

Proyecto BID-MINEC

UR PROYECTO UPSKILLING & RESKILLING

Orientado a la reconversión y mejora de
habilidades digitales y tecnológicas



FrontEnd básico-intermedio con JavaScript

Framework's web



JavaScript

Facilitador: Iván Alvarado

Contenido

1. ~~JSON~~
2. ~~LocalStorage~~
3. ~~Peticiones~~
4. JQuery
5. JQueryUI

4. JQuery

Que es?

- jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

Como usarlo?

- Hay dos versiones de jQuery disponibles para descargar:
 - Versión de producción: esto es para su sitio web en vivo porque ha sido minimizado y comprimido
 - Versión de desarrollo: esto es para pruebas y desarrollo (código sin comprimir y legible)

4. JQuery

CDN de jQuery

- Si no desea descargar y alojar jQuery usted mismo, puede incluirlo desde una CDN (red de entrega de contenido). Google es un ejemplo de alguien que hospeda jQuery:

```
<head>  
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">  
  </script>  
</head>
```

4. JQuery

Sintaxis

- La sintaxis de jQuery está hecha a medida para seleccionar elementos HTML y realizar alguna acción en los elementos.

La sintaxis básica es: \$(selector). acción ()

- Un signo \$ para definir/acceder a jQuery
- Un (selector) para "consultar (o encontrar)" elementos HTML
- Una acción jQuery () que se realizará en los elementos

Ejemplos:

- \$(this).hide()- oculta el elemento actual.
- \$("p").hide()- oculta todos los elementos <p>.
- \$(".test").hide()- oculta todos los elementos con class="test".
- \$("#test").hide()- oculta el elemento con id="test".

4. JQuery

Evento de document

- La mayoría de los métodos de jQuery en los ejemplos están dentro de un evento del objeto document llamado ready

```
$(document).ready(function(){  
  
    // Métodos jQuery... ---->  
  
});
```

```
$(function){  
  
    // Metodos jQuery  
  
});
```

- Esto es para evitar que se ejecute cualquier código jQuery antes de que el documento termine de cargarse (esté listo).
- Es una buena práctica esperar a que el documento esté completamente cargado y listo antes de trabajar con él. Esto también le permite tener su código JavaScript antes del cuerpo de su documento, en la sección de encabezado.
- Estos son algunos ejemplos de acciones que pueden fallar si los métodos se ejecutan antes de que el documento esté completamente cargado:
 - Intentando ocultar un elemento que aún no se ha creado
 - Intentando obtener el tamaño de una imagen que aún no está cargada

4. JQuery

- Los selectores, al menos los más básicos, son parecidos, o iguales, a los que se utilizan en CSS para seleccionar los elementos a los que se desean aplicar ciertos estilos.
- Se entiende que todas las personas que intenten profundizar en el framework jQuery deben haber conocido CSS anteriormente, no habrá ningún problema con ellos.

Selector de etiquetas:

- Simplemente indicamos la etiqueta a la que deseamos referirnos, es decir, la etiqueta que queremos seleccionar. Obtendremos con él todas las etiquetas de la página indicada en el selector.

`$("h1")` //selecciona todos los encabezados de nivel 1

Selector por identificador:

- Sirven para seleccionar los elementos que tengan un identificador dado, que se asigna a las etiquetas a través del atributo id del HTML. Para utilizar este selector se indica primero el carácter "#" y luego el identificador de cuyo elemento se desee seleccionar.

`$("#idelemento")` //selecciona una etiqueta que tiene el atributo id="idelemento"

4. JQuery

Selector por clase:

- Podemos indicar el nombre de una clase (class de CSS) y seleccionar todos los elementos a los que se ha aplicado esta clase. Para ello, como en CSS, comenzamos colocando el carácter "." y luego el nombre de la clase que deseamos seleccionar.

`$(".miclase")` //selecciona todos los elementos que tienen el atributo `class="miclase"`

- **Selector por varias clases:**

Si lo deseamos, podemos indicar varias clases CSS, para obtener todos los elementos que tienen esas clases aplicadas: todas al mismo tiempo. Esto se consigue comenzando por un ".", igual que los selectores de clases, y luego otro "." para separar las distintas clases que queremos utilizar en el selector.

`$(".clase1.clase2")` //selecciona los elementos que tienen `class="clase1 clase2"`

Selector asterisco "*":

- Nos sirve para seleccionar todos los elementos de la página.

`$("*")` //selecciona todos los elementos que tiene la página y todas las listas UL

4. JQuery

Concatenar varios selectores distintos:

- Por último, podemos utilizar varios selectores, para obtener todas las etiquetas que cumplen uno de ellos. No hace falta que cumplan todos los selectores a la vez, sino con que uno de ellos concuerde es suficiente. Para ello colocamos todos los selectores que deseamos, separados por una coma ",".

`$("div,p")` //selecciona todos los elementos división y párrafo

`$(".clase1,.clase2")` //selecciona los elementos que tienen la clase "clase1" o "clase2"

`$("#miid,.miclase,ul)` //selecciona el elemento con id="miid", los elementos con class="miclase" y todas las listas UL

4. JQuery

- Lista de todos los eventos que se pueden usar en JQuery, para ejecutar métodos y funciones. Los eventos son los que desencadenan y disparan las acciones necesarias para ejecutar cualquier función.

| Evento | Como funciona |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| blur() | Conecta / desencadena el evento de desenfoque |
| change() | Conecta / activa el evento de cambio |
| click() | Conecta / activa el evento de clic |
| dblclick() | Conecta / activa el evento de doble clic |
| event.currentTarget | El elemento DOM actual dentro de la fase de burbujeo del evento |
| event.data | Contiene los datos opcionales transferidos a un método de evento cuando el controlador de ejecución actual esta enlazado |
| event.delegateTarget | Devuelve el elemento donde se ha conectado el controlador de eventos jquery actualmente llamado |
| event.isDefaultPrevented() | Devuelve si event.preventDefault () fue llamado para el objeto de evento |
| event.isImmediatePropagationStopped() | Devuelve si event.stopImmediatePropagation () se llama para el objeto de evento |
| event.isPropagationStopped() | Devuelve si event.stopPropagation () fue llamado para el objeto de evento |
| event.namespace | Devuelve el espacio de nombres especificado cuando se desencadena el evento |

4. JQuery

| | |
|----------------------------------|----------------------------------------------------------------------------------------------------------------|
| event.pageX | Devuelve la posición del ratón en relación con el borde izquierdo del documento |
| event.pageY | Devuelve la posición del mouse en relación con el borde superior del documento |
| event.preventDefault() | Evita la acción predeterminada del evento |
| event.relatedTarget | Devuelve el elemento entrado o salido en el movimiento del ratón. |
| event.result | Contiene el ultimo / anterior valor devuelto por un manejador de eventos disparado por el evento especificado |
| event.stopImmediatePropagation() | Evita que se llamen a otros controladores de eventos |
| event.stopPropagation() | Evita que el evento burbujee el árbol DOM, evitando que los manejadores de origen sean notificados del evento. |
| event.target | Devuelve el elemento DOM que ha desencadenado el evento. |
| event.timeStamp | Devuelve el número de milisegundos desde el 1 de enero de 1970, cuando se desencadena el evento |
| event.type | Devuelve que tipo de evento se ha activado |
| event.which | Devuelve la tecla del teclado o el botón del ratón para el evento |
| focus() | Conecta / activa el evento de enfoque |
| focusin() | Conecta un controlador de eventos al evento focus |

4. JQuery

| | |
|--------------|--------------------------------------------------------|
| ocusout() | Conecta un controlador de eventos al evento de enfoque |
| hover() | Conecta dos manejadores de eventos al evento hover |
| keydown() | Conecta / activa el evento keydown |
| keypress() | Conecta / activa el evento de pulsaciones de teclas |
| keyup() | Conecta / activa el evento keyup |
| mousedown() | Conecta / activa el evento mousedown |
| mouseenter() | Conecta / activa el evento mouseenter |
| mouseleave() | Adjunta / activa el evento mouseleave |
| mousemove() | Adjunta / activa el evento mousemove |
| mouseout() | Conecta / activa el evento mouseout |
| mouseover() | Conecta / activa el evento mouseover |

4. JQuery

| | |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| mouseup() | Adjunta / desencadena el evento mouseup |
| off() | Elimina los manejadores de eventos conectados con el método on () |
| on() | Agrega los controladores de eventos a elementos |
| one() | Agrega uno o más controladores de eventos a elementos seleccionados. Este controlador se lo se puede activar una vez por elemento |
| \$.proxy() | Toma una función existente y devuelve una nueva con un contexto particular |
| ready() | Especifica una función a ejecutar cuando el DOM está completamente cargado |
| resize() | Adjunta / activa el evento de cambio de tamaño |
| scroll() | Conecta / activa el evento de desplazamiento |
| select() | Conecta / activa el evento seleccionado |
| submit() | Adjunta / activa el evento de envío |
| trigger() | Activa todos los eventos vinculados a los elementos seleccionados |
| triggerHandler() | Dispara todas las funciones enlazadas a un evento especificado para los elementos seleccionados |

4. JQuery

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Document</title>
5     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
6     </script>
7     <script>
8       $(document).ready(function(){
9         $("p").click(function(){
10           alert("Hizo clic");
11         });
12       });
13     </script>
14   </head>
15   <body>
16     <p>hacer clic</p>
17   </body>
18 </html>
```


4. JQuery

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Document</title>
5     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
6     </script>
7     <script>
8       $(document).ready(function(){
9         $("#p1").on({
10           mouseenter: function(){
11             $(this).css("background-color", "lightgray");
12           },
13           mouseleave: function(){
14             $(this).css("background-color", "lightblue");
15           },
16           click : function(){
17             $(this).css("background-color", "yellow");
18           }
19         });
20       });
21     </script>
22   </head>
23   <body>
24     <p id="p1">hacer click</p>
25   </body>
26 </html>
```

4. JQuery

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
5   </script>
6   <script>
7     $(function(){
8       $("input").focus(function(){
9         $(this).css("background-color", "yellow");
10      });
11      $("input").blur(function(){
12        $(this).css("background-color", "green");
13      });
14    });
15  </script>
16 </head>
17 <body>
18   Nombre: <input type="text" name="fullname"><br>
19   Correo: <input type="text" name="email">
20 </body>
21 </html>
```

4. JQuery

Métodos y funciones para efectos en JQuery

- En la siguiente tabla se listan todos los métodos de JQuery, disponibles para crear efectos y animaciones.

| Método | Función que realiza |
|--------------|--------------------------------------------------------------------------------------------|
| animate() | Ejecuta una animación personalizada en los elementos seleccionados |
| clearQueue() | Elimina todas las funciones restantes en cola de los elementos seleccionados |
| delay() | Establece un retardo para todas las funciones en cola en los elementos seleccionados |
| dequeue() | Elimina la siguiente función de la cola y luego ejecuta la función |
| fadeIn() | Se desvanece en los elementos seleccionados |
| fadeOut() | Desvanece los elementos seleccionados |
| fadeTo() | Desvanece los elementos seleccionados a una opacidad dada |
| fadeToggle() | Alterna entre los métodos fadeIn () y fadeOut () |
| finish() | Detiene, elimina y completa todas las animaciones en cola para los elementos seleccionados |

4. JQuery

Métodos y funciones para efectos en JQuery

| | |
|---------------|--------------------------------------------------------------------|
| hide() | Ocultar los elementos seleccionados |
| queue() | Muestra las funciones en cola en los elementos seleccionados |
| show() | Muestra los elementos seleccionados |
| slideDown() | Desliza (muestra) los elementos seleccionados |
| slideToggle() | Alterna entre los métodos slideUp () y slideDown () |
| slideUp() | Desliza (oculta) los elementos seleccionados |
| stop() | Detiene la animación en ejecución para los elementos seleccionados |
| toggle() | Alterna entre los métodos hide () y show () |

4. JQuery

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Document</title>
5     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
6     <script>
7       $(document).ready(function(){
8         $("#hide").click(function(){
9           $("table").hide(1000);
10          //$("#table").toggle();
11        });
12        $("#show").click(function(){
13          $("table").show();
14        });
15      });
16    </script>
17  </head>
18  <body>
19    <button id="hide">Ocultar</button>
20    <button id="show">Mostrar</button>
21    <table>
22      <tr><td>columna1</td><td>columna2</td><td>columna3</td></tr>
23      <tr><td>columna1</td><td>columna2</td><td>columna3</td></tr>
24    </table>
25  </body>
26 </html>
```

4. JQuery.

```
<script>
$(document).ready(function(){
    $("#hide").click(function(){
        $("table").fadeOut();
        //$("#table").toggle();
    });
    $("#show").click(function(){
        $("table").fadeIn();
    });
});
</script>
```

```
<script>
$(document).ready(function(){
    $("#hide").click(function(){
        $("table").slideUp();
    });
    $("#show").click(function(){
        $("table").slideDown();
    });
});
</script>
```

4. JQuery

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Document</title>
5     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
6     </script>
7     <style>
8     </style>
9     <script>
10      $(document).ready(function(){
11        $("#btnAnimacion").click(function(){
12          $("#div").animate({
13            opacity: '0.5',
14            width: '300px',
15            height : '300px'
16          }).css('background-color', "blue");
17        });
18      });
19    </script>
20  </head>
21  <body>
22    <button id="btnAnimacion">Aplicar animacion</button>
23    <div>
24      <p>Texto</p>
25    </div>
26  </body>
27 </html>
```


4. JQuery

Métodos de JQuery para AJAX

- JQuery posee varios métodos para trabajar con Ajax. Sin embargo, todos están basados en el método \$.ajax, por lo tanto, su comprensión es obligatoria. A continuación se abarcará dicho método.
- Generalmente, es preferible utilizar el método \$.ajax en lugar de los otros, ya que ofrece más características y su configuración es muy comprensible.

\$.ajax

- El método \$.ajax es configurado a través de un objeto, el cual contiene todas las instrucciones que necesita JQuery para completar la petición. Dicho método es particularmente útil debido a que ofrece la posibilidad de especificar acciones en caso que la petición haya fallado o no.
- Además, al estar configurado a través de un objeto, es posible definir sus propiedades de forma separada, haciendo que sea más fácil la reutilización del código. Puede visitar api.jquery.com/jquery.ajax para consultar la documentación sobre las opciones disponibles en el método.

4. JQuery

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>Document</title>
5      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
6      </script>
7      <script>
8        $.ajax({
9          // la URL para la petición
10         url : 'https://mdn.github.io/learning-area/javascript/oajs/json/superheroes.json',
11         // la información a enviar (también es posible utilizar una cadena de datos)
12         data : { id : 123 },
13         // especifica si será una petición POST o GET
14         type : 'GET',
15         // el tipo de información que se espera de respuesta
16         dataType : 'json',
17         // código a ejecutar si la petición es satisfactoria;
18         // la respuesta es pasada como argumento a la función
19         success : function(json) {
20           console.log(json);
21           $('<h1/>').text(json.squadName).appendTo('body');
22           $('<div class="content"/>')
23             .html(json.secretBase).appendTo('body');
24         },
25         // código a ejecutar si la petición falla, son pasados como argumentos a la función
26         // el objeto de la petición en crudo y código de estatus de la petición
27         error : function(xhr, status) {
28           alert('Disculpe, existió un problema');
29         },
30         // código a ejecutar sin importar si la petición falló o no
31         complete : function(xhr, status) {
32           alert('Petición realizada');
33         }
34       });
35     </script>
36   </head>
37   <body>
38     <div>
39       <h1>Texto</h1>
40     </div>
41   </body>
42 </html>
```

4. JQuery

Opciones del método \$.ajax

- El método \$.ajax posee muchas opciones de configuración, y es justamente esta característica la que hace que sea un método muy útil.
- Para una lista completa de las opciones disponibles, puede consultar api.jquery.com/jQuery.ajax; a continuación se muestran las más comunes:
- **async** Establece si la petición será asíncrona o no. De forma predeterminada el valor es true. Debe tener en cuenta que si la opción se establece en false, la petición bloqueará la ejecución de otros códigos hasta que dicha petición haya finalizado.
- **cache** Establece si la petición será guardada en la cache del navegador. De forma predeterminada es true para todos los dataType excepto para script y jsonp. Cuando posee el valor false, se agrega una cadena de caracteres anti-cache al final de la URL de la petición.
- **complete** Establece una función de devolución de llamada que se ejecuta cuando la petición esta completa, aunque haya fallado o no. La función recibe como argumentos el objeto de la petición en crudo y el código de estatus de la misma petición.
- **context** Establece el alcance en que la/las funciones de devolución de llamada se ejecutaran (por ejemplo, define el significado de this dentro de las funciones). De manera predeterminada this hace referencia al objeto originalmente pasado al método \$.ajax.
- **data** Establece la información que se enviará al servidor. Esta puede ser tanto un objeto como una cadena de datos (por ejemplo foo=bar&baz=bim)

4. JQuery

Opciones del método \$.ajax

- **dataType** Establece el tipo de información que se espera recibir como respuesta del servidor. Si no se especifica ningún valor, de forma predeterminada, jQuery revisa el tipo de MIME que posee la respuesta.
- **error** Establece una función de devolución de llamada a ejecutar si resulta algún error en la petición. Dicha función recibe como argumentos el objeto de la petición en crudo y el código de estatus de la misma petición.
- **jsonp** Establece el nombre de la función de devolución de llamada a enviar cuando se realiza una petición JSONP. De forma predeterminada el nombre es callback
- **success** Establece una función a ejecutar si la petición a sido satisfactoria. Dicha función recibe como argumentos la información de la petición (convertida a objeto JavaScript en el caso que dataType sea JSON), el estatus de la misma y el objeto de la petición en crudo.

4. JQuery

Opciones del método \$.ajax

- **timeout** Establece un tiempo en milisegundos para considerar a una petición como fallada.
- **traditional** Si su valor es true, se utiliza el estilo de serialización de datos utilizado antes de jQuery 1.4. Para más detalles puede visitar api.jquery.com/jquery.param.
- **type** De forma predeterminada su valor es GET. Otros tipos de peticiones también pueden ser utilizadas (como PUT y DELETE), sin embargo pueden no estar soportados por todos los navegadores.
- **url** Establece la URL en donde se realiza la petición. La opción url es obligatoria para el método \$.ajax;

4. JQuery

Métodos convenientes

- En caso que no quiera utilizar el método \$.ajax, y no necesite los controladores de errores, existen otros métodos más convenientes para realizar peticiones Ajax (aunque, como se indicó antes, estos están basados el método \$.ajax con valores pre-establecidos de configuración).

Los métodos que provee la biblioteca son:

- \$.get Realiza una petición GET a una URL provista.
- \$.post Realiza una petición POST a una URL provista.
- \$.getScript Añade un script a la página.
- \$.getJSON Realiza una petición GET a una URL provista y espera que un dato JSON sea devuelto.

4. JQuery

Métodos convenientes

Los métodos deben tener los siguientes argumentos, en orden:

- url La URL en donde se realizará la petición. Su valor es obligatorio.
- data La información que se enviará al servidor. Su valor es opcional y puede ser tanto un objeto como una cadena de datos (como foo=bar&baz=bim).Nota: esta opción no es valida para el método \$.getScript.
- success callback Una función opcional que se ejecuta en caso que petición haya sido satisfactoria. Dicha función recibe como argumentos la información de la petición y el objeto en bruto de dicha petición.
- data type El tipo de dato que se espera recibir desde el servidor. Su valor es opcional. Nota: esta opción es solo aplicable para métodos en que no está especificado el tipo de dato en el nombre del mismo método.

4. JQuery

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Document</title>
5     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
6     <script>
7       // obtiene texto plano o html
8       $.get('https://mdn.github.io/learning-area/javascript/ojs/json/superheroes.json', { userId : 1111 }, function(resp) {
9         console.log(resp);
10      });
11      // obtiene información en formato JSON desde el servidor
12      $.getJSON('https://mdn.github.io/learning-area/javascript/ojs/json/superheroes.json', function(resp) {
13        $.each(resp, function(k, v) {
14          //console.log(k + ' : ' + v);
15        });
16      });
17    </script>
18  </head>
19  <body>
20    <div>
21      <h1>Texto</h1>
22    </div>
23  </body>
24 </html>
```

4. JQuery

Métodos de JQuery para AJAX

- Lista de todos los métodos y funciones para AJAX. AJAX es el arte de intercambiar datos con el servidor y actualizar el contenido de cualquier página, sin tener que recargarla.

| Método | Función que realiza |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\$.ajax()</code> | Realiza una solicitud asíncrona AJAX |
| <code>\$.ajaxPrefilter()</code> | Maneje las opciones personalizadas de Ajax o modifique las opciones existentes antes de enviar cada solicitud y antes de que sean procesadas por <code>\$.ajax()</code> |
| <code>\$.ajaxSetup()</code> | Establece los valores predeterminados para futuras solicitudes AJAX |
| <code>\$.ajaxTransport()</code> | Crea un objeto que maneja la transmisión real de datos Ajax |
| <code>\$.get()</code> | Carga datos desde un servidor utilizando una solicitud AJAX HTTP GET |
| <code>\$.getJSON()</code> | Carga los datos codificados en JSON de un servidor que utiliza una solicitud HTTP GET |
| <code>\$.getScript()</code> | Carga (y ejecuta) un JavaScript desde un servidor utilizando una solicitud AJAX HTTP GET |
| <code>\$.param()</code> | Crea una representación serializada de una matriz o un objeto (se puede utilizar como cadena de consulta URL para solicitudes AJAX) |
| <code>\$.post()</code> | Carga datos desde un servidor mediante una solicitud AJAX HTTP POST |

4. JQuery

Métodos de JQuery para AJAX

| | |
|------------------|---------------------------------------------------------------------------------------------|
| ajaxComplete() | Especifica una función que se ejecutará cuando se complete la solicitud AJAX |
| ajaxError() | Especifica una función que se ejecutará cuando la solicitud AJAX finaliza con un error |
| ajaxSend() | Especifica una función que se ejecutará antes de enviar la solicitud AJAX |
| ajaxStart() | Especifica una función que se ejecutará cuando comience la primera solicitud AJAX |
| ajaxStop() | Especifica una función que se ejecutará cuando todas las solicitudes AJAX hayan finalizado |
| ajaxSuccess() | Especifica una función que se ejecutará cuando se complete una solicitud AJAX correctamente |
| load() | Carga los datos de un servidor y coloca los datos devueltos en el elemento seleccionado |
| serialize() | Codifica un conjunto de elementos de formulario como una cadena para su envío |
| serializeArray() | Codifica un conjunto de elementos de formulario como una matriz de nombres y valores |