

TAREA 4: Cargar contenido externo - Cargar contenido Utilizando JSON

CARGAR CONTENIDO EXTERNO

Abra el archivo `/ejercicios/index.html` en el navegador. Realice el ejercicio utilizando el archivo `/ejercicios/js/load.js`. La tarea es cargar el contenido de un artículo de blog cuando el usuario haga click en el título del ítem.

1. Crear un elemento `div` después del título de cada título de artículo de blog y guardar una referencia hacia ellos en el elemento de título utilizando `$.fn.data`.
2. Vincular un evento click al título, el cual utilizará el método `$.fn.load` para cargar en cada `div` creado el contenido apropiado desde el archivo `/ejercicios/data/blog.html`. No olvide de deshabilitar el comportamiento predeterminado del evento click.

Notar que cada título de artículo de blog en `index.html` incluye un enlace hacia el artículo. Necesitará aprovechar el atributo `href` de cada enlace para obtener el contenido propio de `blog.html`. Una vez obtenida el valor del atributo, puede utilizar la siguiente forma para procesar la información y convertirla en un selector para utilizar en conjunto con `$.fn.load`:

- `var href = 'blog.html#post1';`
- `var tempArray = href.split('#');`
- `var id = '#' + tempArray[1];`

Recuerde utilizar `console.log` para asegurarse que está realizando lo correcto.

El código necesario para llevar esto a cabo es el siguiente:

```
$(document).ready(function() {  
    // Selecciona el elemento con el ID 'blog'  
    const $blog = $('#blog');  
  
    // Encuentra todos los elementos 'h3' dentro de #blog  
    $blog.find('h3').each(function() {  
        const $this = $(this);  
        // Crea un nuevo elemento 'div' y lo inserta después del 'h3'  
        const $target = $('<div/>').insertAfter($this);  
  
        // Almacena una referencia al div recién creado en los datos del 'h3' actual  
        $this.data('target', $target);  
    }).click(function(e) {  
        // Cuando se hace clic en un 'h3'  
        const $this = $(this);  
        const $a = $this.find('a'); // Encuentra el elemento 'a' dentro del 'h3'  
        const $target = $this.data('target'); // Obtiene el div almacenado en los datos del 'h3'  
        const href = $a.attr('href'); // Obtiene el atributo href del 'a'  
        const id = '#' + href.split('#')[1]; // Extrae el ID del enlace  
  
        console.log(id); // Imprime el ID en la consola  
        e.preventDefault(); // Previene el comportamiento predeterminado del enlace  
  
        // Carga el contenido del archivo 'blog.html' en el div objetivo usando el ID extraído  
        $target.load('data/blog.html' + id);  
    });  
});
```

De esta forma en cada uno de los enlaces del apartado blog:

Blog

- Blog post 1
- Blog post 2
- Blog post 3

Conseguimos el siguiente efecto al hacer click:

- **Blog post 1**

You got Post 1!

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

- **Blog post 2**
- **Blog post 3**

CARGAR CONTENIDO UTILIZANDO JSON

Abra el archivo `/ejercicios/index.html` en el navegador. Realice el ejercicio utilizando el archivo `/ejercicios/js/specials.js`. La tarea es mostrar los detalles del usuario para un día determinado cuando se selecciona desde la lista desplegable.

1. Añadir un elemento `div` después del formulario que se encuentra dentro del elemento `#specials`; allí será el lugar en donde se colocará la información a obtener.
2. Vincular el evento `change` en el elemento `select`; cuando se realiza un cambio en la selección, enviar una petición Ajax a `/ejercicios/data/specials.json`.
3. Cuando la petición devuelve una respuesta, utilizar el valor seleccionado en el `select` (ayuda: `$.fn.val`) para buscar la información correspondiente en la respuesta JSON.
4. Añadir algún HTML con la información obtenida en el `div` creado anteriormente.
5. Finalmente remover el botón `submit` del formulario.

Para completar este ejercicio es necesario el siguiente código:

```
$(document).ready(function() {  
    // Selecciona el elemento con el ID #specials y elimina el elemento <li> con la clase  
    // .buttons que se encuentra dentro.  
    const $specials = $('#specials');  
    $specials.find('li.buttons').remove();  
  
    // Variable para almacenar la respuesta JSON obtenida de la solicitud AJAX.  
    let cachedResponse = null;  
  
    // Crea un elemento <div> que servirá para mostrar los detalles del usuario.  
    const $details = $('<div/>').appendTo($specials);  
  
    // Función que maneja la respuesta JSON y muestra los detalles correspondientes al día seleccionado.  
    /* usages new */  
    const handleResponse = function (specials, val) {  
        // Obtiene los detalles del usuario basados en el día seleccionado.  
        const daySpecial = specials[val];  
  
        // Construye el contenido HTML para mostrar los detalles del usuario.  
        let html = '<h3>' + daySpecial.title + '</h3>';  
        html += '<p>' + daySpecial.text + '</p>';  
  
        // Coloca el contenido HTML en el elemento <div> creado anteriormente.  
        $details.html(html);  
  
        // Crea una imagen utilizando la URL proporcionada y la agrega al elemento <div>.  
        $('<img/>').attr('src', daySpecial.image).appendTo($details);  
  
        // Establece el color del texto en el elemento <div> utilizando la información proporcionada.  
        $details.css('color', daySpecial.color);  
    };  
});
```

```
// Selecciona el elemento <select> dentro de #specials y establece un evento de cambio.
const $select = $specials.find('select').change(function () {
    // Obtiene el valor seleccionado en el menú desplegable.
    const val = $select.val();

    // Si no se ha seleccionado ningún valor, vacía el elemento <div> que muestra los detalles.
    if (!val) {
        $details.empty();
        return;
    }

    // Verifica si ya se ha realizado una solicitud AJAX y almacenado la respuesta.
    if (cachedResponse) {
        // Si hay una respuesta almacenada, muestra los detalles correspondientes al día seleccionado.
        handleResponse(cachedResponse, val);
    } else {
        // Si no hay respuesta almacenada, realiza una solicitud AJAX para obtener el archivo JSON.
        $.ajax({
            type: 'get',
            dataType: 'json',
            url: 'data/specials.json',
            success: function (specials) {
                // Almacena la respuesta JSON para su uso posterior.
                cachedResponse = specials;
                // Muestra los detalles correspondientes al día seleccionado.
                handleResponse(specials, val);
            }
        });
    }
});
```

De esta forma en la sección specials:

Specials

Choose the day of the week to see the specials:

-

Si seleccionamos uno de los días obtendremos el siguiente resultado:

Specials

Choose the day of the week to see the specials:

-

Twofer Tuesday

Two vegetables for the price of one!.

Notar que cada vez que la selección cambia, se realiza una petición Ajax. ¿Cómo cambiaría el código para realizar solo una petición y guardar la información para aprovecharla cuando se vuelve a cambiar la opción seleccionada?

```
// Selecciona el elemento <select> dentro del elemento con ID #specials y establece un evento de cambio.
const $select = $specials.find('select').change(function () {
    // Obtiene el valor seleccionado en el menú desplegable.
    const val = $select.val();

    // Si no se ha seleccionado ningún valor, vacía el elemento <div> que muestra los detalles.
    if (!val) {
        $details.empty(); // Vacía el contenido del elemento <div> donde se muestran los detalles.
        return;
    }

    // Si hay un valor seleccionado, llama a la función handleResponse con la respuesta almacenada
    // y el valor seleccionado.
    handleResponse(cachedResponse, val);
});

// Realiza una solicitud Ajax para obtener el archivo JSON.
$.ajax({
    type: 'get', // Método HTTP GET
    dataType: 'json', // Tipo de datos esperados en la respuesta
    url: 'data/specials.json', // URL del archivo JSON
    success: function (specials) {
        cachedResponse = specials; // Almacena la respuesta JSON en la variable cachedResponse

        // Obtiene el valor seleccionado actualmente en el menú desplegable.
        const val = $select.val();

        // Verifica si hay un valor seleccionado y si se ha almacenado una respuesta JSON.
        if (val && cachedResponse) {
            handleResponse(val); // Llama a la función handleResponse con el valor seleccionado
        }
    }
});
```