

Código HTML base.

<!doctype html>

<!-- jsontable.html -->

<head>

<title>JSON To Table</title>

<style>

table,td,th {

border:solid 1px;

border-collapse:collapse;

}

</style>

</head>

<body>

<h1>JSON To Table</h1>

<p><form>

<button name=string type=button>Table Using HTML String</button>

<button name=dom type=button>Table Using DOM API</button>

<p><button name=clear type=button>Clear</button>

<p><output name=out></output>

</form>

<script src=jsontable.js></script>

</body>

</html>

La idea es simple: convertir un JSON string en un objeto – atravesar el objeto JSON – crear la estructura del DOM como cadenas u objetos – insertar el nuevo DOM en la página.

La información está codeada pero puede venir de cualquier parte.

// jsontable.js

document.addEventListener("DOMContentLoaded", ()=>{

let jsonstr=`[

{"id":"2", "first\_name":"Sam","last\_name":"Smith", "phone":"111-222-3333","email":"ssmith@yahoo.com","address":"33 Birch Rd","city":"Miami","state":"FL"},

{"id":"3", "first\_name":"Brad","last\_name":"Traversy", "phone":"211-322-4333","email":"brad@test.com","address":"222 South St","city":"Portland","state":"FL"}

]`

let jsonobj=JSON.parse(jsonstr)

let form=document.querySelector("form")

Este es el

POO.

Encapsulamiento: propiedad que tienen los objetos para ocultar información/restringir acceso. Se trata de un principio de responsabilidad única.

Buscar la manera más sencilla para solucionar problemas.

Herencia: Es la característica que nos permite heredar atributos y métodos. No existen contratos a comparación de la interface.

Abstracción: Cómo observamos un objeto y cómo lo definimos lógicamente, cómo lo estamos asimilando. Como interpretamos las cosas.

Analizar el entorno y modelar el sistema.

Generalización:

Particularización/Especialización:

Polimorfismo: Comportamiento de los objetos. Un objeto comportándose diferente en tiempo de ejecución.

Clase: representación del objeto/modelado del objeto.

Objeto: instancia de una clase.

Métodos: Acciones de la clase.

Atributos: Características de la clase.

Interface: Un contrato. Algo que voy a obligar a que un objeto cumpla. En caso de que no se cumpla, el objeto no existiría.

En las interfaces solo hay métodos SIN implementar.

Objetos compuestos:

Métodos abstractos:

Objetos abstractos:

Clase abstracta: Es una clase que no está completa.

Sobrescritura de métodos: Solamente se da cuando hay una herencia.

Sobrecarga de métodos: Definir un método varias veces con diferentes parámetros.

Singleton: Es un patrón de diseño. Un objeto único.

Bean: La clase más simple que se pueda encontrar en un lenguaje de programación.

Reflexión:

Introspección: En tiempo de ejecución.

Casting de componentes:

Leer patrones de diseño/patrones arquitectónicos.

JS

Función de orden superior:

Cuando invocamos a this en una función cualquiera, apunta al objeto global (Window). Cuando se ocupa como un método, hace referencia a un objeto. En una función anidada hay un error, porque la toma como si se definiera como en una función cualquiera.

This That en javascript

Arrow functions =>

Un archivo JSON debe llevar comillas dobles.

Callback:

Todas las funciones que no retornan un valor es como si tuviera “undefined”.

El load se dispara del XMLHttpRequest, se mantendrá a la escucha.

Por dentro de la función request se hace un callback.

AddEventListener ejecuta la función onLoadCallback, por dentro esta es la función que hace el callback.

Patrón de diseño observador.

Un elemento puede tener atributos y propiedades. No es lo mismo los atributos que las propiedades.

Los atributos son en teoría strings. Sirven para definir en código HTML cómo se comporta un elemento, es más bien un indicador.

Las propiedades son las que realmente hacen el cambio de estados, estilos, etc.

classList recientemente en el DOM, lo controla como una lista. Trae todo como una lista.

className contraparte de setAttribute. Pone todo el string del atributo class.

setAttribute para colocar un atributo.

append puedes pasarle strings y convierte a tags

appendChild solo recibe nodos.

Documents fragments

Para ahorrar cálculos de los elementos. Funciona como un contenedor temporal.

Template es un document fragment.