

Programación con JavaScript II

Sesión sincrónica 5

AJAX



Bienvenida y actividad de bienestar

Duración: 10 minutos.

Nombre de la práctica: Interacciones positivas.

Descripción de la práctica: Reflexionarás sobre las cualidades positivas que aprecias de las personas con las que interactúas diariamente.

Palabras clave: Relaciones positivas.

Instrucciones para el participante: Puedes obtener mayor gozo de los momentos que compartes con tus colegas si te tomas el tiempo para pensar en lo que valoras y aprecias de ellos. Diversas investigaciones muestran que enfocarse en lo positivo que sucede diariamente ayuda a incrementar nuestra felicidad y lo mismo aplica a todas nuestras relaciones cercanas.

El psicólogo John Gottman sugiere que, para tener relaciones felices con alguna persona, es necesario aspirar a tener cinco interacciones positivas por cada interacción negativa que se tenga con ella. Enfócate en tus compañeros y/o colegas y piensa en las siguientes preguntas. En cada caso, anota ejemplos específicos.

1. ¿Qué te atrajo de tus compañeros cuando se conocieron?
2. ¿Qué cosas han disfrutado al hacerlas juntos?
3. ¿Qué cosas realmente aprecias de ellos en este momento?
4. ¿Cuáles son sus fortalezas?

Ahora, cuando estés con tus compañeros, lo más importante es que te tomes el tiempo para darte cuenta y reconocer estas cualidades, sus fortalezas, las cosas que ellos hacen que realmente aprecies, así como los momentos agradables que han compartido.

Piensa en estas declaraciones:

- “Realmente me encanta cuando ellos...”
- “Son tan buenos para...”
- “Viéndolos hacer..., me recuerda ese fantástico día cuando nosotros...”

Aunque realizar dicho análisis con todas las personas que conoces resulta poco práctico, puedes usar los mismos principios para mejorar tus relaciones en general. Por ejemplo, antes de pasar tiempo con alguien, tómate un momento para pensar en aquellas cosas que te gustan, aprecias o admiras de esa persona o cómo te hacen sentir bien. Asimismo, después de pasar tiempo con esa persona, piensa en las cosas que apreciaste o lo que disfrutaste del tiempo que pasaron juntos.

Actividad guiada

Parte 1

Duración: 75 minutos.

En esta sesión pondrás en práctica estructuras AJAX que permitan generar respuestas inmediatas sin recarga de la página. Para lograr este objetivo, realizarás los siguientes ejercicios:

Considera que, para llevarla cabo, se requiere Visual Studio Code, y para trabajar con AJAX, es necesario tener acceso a un repositorio de un servidor web o bien usar la extensión LIVE SERVER, en donde alojarás los archivos para efectuar el ejercicio.

Ejercicio 1

Responde a las siguientes preguntas:

1. Describe una situación en la que consideres que implementar AJAX es una alternativa adecuada.
2. ¿Por qué elegirías una implementación AJAX con API Fetch en lugar de usar el objeto XMLHttpRequest (XHR)?
3. Explica por qué es importante usar `catch()` en una implementación API Fetch.
4. Analiza el siguiente código y completa las instrucciones necesarias para implementar un objeto XMLHttpRequest, considerando el método GET para sustituir las líneas html por el contenido del archivo "ajax_info.txt".

```
function CargaDocumento() {  
  var xhttp = new _____  
  xhttp.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200) {  
      document.getElementById("demo").innerHTML =  
        this.____;  
    }  
  };  
  xhttp.open("____", "____", true);  
  xhttp.send();  
}
```


Actividad guiada

Parte 1

5. Reescribe el código cambiando las estructuras necesarias para convertirlo en solicitudes API Fetch.

Ejercicio 2

1. Live Server es una extensión que permite simular la respuesta de un servidor web de manera local, de tal suerte, que podamos recargar páginas de manera estática o dinámica.
2. Descarga la extensión Live Server para Visual Studio Code desde la página oficial de Visual Studio <https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer> y sube a Visual Studio un archivo con el nombre de ActividadGuiadaTema4.html que contenga el siguiente código (GitHub, 2022):

Actividad guiada

Parte 1

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">
  <title>Uso de Fetch</title>

  <style>
    body{
      background-color: cornflowerblue;
    }
    table thead {
      background:#302b63;
      color:white;
    }
  </style>
</head>
<body>
  <div class="container mt-4 shadow-lg p-3 mb-5 bg-body rounded">
    <table class="table table-bordered table-striped">
      <thead>
        <tr>
          <th>Id</th>
          <th>Name</th>
          <th>Email</th>
        </tr>
```

Actividad guiada

Parte 1

```
</thead>
  <tbody id="data">
  </tbody>
</table>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
crossorigin="anonymous"></script>

<script>
let url = 'https://jsonplaceholder.typicode.com/users/';
fetch(url)
  .then( response => response.json() )
  .then( data => mostrarData(data) )
  .catch( error => console.log(error) )

const mostrarData = (data) => {
  console.log(data)
  let body = ""
  for (var i = 0; i < data.length; i++) {
    body+=`<tr><td>${data[i].id}</td><td>${data[i].name}</td><td></td></tr>`
  }
  document.getElementById('data').innerHTML = body
  //console.log(body)
}
</script>
</body>
</html>
```

Actividad guiada

Parte 1

- 6. Analiza el código, específicamente la instrucción `fetch()`, y describe los datos que contiene la estructura del archivo json.
- 7. Haz clic derecho sobre el código y usa la opción **Open with Live Server** (ALT + L, ALT+O).
- 8. Compara el resultado obtenido con la siguiente imagen, donde se puede observar que el código toma el archivo JSON del servidor `jsonplaceholder.typicode.com` y lo muestra en una tabla.

Id	Name	Email
1	Leanne Graham	
2	Ervin Howell	
3	Clementine Bauch	
4	Patricia Lebsack	
5	Chelsey Dietrich	
6	Mrs. Dennis Schulist	
7	Kurtis Weissnat	
8	Nicholas Runolfsdottir V	
9	Glenna Reichert	
10	Clementina DuBuque	

Ejercicio 3

- 1. Analiza el código usado en el ejercicio 2, después, encuentra el lugar correcto dónde se deba incluir la siguiente línea de código que agrega el correo electrónico a la tabla `${data[i].email}`
- 2. Ejecuta el código con Live Server para que se obtenga el mail de las personas de la lista del ejercicio anterior. Una vez añadido, el resultado quedará como se muestra a continuación:

Actividad guiada

Parte 1

Id	Name	Email
1	Leanne Graham	Sincere@april.biz
2	Ervin Howell	Shanna@melissa.tv
3	Clementine Bauch	Nathan@yesenia.net
4	Patricia Lebsack	Julianne.OConner@kory.org
5	Chelsey Dietrich	Lucio_Hettinger@annie.ca
6	Mrs. Dennis Schulist	Karley_Dach@jasper.info
7	Kurtis Weissnat	Telly.Hoeger@billy.biz
8	Nicholas Runolfsdottir V	Sherwood@rosamond.me
9	Glenna Reichert	Chaim_McDermott@dana.io
10	Clementina DuBuque	Rey.Padberg@karina.biz

3. Ahora incluye el resto de los datos disponibles en el archivo json. <https://jsonplaceholder.typicode.com/users>

Consideraciones:

Instructor: Es importante no asumir que el aprendiz conoce todos los temas necesarios para realizar la práctica, por lo que se recomienda detectar sus áreas de oportunidad y retroalimentarlas, para que, de ese modo, todos logren realizarla.

Aprendedor: Recuerda, antes de iniciar esta práctica, repasa los conceptos explicados en el tema. Es importante tener configurada la extensión Live Server y contar con acceso a internet.

Receso

Duración: 10 minutos.

Programación con JavaScript II

Actividad guiada

Parte 2

Duración: 75 minutos.

Ejercicio 4

Objetivo: Crear una página que, al seleccionar un estado de la República Mexicana, obtenga los códigos postales y las ciudades.

1. Desde la siguiente referencia, descarga los códigos postales del Servicio Postal Mexicano en formato XML:
https://www.correosdemexico.gob.mx/SSLServicios/ConsultaCP/CodigoPostal_Exportar.aspx

Puedes revisar la estructura del archivo XML bajándolo aquí.

2. Crea la siguiente interfaz haciendo uso de HTML, CSS, JavaScript.

Estado:

----- T o d o s -----

Municipio:

----- T o d o s -----

Códigos postales				
Código Postal	Municipio	Estado	Ciudad	Clave de Oficina

3. El primer combobox para el código postal, deberá contener los estados de la República Mexicana ordenados ascendentemente de forma alfabética. Puedes usar:
<select><option value...></option></select>
4. El combobox del municipio deberá contener solo los del Estado que previamente se haya seleccionado.
Al usar AJAX se deberán cargar los datos del Código Postal + Municipio + Estado + Ciudad + Clave de Oficina en una tabla. Recuerda que puedes utilizar las etiquetas <div></div> para reemplazar los datos tomados de la estructura XML. El evento del combo box del municipio será <onchange="myFunction()">
Considera que los datos deberán estar almacenados en el servidor Web.

Actividad guiada

Parte 2

5. Al seleccionar “TODOS” los municipios, se deberán mostrar la totalidad de los códigos postales.
6. Incluye la función de ordenar la tabla de datos al hacer clic sobre el título de cada una de las columnas.

Consideraciones:

Instructor:

En esta actividad es necesario guiar al participante en la adquisición de un web server que sirva de práctica. Se aplica el criterio de usar el objeto XMLHttpRequest (XHR) o bien API Fetch.

Dado que la cantidad de información es considerable, se puede tomar una muestra de los estados de la República Mexicana y sus municipios como ejemplo, y dejar al participante el reto de completar la aplicación.

Se deja a criterio del instructor si en lugar de usar combo boxes en cascada, solo es un `<input type="text" id="fname" onchange="myFunction()">` para que escriban un estado de la República.

Aprendedor:

Es importante que, a través del objeto XMLHttpRequest() y la de API FETCH, reconozcas la diferencia entre una implementación de peticiones y una API.

Repasa en el tema dichas diferencias, además de sus ventajas.

Cierre

Duración: 10 minutos.

Ahora has aprendido a usar dos métodos AJAX para actualizar las páginas HTML sin la necesidad de un proceso de recarga, y esto, da la sensación de tener un tiempo de respuesta muy corto. Tanto el uso del objeto XMLHttpRequest como la estructura API Fetch, son muy eficientes para lograr esta comunicación entre archivos o información de la base de datos de un servidor.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.