

Paso a paso Html

Linkear todo

Body Cal llamar al `onsubmit` lo que hay en el js)

`autocomplete "off"` = NO historial busqueda

`toda la pokdex = data-pokemonCard`

`Nombre del pokemon = data-pokemonName`

`imagen del pokemon = data-poke-ContainerImage`

`Nº del pokemon = data-pokeID`

`¿Qué es? y tipo = data-pokeTypes`

`descripciones varias = data-pokeStats`

Paso a paso Js (Lo dividi por descripción de lo que se hace)

1. `query Selector` para llamar data del html

2. `Mapear diccionario` (const) mapeando tipo de poke y dar color

3. `Crear función Search` (se llama en el `onsubmit`)

3.1. `cancela submit` con `preventDefault`

3.2. `event.target.pokemon`: obtener valor input (`html "pokemon"` línea 14)

3.3. `fetch` = Nombre API = `to LowerCase` (Casi no importa si escriben con mayus o minus)

3.4. `then`: obtener `responsive` =

Crear función `renderPokemonData` la cual envia la response

4. `Crear función` (const) recibe la data

4.1. obtener los `sprites` (img del poke)

4.2. " `types` y `stats`

4.3. `Poke name`, `PokeImg`, `pokeID` (obtiene img, nombre y ID)

5. `Se encarga de darle el color` relacionado al pokemon tanto al fondo de la img como al texto del type

6. `función renderPokemonTypes` = recibe `types` =)

6.1. `innerHTML` = borrar lo que tenia y poner nuevos

6.2. `type` por `poke` recibido

6.3 element = type TextElement (con div)

6.4.  .style.color = esto le da color al type específico del pokemon

6.5 appendChild = crea los + types

7. Función renderPokemonStats = imprime los stats del Pokemon

7.1. Inner HTML = Borra lo anterior

7.2. stats.forEach = imprime estadísticas al pokemon

7.3 elemento statElement se le da el valor del Api

7.4 appendChild llama la ^(stat.stat.name y stat.base_stat)

info y por último a los pokestats se les da toda la info a imprimir