CLASE NRO. 11 – ARRAYS / VECTORES.

Tipo de Clase: Teórico Práctica

Duración: 120 minutos

Objetivo de la clase:

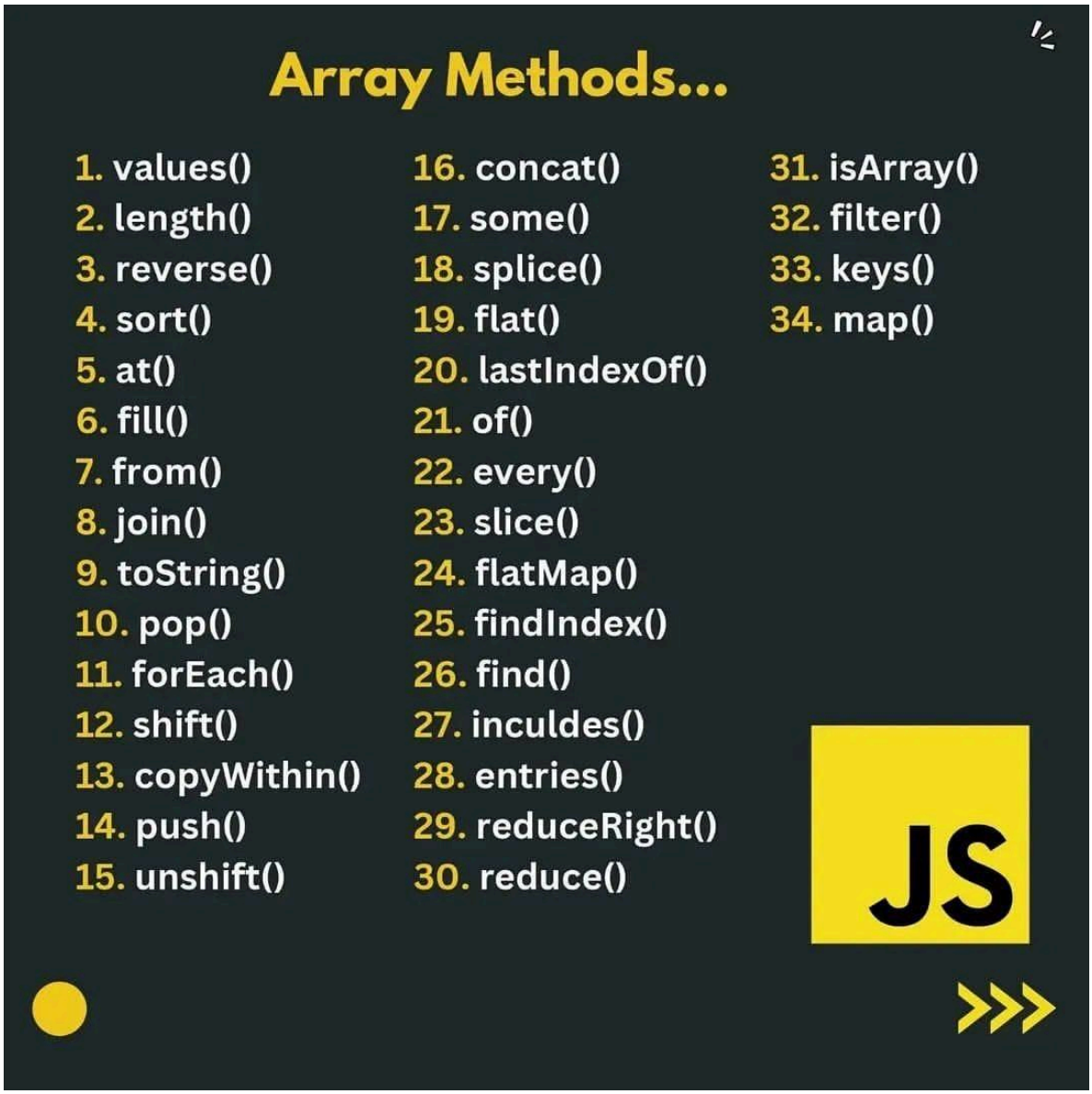
Al finalizar la clase, el alumno deberá comprender que la forma de declarar literalmente vectores, acceder a posiciones de un vector para cargar elementos, mostrar el contenido de una posición del vector, formas de recorrido de un vector, funciones predeterminadas que tienen los vectores. Estos son los puntos que se deberían observar en la clase

- Declarar un vector literal
- Mostrar el contenido de un vector
- Acceder a una posición del vector y modificarlo
- Agregar al final **push()**
- Eliminar el último valor y devolverlo **pop()**
- Eliminar el primer elemento del vector y devolverlo **shif()**
- Agregar al principio **unshif()**
- Eliminar elementos contiguos **splice()**
- Devolver la ubicación de un elemento **indexOf() y lastIndex()**. Uno retorna la primera ubicación y el otro devuelve la última posición.
- Devolver un elemento y la ubicación, pero con criterios más flexibles y definidos por el usuario como **find y findIndex**.
- Filtrar y seleccionar y devolver un conjunto de elementos del vector que cumplan con una condición específica. **filter**
    - Forma 1: con una arrow function que no tiene la palabra los () para el parámetro, no tiene las llaves {} del cuerpo y no lleva la palabra return;
    - Forma 2: con una arrow function completa que tiene los () y la palabra return y las llaves del cuerpo {}
    - Forma 3: con una función anónima clásica definida dentro del parámetro de filter
    - Forma 4: con una función declarada previamente y pasarle como parámetro la función dentro del filter.
- Recorrer tradicionalmente un vector con un ciclo **for**
- Recorre un vector con **vector.forEach()**
- Transformación de los elementos de un vector con **vector.map()** recorre los elementos, aplica la función pasada como parámetro y devuelve cada elemento transformado, a todos ellos los devuelve y los agrega en un vector de salida.
- Funciones reductoras con **reduce()**. Que permiten acumular valores, encontrar el mayor, el menor, etc.
- Función de ordenamiento **sort()** con las particularidades cuando son valores string o cuando son valores númericos.

- Transformar un vector y darlo vuelta “al revés” con **reverse()**.
- Particularidad de copiar variables o constantes que contienen vectores. Es decir, se crean referencias a los elementos y no se realiza una copia del elemento. Dado este problema surge la necesidad de hacer una clonación de esos elementos y sale **spreadOperator también se puede usar slice() pero es más completo y flexible spread.** El problema de la copia por referencia ocurre con Vectores y con Objetos.
- spreadOperator …
    - spread para copiar o clonar vectores
    - spread para unir en un vector dos o más vectores
    - spread para crear una copia ampliada de un vector
    - spread para clonar un objeto o ampliarle las propiedades (tipo fusión)
    - convertir lista de nodos ó elementos que se obtienen por querySelectorAll en un array
    - spread como una forma de pasar parámetros en una función, indicándole que los parámetros p1, p2, p3 están en el vector que tiene justamente 3 posiciones. Esto sucede en la invocación de las funciones.
    - Spread como forma de declarar una función que recibirá múltiples parámetros, de esta forma internamente la función los trata como si fueran un vector y en la invocación se puede hacer invocacion_funcion(p1,p2,p3,p4,p5).
    - Convertir colecciones o listas en Arrays

-

Definición: es una estructura de datos lineal que permite el almacenamiento de múltiples valores y la posibilidad de accederlos a todos mediante un solo identificador, en este caso el nombre del vector.



LINK PROYECTO:

https://drive.google.com/file/d/1fWX-F2PzUfbw2dp0pGrPidR7lo2jldE8/view?usp=sharing