# JS

Js= programación del lado del cliente = en el navegador

Js= interpretado = no compilado , di

This -> no es una variable es una palabra reservada que hace referencia al elemento padre de el contexto que me encuentrec

npm init vite@latest NAME -- -- template vallina

Operaciones

Suma= n+m

Resta=n-m

Milti= n\*m

Divi= m? n/m ”no se puede dividir”

Return `suma ${suma},`resta ${resta},,multi ${multi},${divi }´

Array

Const miArray ={1,2,3,4,5,6,7,8,9};

Const miArray2 = new Array(1,2,3,4);

Const miArray3 =[];

Meter elementos en la posición que queramos no se puede sobre escribir a otros elementos con Marray [0]=‘hola”

miArray2[miArray2.lenght]=11

miArray2.push añade un elemento al final

miArray2.pop elimina el ultimo elemento

miArray2.unshift(0)¸ mete un elemento al principio del array

miArray2.shift() saca un elemento del principio del array

miArray2.slice((stard) 1/(end) 4) coge la posición que indica stard astas la end “”niño gua”””

en negativos empieza al reves

miArray2.splice()

map

estructura basada en clave valor

ejemplo

### //tamaño

.size

##### //verificar la existencia de un elemento dentro

.has

### //borrar

.delete(“perro”);

##### //siquiero recorrer las clves llamo a keys ,valores values y pra los dos'cuidado te llevas un array' es entries

for (let mes of newMap2.keys()){ console.log(mes); }

for (let indicepluvio of newMap2.values()){ console.log(indicepluvio); }

for (let [mes,indicepluvio] of newMap2.entries()){ console.log(`mes : ${mes}---> indice plubiometrico : ${indicepluvio}`); }

newMap2.forEach(()=>);

ejemplo

newMap2.forEach((indicepluvio,mes)=>  console.log(`mes : ${mes}--->

newMap2.forEach((indicepluvio,mes)=>  console.log(`mes : ${mes}---> indice plubiometrico : ${indicepluvio}`));

/// a=[["enero","15"],["febrero","27"],["enero","45"]]

//...a ---> QUITAN LA ESTRUCTURA  ES DECIR DA CADA ARRAY POR SEPARADO/["enero","15"] / ["febrero","27"] / ["enero","45"]

//[...a] --->[["enero","15"] , ["febrero","27"] , ["enero","45"]]<-- Array

Objeto

Son pares de clave valor mas potente que los map

LOCALSTORAJE

 /// isetrar elementos en el local storaje

 localStorage.setItem("NOMBRE","Jairo");

# para meter objetos

 const Usuario= {"name":"Jairo","cp":18270};

 localStorage.setItem("User",JSON.stringify(Usuario));

# para sacar

 alert(localStorage.getItem("nombre") );

## borrar y Borrar todo

 localStorage. removeItem("NOMBRE")

 localStorage.clear();

set

const miset= new set(‘loro’,’pato’,’perro’);

### //añadir

.add(“gato”);

### //borrar

.delete(“perro”);

### //tamaño

.size

### //verificar la existencia de un elemento dentro de mi set :

Miset.has(“loro”);

### //para recorrer usaremos el for … of

For( let animal of miset){

Consoloe.log(animal);

}

### //convertir en array

arrayAnimales=Array.from(animales)

### //Stread operator para convertir de set a Array.

arrayAnimales2= [ …animales ];

ejemplo

const a= [1,0,2,3,2,1,4,5,2,1]

const aSet=new set(a);

const a1 = Array.from(aSet)

### //operaciones con set

### //UNIÓN

Const c= new set([1,2,3])

Const d= new set([5,2,1])

Const cd = new set ([ …c , …d ])

### //DIFERENCIA 🡪 c-d //FILTER

Const c\_d= new set( [ …c ].filter( (num) =>!d.has(num) ) )

### //INTERSECCIÓN -🡪 c\_i\_d

Const c\_d= new set( [ …c ].filter( (num) =>d.has(num) ) )

# REACT

Crea el proyecto en recat => npm créate vite@latest .

Instalar dependencias => npm i

Abre la carpeta en el visual=> npm code .

Iniciar el servidor de react => npm run dev0

Esneapers => rafce