

---

## Apuntes sobre Maps en JavaScript

**Título:** Maps (Mapas) en JavaScript

**Concepto:** Un Map es una colección de pares clave-valor. A diferencia de los objetos, las claves pueden ser de *cualquier* tipo de dato (objetos, funciones, otros Maps, etc.), no solo strings.

### Declaración e Inicialización:

- Se crea un nuevo Map con el constructor `Map()`.
- Se puede inicializar con un array de arrays, donde cada sub-array es un par [clave, valor].

### Métodos Principales:

- **set(clave, valor):** Agrega o actualiza un par clave-valor. Retorna el propio Map (permite encadenamiento).
  - Ejemplo: `miMap.set('nombre', 'Juan').set('edad', 30);`
- **get(clave):** Devuelve el valor asociado a la clave. Si no existe, retorna `undefined`.
  - Ejemplo: `let nombre = miMap.get('nombre'); // nombre = 'Juan'`
- **has(clave):** Verifica si el mapa contiene la clave. Retorna `true` o `false`.
  - Ejemplo: `if (miMap.has('edad')) { ... }`
- **delete(clave):** Elimina el par clave-valor. Retorna `true` si la clave existía y fue eliminada, `false` en caso contrario.
  - Ejemplo: `miMap.delete('edad');`
- **clear():** Elimina todos los pares clave-valor.
- **size:** Propiedad (no método) que devuelve la cantidad de pares clave-valor.
  - Ejemplo: `let cantidad = miMap.size;`

### Iteración:

- **forEach((valor, clave, mapa)=> { ... }):** Itera sobre cada par. El callback recibe valor, clave y el mapa.
- **for...of:** Itera sobre el mapa, devolviendo un array [clave, valor] en cada iteración.
  - Ejemplo: `for (let [clave, valor] of miMap) { ... }`
- **keys():** Retorna un iterador para las claves.

- 
- **values()**: Retorna un iterador para los valores.
  - **entries()**: Retorna un iterador para las entradas (arrays [**clave**, **valor**]). Es el iterador por defecto para **for...of**.

#### **Conversión desde/hacia Objetos:**

- **Objeto a Map:** **const** miMap = **new** Map(Object.entries(miObjeto));
- **Map a Objeto:** **const** miObjeto = Object.fromEntries(miMap);

#### **Casos de Uso:**

- Claves que no son strings.
- Orden de inserción predecible.
- Cachés.
- Almacenar datos relacionados con objetos.
- Representar grafos.