**Sets**

* Se creaza asa:

const *set* = new *Set*();

sau asa

const *set* = new *Set*([1,2,3,4,5]);

* Valorile duplicate pur si simplu nu sunt puse in set si gata
* Constructori:

Set()

Set([])

Set(string) – fiecare litera e un element

* Sets sunt iterabile, deci au metodele values(), keys(), entries()
* **Ordinea elementelor e pastrata**
* Sunt iterable, deci putem folosi for of
* Operatorul ... tot merge pe sets
* const *set* = new *Set*([1,2,3,4,5])  
    
  const *array* = [...*set*]
* Totusi, aici apare si o problema cu faptul ca putem pune orice obiect ca key:
* const *map* = new *Map*();  
    
  *map*.set([1,2], 'value');  
  *console*.log(*map*.get([1,2]))

undefined

Nu va compara valorile din arrays, ci doar referintele

* **Fields**:

- size

* **Methods:**

- has(element)

- add(element)

- delete(element)

- clear() – sterge toate elementele

- entries()

- keys()

- values()

**keys() si values() returneaza ambele doar valorile propriu zise!!!**

- forEach()

- subSet(set) – returneaza true daca setul curent contine tot ce are si set

- union(set) – returneaza un nou set

- intersection(set)

- difference(set)

**Maps**

* Maps pastreaa key + value, insa key pot fi de orice tip, chiar si alt map
* const *map* = new *Map*();
* **Constructori:**

Map()

Map([[key, value], [key, value],...])

Map(entries)

E usor sa cream un map din entries:

const *obj* = {  
 'monday' : 10,  
 'tuesday' : 20,  
 'friday' : 30  
}  
  
const *map* = new *Map*(*Object*.entries(*obj*));

* Sunt interable
* const *map* = new *Map*(*Object*.entries(*obj*));  
    
  for(const [key, value] of *map*)  
   *console*.log(key, value);

Si nu mai trebuie Object.entries() in for()

* Un map poate fi convertit usor in array cu ...

const *map* = new *Map*(*Object*.entries(*obj*));  
  
const *array* = [...*map*];

* Map mereu memoreaza ordinea de inserare la iteme
* **Fields:**

- size

* Metode:

- **set(key, value)** – returneaza si mapul la urma

*map*.set("key1",'value1');  
*map*.set(1,2);

Datorita faptului ca returneaza map, putem face un chain:

*map*.set("key1",'value1').set(1,2);

- **get(key)**

- has(key)

- delete(key)

- clear()

- entries() – returneaza un array ce contine un array pentru fiecare key value

*map*.set(1,10).set(2,20).set(3,30);  
*console*.log(*map*.entries());

[Map Entries] { [ 1, 10 ], [ 2, 20 ], [ 3, 30 ] }

- keys()

- values()