**An Introduction to Iterators and Generators in JavaScript**

## Iterators and generators are two different concepts, but they're used in similar ways. They're used to iterate through arrays and objects in JavaScript. Iterators

Iterators are like advanced loops that can be paused. Iterators consist of the **next()**function, which returns the value and the done status. The **value**field is the value in the array at a given index. **done**is the boolean value that returns the status of completion for the iteration through the loop.

Here's an example that demonstrates how iterators:

function fruitIter(fruits){  
 let index = 0;  
 return {  
 next: function(){  
 return index &lt; fruits.length ? { value: fruits[index++], done: false } : {done: true}  
 }  
 }  
}  
const fruitsArray = ["Mango", "Banana", "Grapes"];  
const fruits = fruitIter(fruitsArray);  
console.log(fruits.next().value);

**Output**:

Mango

When you pass the **fruitsArray** array in the **fruitIter()** method, it returns an iterator that's stored in the **fruits** variable. The **index** variable in the **fruitIter()** method gets initialized to 0. This method returns the **next()** function that helps in [**looping through the array**](https://www.makeuseof.com/for-while-do-while-java/). The **next()** function checks whether the index is less than **fruitsArray** length. If so, it returns two variables: the fruit name at that index, and the**done** status. While returning these values, it also increments the index value.

To check how this method works and print the fruit name, you need to call the **next()** function on the **fruits** iterator and get access to its value.

Generators are similar to Iterators but they return multiple values. These values are called yield values. Generator functions are written using the **function\* syntax**. **\*** denotes that it's not a normal function, but a generator. Here's an example of generators:

function\* printFruits(){  
 yield "Mango";  
 yield "Banana";  
 yield "Grapes";  
}  
const fruit = printFruits();  
console.log(fruit.next());

**Output**:

{value: 'Mango', done: false}

In this example, **yield** is the iterator. When you call the function **printFruits()**, and print **fruit.next()**, it gives you an object where you get the value. The **done**status denotes whether all values have been iterated through.

Refernces

https://www.makeuseof.com/javascript-iterators-generators/