## WeakMap, WeakSet

## WeakMap

## WeakMap – это Map, у которого ключи - объекты, а не примитивные значения. Если мы используем объект в качестве ключа и если больше нет ссылок на этот объект, то он будет удалён из памяти (и из объекта WeakMap) автоматически (в отличие от Map).

## *Методы и свойства*:

## new WeakMap([iterable]) – создание WeakMap

## map.set(key, value) – записывает по ключу key значение value.

## map.get(key) – возвращает значение по ключу или undefined, если ключ key отсутствует.

## map.has(key) – возвращает true, если ключ key присутствует в коллекции, иначе false.

## map.delete(key) – удаляет элемент по ключу key.

WeakMap не поддерживает перебор и методы keys(), values(), entries(), так что нет способа взять все ключи или значения из неё.

Применение

Счетчик посетителей. Если посетитель удален, то в WeakMap он также удалится, дополнительная очистка не нужна.

let visitsCountMap = new WeakMap(); // map: пользователь => число визитов

// увеличиваем счётчик

function countUser(user) {

let count = visitsCountMap.get(user) || 0;

visitsCountMap.set(user, count + 1);

}

Другое применение - для кеширования.

**WeakSet**

Объект WeakSet аналогичен Set, но мы можем добавлять в WeakSet только объекты (не примитивные значения).

Как и WeakMap не поддерживает перебор и методы keys(), values(), entries

Его основные методы это:

* new Set(iterable) – создание.
* set.add(value) – добавляет значение (если оно уже есть, то ничего не делает), возвращает тот же объект
* set.delete(value) – удаляет значение, возвращает true, если value было в множестве на момент вызова, иначе false.
* set.has(value) – возвращает true, если значение присутствует в множестве, иначе false.

let visitedSet = new WeakSet();

let john = { name: "John" };

let pete = { name: "Pete" };

let mary = { name: "Mary" };

visitedSet.add(john); // John заходил к нам

visitedSet.add(pete); // потом Pete

visitedSet.add(john); // John снова

// visitedSet сейчас содержит двух пользователей

// проверим, заходил ли John?

alert(visitedSet.has(john)); // true

// проверим, заходила ли Mary?

alert(visitedSet.has(mary)); // false

john = null;

// структура данных visitedSet будет очищена автоматически