Coding Bootcamp

Sprint 1

Temario

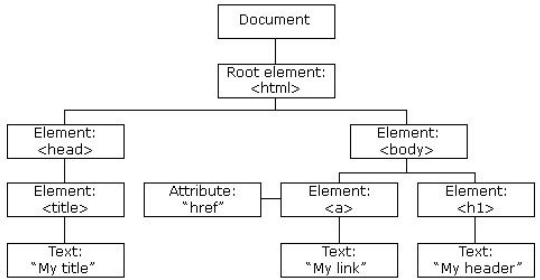
Temario

- DOM.
- jQuery.
- Eventos.

DOM

¿Qué es el DOM?

- DOM = Document Object Model.
  Documentación: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction
- Cuando se carga una página, el browser crea un DOM de la página a partir del HTML. Es una representación del documento que tiene una estructura jerárquica con forma de árbol.
- DOM ≠ HTML.
- DOM ≈ El código que se ve en Dev Tools.
- DOM es una API.
- JavaScript permite modificar el DOM (pero no modificar el HTML).



Objeto `window`

Escribir en la consola:

```
window;
```

El *objeto* `window` representa la ventana abierta del navegador. También se le llama BOM (Browser Object Model) y permite que JavaScript interactúe con toda la ventana, no sólo con el documento HTML.

Esto permite hacer consultas, como por ejemplo, obtener el ancho de la ventana (incluyendo la *scrollbar*) o información de la ubicación (URL):

```
window.innerWidth;
window.location;
```

Objeto `document`

Escribir en la consola:

```
document;
```

El *objeto* `document` es una referencia al documento HTML dentro de la ventana.

Es el objeto con el que interactuaremos más frecuentemente.

Es quien nos permite acceder y modificar el DOM.

Nota: También se puede acceder a `document` de esta forma:

```
window.document;
```

Manipular el DOM con JS (1/3)

El DOM cuenta con un montón de funciones (métodos) que permiten manipularlo.

Por ejemplo, la función `querySelector` permite seleccionar un elemento de la página web.

```
var titulo = document.querySelector("h1");
```

Luego es posible manipular dicho elemento.
Por ejemplo, es posible cambiarle el texto:

```
titulo.textContent = "Cursos de Programación";
```

Esto es manipulación de DOM usando JavaScript “puro”, sin uso de ninguna librería ni framework. Ej: jQuery.

Manipular el DOM con JS (2/3)

El código de la diapositiva anterior, también se podría haber escrito en una sola línea de código. 👉 ¡Ingresar a https://ha.edu.uy y probarlo!

```
document.querySelector("h1").textContent = "Cursos de Programación";
```

Probar también:

```
document.querySelector("p").style.color = "blue";
document.querySelector("p").style.fontSize = "4rem";
document.querySelector(".row").style.border = "10px solid red";
```

Manipular el DOM con JS (3/3)

¿Qué se puede hacer con JavaScript y el DOM?

- Modificar todos los elementos HTML en la página.
- Modificar todos los atributos HTML en la página.
- Modificar todos los estilos CSS en la página.
- Remover elementos HTML y atributos.
- Agregar nuevos elementos HTML y atributos.
- Reaccionar a eventos que suceden en la página.

jQuery

jQuery

- Es una *librería* de JavaScript que se creó en 2006.
  Una librería es una colección de métodos reutilizables (código pre-escrito por alguien más) que se puede reutilizar para acelerar nuestro trabajo.

- Con jQuery no se puede hacer más de lo que permite hacer JS, pero se obtienen funcionalidades extra que simplifican mucho el trabajo. Es decir, se pueden hacer un montón de cosas sin tener que escribir un montón de código.

- ¡Es *open source*! (con una gran comunidad de desarrolladores).

jQuery: la librería JS más popular del mundo

- La versión 3.4 tiene ≈ 10.600 líneas de código (≈90 kB).
- ¿Qué permite hacer?
  - Manipular datos.
  - Manipular el **DOM**.
  - Manejar eventos.
  - Realizar llamadas AJAX.
  - Realizar animaciones.
- Documentación: http://api.jquery.com/
- Incluso Bootstrap usa jQuery.
  Aunque esto cambiará en la versión 5 de Bootstrap.

¿Por qué usar jQuery?

Sin jQuery, es decir, con JavaScript “puro”:

```
document.querySelector("img#profile").style.display = "none";
```

Con jQuery:

```
$("img#profile").hide();
```

Notar que `hide` es una función definida por jQuery.

…¿y si quisiésemos ocultar todas las imágenes?

Sin jQuery, es decir, con JavaScript “puro”:

```
var images = document.querySelectorAll("img");
for (var i = 0; i < images.length; i++) {
   images[i].style.display = "none";
}
```

Con jQuery:

```
$("img").hide();
```

Notar que `hide` es una función definida por jQuery.

Otro ejemplo de porqué usar jQuery

Sin jQuery (JavaScript “puro”):

```
var box = document.createElement("div");

box.textContent = "Hola Mundo!";

box.className = "special";

box.style.cssFloat = "right";

box.style.backgroundColor = "red";

box.style.color = "blue";

document.querySelector("body").appendChild(box);
```

Con jQuery:

```
var box = $("<div></div>");

box.text("Hola Mundo!");

box.addClass("special");

box.css("float", "right");

box.css("background-color", "red");

box.css("color", "blue");

$("body").append(box);
```

👉 Acá la diferencia no es tanto la cantidad de código, sino la claridad. jQuery resulta más intuitivo.

Funcionamiento básico de jQuery (1/2)

```
$("p").addClass("importante");
```

1. Seleccionar uno o varios elementos.

2. Usar una función (método) de jQuery para manipular el o los elementos seleccionados.
En este ejemplo, se usó el método `addClass`.

En este caso se le asigna la clase "importante" a todos los elementos `p` de la página.

Funcionamiento básico de jQuery (2/2)

```
$(selector).unaAccion();
```

`$`	Es el nombre de una variable que apunta al objeto jQuery, que contiene toda la funcionalidad de la librería. Probar escribir `$` en la consola para ver si jQuery está instalado. También se puede escribir `jQuery` en lugar de `$`.
`(selector)`	Ingresar el selector entre comillas (simples o dobles) para seleccionar un elemento en el DOM. El resultado retornado es una “colección jQuery”.
`unaAccion()`	Alguna (función) método jQuery que realiza alguna acción.

jQuery – Instalación

Método 1: Descargar jQuery de https://code.jquery.com/jquery/ y copiarlo en nuestro proyecto.

```
<body>
    ...
    <script src="js/jquery.min.js"></script>
</body>
```

Método 2: Linkear a archivo en servidor remoto (CDN).

```
<body>
    ...
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</body>
```

¿Cómo se seleccionan estos elementos?

```
<p>
    Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
</p>

<img src="logo.png">

<p class="intro">Lorem ipsum dolor sit amet</p>

<div id="content">
    <p class="intro">Lorem ipsum dolor sit amet</p>
</div>
```

Probar las siguientes funcionalidades:

Entrar a https://ha.edu.uy y probar las siguientes sentencias en consola:

```
$("h1").fadeOut(5000);

$("h1").fadeIn();

$("p").slideUp();

$("p").slideDown();

$("#intro").slideToggle();

$("#sub-intro").fadeToggle();
```

Otros ejemplos de uso de jQuery

```
var h = $("#hack-academy"); // Selecciona el elemento con id="hack-academy".

h.addClass("importante"); // Se le agrega la clase "importante" al elemento.

h.removeClass("importante"); // Se le quita la clase "importante" al elemento.

h.css("color", "blue"); // Se colorea de azul el texto del elemento.

h.text("Algún texto..."); // Setea el texto del elemento.

h.hide(); // Se oculta el elemento.

h.show(); // Se muestra el elemento.

h.slideToggle();

h.fadeToggle();
```

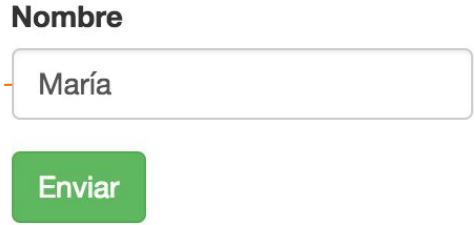
jQuery y Formularios

jQuery y Formularios

Con jQuery es muy fácil **obtener el valor de un campo** (`input`, `select`, `textarea`) de un formulario. Para ello se usa una función llamada `val`. Docs: https://api.jquery.com/val.

```
$("#btn-enviar").on("click", function () {
    var nombre = $("#name").val();
    if (nombre === "María") {
        alert("¡Bienvenida María!");
    } else {
        alert("Lo siento, no te conozco.");
    }
});
```

Ejemplo: un usuario ingresa su nombre en un campo de texto y el sistema realiza alguna validación.



Eventos

(Esto en realidad es un tema de JavaScript, no específicamente de jQuery)

¿Qué es un evento?

- Es “algo” que sucede mientras un usuario interactúa con una página web.
- Ejemplos de eventos:
  - El usuario presiona una tecla.
  - El usuario hace click en un botón.
  - El usuario agranda o achica una ventana.
  - Una página se terminó de cargar.

JavaScript permite detectar eventos.

Documentación: https://developer.mozilla.org/en-US/docs/Web/Events.

Tipos de eventos (1/3)

- Generados con el mouse (MouseEvent):
  - `click`
  - `dblclick`
  - `mouseenter`
  - `mouseleave`
  - `mouseover`
  - `mousemove` (http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_event_mouseenter_mouseover)
  - etc.

Probablemente el evento más usado.

Tipos de eventos (2/3)

- Generados con el teclado (KeyboardEvent):
  - `keydown`
  - `keypress` (produce un caracter)
  - `keyup`
- Generados con el browser/ventana (UIEvent):
  - `load`
  - `resize`
  - `scroll`
  - etc.

Tipos de eventos (3/3)

- Generados en un formulario:
  - `change`
  - `reset`
  - `submit`
  - `input`
- Otros:
  - `focusin`
  - `focusout`

Detectar un evento con JS (con jQuery) (1/2)

```
$("#btn-guardar").on("click", function() {
    /*
        Bloque de código que se ejecuta cuando el usuario
        hace click en el elemento de id btn-guardar.
    */
});
```

`on` es una función que recibe dos parámetros. El primero es un string y el segundo es una función.

En este ejemplo:

- El string que está pasando como primer argumento indica que se quiere “escuchar clicks” que sucedan sobre un botón de la página llamado `btn-guardar`.
- La función que se está pasando como segundo argumento es particular porque no tiene nombre. Es lo que se conoce como función anónima y es un caso particular de *Function Expression*.
  Esta función se ejecutará cada vez que un usuario haga click sobre el botón `btn-guardar`.

Detectar un evento con JS (con jQuery) (2/2)

```
$("#btn-guardar").on("click", guardar);
```

```
function guardar() {

    // Código para guardar datos...

    console.log("¡Se guardaron los datos!");

}
```

En lugar de usar una función anónima, se podría haber usando una función con un nombre, la cual luego también se puede llamar desde otros lugares del código.

Detectar un evento con JS (sin jQuery)

```
document.querySelector("#btn-guardar").addEventListener("click", guardar);
```

```
function guardar() {

    // Código para guardar datos...

    console.log("¡Se guardaron los datos!");

}
```

También se pueden detectar eventos usando JavaScript “puro”, sin necesidad de contar con una librería externa como jQuery. No lo vamos a usar en el curso.

Detectar un evento con JS (con jQuery) – Sintaxis

Sintaxis básica de la función `on`:

```
$("#boton").on("click", function() {...});
```

1. Seleccionar uno o varios elementos.
2. Nombre del evento que se quiere detectar.
3. Función que se llama al detectar el evento.
   A esta función se le llama *callback* o *handler*.

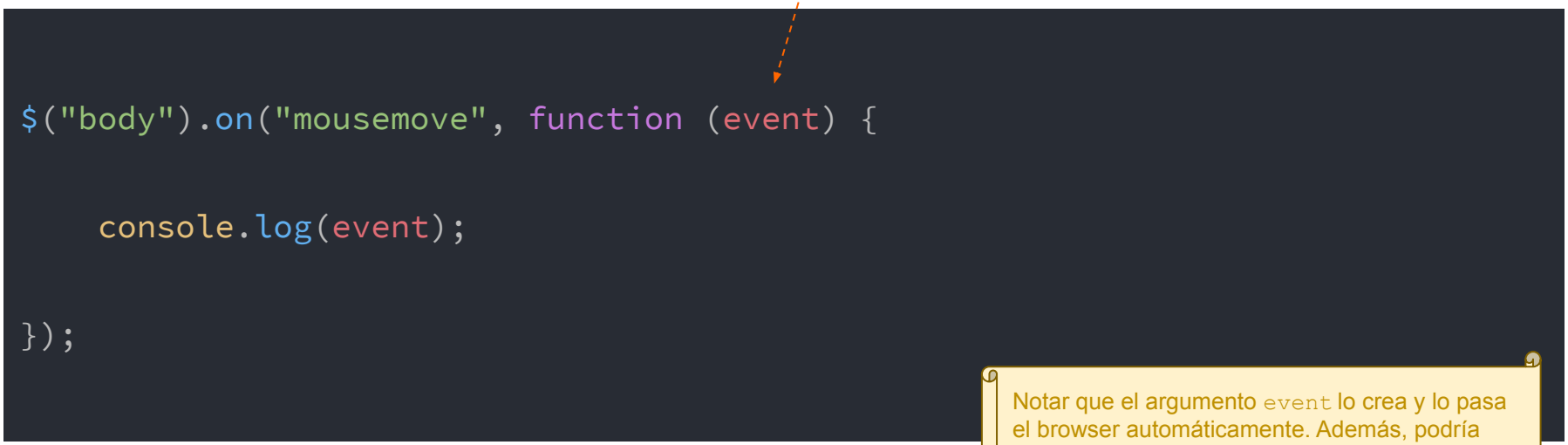
Documentación de la función `on`: https://api.jquery.com/on/.

Otras funciones jQuery de eventos: https://api.jquery.com/category/events/.

AVANZADO

Capturar el *objeto* de tipo Event

En caso de ser necesario, es posible “capturar” el *objeto* Event que se crea (automáticamente) cuando se genera el evento. Luego se puede analizar dicho objeto.



```
$("body").on("mousemove", function (event) {

    console.log(event);

});
```

Notar que el argumento `event` lo crea y lo pasa el browser automáticamente. Además, podría tener otro nombre, como por ejemplo: `e`.