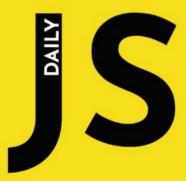


Day 27, Today's Topic

Weak map



/* ------ */
/* ----- Daily JS - Day 27 ----- */
/* ----- */

Day 27: WeakMaps

Welcome back to the 27th day of the DailyJS series. Yesterday we discussed about the concept of Map in JS, and today's topic is `WeakMap`.

> The WeakMap object is a collection of key/value pairs in which the keys are weakly referenced.

The WeakMap object is a collection of key/value pairs in which the keys are weakly referenced. The keys must be objects and the values can be arbitrary values.

I know, you might be confused by the above lines.

Explaining it in layman terms, Weak map is very similar to map, it is also a data structure which allows us to hold key-value pairs, with the difference being that the keys of a WeakMap are of type `Object` only, unlike `Map`, using primitive data types as keys is not allowed.

```
/* ========= madhavbahl.tech/dailyjs/day27 ========= */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======== */
```

```
. .
/* ------ */
/* ============ Daily JS - Day 27 =========== */
/* ----- */
 ## Syntax
 Very similar to Map, a WeakMap can be created using the
 constructor, and it also comes with several methods like
 set and get.
 ···js
    let weakMap = new WeakMap ();
 Some important methods -

    WeakMap.prototype.set(key, value)

   WeakMap.prototype.get(key)
   WeakMap.prototype.delete(key)
   4. WeakMap.prototype.has(key)
 But why is it used?
/* ======== madhavbahl.tech/dailyjs/day27 ========= */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */
```

/* ============== Daily JS - Day 27 ============== */ ## Use cases of WeakMap Source: https://stackoverflow.com/questions/29413222/ what-are-the-actual-uses-of-es6-weakmap - Keeping private data about a specific object and only giving access to it to people with a reference to the Map. A more ad-hoc approach is coming with the private-symbols proposal but that's a long time from now. - Keeping data about library objects without changing them or incurring overhead. - Keeping data about a small set of objects where many objects of the type exist to not incur problems with hidden classes JS engines use for objects of the same type. - Keeping data about host objects like DOM nodes in the browser. - Adding a capability to an object from the outside (like the event emitter example in the other answer). /* ======== madhavbahl.tech/dailyjs/day27 ========= */ /* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */

```
/* ================= */
/* =========== Daily JS - Day 27 =========== */
/* ------*/
   /**
   * WeakMap in JS
   */
  var weakMap = new WeakMap ();
  // weakMap.set ("a", 1); - ERROR!
  // Set
  var myObj = {};
  weakMap.set (myObj, 'My Object');
  console.log (weakMap);
  // Get
  var val = weakMap.get (myObj);
  console.log ("Value is: ", val);
  // Has
  console.log ("Has myObj?", weakMap.has (myObj));
  // Delete
  weakMap.delete (myObj);
/* ======== madhavbahl.tech/dailyjs/day27 ========= */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */
```

.



Thank you!

Feel free to reach out...

Email: theleanprogrammer@gmail.com

Web: madhavbahl.tech/

Github: github.com/MadhavBahlMD

LinkedIn: linkedin.com/in/madhavbahl/

Insta: instagram.com/theleanprogrammer/

