Challenge Question: Arrays and Objects

Manipulation of objects and arrays can be quite complex and challenging at times, especially when they become nested within one another.

Earlier in your bootcamp, you learned about dealing with arrays of objects. For this challenge question, you'll be asked to research more information about how arrays and objects work together and share your findings.

Please research the following questions and provide your answer below.

**Question #1**: What is at least one way to add an object to an existing array? What is at least one way to add an array to an existing object?

The push() method will both add an object to and existing array as well as add an array to an existing object.

**Example code 1:**

**let arrayFood = [‘bread’, ‘cheese’, ‘milk’];**

**let fridge = {**

**brand: 'GE',**

**purchased: 2021,**

**icemaker: 'yes'**

**};**

**arrayFood.push(fridge);**

**Example code 2:**

**let vet = {**

**arrayPets : [‘cat’, ‘dog’, ‘bird’];**

**arrayVisitTypes : [‘annual’, ‘urgent’, ‘routine’]**

**};**

**let arrayAdditionalVisitTypes = [ ‘vaccince’, ‘blooddraw’, ‘urinesample’];**

**Vet[‘arrayVisitTypes’].push(arrayAdditionalVisitTypes);**

**Question #2:** How can you find the length of an array? How can you find the number of properties in an object?

**Answer part 1:**

let x = array.length

console.log(x)

**Answer part 2:**

const books = {

name: "Nana Upstairs Nana Downstairs",

published: 1973,

Author: "Tomie dePaola",

Category: "Children’s book",

}

let count = Object.keys(books).length

console.log(count)

**Question #3:** Let's say you have a MongoDB collection of documents, and you've used a Mongoose/MongoDB Node Driver method such as *collection*.find() that returns the documents as objects in an array. How would you go about checking to see if a specific object already exists in that array?

db.users.find({recipe: {$elemMatch: {recipe: ‘chicken pot pie’ , dairy: ‘real cream’}}})