LESSON24

GARBAGE COLLECTION
SET/MAP

GARBAGE COLLECTION

- V8
- Background process
- Working with the heap
- Reachability
- Leads to memory leak
- Unpredictable

HOW GARBAGE COLLECTION WORKS

- mark-and-sweep algorithm

- Root objects
- Child objects by links
- … Child objects by links until finish
- Mark
- Delete others

REACHABILITY (ДОСТИЖИМОСТЬ)

- Base reachable
  - Локальные переменные и параметры текущей функции.
  - Переменные и параметры других функций в текущей цепочке вложенных вызовов.
  - Глобальные переменные.
  - (некоторые другие внутренние значения)

MEMORY LEAK

- Global variables
- Timers and callbacks (removeEventListener)
- Links to DOM elements
- Closures

EXAMPLE

```
var theThing = null;

var replaceThing = function () {
  var originalThing = theThing;

  var unused = function () {
    if (originalThing) console.log("hi");
  };

  theThing = {
    longStr: new Array(1000000).join("*"),
    someMethod: function () {
      console.log(someMessage);
    },
  };
};

setInterval(replaceThing, 1000);
```

SET & MAP

- Set - collection of unique values;
- Map - key/value collection, but “key” can be any type;

SET

- Collection of unique values
- new Set(iterable);
- Sets are array-like and iterable;
- .forEach()

SET METHODS

- `set.add(value)` - добавляет значение (если оно уже есть, то ничего не делает), возвращает тот же объект `set`.
- `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`.
- `set.clear()` - удаляет все имеющиеся значения.
- `set.size` - возвращает количество элементов в множестве.

MAP

- Key/value collection
- new Map(entries);
- Алгоритм сравнения ключей SameValueZero (NaN === NaN)
- .forEach()

MAP METHODS

- `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`.
- `map.clear()` - очищает коллекцию от всех элементов.
- `map.size` - возвращает текущее количество элементов.

WEAK MAP

- new WeakMap()
- Data can be deleted by garbage collector
- Only objects as a key
- Doesn’t support keys(), values(), entries()

WEAK MAP METHODS

- weakMap.get(key)
- weakMap.set(key, value)
- weakMap.delete(key)
- weakMap.has(key)

EXAMPLE

```
let john = { name: "John" };

let weakMap = new WeakMap();
weakMap.set(john, "...");

john = null;
```

WEAK SET

- new WeakSet()
- Data can be deleted by garbage collector
- Only objects as a values
- Doesn’t support keys(), size

WEAK SET METHODS

- weakSet.add(key, value)
- weakSet.delete(key)
- weakSet.has(key)

GENERATORS

- *
- yield
- .next()