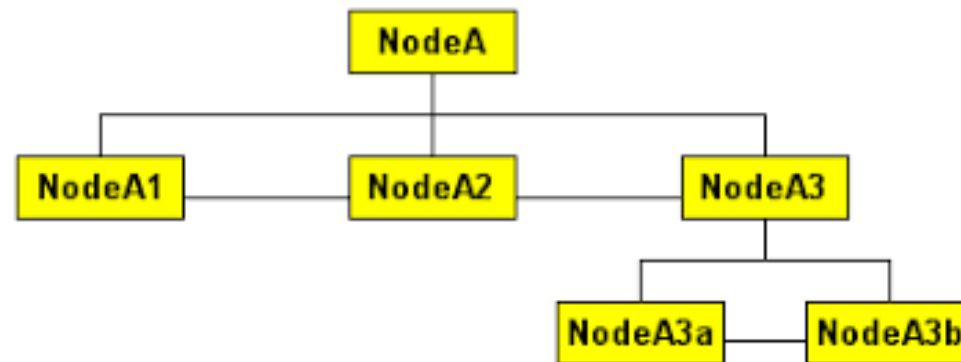
Uso de JQuery

Introducción a DOM



Document Object Model (DOM)

- DOM es un conjunto de utilidades que permiten manipular documentos XML y HTML.
- DOM transforma un archivo HTML en un árbol de nodos jerárquico que puede ser manipulado.
- El navegador web se encarga de generar automáticamente un árbol de nodos a partir de un archivo HTML.



Documento Object Model (DOM)

- Los nodos representan todos los elementos contenidos en la página web.

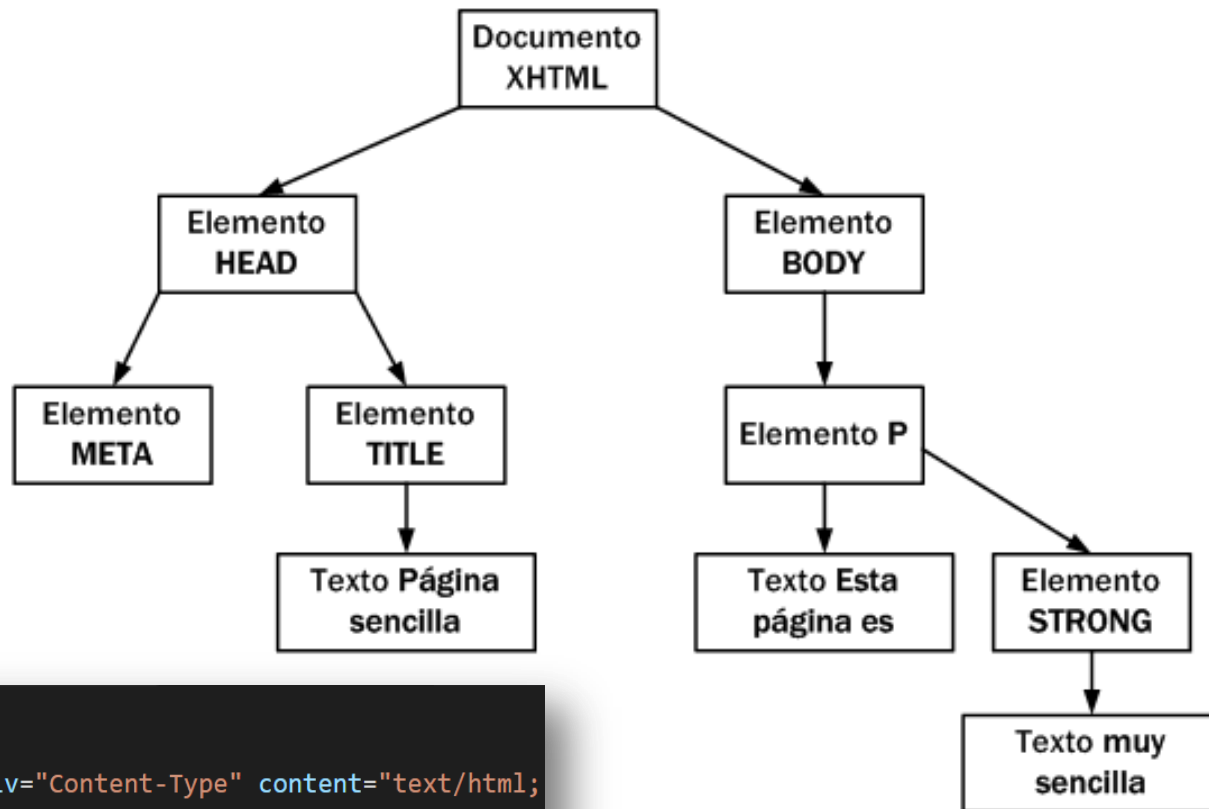
| Nodos | Definición del nodo (elemento HTML) |
|------------------|-------------------------------------------------------------------------------------------|
| Document: | Representa toda la página HTML. Es el nodo raíz. |
| Element: | Representa el contenido definido por un par de etiquetas de apertura y cierre (tag HTML). |
| Attr: | Representa atributos de un elemento, atributo=valor. |
| Text: | Almacena el contenido de texto que se encuentra entre una etiqueta de apertura y cierre. |
| Comment: | Representa los comentarios en el documento HTML. |

JavaScript

Programming Language



Documento Object Model (DOM)



```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html;"
4 <title>Página sencilla</title>
5 </head>
6
7 <body>
8 <p>Esta página es <strong>muy sencilla</strong></p>
9 </body>
10 </html>
```

JavaScript
Programming Language



Métodos de acceso directo

- Funciones básicas de acceso directo a los nodos:

| Funciones | Descripción |
|------------------------------------------|------------------------------------------------------------------------|
| getElementById(id) | Devuelve el elemento HTML con un id específico. |
| getElementsByName(name) | Devuelve los elementos con un name específico (array). |
| getElementsByTagName(tagname) | Devuelve elementos con un tagname específico (array). |
| getElementsByClassName(classname) | Devuelve elementos con un nombre de clase específico (array). |

JavaScript

Programming Language



Métodos de acceso directo

- Funciones básicas de acceso directo a los nodos:

| Funciones | Descripción |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| <code>getAttribute(attributeName)</code> | Devuelve el valor del atributo con nombre <code>attributeName</code> . |
| <code>querySelector(selector)</code> | Devuelve un único elemento de acuerdo con el selector. |
| <code>querySelectorAll(selector)</code> <code>var elem = document.querySelector('#page');</code> <code>var elem = document.querySelector('.main');</code> | Devuelve un (array) con los elementos que correspondan el selector. |

OBSERVACIÓN: Por lo general, los métodos de búsqueda devuelven como resultado un dato de tipo [HTMLCollection](#) o [NodeList](#).



Propiedades de Acceso a contenidos

| Propiedades | Descripción |
|--------------------|-----------------------------------------------------------------------------------------------------------------|
| innerHTML | Devuelve el HTML interior de un elemento; también permite modificar y cambiar contenido (<div>). |
| outerHTML | Similar a innerHTML , pero incluye código HTML del elemento (HTML completo del elemento). |
| textContent | Devuelve contenido de texto, también permite modificar. |
| nodeName | Devuelve el nombre del nodo (elemento HTML). |
| tagName | Devuelve el nombre de la etiqueta HTML correspondiente al nodo. |
| title | Devuelve o permite modificar el valor del atributo title de un nodo. |
| attributes | Devuelve un objeto con todos los atributos que posee un nodo |

JavaScript

Programming Language



Manipulación de nodos

- Funciones básicas de manipulación de nodos:

| Funciones | Descripción |
|---------------------------------------|----------------------------------------------------------------------------------|
| createElement(tagname) | Crea un elemento con el nombre del parámetro <i>tagname</i> . |
| createTextNode(text) | Crea un texto que puede ser añadido a un nodo |
| createTextAttribute(attribute) | Crea un atributo que puede ser añadido a un elemento. |
| nodoPadre.appendChild(node) | Añade un nodo, como hijo de otro nodo. |
| removeAttribute(attribute) | Elimina el atributo de nombre <i>attribute</i> del nodo desde el que es llamado. |

JavaScript

Programming Language



Manipulación de nodos

- Funciones básicas de manipulación de nodos:

| Funciones | Descripción |
|------------------------------------------------|-----------------------------------------------------------------------------------|
| removeChild(<i>child</i>) | Elimina nodo hijo de acuerdo al parámetro <i>child</i> . |
| replaceChild(<i>new</i>, <i>old</i>) | Reemplaza el nodo <i>old</i> por el parámetro <i>new</i> . |
| insertBefore(<i>new</i>, <i>target</i>) | Inserta un elemento <i>new</i> antes del indicado en el parámetro <i>target</i> . |
| | |



Uso de JQuery



Fundamentos. ¿Qué es JQuery?

- **JQuery** es una librería multiplataforma de Javascript que permite simplificar la escritura de programas (scripts).
- Permite manipular DOM, manejar eventos, etc.
- Ventajas que ofrece:
 - ☐ Fácil escritura y comprensión de códigos.
 - ☐ Soportado por la mayoría de los navegadores.
 - ☐ Se combina con sentencias Javascript nativo.
 - ☐ Facilita la utilización de Ajax.



Versiones JQuery

- **Jquery.**

Sentencias básicas DOM, eventos



- **Jquery user interface.**

Interfaces gráficas interactivas



- **JQuery Mobile.**

Aplicaciones móviles



Instalación de JQuery?

- JQuery CDN (*content delivery network*).

```
<script src="https://code.jquery.com/jquery-3.4.1.js" integrity="sha256-WpOohJOqMqqyKL9FccASB9O0KwACQJpFTUBLTYOVvVU=" crossorigin="anonymous"></script>
```



```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```



- Descarga de JQuery.

☐ <http://jquery.com/download/>

☐ <https://code.jquery.com/jquery-3.4.1.min.js>



JavaScript

Programming Language



Sintaxis Básica de JQuery

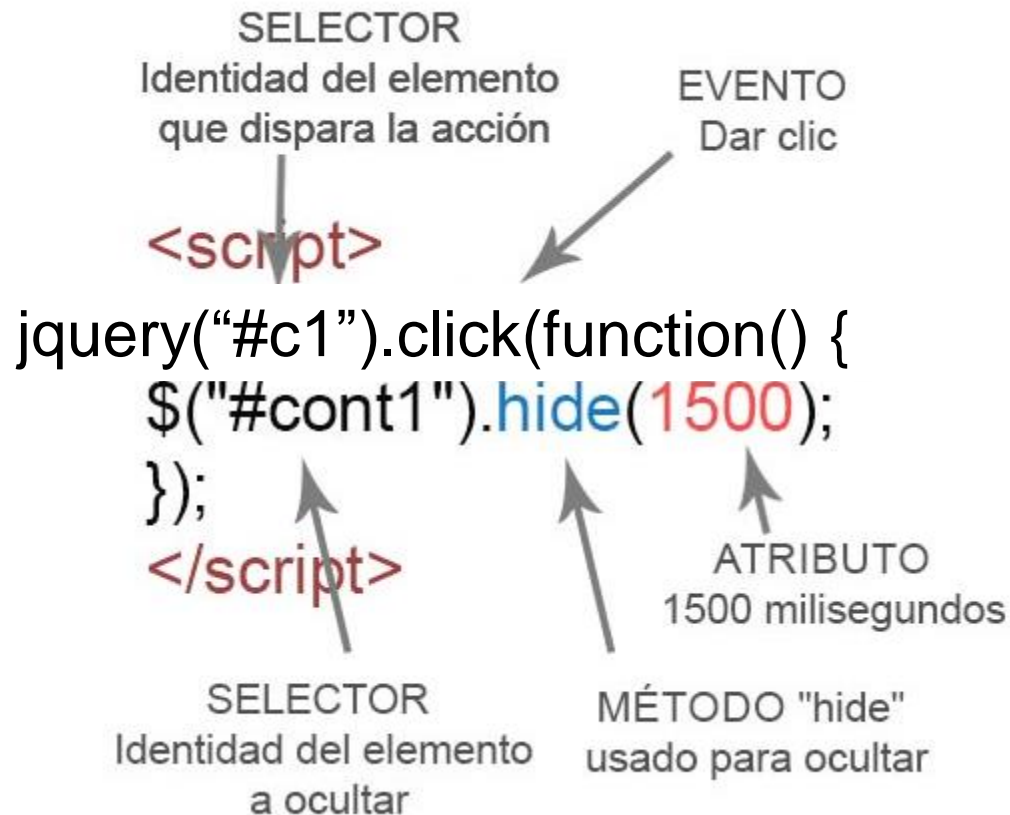
- Consiste en una estructura de cuatro elementos:
 - ❑ Función Query → representada por **\$**
 - ❑ Un selector como parámetro → colocado entre (**" "**).
 - ❑ Uno o varios **métodos** de la librería.
 - ❑ Parámetros del método (**opcionales**).

```
$('elemento').metodo();
```

```
<script type="text/javascript">  
$(document).ready( function(){  
    console.log ("Esto es JQuery");  
});  
</script>
```



Sintaxis Básica de JQuery



Selectores Básicos de JQuery

```
$('p').metodo();
```

- Selección por medio de nombre del elemento.

```
$('#idp1').metodo();
```

- Selección de elemento mediante ID.

```
$('.miClase').metodo();
```

- Selección mediante nombre de clase.



Selectores Básicos de JQuery

```
$('div.miClase').metodo();
```

- Selección mediante tipo de elemento y nombre de clase.

```
$(":text").metodo();
```

- Selecciona elementos de entrada de tipo = “texto”.

```
$("[href='default.htm']").metodo();
```

- Selección mediante atributos.

```
$("div li").metodo();  
$('.contenedor h1').metodo();
```

- Selección de elementos hijos de un elemento padre.



Selectores Básicos de JQuery

```
$("[href]").metodo();
```

- Selección de elementos mediante un atributo.

```
$(this).metodo();
```

- Selecciona el elemento actual (o en uso).

```
$("*").metodo();
```

- Selección mediante atributos.



Manipulación DOM con JQuery

| Propiedades | Descripción |
|-----------------------------|----------------------------------------------------------------|
| <code>.html()</code> | Establece o devuelve el contenido de elementos seleccionados. |
| <code>.text()</code> | Establece o devuelve el contenido de texto de los elementos. |
| <code>.replaceWith()</code> | Sustituye elementos seleccionados por nuevos contenidos. |
| <code>.remove()</code> | Elimina los elementos seleccionados por completo. |
| <code>.prepend()</code> | Inserta contenido al principio de los elementos seleccionados. |
| <code>.append()</code> | Inserta contenido al final de los elementos seleccionados |
| <code>.addClass()</code> | Añade una clase, sin eliminar las ya existentes. |
| <code>.removeClass()</code> | Elimina una clase específica. |
| <code>.css()</code> | Establece o devuelve una o más propiedades de estilo. |

JavaScript

Programming Language



Manipulación DOM con JQuery

| Propiedades | Descripción |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .val() | Establece o devuelve el valor de los <i>input field</i> . |
| .attr() | Devuelve el valor del atributo de un elemento. |
| .each() | Ejecuta una función para cada elemento coincidente. Suele utilizarse para hacer ciclos: <code>\$(selector).each(function(indice,elemento));</code> <code>\$.each(array, function(indice,valor));</code> <code>\$.each(objeto, function(propiedad,valor));</code> |
| .get() | Selecciona una colección de objetos HTML. |
| | |
| | |

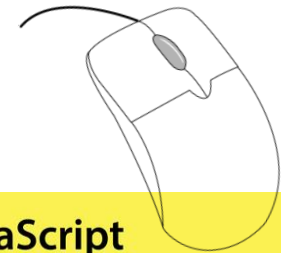
JavaScript

Programming Language



Eventos con JQuery

| Propiedades | Descripción |
|---------------|--------------------------------------------------------------------------------------------------------------------|
| .click() | Activa una función con el evento click |
| .dblclick() | Activa una función con el evento doble click |
| .mouseenter() | Activa una función cuando el mouse pasa sobre el elemento |
| .mouseleave() | Activa una función cuando el mouse abandona el elemento |
| .keyup | Activa una función cuando se dejar de presionar una tecla |
| .keypress | Activa una función cuando se presiona una tecla |
| .change | Activa un función cuando el evento cambio ocurre. <code>\$(selector).change(function) { //instrucción };</code> |



JavaScript
Programming Language



Efectos animados con JQuery

| Propiedades | Descripción |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>.fadeIn()</code> | Entrada o aparición de elementos de forma gradual. |
| <code>.fadeOut()</code> | Desaparición de elementos de forma gradual. |
| <code>.hide()</code> | Ocultar un elemento HTML |
| <code>.show()</code> | Muestra un elemento HTML. |
| <code>.delay()</code> | Genera un retraso en la ejecución de efectos. |
| <code>.toggle()</code> | Alternar dos efectos de mostrar y ocultar alternadamente. |
| <code>.slideUp()</code> | Desliza el elemento seleccionado hacia arriba (oculta). <code>\$(selector).slideUp(velocidad_milisegundos);</code> |
| <code>.slideDown()</code> | Desliza el elemento seleccionado hacia abajo (muestra). |
| <code>.slideToggle()</code> | Desliza el elemento hacia arriba y abajo, alternadamente |



JavaScript

Programming Language



AJAX JQuery

- El método `ajax()` de jQuery es utilizado para realizar una solicitud AJAX o una solicitud HTTP asíncrona.
- **Sintaxis:**

Parámetros

```
$.ajax({nombre: valor, nombre: valor});
```

- ✓ **type:** Especifica el tipo de solicitud (Ej. GET, POST).
- ✓ **url:** Especifica el url hacia el cual se envía la solicitud ajax.
- ✓ **async:** Indica si la solicitud debe manejarse de forma asíncrona o no (valor por defecto: true).
- ✓ **dataType:** Tipo de dato esperado como respuesta del servidor: json, text, etc.
- ✓ **success:** Debe ejecutarse cuando la solicitud sea exitosa.
 - **data:** Contiene los datos devueltos del servidor.
 - **status:** Indica que la conexión fue exitosa.
 - **xhr:** Crea un objeto XMLHttpRequest.



AJAX JQuery

- Ejemplo:

```
$.ajax({dataType: "json",url: "menu.json",success:  
function(data, status, xhr)
```

La variable **xhr** almacena un objeto de tipo **XMLHttpRequest** que almacena datos de una conexión a un servidor: readyState, status, statusText, responseJSON,.responseText, etc.

```
► Object { readyState: 4, getResponseHeader: getResponseHeader() ⚡, getAllResponseHeaders: getAllResponseHeaders() ⚡, setRequestHeader: setRequestHeader() ⚡, overrideMimeType: overrideMimeType() ⚡, statusCode: statusCode() ⚡, abort: abort() ⚡, state: state() ⚡, always: always() ⚡, catch: catch() ⚡, ... }
```



AJAX JQuery

- **readyState:** Almacena el estado **XMLHttpRequest**.
 - 1: Conexión de servidor establecida.
 - 2: Solicitud recibida.
 - 3: Procesando solicitud.
 - 4: Solicitud terminada y respuesta está lista.
- **Status:** Devuelve un número de estado.
 - 200: "OK".
 - 403: "Forbidden".
 - 404: "Page not found"
- **statusText:** Devuelve el estado en formato texto. Ej. "OK".
- **responseText:** Devuelve los datos de respuesta como cadena de texto.
- **responseJSON:** Devuelve los datos de respuesta en formato JSON.



Notación JSON

- JSON es un formato de intercambio datos, basado en texto, que utiliza la sintaxis de objetos de JavaScript.
- Puede ser utilizado de forma independiente de Javascript.
- Debe ser convertido a un objeto nativo Javascript para poder acceder a sus datos (*parsing*).
- Un objeto JSON se almacena en un archivo ***.json**.



Método \$.getJSON

- El método `getJSON()` de jQuery es utilizado para cargar datos codificados en JSON desde el servidor mediante una solicitud GET HTTP.
- **Sintaxis:**

Parámetros

```
$.getJSON(url [, data ] [, success ]);
```

- ✓ **url:** URL a la que se envía la solicitud.
- ✓ **data:** Un objeto simple o cadena que se envía al servidor con la solicitud.
- ✓ **success:** Función que se ejecuta si la solicitud es exitosa.



Método \$.getJSON

- Ejemplo:

```
$.getJSON( "peliculas.json",function(data, status, xhr){  
  //aquí las instrucciones a ejecutar con los datos JSON  
})
```



Método \$.load()

- El método \$.load() de jQuery carga los datos del servidor y coloque el HTML devuelto en los elementos coincidentes.
- **Sintaxis:**

Parámetros

```
$(selector).load(url , function(responseTxt, statusTxt, xhr));
```

- ✓ **url:** URL que se desea cargar .
- ✓ **función:** Función que se ejecuta si la solicitud es exitosa.
 - **responseTxt:** Almacena el contenido resultante de una llama exitosa.
 - **statusTxt:** Indica que la conexión fue exitosa. Muestra estado de la llamada.
 - **xhr:** Crea un objeto XMLHttpRequest.



Solicitudes HTTP \$.get()

- El método \$.get() solicita datos de un recurso específico.
- **Sintaxis:**

Parámetros

```
$.get(url , function(data, status));
```

- ✓ **url:** URL a la que se envía la solicitud.
- ✓ **data:** Guarda el contenido de la página solicitada.
- ✓ **status:** Almacena el estado de la solicitud. Ej. "Success".



Solicitudes HTTP \$.post()

- El método \$.post() envía datos para ser procesados.
- **Sintaxis:**

Parámetros

```
$.post(url , datos , function(data, status));
```

- ✓ **url:** URL a la que se envía la solicitud.
- ✓ **datos:** Datos que se envían con la solicitud.
- ✓ **data:** Almacena contenido de la página solicitada (respuesta).
- ✓ **status:** Almacena el estado de la solicitud. Ej. "Success".

