**First of all**

**What is an iterator object or iterator**

In JS an object is iterable when it implements a **next()** method

In JS not every objects are iterable

Meaning it must have a next( ) method in it, which again returns us the object that must have two properties

value : it returns the current value for each iteration

done : it is Boolean i.e either true or false, it returns true when we are done traversing

We often say iterator object or also simply iterator

**What is [Symbol.iterator](), nothing but a method having fixed name and fixed return type introduced by ECMA to make custom objects iterable**

Users wanted to traverse through their custom made objects

So they could use features like for…of() loop and … spread operator

So ECMA said

Okey fine, Users could make their custom objects iterable, by inserting a property in their custom object having key of [Symbol.iterator] and value a function

Meaning users have to introduce a method that have a fixed name of [Symbol.iterator]() and have a fixed return type i.e an iterator object

And that iterator object can have any properties but that must have a next() method, which again returns another object

Having two properties

value : and

done :

**How for…of ( ) loop and … spread operator works ?**

First yenie haru jun object maa operated vaa xan tyo object bhitra [Symbol.iterator]() method lai khojxan raw call garxan

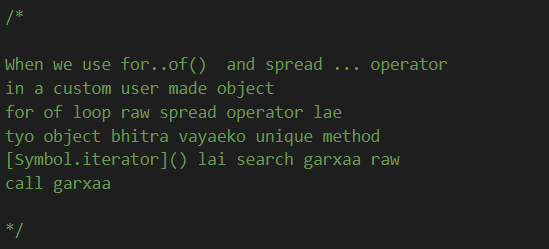
[Symbol.iterator]() method lae iterator object dinxaa

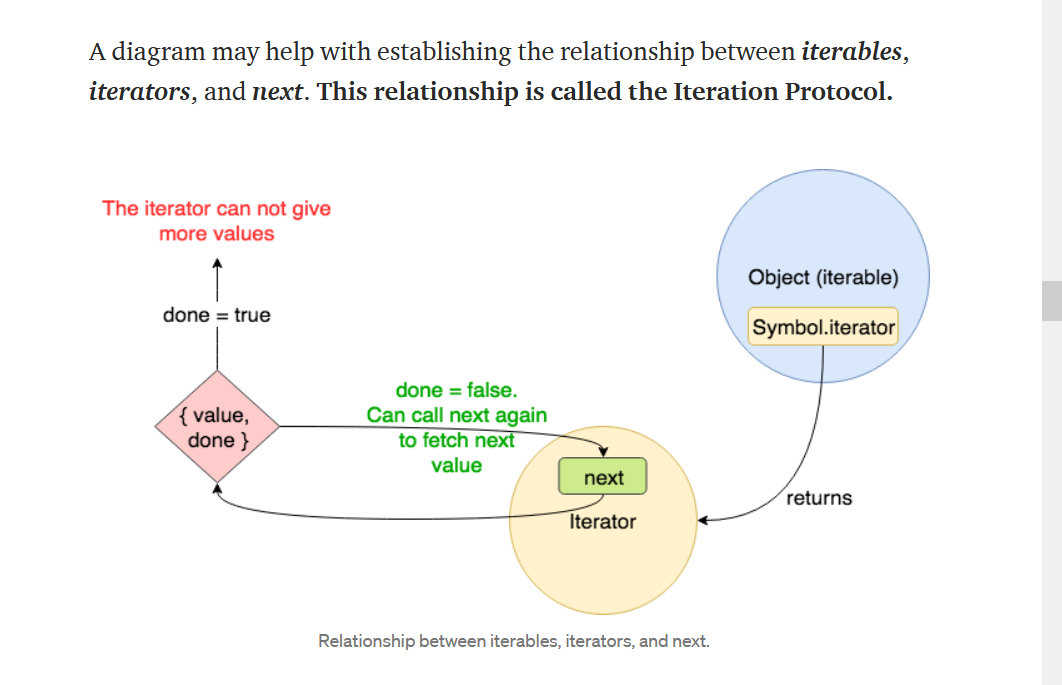
And they again call the next() method present inside of that iterator object

Next() method returns an object having two properties

value : it contains current value of each iteration

done : it contains true or false, which says whether iteration finished or not





see this <https://codeburst.io/a-simple-guide-to-es6-iterators-in-javascript-with-examples-189d052c3d8e>

<https://javascript.info/iterable>

See an example



