



IT Academy
by KIBERNUM

Programa: Desarrollo de Aplicaciones Front-End

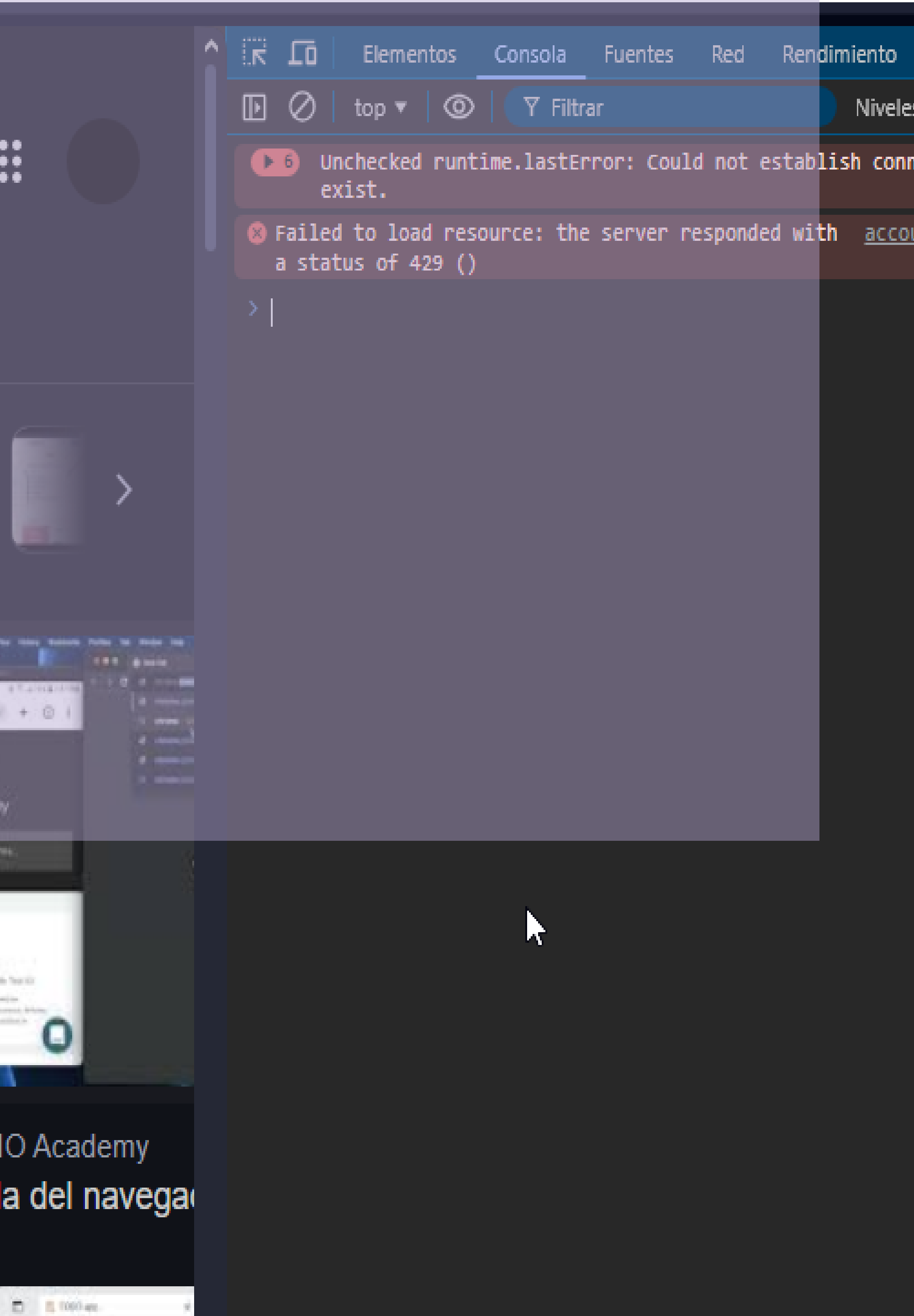
Módulo 2: Fundamentos de Desarrollo Front-End



Aprendizaje esperado

Utilizar código Javascript para la personalización de eventos sencillos dentro de un documento html dando solución al problema planteado.





Ejecutar y Depurar Código JavaScript en la consola

- ◆ ¿Qué es la consola del navegador?
 - Es una herramienta que permite interactuar con el JavaScript de una página web en tiempo real.
 - Se puede usar para probar código, depurar errores y verificar valores.
 - Compatible con todos los navegadores modernos.



Cómo abrir la consola en diferentes navegadores

◆ Atajos de teclado:



Google Chrome: F12 o Ctrl + Shift + i (Cmd + Option + I en macOS).



Firefox: F12 o Ctrl + Shift + K (Cmd + Option + K en macOS).



Microsoft Edge: F12 o Ctrl + Shift + i.



Safari (macOS): Activar herramientas de desarrollo en Preferencias > Avanzado y luego abrir con Cmd + Option + C.

Ejemplo: Ejecutando código en la consola

- ✓ Abre la consola del navegador
- ✓ Escribe la siguiente línea y presiona enter

```
console.log("Hola desde la consola");
```




Depurando el código JavaScript con la consola

- ◆ Uno de los métodos más comunes para depurar es usar `console.log()` para ver el valor de las variables o el estado de la ejecución en diferentes partes de tu código.
- ◆ Ejemplo: imprimir valores con `console.log()`

```
let a = 10;  
let b = 20;  
let resultado = a + b;  
console.log("El resultado es: " + resultado);
```




Detección de errores con la consola

- ¿Qué sucede cuando hay errores en el código?
- ❑ Cuando el código tiene errores, la consola del navegador muestra mensajes de advertencia o error.
- ❑ Esto ayuda a identificar y corregir problemas rápidamente.
- ◆ Ejemplo:

```
console.log("Hola desde la consola");
```


Selectores, Manipulación y Eventos

Ahora veremos conceptos básicos de JavaScript, enfocándonos en la interacción con el DOM (Document Object Model) para crear páginas web dinámicas e interactivas.

Seleccionando Elementos con getElementById

- ◆ **Sintaxis:** El método `getElementById` es simple y directo. Se utiliza para seleccionar un elemento del DOM por su ID único. La sintaxis es:

```
var element = document.getElementById("idDelElemento");
```

- ◆ **Ejemplo:** Supongamos que tenemos un div con el ID "miDiv". Para seleccionarlo, usaríamos:

```
var miDiv = document.getElementById("miDiv");
```

La variable `miDiv` ahora contiene una referencia al div, permitiéndote manipularlo.

Manipulando Texto y Valores

- ◆ **innerHTML**: te permite obtener o establecer todo el contenido HTML dentro de un elemento, incluyendo texto y otros elementos HTML.

```
var contenidoHTML = document.getElementById("demo1").innerHTML;  
console.log(contenidoHTML);
```

- ◆ **textContent**: te permite obtener o establecer solo el texto dentro de un elemento, sin procesar el HTML.

```
var texto = document.getElementById("demo2").textContent;  
console.log(texto);
```

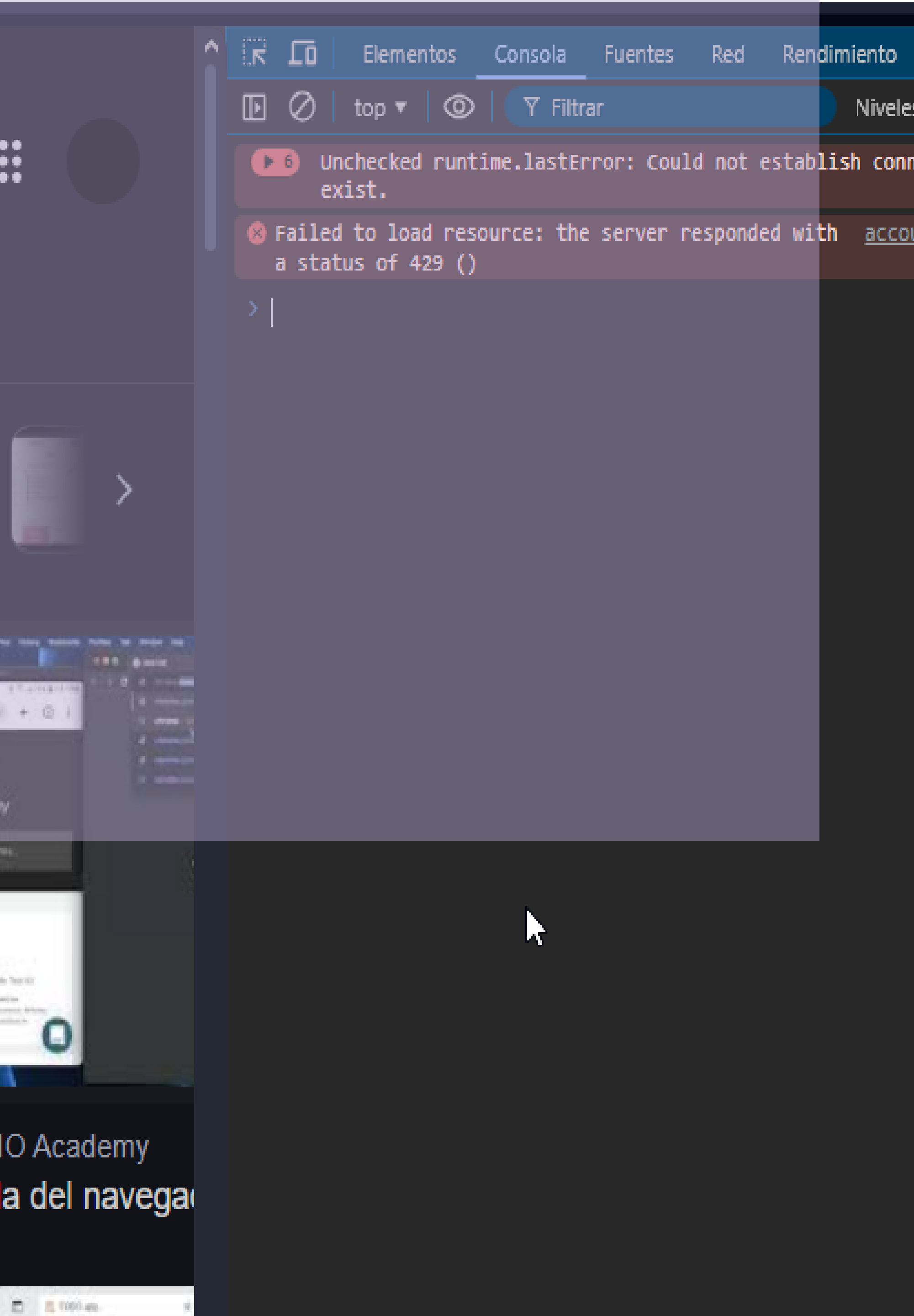

Manipulando Valores en Formularios

- ◆ **Obteniendo Valores:** Para obtener el valor de un elemento de formulario, como un campo de entrada, se utiliza la propiedad value.

```
var input = document.getElementById("miInput");  
console.log(input.value); // Muestra el valor actual
```

- ◆ **Modificando Valores:** Para modificar el valor de un elemento de formulario, simplemente se asigna un nuevo valor a la propiedad value.

```
input.value = "Nuevo valor";
```

Actividad Práctica Guiada

Nombre del ejercicio:

Manipulación del DOM con getElementById, textContent e innerHTML.

Objetivo:

Tomar el texto ingresado por el usuario y:

1. mostrar un saludo,
2. agregar una entrada formateada a un historial,
3. finalizar mostrando un resumen con alert.

Requisitos previos:

- Conocer getElementById, textContent, innerHTML y onclick.
- 3 archivos: index.html, styles.css, script.js.
- Un navegador y editor de código.

Actividad Práctica Guiada

Código HTML base

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Manipulación del DOM</title>
  <link rel="stylesheet" href="styles.css" />
</head>
<body>
  <main class="container">
    <h1>Formulario interactivo</h1>
    <p>Ingresa tu nombre y usa los botones para ver el efecto en el DOM.</p>

    <label for="nombre">Nombre:</label>
    <input id="nombre" type="text" placeholder="Escribe tu nombre" />

    <div class="acciones">
      <button id="btnMostrar" onclick="mostrar()">Mostrar saludo</button>
      <button id="btnFormatear" onclick="formatear()">Agregar al historial</button>
      <button id="btnFinal" onclick="finalizar()">Finalizar</button>
    </div>

    <section class="salida">
      <h2>Salida</h2>
      <p id="salida"></p>
    </section>

    <section class="historial">
      <h2>Historial</h2>
      <div id="historial"></div>
    </section>
  </main>

  <script src="script.js"></script>
</body>
</html>
```


Actividad Práctica Guiada

Código CSS base

```
* {
  box-sizing: border-box;
}

body {
  font-family: system-ui, -apple-system, Segoe UI, Roboto, Arial, sans-serif;
  background: #f7f7fb;
  margin: 0;
  padding: 24px;
}

.container {
  max-width: 720px;
  margin: 0 auto;
  background: #fff;
  padding: 24px;
  border-radius: 12px;
  box-shadow: 0 6px 18px rgba(0, 0, 0, .06);
}

label {
  display: block;
  margin: 12px 0 6px;
  font-weight: 600;
}

input[type="text"] {
  width: 100%;
  padding: 12px;
  border: 1px solid #ddd;
  border-radius: 8px;
}
```

```
.acciones {
  margin: 16px 0;
  display: flex;
  gap: 12px;
  flex-wrap: wrap;
}

button {
  padding: 10px 14px;
  border: none;
  border-radius: 10px;
  cursor: pointer;
  background: #4f46e5;
  color: #fff;
  font-weight: 600;
}

button:hover {
  filter: brightness(1.05);
}

.salida,
.historial {
  margin-top: 16px;
}

#salida {
  min-height: 24px;
  font-size: 1.05rem;
}

.historial-item {
  background: #f3f4ff;
  padding: 10px 12px;
  border-radius: 10px;
  margin: 8px 0;
}
```

Actividad Práctica Guiada

Mostrar saludo con.textContent

```
// script.js - Paso 1: Mostrar saludo usando.textContent
function mostrar() {
  const nombreInput = document.getElementById('nombre');
  const salida = document.getElementById('salida');

  const nombre = nombreInput.value.trim();
  if (nombre === '') {
    salida.textContent = 'Por favor, escribe tu nombre arriba.';
    return;
  }

  // Cambiamos solo el texto, sin HTML
  salida.textContent = `¡Hola, ${nombre}!`;
}
```

Qué aprendemos:

- `getElementById('salida')` para seleccionar elementos.
- `textContent` para escribir **solo texto** seguro (sin etiquetas).

Actividad Práctica Guiada

Agregar entrada con innerHTML

```
// Paso 2: Agregar una línea al historial usando innerHTML
let contador = 0;

function formatear() {
  const nombre = document.getElementById('nombre').value.trim();
  const historial = document.getElementById('historial');

  if (!nombre) return;

  contador++;
  // Agregamos HTML con formato (negritas, énfasis)
  const item = `

<strong>${contador}</strong> Guardado: <em>${nombre}</em></p>`;
  historial.innerHTML += item;
}


```

Qué aprendemos:

- innerHTML permite **insertar HTML** con formato.
- Diferencia práctica con `textContent`.

Actividad Práctica Guiada

Resumen final con alert + reset

```
// Paso 3: Mostrar resumen con alert y resetear el formulario/DOM
function finalizar() {
  const nombre = document.getElementById('nombre').value.trim();
  const mensajeBase = nombre ? `¡Listo, ${nombre}!` : '¡Listo!';
  alert(`${mensajeBase}\nEntradas en el historial: ${contador}`);

  // Limpiar interfaz
  document.getElementById('salida').textContent = '';
  document.getElementById('historial').innerHTML = '';
  document.getElementById('nombre').value = '';

  // Reiniciar contador
  contador = 0;
}
```

Qué aprendemos:

- Lectura del valor actual del input.
- Uso de alert para un **resumen final**.
- Reset del contenido con textContent, innerHTML y del input.

Actividad Práctica Guiada

Salida esperada

The screenshot shows a web application interface with a dark theme. At the top, there are browser tabs: 'eación...', 'Sesión 4- Act...', 'tos Cód...', and 'Códig...'. The main content area has a title 'Formul...' and a description 'Ingresa tu nombre y usa los botones para ver el efecto en el DOM.' Below this is a form with a label 'Nombre:' and a text input field containing 'Catalina'. There are three buttons: 'Mostrar saludo', 'Agregar al historial', and 'Finalizar'. Below the form, there is a section titled 'Salida' showing '¡Hola, Catalina!'. At the bottom, there is a section titled 'Historial' with a list item '1. Guardado: Catalina'. A modal dialog is open in the center, titled '127.0.0.1:5500 dice', containing the text '¡Listo, Catalina!' and 'Entradas en el historial: 1', with an 'Aceptar' button.

127.0.0.1:5500 dice

¡Listo, Catalina!

Entradas en el historial: 1

Aceptar

Formul

Ingresa tu nombre y usa los botones para ver el efecto en el DOM.

Nombre:

Catalina

Mostrar saludo Agregar al historial Finalizar

Salida

¡Hola, Catalina!

Historial

1. Guardado: *Catalina*



IT Academy

by KIBERNUM