31/03/2025

Se tiene un index.html el cual en el header tiene una lista de botones los cuales se recuperan en el main.js y se le asigna a cada uno una funcion que extrae datos de la url del api a la que pertencen las funciones viene de un archivo aparate

se traen las funciones con el

impor{} from xxxxx

El archivo que tiene las funciones con las funciones las api se llamam api.js y funcionan con el fectch y .then . Chacth

fetch toma el endpoint y por defecto realizan la funcion “GET” se crean una para cada enpoit ya que la estructura del api son difrentes ,

fecth user se comenta pues es de prueba

fectch pokemon el api nos ofrece la oportunidad de elegir la cantidad de datos que queremos traer, con limit y offset los cuales establecemos en 20 y 0 respectvamenete ,

realizamos la peticion :

- manejamos con then la respuesta y si trae un error los manejamos con un creador de errores, si no nos retronara una respuesta que viene en un formato que debemos pasar a json .

Y haci con then enviamos la data con eligiendo el campo de results e invocamos a la funcion getPokemon

-si no hay errores desde la primer peticion los capturamos con cathc.

Fetch joke la misma estructura que la anterior pero aca no elegimos ningun campo de la data y llamamos a la funcion getJoke .

Getpokemon elije un contenedor del html el cual esta vacio,

con la data resuelta con el metodo forEach que maneja arrays u objetos y toma cada elemento y le aplica el codigo que escribamos , etonces por cada elemento crea una variable boton que entra al documento crea un elemento button, entra a la lista de clases y añade la clase btn , entra al contenido e imprime . Name que es la estructura del api para darnos del nombre , le añade un eventlistener con la funcion getDataPoke y el elemento creado se añade a un boton

01/04/2025

se crea getNameaApi toma la data que viene de fecthUser , la cual toma el obtiene el contenedor del DOM con el id “user”, como es un arrray *data* usamos forEach y por cada item del array se crea un elmento con create Element se le añadde clases ccon classList.add establecemos atributos con setAtribute y ya al elemento recuperado del DOM le agregamos los hijos de esta funcion.

GETJOKE FUNCION : recuperamos el div con el id  *joke* , como la api nos devuelve dos tipos de chistes debemos manejarlos con *hasOwn* si el objeto solo tiene un *joke*  en la respuesta creara un elemento h4 con el joke , si no si mas tiene un setup que es un chiste de dos partes crea el encabezado en un h4 y tiempo despues son settiemout crea el resto del chiste con h6 amabas llaman create element que esta dentro de la misma funcion y

div imPokemon que va a traer la imagen del pokemon el nombre y los poderes

con getDataPoke se van a traer los datos y colocar en el api

*elemento.innertHtml o textContent o si es array= “ “ se puede resetear el contenido cada vez que invoquemos una funcion*

*-*showJokeBox.innerHTML = ""; // *Limpiar el contenido previo*

*se resetea el contenido de showJokeBox .*