# Section 11 - For...of loop

Starting from ES6, we have a new way to loop over arrays, and that’s using the **for-of loop**.

It is very similar to the for-in loop, but instead of the in keyword, we are using the **of** keyword.

We are going to use the example of the colors array to illustrate how easy the for-of loop is with arrays.

## Syntax

for(let item of array){

//code to run

}

\* Note the **of** keyword, instead of the **in** keyword

Example

let colors = [‘red’, ‘green’, ‘blue’];

for(let color of colors){

console.log(color); //will log ‘red’, ‘green’ and ‘blue’

}

So here's an example, for(let color of colors), you can simply log this color on the console.

console.log(color);

You can see with this new for loop, **we don't have to deal with the index**.

So in each iteration this color, which is our **loop variable,** will hold one of the elements in this array.

In the console, you will see red green and blue.

**Warning**

for...in is intended for iterating over object keys, not array indexes. Using it to loop through an array is generally discouraged. It also includes properties from the prototype, so it may be necessary to check if the key is within the object using has Own Property. If any attributes in the object are defined by the define Property/define Properties method and set the param enumerable:false, those attributes will be inaccessible.

"for ... of" loop

The advantages from the for...of loop are:

This is the most concise, direct syntax yet for looping through array elements

It avoids all the pitfalls of for...in

Unlike forEach(), it works with break, continue, and return

Support of for...of in other collections

Strings for...of will treat a string as a sequence of Unicode characters

const string ="abc";

for(let chr of string)

{ console.log(chr);}

Expected output: