APIs

Manipulación de elementos del DOM con jQuery

{desafío} latam_

Crear plugins propios con jQuery para ampliar las funcionalidades de una aplicación web.

- Unidad 1:
  Introducción al lenguaje JavaScript
- Unidad 2:
  Funciones y Ciclos
- Unidad 3:
  Arrays y Objetos
- Unidad 4:
  APIs

Te encuentras aquí

{desafío}
latam_

¿Qué aprenderás en esta sesión?

- *Desarrolla algoritmos transformando código de JavaScript a jQuery para utilizar correctamente la sintaxis de la librería.*
- *Crea un script que manipule elementos del DOM con selectores utilizando la librería jQuery para resolver un problema planteado.*
- *Crea un script que permita la selección y manipulación de elementos del DOM utilizando la librería jQuery para resolver un problema planteado.*

De acuerdo a lo aprendido, ¿cuál es la principal diferencia entre los métodos find y filter?

{desafío} latam_

¿Qué método permite verificar que existe algún elemento dentro de un arreglo que cumpla una expresión determinada?

/* Fundamentos de jQuery */

Fundamentos de jQuery

¿Qué es jQuery?

- jQuery es una librería de JavaScript, que permite crear páginas web dinámicas de forma muy rápida y con la menor cantidad de código posible en JavaScript. El lema principal de jQuery es “escribir menos, hacer más”.

- Ejemplo de código en JavaScript

```
document.getElementById('test');
```

- Ejemplo de código en jQuery

```
$('#test');
```

Demostración - “Transformando JavaScript a jQuery”

Ejercicio guiado

Transformando JavaScript a jQuery

El siguiente código, escrito con la sintaxis de JavaScript puro, obtiene un elemento con id ‘text’, una vez que el usuario realiza click. Se solicita transformar el código JavaScript con la librería de jQuery, para evidenciar sus diferencias. El código es el siguiente:

```
let text = document.getElementById('texto');
text.addEventListener('click', function(){
   document.write('click sobre el texto');
});
```

Ejercicio guiado

Transformando JavaScript a jQuery

- **Paso 1:** Llevar la instrucción `let text = document.getElementById('texto');` a jQuery, utilizando el método o símbolo especial "$" que permite abreviar el document.getElementById, indicando en los paréntesis del método que estaremos buscando, un selector denominado "texto" el cual es un "id". Ese resultado lo guardamos en la variable "text".

```
let text = $('#texto');
```

Ejercicio guiado

Transformando JavaScript a jQuery

- **Paso 2:** Como ya se encuentra el elemento guardado en una variable, se puede utilizar directamente para asignarle un evento, en este caso, el evento “click” de JavaScript. El cual, en jQuery viene asignado con una función que se ejecutará cuando el evento se active y realizará todos los procesos que contenga dentro de ella. Es decir, el evento “click” pasa a ser un método en jQuery exclusivo para ese tipo de evento.

```
let text = $('#texto');
text.click(function(){
    document.write('click sobre el texto');
});
```

Fundamentos de jQuery

¿Por qué utilizamos jQuery?

- Licencia MIT y Open Source.
- Soporte
- Compatibilidad con todos los navegadores web en el mercado.
- Manipulación del modelo de objeto de dominio.
- Manipulación dinámica de estilos CSS.
- Integración con AJAX.
- Creación de animaciones de forma intuitiva y sencilla.
- Compatibilidad con una lista de Plugins creados a partir de jQuery.
- Librería ligera.
- Capacidad de optimización de tiempo y orden de código.

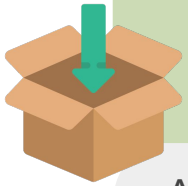
/* Integrando jQuery en un proyecto */

Integrando jQuery en un Proyecto

- Es importante señalar que, dependiendo del proyecto en el cual se esté trabajando, se puede integrar jQuery.

Utilizar un recurso en un servidor remoto (CDN)

Importar la librería jQuery como un <script> en la cabeza del archivo principal o main de un proyecto, referenciando el recurso de tipo text/JavaScript.



Descargar la librería y añadirla manualmente al proyecto.

Al momento de subir tu página web, este archivo debe estar cargado, ya sea en el mismo servidor de la página web, o bien, en otro servidor propio donde se manejen los archivos.

Demostración - “Implementando jQuery mediante CDN”

{desafío} latam_

Ejercicio guiado

Implementando jQuery mediante CDN

Implementar la librería jQuery mediante el uso de CDN con el siguiente código, que obtiene el texto del elemento con id ‘texto’ cuando se recibe el evento click:

```
let text = $('#texto');
text.click(function(){
    document.write('click sobre el texto');
});
```

- **Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crea dos archivos, un index.html y un script.js.

Ejercicio guiado

Implementando jQuery mediante CDN

- **Paso 2:** En el index.html, escribir la estructura básica de un documento HTML como se muestra a continuación:

```
<!DOCTYPE html>
<html lang="en">
<head>
     <meta charset="UTF-8">
     <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
     <title>Document</title>
</head>
<body>
     <h4>Este es un documento HTML
con JavaScript</h4>
   <script src="script.js"></script>
</body>
</html>
```

Ejercicio guiado

Implementando jQuery mediante CDN

- **Paso 3:** Lo primero es incorporar el enlace o CDN de jQuery, que nos permita incluir esta librería en nuestro ejemplo. Puedes conseguir los enlaces en cualquiera de las opciones que facilita la página de jQuery en la sección de descarga. Para este ejemplo, utilizaremos el enlace que se encuentra con la versión: 3.x.

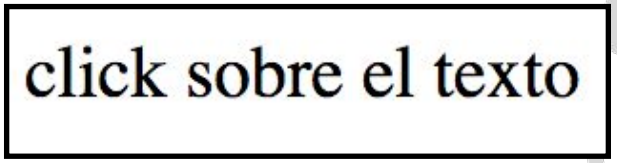
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
   <script
src="https://code.jquery.com/jquery-3.7.1
.min.js"></script>
   <title>Document</title>
</head>
<body>
   <h4 id="texto">Este es un documento
HTML con JavaScript</h4>
   <script src="script.js"></script>
</body>
</html>
```

Ejercicio guiado

Implementando jQuery mediante CDN

- **Paso 4:** En el archivo script.js, agregar el código dispuesto al inicio del ejemplo para poder ejecutarlo en el siguiente paso y ver el resultado. El código consiste en seleccionar un elemento mediante el tipo de selector “id”, guardarlo en una variable y cuando se haga un clic sobre ese elemento se activa un mensaje que se mostrará en el documento:

```
let text = $('#texto');
text.click(function(){
    document.write('click sobre el texto');
});
```



/* Manipulación de elementos del DOM con jQuery */

¿Qué es Modelo de Objetos de Documento?

- Recordemos que los objetos del DOM modelan tanto la ventana del navegador como el historial, el documento o página web y todos los elementos que pueda tener dentro la propia página, como párrafos, divisiones, tablas, formularios y sus campos, etc.

- A través del DOM se puede acceder, por medio de JavaScript, a cualquier elemento, es decir a sus correspondientes objetos para alterar sus propiedades o invocar a sus métodos.

Manipulando DOM con jQuery

- Al igual que con JavaScript, jQuery nos permite manipular cualquiera de los elementos del DOM, pero de manera muy sencilla utilizando el objeto $, el cual es acompañado de algún selector como los que se utilizan en CSS.

- Para asegurarse que todos los elementos HTML se han cargado completamente en el DOM, jQuery dispone de un método llamado $(document).ready,

```
$(document).ready(function() {
    // Escribe acà el código que desees que se ejecute
una vez cargados los elementos HTML del DOM
});
```

Manipulando DOM con jQuery

Selector ID

- El selector de ID es uno de los selectores más utilizados, tanto en jQuery como en JavaScript puro.

- La forma correcta de escribir un selector con jQuery es:

```
$(document).ready(function() {
    $('#some');
});
```

En donde #some es el id de algún componente HTML

Demostración - “Seleccionar elementos por id con jQuery”

Ejercicio guiado

Seleccionar elementos por id con jQuery

Seleccionar mediante jQuery los elementos que se encuentran en un archivo HTML, guardarlos en una variable y mostrar las variables mediante un console.log, siendo estos elementos una etiqueta <p>, una etiqueta <input> y una etiqueta <button>, cada uno con un id distinto, como se muestra a continuación:

```
<p id="texto">Este es un documento HTML con
JavaScript</p>
<input id="entrada" type="text" placeholder="Ingresa el
texto aquí">
<button id="btn">Clic Aquí</button>
```

Ejercicio guiado

Seleccionar elementos por id con jQuery

- **Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crea dos archivos, un index.html y un script.js.

- **Paso 2:** En el index.html, escribir la estructura básica de un documento HTML, agregar el CDN para tener acceso a jQuery, enlaza el archivo externo de JavaScript y finalmente agrega las etiquetas indicadas al inicio del ejercicio, de esta forma:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport"
content="width=device-width, initial-scale=1.0">
   <script
src="https://code.jquery.com/jquery-3.7.1.min.js"
></script>
   <title>Document</title>
</head>
<body>
   <p id="texto">Este es un documento HTML con
JavaScript</p>
   <input id="entrada" type="text"
placeholder="Ingresa el texto aquí">
   <button id="btn">Clic Aquí</button>
   <script src="script.js"></script>
</body>
</html>
```

Ejercicio guiado

Seleccionar elementos por id con jQuery

- **Paso 3**: En el archivo script.js, agregar el método llamado $(document).ready que permite ejecutar nuestro jQuery una vez que todos los elementos hayan sido cargados y así evitar posibles errores. Dentro del método, será donde agregaremos el código para seleccionar los elementos del DOM que poseen id:

```
$(document).ready(function() {
   let texto = $('#texto');
   let entrada = $('#entrada');
   let btn = $('#btn');

   console.log(texto);
   console.log(entrada);
   console.log(btn);
});
```

Ejercicio guiado

Seleccionar elementos por id con jQuery

- **Paso 4:** Al ejecutar el código anterior, el resultado en la consola sería:

```
Object { 0: p#texto, length: 1 }
Object { 0: input#entrada, length: 1 }
Object { 0: button#btn, length: 1 }
```

- En el resultado anterior, se puede observar como jQuery almacena un objeto por cada elemento que guardamos en las variables, cada objeto en su interior en la primera posición (0), agrega la etiqueta que contiene el id que almacenamos, mientras que en la última posición, agrega el largo que contendrá ese elemento.

Manipulando DOM con jQuery

Selector de Clase

- El selector de clase es otro de los métodos más utilizados, tanto en jQuery como en JavaScript puro.

- La forma correcta de escribir un selector de clase con jQuery es:

```
$(document).ready(function() {
   $('.some');
});
```

En donde .some es la clase de algún componente HTML.

Demostración - “Selectores de clase con jQuery”

{desafío}
latam_

Ejercicio guiado

Selectores de clase con jQuery

Seleccionar mediante los selectores de clases de jQuery los elementos que se encuentran en un archivo HTML, guardalos en una variable y muestra las variables mediante un console.log, siendo estos elementos una etiqueta <p>, una etiqueta <input> y una <button>, como se muestra a continuación:

```
<p class="texto">Este es un documento HTML con JavaScript</p>
<input class="texto" type="text" placeholder="Ingresa el
texto aquí">
<button class="texto">Clic Aquí</button>
```

Ejercicio guiado

Selectores de clase con jQuery

- **Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crea dos archivos, un index.html y un script.js.

- **Paso 2:** En el index.html, escribir la estructura básica de un documento HTML, agregar el CDN para tener acceso a jQuery, enlazar el archivo externo de JavaScript y finalmente agregar las etiquetas indicadas al inicio del ejercicio, como se muestra a continuación:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
   <script
src="https://code.jquery.com/jquery-3.7.1.m
in.js"></script>
   <title>Document</title>
</head>
<body>
   <p class="texto">Este es un documento
HTML con JavaScript</p>
   <input class="texto" type="text"
placeholder="Ingresa el texto aquí">
   <button class="texto">Clic Aquí</button>
   <script src="script.js"></script>
</body>
</html>
```

Ejercicio guiado

Selectores de clase con jQuery

- **Paso 3:** En el archivo script.js, agregar el método llamado $(document).ready que permite ejecutar nuestro jQuery una vez que todos los elementos hayan sido cargados y así evitar posibles errores.

```
$(document).ready(function() {
   let texto = $('.texto');

   console.log(texto);
});
```

```
Object { 0: p.texto, 1: input.texto, 2: button.texto, length: 3}
```

Manipulando DOM con jQuery

Selección de Elementos en general

- Es posible seleccionar elementos del DOM a través de características como el ID y la clase, pero en códigos más complejos tendremos que combinar selectores para obtener los resultados esperados.

- Si se quisiera seleccionar un div con clase some del siguiente HTML:

```
<div class="another"></div>
<div class="some"></div>
<div></div>
```

- Para acceder al elemento con clase some podríamos hacerlo con el selector de css div.css.

```
$(document).ready(function() {
    $('div.some');
});
```

Demostración - “Accediendo a los elementos del DOM”

Ejercicio guiado

Accediendo a los elementos del DOM

Utilizando la selección de elementos generales de jQuery, se solicita identificar la primera etiqueta <a> sin agregar clases o id al elemento que contiene el HTML. Finalmente, para mostrar el resultado, se debe agregar un color de fondo al texto 'jQuery supports'. Utilizar el siguiente código:

```
<ul class="my-list">
   <li>
      <a href="http://jQuery.com">jQuery
supports</a>
      <ul>
         <li><a href="css1">CSS1</a></li>
         <li><a href="css2">CSS2</a></li>
         <li><a href="css3">CSS3</a></li>
         <li>Basic XPath</li>
      </ul>
   </li>
   <li>jQuery also supports
      <ul>
         <li>Custom selectors</li>
         <li>Form selectors</li>
      </ul>
   </li>
</ul>
```

Ejercicio guiado

Accediendo a los elementos del DOM

- **Paso 1:** Crear una carpeta en tu lugar de trabajo favorito y dentro de ella crea dos archivos, un index.html y un script.js.

- **Paso 2:** En el index.html, escribir la estructura básica de un documento HTML, agregar el CDN para tener acceso a jQuery, enlaza el archivo externo de JavaScript y finalmente agregar las etiquetas indicadas al inicio del ejercicio, como se muestra a continuación:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
   <title>Document</title>
</head>
<body>
   <ul class="my-list">
      <li>
         <a href="http://jQuery.com">jQuery supports</a>
         <ul>
            <li><a href="css1">CSS1</a></li>
            <li><a href="css2">CSS2</a></li>
            <li><a href="css3">CSS3</a></li>
            <li>Basic XPath</li>
         </ul>
      </li>
      <li>jQuery also supports
         <ul>
            <li>Custom selectors</li>
            <li>Form selectors</li>
         </ul>
      </li>
   </ul>
   <script src="script.js"></script>
</body>
</html>
```

Ejercicio guiado

Accediendo a los elementos del DOM

- **Paso 3:** En el archivo script.js, implementar los atributos y etiquetas que ya tiene nuestro HTML, ya que no podemos realizar modificación alguna
  - El primer elemento será la etiqueta **<ul>** que posee una clase denominada **“my-list”**. Siendo específicos en el sector indicando, que sería la etiqueta ul que contenga la clase **“my-list”**, quedando **“ul.my-list”.**
  - Luego, para indicar que seleccionaremos el primer elemento hijo de la etiqueta **<ul>**, utilizamos el selector especial de CSS “>”, este selector nos permite seleccionar sólo hijos directos de un elemento padre. En este caso “**ul.my-list > li**” seleccionará solo los dos li descendientes del padre ul con la clase my-list.
  - Luego, la siguiente parte del selector “**li > a**” nos seleccionará los elementos que sean hijos directos de li y sólo cumpliría esta condición el siguiente elemento: **jQuery supports:**

Ejercicio guiado

Accediendo a los elementos del DOM

```
$(document).ready(function() {
    $('ul.my-list > li > a');
    console.log($('ul.my-list > li > a'));
});
```

- **Paso 4:** Al ejecutar el código anterior, el resultado en la consola sería:

```
Object { 0: a, length: 1, prevObject: {…} }
```

Ejercicio guiado

Accediendo a los elementos del DOM

- **Paso 5**: Para darle un poco más de colorido a nuestro elemento seleccionado, trabajaremos con el método CSS de jQuery, al cual le indicamos la propiedad y valor de CSS que deseamos agregar como un elemento en línea al elemento con el atributo style de HTML.

```
$(document).ready(function() {
     $('ul.my-list > li > a').css('background','red');
});
```

¿Existe algún concepto que no hayas comprendido?

Volvamos a revisar los conceptos que más te hayan costado antes de seguir adelante.

Próxima sesión...

- *Desarrolla algoritmos transformando código de JavaScript a jQuery para utilizar correctamente la sintaxis de la librería.*
- *Crea un script que manipule elementos del DOM con selectores utilizando la librería jQuery para resolver un problema planteado.*
- *Crea un script que permita la selección y manipulación de elementos del DOM utilizando la librería jQuery para resolver un problema planteado.*

{desafío}
latam_

{desafío}
latam_

Academia de
talentos digitales