Consuming Iterables

iterating an iterable object and an introduction to data consumers

Both the `for-of` loop and the spread operator can be used to perform iteration on an iterable object.

```
for ( let element of iterableObject ) {
    console.log( element );
}

console.log( [...iterableObject] );
```

Using the countdown example, we can print out the result of the countdown in an array:

```
let countdownIterator = {
  countdown: 10,
  next() {
    this.countdown -= 1;
    return {
      done: this.countdown === 0,
      value: this.countdown
    };
  }
};

let countdownIterable = {
  [Symbol.iterator]() {
    return Object.assign( {}, countdownIterator )
  }
};

console.log([...countdownIterable]);
//> [9, 8, 7, 6, 5, 4, 3, 2, 1]
```

Language constructs that consume iterable data are called **data consumers**.

We will learn about other data consumers soon. But first, let’s study Iterables in a bit more detail.