

Ejemplo 1: Agenda telefónica simple

Descripción:

Usamos un Map para guardar **nombres como claves y números de teléfono como valores.**

Código:

```
// Crear el mapa
const agenda = new Map();

// Agregar contactos
agenda.set("Ana", "7777-1111");
agenda.set("Beto", "7777-2222");
agenda.set("Carla", "7777-3333");

// Obtener un número específico
console.log("Teléfono de Ana:", agenda.get("Ana"));

// Verificar si un contacto existe
console.log("¿Existe Pedro?", agenda.has("Pedro"));

// Mostrar todos los contactos
for (const [nombre, telefono] of agenda) {
  console.log(nombre, "=>", telefono);
}
```

Explicación:

- `.set(clave, valor)` agrega un contacto.
- `.get(clave)` busca el valor de esa clave.

- `.has(clave)` verifica si existe.
- El ciclo `for...of` recorre todos los elementos del Map.

Salida esperada:

Teléfono de Ana: 7777-1111

¿Existe Pedro? false

Ana => 7777-1111

Beto => 7777-2222

Carla => 7777-3333

Ejemplo 2: Registro de calificaciones

Descripción:

Cada estudiante tiene un promedio final.

Usamos un Map para almacenar sus nombres y notas.

Código:

```
const calificaciones = new Map();

// Agregar datos
calificaciones.set("Luis", 8.5);
calificaciones.set("María", 9.2);
calificaciones.set("Pedro", 7.9);

// Mostrar todos los estudiantes
for (const [nombre, nota] of calificaciones) {
  console.log(` ${nombre} tiene una nota de ${nota}`);
}
```

```
// Calcular el promedio general  
  
let suma = 0;  
  
for (const nota of calificaciones.values()) {  
  
    suma += nota;  
  
}  
  
const promedio = suma / calificaciones.size;  
  
console.log("Promedio general:", promedio.toFixed(2));
```

Explicación:

- .values() devuelve solo los valores del Map.
- .size indica cuántos elementos hay.
- Se recorre para sumar las notas y calcular el promedio.

Salida esperada:

Luis tiene una nota de 8.5

María tiene una nota de 9.2

Pedro tiene una nota de 7.9

Promedio general: 8.53

Ejemplo 3: Lista de productos con precios

Descripción:

Guardamos productos (clave) y sus precios (valor).

Luego calculamos el total de una compra.

Código:

```
const productos = new Map([  
    ["Manzana", 0.5],  
    ["Pan", 1.2],  
    ["Leche", 1.8],
```

```
["Huevos", 2.0]
});

// Mostrar productos disponibles
for (const [nombre, precio] of productos) {
    console.log(` ${nombre} cuesta $$ {precio}`);
}

// Calcular total de una compra
const compra = ["Pan", "Leche", "Huevos"];
let total = 0;

for (const item of compra) {
    total += productos.get(item);
}

console.log("Total a pagar: $", total);
```

Explicación:

- Se crea el Map con varios pares [clave, valor] directamente.
- .get(item) obtiene el precio de cada producto.
- Se suman los precios para calcular el total.

Salida esperada:

Manzana cuesta \$0.5

Pan cuesta \$1.2

Leche cuesta \$1.8

Huevos cuesta \$2

Total a pagar: \$ 5

Ejemplo 4: Red de amigos (modelo de grafo con Map)

Descripción:

Representamos quiénes son amigos de quién (como un grafo no dirigido).

Código:

```
const amigos = new Map();
```

```
amigos.set("Ana", ["Beto", "Carla"]);
```

```
amigos.set("Beto", ["Ana", "Diego"]);
```

```
amigos.set("Carla", ["Ana"]);
```

```
amigos.set("Diego", ["Beto"]);
```

```
// Mostrar los amigos de cada persona
```

```
for (const [persona, lista] of amigos) {  
    console.log(` ${persona} tiene como amigos a: ${lista.join(", ")}`);  
}
```

```
// Verificar si Ana es amiga de Carla
```

```
console.log("¿Ana y Carla son amigas?", amigos.get("Ana").includes("Carla"));
```

Explicación:

- Se guarda cada persona como **clave**, y un **array de amigos** como valor.
- `.join(", ")` convierte el arreglo en una cadena separada por comas.
- `.includes(valor)` revisa si un elemento está en el arreglo.

Salida esperada:

Ana tiene como amigos a: Beto, Carla

Beto tiene como amigos a: Ana, Diego

Carla tiene como amigos a: Ana

Diego tiene como amigos a: Beto

¿Ana y Carla son amigas? true

Ejemplo 5: Contador de visitas en una página

Descripción:

Simulamos cuántas veces se visitan diferentes páginas de un sitio web.

Cada página es una **clave**, y el número de visitas es el **valor**.

Código:

```
const visitas = new Map();
```

```
function registrarVisita(pagina) {  
    if (visitas.has(pagina)) {  
        visitas.set(pagina, visitas.get(pagina) + 1);  
    } else {  
        visitas.set(pagina, 1);  
    }  
}
```

```
// Simular visitas  
registrarVisita("Inicio");  
registrarVisita("Inicio");  
registrarVisita("Contacto");  
registrarVisita("Inicio");  
registrarVisita("Tienda");
```

```
registrarVisita("Tienda");

console.log("Visitas totales por página:");
for (const [pagina, cantidad] of visitas) {
    console.log(` ${pagina}: ${cantidad} visitas`);
}
```

Explicación:

- Si la página **ya existe** en el Map, se suma +1 a su contador.
- Si no existe, se crea con valor 1.
- .has() verifica si ya hay registro previo.

Salida esperada:

Visitas totales por página:

Inicio: 3 visitas

Contacto: 1 visitas

Tienda: 2 visitas