Iterables and Looping

Let's have a look at the new type of loops the ES6 introduced.

WE'LL COVER THE FOLLOWING

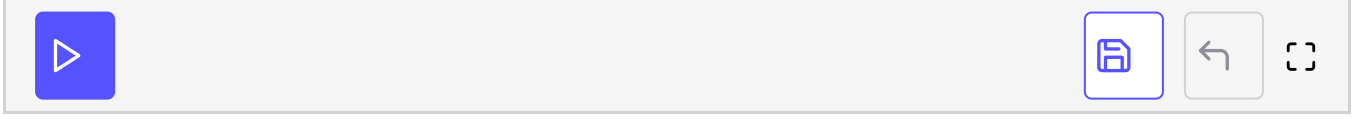
The `for of` loop

ES6 introduced a new type of loop, the `for of` loop. Let's take a look at how it is used.

Iterating over an array

Usually, we would iterate using the following method:

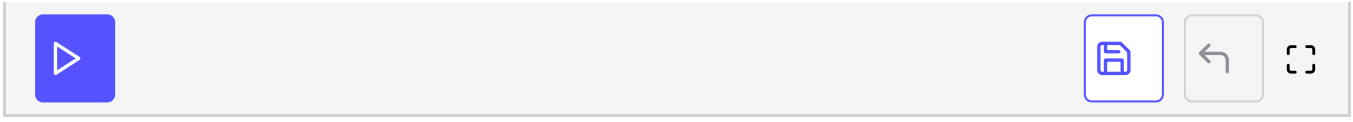
```
var fruits = ['apple','banana','orange'];
for (var i = 0; i < fruits.length; i++){
  console.log(fruits[i]);
}
// apple
// banana
// orange
```



This is a normal loop where at each iteration we increase the value of `i` by 1 as long as it is less than `fruits.length`. At that point the loop will stop.

Look at how we can achieve the same with a `for of` loop:

```
const fruits = ['apple','banana','orange'];
for(const fruit of fruits){
  console.log(fruit);
}
// apple
// banana
// orange
```

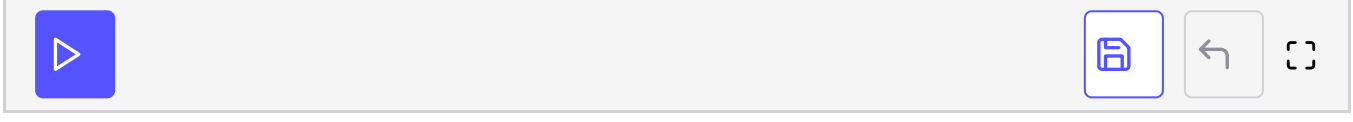


Iterating over an object

Objects are **non iterable**, so how do we iterate over them? We have to first grab all the values of the object using something like `Object.keys()` or the new ES6 function: `Object.entries()`.

```
const car = {
  maker: "BMW",
  color: "red",
  year : "2010",
}

for (const prop of Object.keys(car)){
  const value = car[prop];
  console.log(prop,value);
}
// maker BMW
// color red
// year 2010
```



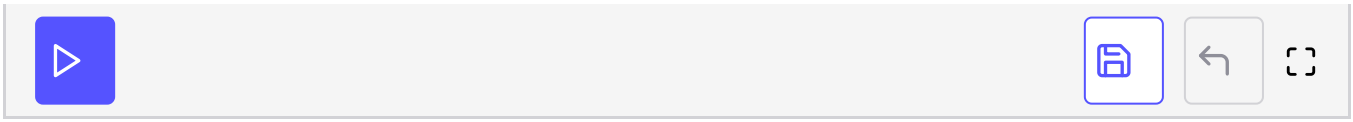
The `for in` loop

Even though it's not a new ES6 loop, let's look at the `for in` loop to understand what differentiates it to the `for of` loop.

The `for in` loop is a bit different because it will iterate over all the enumerable properties of an object in no particular order.

It is therefore suggested not to add, modify or delete properties of the object during the iteration. There's no guarantee that they will be visited, or if they will be visited before or after being modified.

```
const car = {
  maker: "BMW",
  color: "red",
  year : "2010",
}
for (const prop in car){
  console.log(prop, car[prop]);
}
// maker BMW
// color red
// year 2010
```



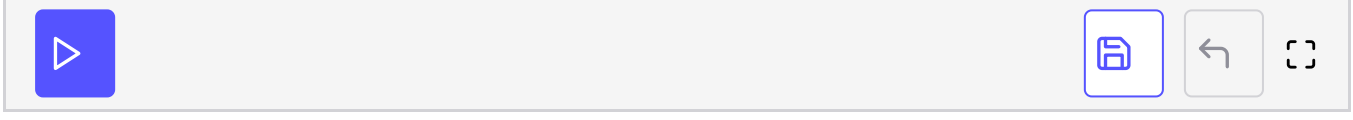
Difference between `for of` and `for in`

The first difference we can see is by looking at this example:

```
let list = [4, 5, 6];

// for...in returns a list of keys
for (let i in list) {
   console.log(i); // "0", "1", "2",
}


// for ...of returns the values
for (let i of list) {
   console.log(i); // "4", "5", "6"
}
```



`for in` will return a list of keys whereas the `for of` will return a list of values of the numeric properties of the object being iterated.

In the next lesson, we will solve a quiz and a coding challenge to test the concepts covered in this lesson