

TAREA 2: Crear sugerencia - Añadir navegación por pestañas

CREAR UNA "SUGERENCIA" PARA UNA CAJA DE INGRESO DE TEXTO

Abra el archivo `/ejercicios/index.html` en el navegador. Realice el ejercicio utilizando el archivo `/ejercicios/js/inputHint.js` o trabaje directamente con Firebug. La tarea a realizar es utilizar el texto del elemento `label` y aplicar una "sugerencia" en la caja de ingreso de texto. Los pasos a seguir son los siguientes:

Constantes necesarias:

```
// Constante para almacenar el elemento input.input_text.  
const input = $('#input_text');  
  
// Constante para almacenar el elemento label.  
const label = $('#label');  
  
// Constante para almacenar el texto del elemento label.  
const textLabel = label.text();
```

1. Establecer el valor del elemento `input` igual al valor del elemento `label`;
2. Añadir la clase `"hint"` al elemento `input`;
3. Remover el elemento `label`;
4. Vincular un evento `focus` en el `input` para remover el texto de sugerencia y la clase `"hint"`;
5. Vincular un evento `blur` en el `input` para restaurar el texto de sugerencia y la clase `"hint"` en caso que no se haya ingresado algún texto.

```
Alberto Martínez Pérez *  
$(document).ready(function() {  
    // Establece el valor del elemento input igual al valor del elemento label.  
    input.val(textLabel);  
  
    // Añade la clase 'hint' al elemento input.  
    input.addClass('hint');  
  
    // Remover el elemento label.  
    label.remove();  
  
    // Vincular un evento focus.  
    input.focus(function() {  
        input.val('')  
        .removeClass('hint');  
    });  
  
    // Vincular un evento blur.  
    input.blur(function() {  
        if (input.val() === '') {  
            input.val(textLabel)  
            .addClass('hint');  
        }  
    });  
});
```

De esta forma cuando entramos ahora veremos lo siguiente en la sección del input de búsqueda:

Si hacemos focus en el input el texto desaparece:

Y vuelve a aparecer si nos salimos (blur) sin haber realizado cambios:

¿Qué otras consideraciones debe considerar si se desea aplicar esta funcionalidad a un sitio real?

Estamos creando un elemento `<label>` en el documento HTML que siempre vamos a destruir en la carga del documento, por lo que no tiene mucho sentido hacer esto. Sería mejor tener una variable con un texto por defecto el cual se muestre como contenido "placeholder" del input de búsqueda al inicio o cuando se hace blur y el input no tiene texto.

Además, sería importante tener la compatibilidad del navegador con el contenido jQuery.

AÑADIR UNA NAVEGACIÓN POR PESTAÑAS

Abra el archivo `/ejercicios/index.html` en el navegador. Realice el ejercicio utilizando el archivo `/ejercicios/js/tabs.js` o trabaje directamente con Firebug. La tarea a realizar es crear una navegación por pestañas para los dos elementos `div.module`.

Los pasos a seguir son los siguientes:

1. Ocultar todos los elementos `div.module`;
2. Crear una lista desordenada antes del primer `div.module` para utilizar como pestañas;
3. Interactuar con cada `div` utilizando `$.fn.each`. Por cada uno, utilizar el texto del elemento `h2` como el texto para el ítem de la lista desordenada;
4. Vincular un evento click a cada ítem de la lista de forma que:
 - Muestre el `div` correspondiente y oculte el otro;
 - Añada la clase `"current"` al ítem seleccionado;
 - Remueva la clase `"current"` del otro ítem de la lista;
5. Finalmente, mostrar la primera pestaña.

```
$(document).ready(function() {
    // Constantes que almacenan los elementos div con clase module y la lista desordenada.
    const modulos = $('div.module');
    const listado = $('<ul class="tabs"></ul>').insertBefore(modulos.first());

    // Ocultar todos los elementos div.module excepto el primero.
    modulos.not(':first').hide();

    // Iterar sobre cada div.module y crear la lista de pestañas.
    modulos.each(function(index) {
        const titulo = $(this).find('h2').text();
        $('<li><a href="#tab' + index + '"> + titulo + '</a></li>').appendTo(listado);
    });

    // Mostrar la primera pestaña.
    listado.find('li:first').addClass('current');

    // Manejar el cambio de pestañas al hacer clic.
    listado.on('click', 'li', function(e) {
        e.preventDefault();
        const index = $(this).index();

        modulos.hide().eq(index).show(); // Oculta todos los div.module y muestra el correspondiente al índice.
        $(this).addClass('current').siblings().removeClass('current'); // Gestiona las clases "current".
    });
});
```

Si accedemos a la web veremos lo siguiente:

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

Si hacemos clic en Specials, aparecerá el contenido correspondiente:

- [Blog](#)
- [Specials](#)

Specials

Choose the day of the week to see the specials:

-
-