



IT Academy
by KIBERNUM

Programa: Desarrollo de Aplicaciones Front-End

Módulo 2: Fundamentos de Desarrollo Front-End



Aprendizaje esperado

Utilizar biblioteca JQuery para la incorporación de elementos dinámicos en un documento html mediante el manejo básico de objetos y eventos del DOM.





¿Qué es un plugin?

Un plugin es una extensión que agrega funcionalidades específicas a jQuery o Bootstrap.

- ❑ Ventajas: Permite realizar tareas especializadas sin necesidad de escribir código desde cero.
- ❑ Ejemplos comunes: Carruseles de imágenes, modales pop-ups, validaciones de formularios.

Plugins populares de Bootstrap y jQuery

DOM



- ✓ Modal Pop-ups: Ventanas emergentes para mostrar información o capturar datos.
- ✓ Carruseles de imágenes: Desplazamiento dinámico de imágenes.
- ✓ Validaciones de formularios: Aseguran que los usuarios ingresen datos correctos.

¿Cómo funcionan los plugins?

DOM



Estructura interna de un plugin en jQuery

Un **plugin** en jQuery es una función que se “acopla” al objeto principal de jQuery (\$), permitiendo **extender sus capacidades** y reutilizar código de manera sencilla.

La estructura básica es:

```
(function($){  
  $.fn.miPlugin = function(opciones) {  
    // Código del plugin  
    return this.each(function(){  
      // Lógica que se aplicará a cada elemento seleccionado  
    });  
  };  
}(jQuery));
```

DOM



¿Cómo funcionan los plugins?

Estructura interna de un plugin en jQuery

```
(function($){  
  $.fn.miPlugin = function(opciones) {  
    // Código del plugin  
    return this.each(function(){  
      // Lógica que se aplicará a cada elemento seleccionado  
    });  
  };  
})(jQuery));
```

- (function(\$){ ... }(jQuery)); → crea una función autoejecutable para evitar conflictos con otras librerías.

- \$.fn → es donde se almacenan todas las funciones que pueden usarse como métodos de jQuery.
- this.each() → garantiza que el plugin se aplique a **cada elemento** seleccionado (por ejemplo, todos los <div> o <input> seleccionados con \$()).

DOM



Usar vs Crear un plugin

Usar un plugin existente

- Ideal cuando **ya existe una solución probada** que cumple con tus necesidades (por ejemplo, un carrusel o un modal).
- Ahorra tiempo y reduce errores.
- Suelen estar **bien documentados y actualizados**.

Ejemplo:

Usar el carrusel de Bootstrap o el plugin slick.js para deslizar imágenes.

Crear tu propio plugin

- Útil cuando necesitas una funcionalidad **muy específica** que no encuentras en otros plugins.
- Permite **personalizar completamente el comportamiento** y estilo.
- Ideal para proyectos donde la **coherencia de código** es importante (por ejemplo, mantener la misma estructura o estilo de todos los componentes).

DOM



Usar vs Crear un plugin

- Cuándo conviene cada opción

Situación	Recomendación
Necesitas una funcionalidad común (modales, sliders, tooltips)	Usa un plugin existente
Requieres un comportamiento muy específico o personalizado	Crea tu propio plugin
Buscas mantener código ligero y sin dependencias externas	Desarrolla uno propio
Priorizas rapidez y compatibilidad	Usa un plugin probado

DOM



Plugins más usados en Bootstrap

Plugin	Descripción	Ejemplo de uso
Modal	Ventanas emergentes	<code>.modal('show')</code>
Carousel	Galería de imágenes	<code>.carousel('next')</code>
Tooltip	Mensaje flotante al pasar el cursor	<code>[data-bs-toggle="tooltip"]</code>
Collapse	Mostrar/ocultar contenido	<code>.collapse('toggle')</code>

Errores comunes con JQuery y Bootstrap

DOM



- No incluir las librerías en el orden correcto.
- No esperar a que el DOM esté listo (`$(document).ready()`).
- Mezclar versiones incompatibles de Bootstrap/jQuery.

Ejemplo: Carrusel Dinámico con Bootstrap y jQuery

¿Qué aprenderemos?

- ✓ Cómo usar el carrusel de Bootstrap.
- ✓ Cómo personalizar su velocidad con jQuery.
- ✓ Cómo detectar eventos de navegación.
- ✓ Cómo pausar y reanudar el carrusel con el mouse.

Código HTML

```
<body>
  <!-- Carrusel de Bootstrap -->
  <div id="miCarrusel" class="carousel slide" data-bs-ride="carousel">
    <div class="carousel-inner">
      <div class="carousel-item active">
        
      </div>
      <div class="carousel-item">
        
      </div>
      <div class="carousel-item">
        
      </div>
    </div>
    <!-- Controles de navegación -->
    <button class="carousel-control-prev" type="button" data-bs-slide="prev">
      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
      <span class="visually-hidden">Anterior</span>
    </button>
    <button class="carousel-control-next" type="button" data-bs-slide="next">
      <span class="carousel-control-next-icon" aria-hidden="true"></span>
      <span class="visually-hidden">Siguiete</span>
    </button>
  </div>

  <!-- Incluir jQuery -->
  <script src="https://code.jquery.com/jquery-3.7.1.min.js" integrity="sha256-/JqT3SQfawRcv/BIHPThkBs00EvtFFmqPF/LYI/Cxo=" crossorigin="anonymous"></script>
  <!-- Incluir Bootstrap JS -->
  <script src="./assets/js/script.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcxpyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>
</body>
```

Ejemplo: Carrusel Dinámico con Bootstrap y jQuery

¿Qué aprenderemos?

- ✓ Cómo usar el carrusel de Bootstrap.
- ✓ Cómo personalizar su velocidad con jQuery.
- ✓ Cómo detectar eventos de navegación.
- ✓ Cómo pausar y reanudar el carrusel con el mouse.

Código JS

```
$(document).ready(function() {  
    $("#miCarrusel").carousel({  
        interval: 3000  
    });  
  
    $('.carousel-control-next').click(function(){  
        console.log("Siguiete imagen")  
    });  
  
    $('.carousel-control-prev').click(function(){  
        console.log("Imagen anterior")  
    });  
});
```

Discusión en Grupo

💬 Discusión en Parejas: ¿Cuándo usar jQuery en lugar de JavaScript puro?

◆ Pregunta clave: ¿En qué situaciones preferirías usar jQuery en lugar de JavaScript puro?

👥 Actividad en pares:

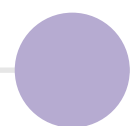
✓ Reflexiona y escribe tu respuesta.

✓ Explica tu punto de vista a tu compañero.

✓ Comparte una conclusión con el grupo.

📌 Posibles puntos de discusión:

¿Cuándo es más rápido usar jQuery? ¿Es más fácil de entender que JavaScript puro? ¿Hay casos donde NO deberíamos usarlo?





Resumen

- ◆ Introducción a jQuery
 - ✓ Qué es jQuery y por qué es útil.
- ◆ Cómo incluir jQuery en un sitio web
 - ✓ Descarga e instalación local.
 - ✓ Uso de CDN para incluir jQuery en línea.
- ◆ Manipulación del DOM con jQuery
 - ✓ Selección de elementos (`$()` y selectores avanzados).
 - ✓ Modificación de contenido (`.text()`, `.html()`, `.val()`).
 - ✓ Cambio de atributos y estilos dinámicamente.



Resumen

- ◆ Manejo de eventos con jQuery
 - ✓ Tipos de eventos (click, hover, keypress, focus, etc.).
 - ✓ Uso de .on() para delegar eventos.
- ◆ Uso de Bootstrap con jQuery
 - ✓ Cómo jQuery facilita la interacción con componentes de Bootstrap.
- ◆ Integración de plugins jQuery con Bootstrap
 - ✓ Qué son los plugins y cuándo utilizarlos.

¿Cómo crees que podrías mejorar la interactividad de tus proyectos usando jQuery?



Conceptos Fundamentales

Usar
document

✓ Siempre coloca tu código dentro de `$(document).ready()`. Esto asegura que el DOM esté completamente cargado antes de que cualquier manipulación o evento se ejecute.

Eventos sin
prefijo on

✓ A diferencia de JavaScript puro, donde los eventos se escriben con el prefijo on (como onclick), en jQuery puedes simplemente usar el nombre del evento sin necesidad de agregar el prefijo on.

No usar
!important

✓ Evita usar !important al agregar estilos mediante jQuery, ya que esto puede dificultar el mantenimiento del código y puede generar conflictos con otras reglas CSS.

Conceptos Fundamentales

Selectores en JQuery

✓ No abuses de selectores complejos o redundantes. Siempre que sea posible, selecciona los elementos de manera más eficiente utilizando clases o IDs específicos, en lugar de recorrer todo el DOM.

No usar Jquery para todo

✓ Evita sobrecargar tu proyecto con jQuery si las tareas pueden resolverse con JavaScript puro.



IT Academy

by KIBERNUM