

DOM

Cada etiqueta HTML es un objeto que puede tener padres e hijos

EL DOM puede ser modificable (añadir, eliminar modificar elementos)

Existen 12 tipos de nodos (se utilizan principalmente 5).

Document (Nodo raíz)

Element (Etiquetas HTML que pueden ser padres e hijos)

Attr (Atributos)

Text (Texto)

Comment (Commentarios)

Operaciones básicas en el DOM

Obtener

`getElementByTagName()`

`getElementById()`

`getElementByClassName()`

Los 3 métodos anteriores son de “document”

Cambiar

`element.innerHTML = nuevo HTML`

`element.attribute = nuevo valor`

`element.style.property = nuevo estilo`

`element.setAttribute = nuevo atributo, nuevo valor`

Crear y eliminar

`document.createElement(element)`

`document.removeChild(element)`

`document.appendChild(element)`

`document.replaceChild(element)`

Eventos

Suceso detectada de una acción preprogramada

Uso de eventos

- Como un atributo de HTML (onEvent="function()")
- En js (document.getElementById('main-button').onclick = function_name;)

Tipos

Eventos de entrada (ejemplos):

- Onblur
- Onchange
- OnFocus
- Onkeydown (Cuando presiona la tecla)
- Onkeypress (cuando suelta la tecla)

Eventos de click (ejemplos):

- Onclick
- Ondblclick

Eventos de carga:

- No ejemplos

Patrones de diseño

Soluciones probadas, expresivas y fáciles de mantener para resolver problemas comunes en el desarrollo de software.

Arquitectura de software

Estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos.

MVC

Patrón de diseño que separa la lógica de la aplicación, la vista y los datos.

Ventajas:

- Mantenible

- Pauta para desarrollo colaborativo
- Estandarización de código
- Cambios independientes de acuerdo al problema

Modelo

Representación de los datos con los que la aplicación va a interactuar.

- Consulta la DB
- Consulta de apis externas
- CRUD de datos

Vista

Representación gráfica de los datos al usuario

- Mostrar los datos al usuario final
-
-

Controlador

Gestiona el flujo de información entre el modelo y la vista

- Contesta peticiones al usuario (vista)
- Responde información estructurada por el modelo

Fundamentos de MVC

- Convención sobre configuración: Preocupate por codificar, no por configurar
- No repitas(principio “DRY”): Su principal ventaja es mantener el código centralizado