# Iterating an Object with Standard For/While

This lesson delves further into looping over an object's property names and values with a traditional for or while loop.

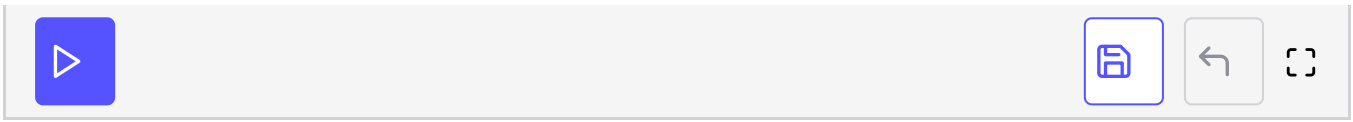
WE'LL COVER THE FOLLOWING


- For Loop
- While Loop


## For Loop #

You can also loop with the standard `for` loop. Using an index works as well for an array, but again, but it won't let you loop an object without using `object.keys()` or `object.entries()` which both returns array. Note that the `entries` function requires using a lib called `es2017.object` which must be set in the TypeScript compiler.

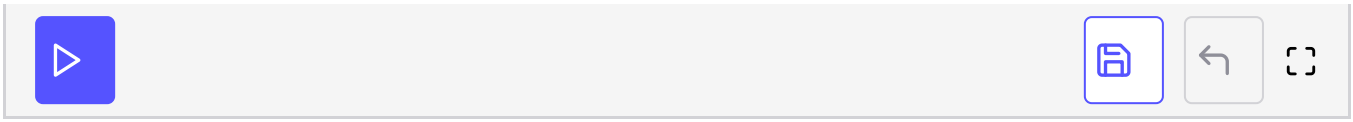
```
let listArrayPrimitive = { m1: "valuem1", m2: 2 };
const keys = Object.keys(listArrayPrimitive);
const entries = Object.entries(listArrayPrimitive); // require to have "lib": [ "es2017.objec
console.log("keys", keys);
console.log("entries", entries);
```



The list of keys will be of type `string[]` but the list of `entries` will be strongly typed as the type found in the object. In the example, it will be `(number | string)[]` because the object `listArrayPrimitive` contains one property with a string value and one with a number.

```
let listArrayPrimitive = { m1: "valuem1", m2: 2 };
const keys = Object.keys(listArrayPrimitive);
for (let index = 0; index < keys.length; index++) {
    const key = keys[index];
    const value = (listArrayPrimitive as any)[key];
```

```
    console.log(`The property name ${key} has the value ${value}`);
}
```

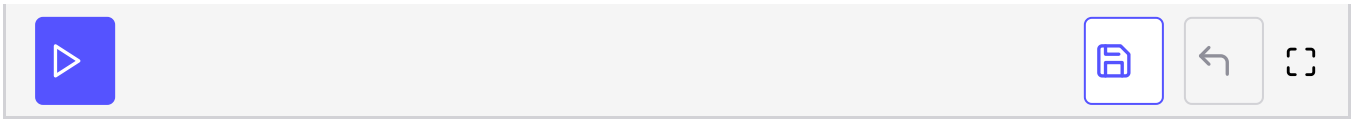


The loop was performed using a `for` that goes from `0` to the length of the array. You can extract the value with an index signature. `any` is forced because the type does not have a signature allowing access by name.

## While Loop #

Similarly, you can use a `while` loop. This time, the example is

```
let listArrayPrimitive = { m1: "valuem1", m2: 2 };
const entries: any[] = Object.entries(listArrayPrimitive); // require to have "lib": [ "es201
while (entries.length > 0) {
    const val = entries.shift();
    console.log(`The property name ${val[0]} has the value ${val[1]}`);
}
```



A `for` and a `while` loops are interchangeable. TypeScript does not have a lot of difference else than defining the type when looping with the `for` which can be inferred while initializing to `0`.