

Context

- What is an object?
 - An object is one of the built in data types in JavaScript.
 - An object is a collection of properties and methods.
 - A property is a name:value pair, the value of which can be accessed by referencing the name the value is associated with, or by using a getter method.
 - The value of a property has it's own data type, and can be another object (a nested object), a string, integer, etc...
 - The properties of an object can be changed directly by referencing the property, or by using a setter method.
 - The contents of an object are mutable, even if the object is declared using const. What is an array?
- What is an array?
 - In JavaScript, an array is one of the built in object data types, and thus it has methods associated with it.
 - An array is a collection of items, with each item stored within the array having an associated index that
 - The items within an array can be any other data type, including other arrays or objects.
 - An array that contains another array can be called a 2-dimensional array.
- Discuss the difference between an array and an object.
 - Both arrays and objects store collections of data, are mutable even when declared as constants, and can store a mix of data types.
 - The data stored within an array is referenced using a non-negative integer value, whereas within an object data is referenced using the name of the property the data is assigned to.
 - Both objects and arrays have built in methods, however arrays cannot have new methods added to them during and after creation whereas objects can.
 - While an array object can only be created by declaring an array, objects can either be created by declaring them or by the use of a class.

Use of Arrays and Objects

How can arrays be used in loops

- A for loop:
 - An array can be used within a for loop:
 - To loop over each element within the array.
 - To modify the array a set number of times.

```
1 // ----- Array in a for loop ----- \\
2 let x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0];
3 let n = 0;
4
5 for (let i in x) {
6   console.log(n, x[n]);
7   n++;
8 }
```

- A while loop:
 - An array can be used within a while loop:
 - To loop over each element of the array.
 - To search through the array for an item that meets a condition, or while a condition is met.

```
11 // ----- Array in a while loop ----- \\
12 let y = [1, "xyz", 3, 4, 5, "xyz", 7, 8, 9, 0];
13 let index = 0;
14
15 while (typeof y[index] !== "string" && index < y.length) {
16   j = y[index];
17   index++;
18 }
19 console.log("String detected at index: ", index);
```

How can objects be used in loops

- A for loop:
 - An object can be used within a for loop:
 - To loop over each property and method of that object.
 - To modify the object a set number of times.

```
22 // ----- Object in a for loop ----- \\
23 const a = {
24   d: 1,
25   e: 2,
26   f: 3,
27   g: 4,
28   h: 5,
29 }
30
31 for (let property in a) {
32   console.log(property, a[property]);
33 }
```

- A while loop
 - An object can be used in a while loop:
 - To loop over each property or method of that object.
 - To search through the object until a property meets a condition, or while a condition is met.
 - To modify an object until a condition is met.
 - To call an objects method until a condition is met.

```
35 // ----- Object in a while loop ----- \\
36 const b = {
37   one: "q",
38   two: 2,
39   three: undefined,
40   four: 4,
41   five: "t",
42 }
43
44 keys = Object.keys(b);
45
46 let count = 0;
47 let k = keys[0];
48 while (count < keys.length && typeof b[k] !== "undefined") {
49   k = keys[count];
50   count++;
51 }
52
53 console.log("The value of the key:", k, "is undefined");
```

Appendix

- The primary challenge I faced in this task was going back to basics. In having to consider how to use an array or object in a way you would not normally use them, you are forced to think outside of the box to find a solution to a seemingly simple problem.
- I also found it useful going back over the definitions of objects and arrays, and more specifically them within the confines of JavaScript to better understand their implementation within the language.

- General:
 - <https://developer.mozilla.org/en-US/docs/Glossary/Primitive>
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Inheritance_and_the_prototype_chain
 - <https://www.edureka.co/blog/data-types-in-javascript/#:~:text=An%20array%20in%20JavaScript%20is,the%20index%20starts%20from%200.>
 - https://medium.com/@zac_heisey/objects-vs-arrays-42601ff79421#:~:text=Objects%20represent%20a%20special%20data,store%20a%20list%20of%20values.
 - <https://chat.openai.com/>
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures
- Arrays:
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array
 - <https://stackoverflow.com/questions/61517204/what-do-we-mean-by-defining-a-const-array-in-js>
- Objects:
 - https://www.w3schools.com/js/js_objects.asp
 - <https://www.techtarget.com/searchapparchitecture/definition/object>
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object
 - <https://stackoverflow.com/questions/1568091/why-use-getters-and-setters-accessors>
 - https://www.w3schools.com/js/js_classes.asp
 - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes>
 - <https://www.codecademy.com/courses/introduction-to-javascript/lessons/objects/exercises/nested-objects>