

Auxiliar N°2

JS

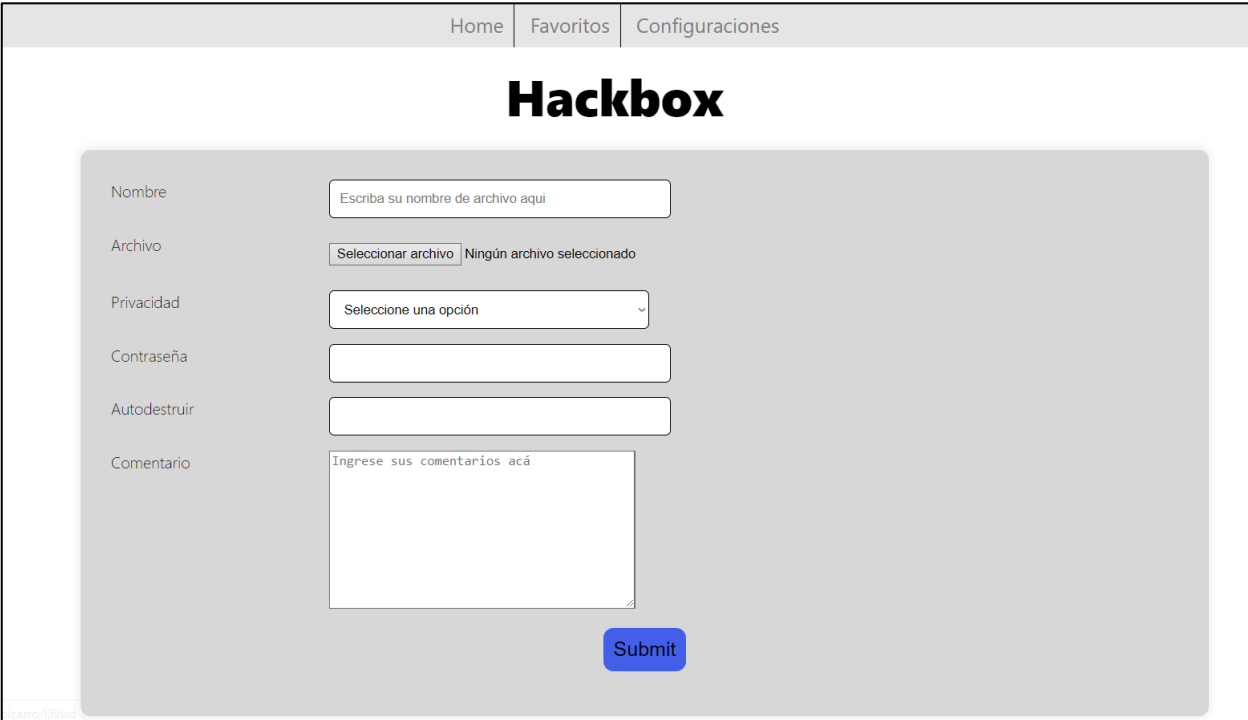
Desarrollo de Aplicaciones web
Auxiliar: Pablo Pizarro – Gabriel Iturra

Sobre hoy

- JS
 - Breve descripción del lenguaje de programación
 - Acceso al DOM
 - Modificación de formularios

Problema de hoy

- Añadir funcionalidad a nuestro Hackbox. En el envío del formulario, validar que el formulario cumpla con los requerimientos.
- Alertar al usuario de los errores con JS.



The screenshot displays the Hackbox web application interface. At the top, there is a navigation bar with links for 'Home', 'Favoritos', and 'Configuraciones'. The main heading 'Hackbox' is centered. Below it, a form is presented with the following fields:

- Nombre:** A text input field with the placeholder text 'Escriba su nombre de archivo aqui'.
- Archivo:** A file selection interface showing a button 'Seleccionar archivo' and the status 'Ningún archivo seleccionado'.
- Privacidad:** A dropdown menu with the placeholder text 'Seleccione una opción'.
- Contraseña:** A text input field.
- Autodestruir:** A text input field.
- Comentario:** A text area with the placeholder text 'Ingrese sus comentarios acá'.

A blue 'Submit' button is located at the bottom right of the form area.

Javascript

- Lenguaje de programación interpretado, orientado a objetos.
- Similar a Python.

```
function Object(){  
  this.variable = 1;  
  this.variable2 = "String";  
  this.lista = [1, 2, "String", false];  
  this.method = function(){  
    let a = 1; // Variable local  
    if (a === 1){  
      return 2;  
    } else {  
      return false;  
    }  
  }  
}
```

- Definición de objetos se realiza con *funciones*.
- Para declarar variables locales se utiliza **let** y globales **var**.
- Funcionamiento de listas es idéntico a Python.
- Para crear objetos utilizar **new Object(...)**

Javascript

- La estructura de objetos en Javascript se denomina JSON (Javascript object notation).

```
let modelo = {  
  enabled: false,  
  mesh: {  
    "mesh-data": null,  
    "list-vertex": new VertexList(),  
    "list-faces": [[1,2,3], [4,5,6]],  
    "list-edges": edges[0],  
  }...  
}
```

```
let a = modelo.enabled; // false  
let b = modelo["mesh"]["mesh-data"];  
modelo["mesh"]["list-edges"] = null;
```

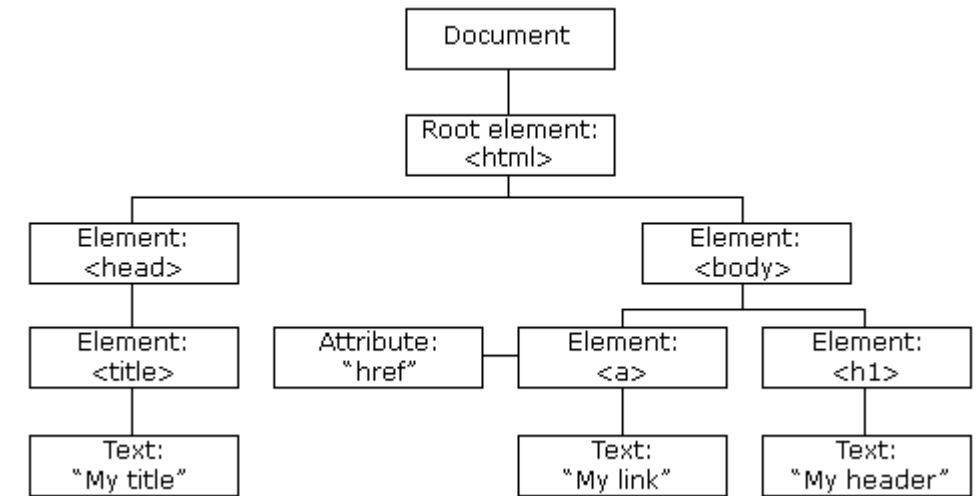
- Objetos similares a los diccionarios de Python.

- Ver tutorial: <https://www.w3schools.com/js/>

Accediendo al DOM

Con JS:

- Puede cambiar todos los elementos HTML de la página
- Puede cambiar todos los atributos HTML de la página
- Puede cambiar todos los CSS de la página
- Puede eliminar cualquier elemento o atributo HTML
- Puede añadir nuevos elementos o atributos HTML
- Puede reaccionar a todos los eventos
- Puede crear nuevos eventos HTML



Event Handlers	
Event Handler	Description
onabort	User cancels a page load.
onblur	An element loses focus because the user focuses on a different element.
onchange	User changes the contents of a form element or selects a different check box, radio button, or menu item.
onclick	User clicks an element with the mouse.
ondblclick	User double-clicks an element with the mouse.
onerror	Browser encounters an error in the scripts or other instructions on the page.
onfocus	An element becomes the focus of the user's attention, like a form field when you start typing in it.
onkeydown	User presses and holds a key on the keyboard.
onkeypress	User presses and immediately releases a key on the keyboard.

Accediendo al DOM

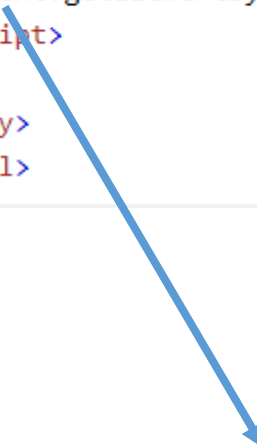
- Acceder al contenido de una etiqueta:

```
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```



A través de **document** se accede a todas las propiedades y métodos del DOM

DOM Document

Encontrar elementos HTML

Método	Descripción
<code>document.getElementById(<i>id</i>)</code>	Encuentra un elemento por su ID
<code>document.getElementsByTagName(<i>name</i>)</code>	Encuentra un elemento por un tag
<code>document.getElementsByClassName(<i>name</i>)</code>	Encuentra un elemento por un nombre de clase

Cambiar elementos HTML

Propiedad	Descripción
<code>element.innerHTML = <i>new html content</i></code>	Cambia el HTML interno de un elemento
<code>element.attribute = <i>new value</i></code>	Cambia el valor atributo de un elemento
<code>element.style.property = <i>new style</i></code>	Cambia la propiedad de un elemento
Method	Descripción
<code>element.setAttribute(<i>attribute</i>, <i>value</i>)</code>	Establece un atributo y un valor

DOM Document

Agregar o eliminar elementos

Método	Descripción
<code>document.createElement(<i>element</i>)</code>	Crea un elemento HTML
<code>document.removeChild(<i>element</i>)</code>	Elimina un elemento HTML
<code>document.appendChild(<i>element</i>)</code>	Agrega un elemento HTML
<code>document.replaceChild(<i>new</i>, <i>old</i>)</code>	Reemplaza un elemento HTML
<code>document.write(<i>text</i>)</code>	Escribe un texto en el stream de salida

Manejar eventos en HTML

Método	Descripción
<code>document.getElementById(<i>id</i>).onclick = function(){<i>code</i>}</code>	Atrapa un evento

