

# JavaScript Generators and Iterators

---

WHAT IS AN ITERATOR?



**Marques Woodson**

STAFF SOFTWARE ENGINEER

@mwq27 [www.marqueswoodson.com](http://www.marqueswoodson.com)



# New ES6 Data Structures

**Maps**

**Sets**

**WeakMaps**

**WeakSets**





# Loops – For and ForEach

Iterate over each element in an array



# Standard Loops

```
for ( let i = 0; i < 10; i++ ) {  
  console.log(i * 2);  
}
```

---

```
[1,3,5,7,9].forEach(value => {  
  console.log(value * 2);  
});
```



# Standard Loops

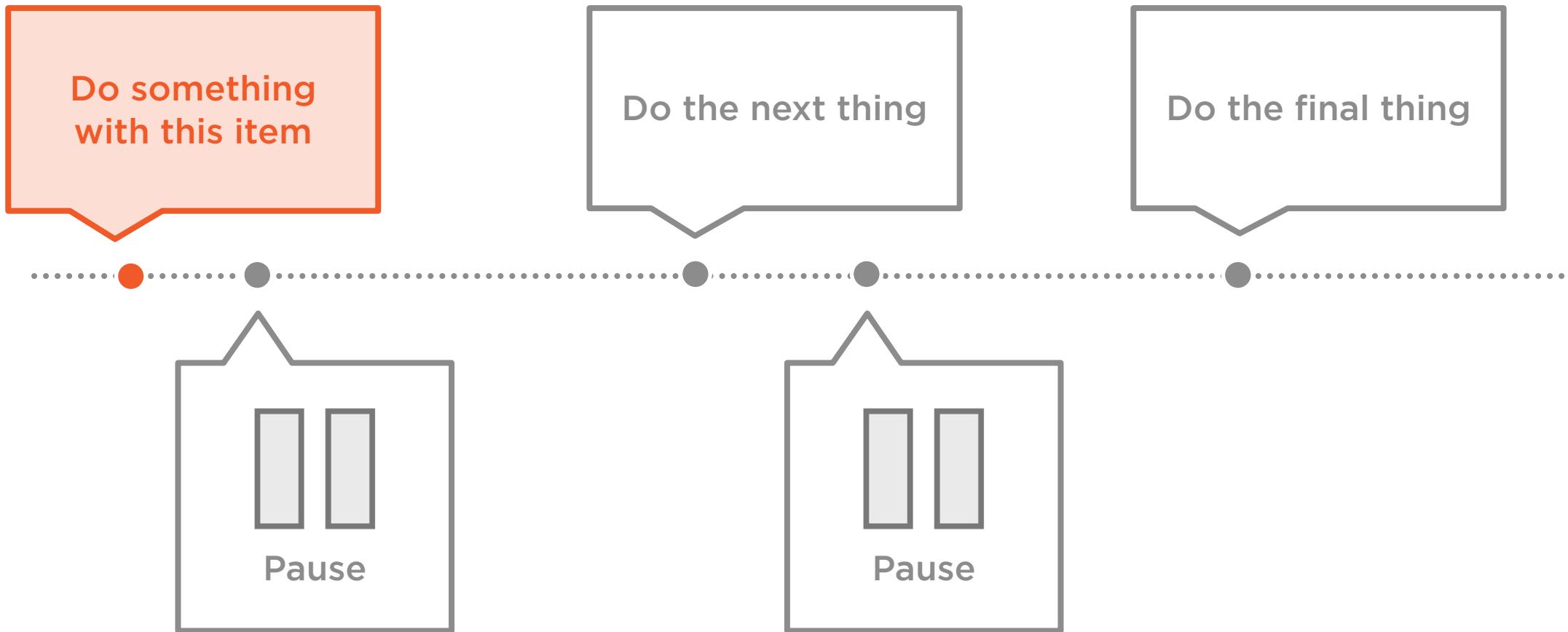
```
for ( let i = 0; i < 10; i++ ) {  
  if (i > 2) break;  
  console.log(i * 2);  
}
```

---

```
[ 1, 3, 5, 7, 9 ].forEach(value => {  
  if (i > 2) break;  
  console.log(value * 2);  
});
```



# Pause While Iterating

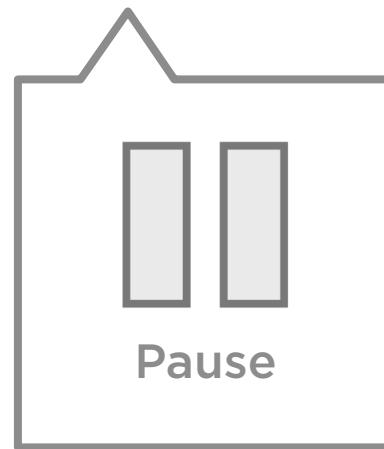
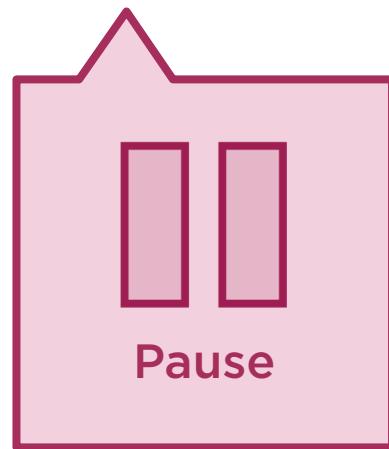


# Pause While Iterating

Do something  
with this item

Do the next thing

Do the final thing

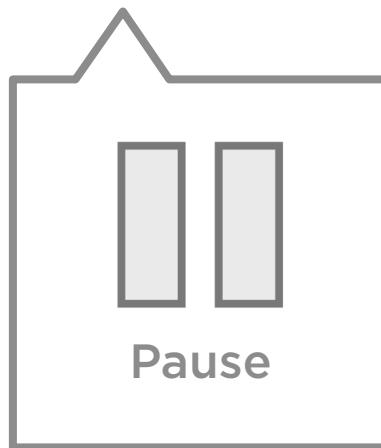
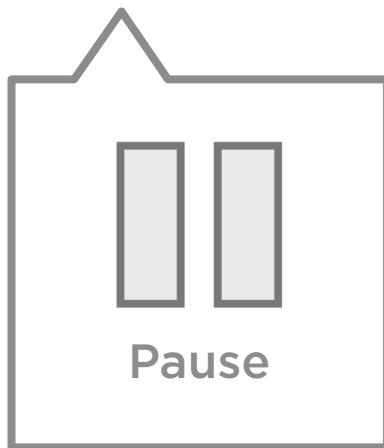


# Pause While Iterating

Do something  
with this item

Do the next thing

Do the final thing

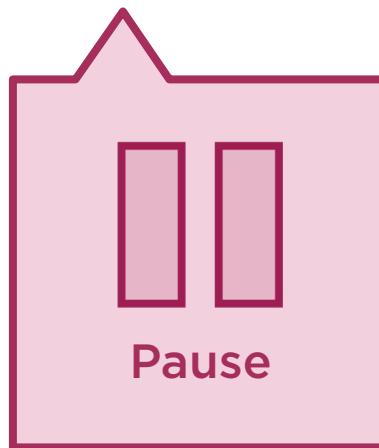
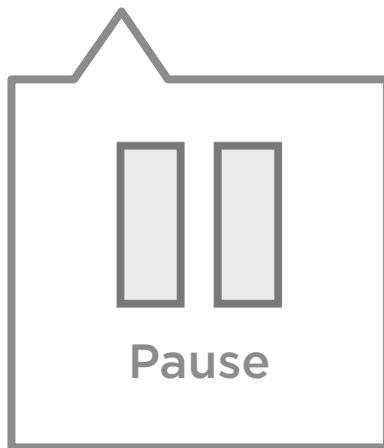


# Pause While Iterating

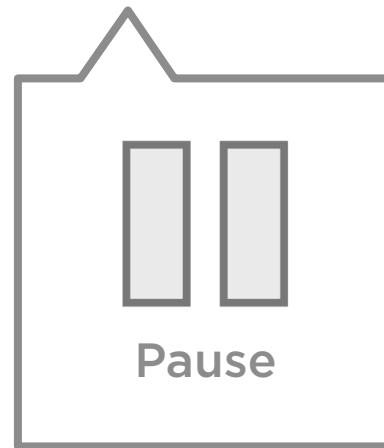
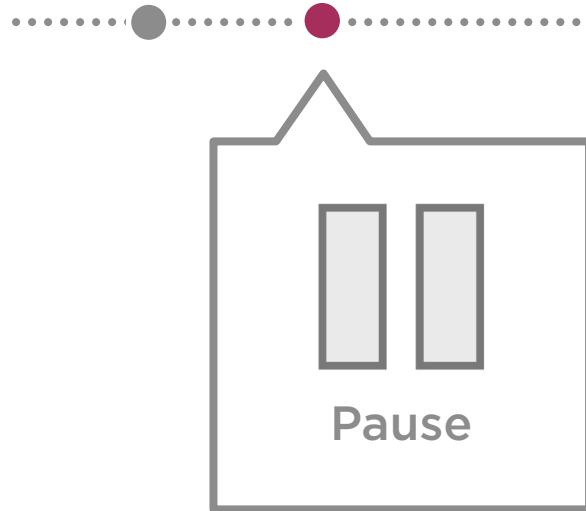
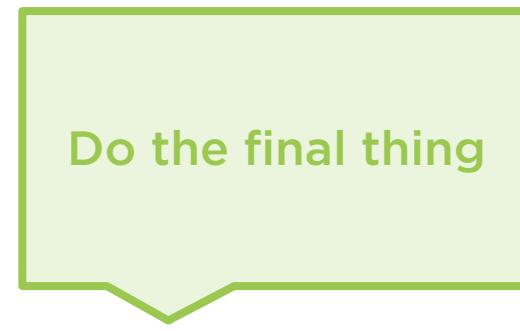
Do something  
with this item

Do the next thing

Do the final thing



# Pause While Iterating



An iterator lets you iterate through a collection's contents one at a time, pausing at each item.



# Module Overview



**Differences with loops**

**What is an iterable?**

**Utilizing built-in iterables**

**Custom iterators**



# Iterator

An Iterator is any object that implements the Iterator protocol by having a **next()** method that returns a **value** property and a **done** property.



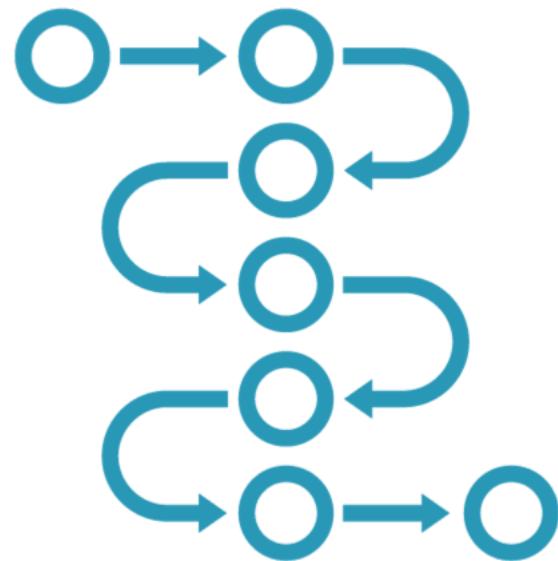


# Console Log-My-Food

A CLI application that puts  
your health first



# What exactly Is an iterable?



An array is a built-in iterable

There are other built-in iterables  
(strings, maps, and sets)

Iterables implement the  
`@@iterator` method

`Symbol.iterator` a Well-Known  
Symbol in Javascript



# for..of Loops



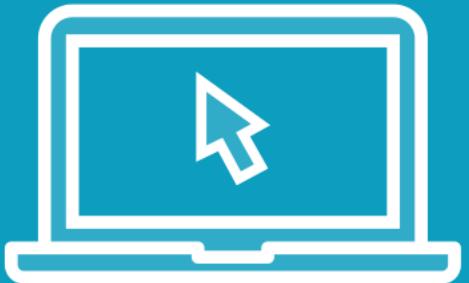
This type of loop **ONLY** works with iterable objects

**Don't confuse this with a for..in loop**

**for..of loop iterates over VALUES, for..in iterates over enumerable properties**



Demo



**Run this app from the command line**

**Prompt the user asking what they ate, and how much**

**Show the user how many calories are in that food**

**Setup database**



# Recap of an iterable

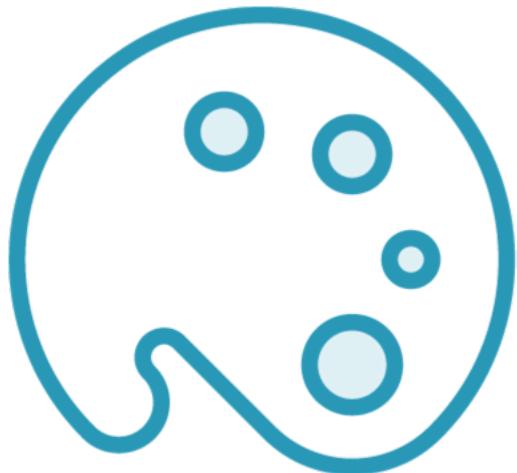
**It can be enumerated with a for..of loop**

**It adheres to the Iterator Protocol**

**It returns an object with an object that has a value and done property**



# Creating Custom Iterators



**So far we've only seen the  
Symbol.iterator execution**

**Let's define the Symbol.iterator  
function and the 'next' method**

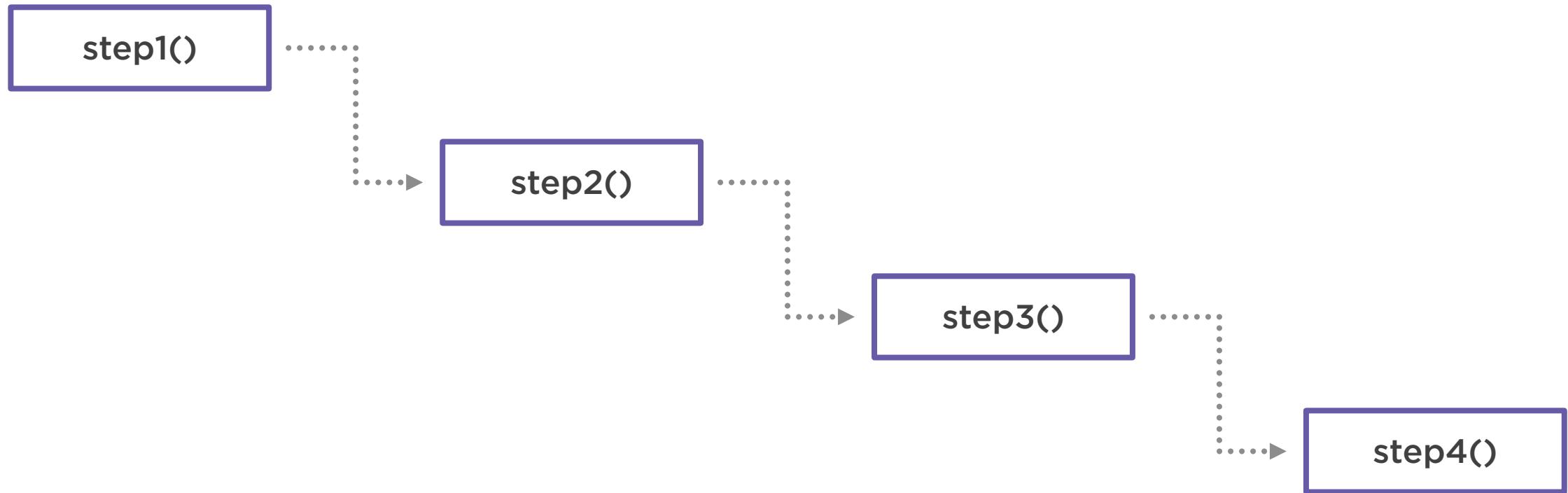
**What are we going to use our custom  
iterator for???**

**We have a new feature request!**

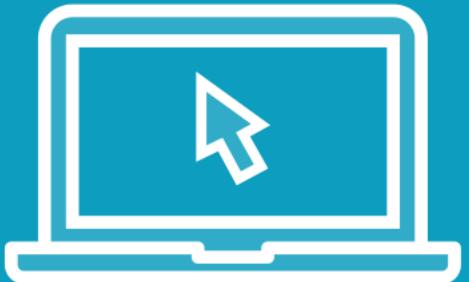
**Implement readline.prompt and line  
event listener**



# Sequence of Function Calls



# Demo



**Use follow-up questions**

**Specify how many servings were eaten**

**Create iterator to execute functions**

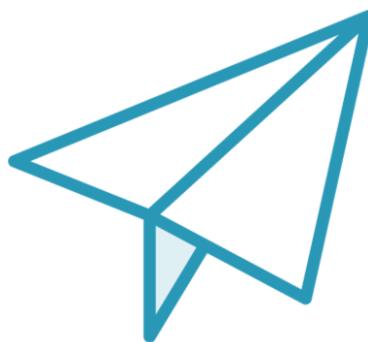


# Iterator Interface



**property: next**

Function  
Required



**property: throw**

method  
Optional



**property: return**

method  
Optional



# Iterator Result Interface



## **property: done**

Boolean

Required



## **property: value**

Anything

If value is undefined, may be absent

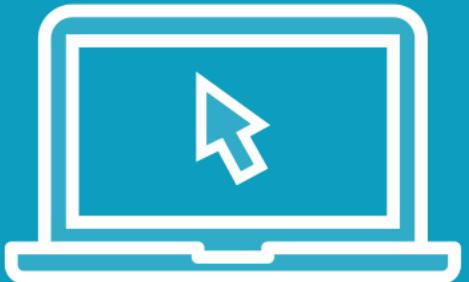


# Return is Not Available on Arrays

```
const arr = [0, 1, 2];
const it = arr[Symbol.iterator]();
it.return(); // return is not a function!
```



# Demo



**Implement the return method**

**Implement the throw method**

**Save log to database**



# Summary



**Defined iterables and iterators**

**Built in iterables**

**Creating custom iterators**

**Custom iteration of functions**

**Implemented optional methods on  
custom iterator**

