JavaScript – Arrays

1. Create an array called `practiceFile` with the following entry: 273.15. Use the `push` method to add the following elements to the array. Add items a & b one at a time, then use a single `push` to add the items in part c. Print the array after each step to confirm the changes.

   a. 42
   b. "hello"
   c. `false`, -4.6, "87"

---

2. `push`, `pop`, `shift` and `unshift` are used to add/remove elements from the beginning/end of an array. **Bracket notation** can be used to modify any element within an array. Starting with the `cargoHold` array `['oxygen tanks', 'space suits', 'parrot', 'instruction manual', 'meal packs', 'slinky', 'security blanket']`, write statements to do the following:

   a. Use bracket notation to replace `'slinky'` in the array with `'space tether'`. Print the array to confirm the change.
   b. Remove the last item from the array with `pop`. Print the element removed and the updated array.
   c. Remove the first item from the array with `shift`. Print the element removed and the updated array.
   d. Unlike `pop` and `shift`, `push` and `unshift` require arguments inside the `()`. Add the items 1138 and '20 meters' to the array - the number at the start and the string at the end. Print the updated array to confirm the changes.
   e. Use a template literal to print the final array and its length.

---

3. The `splice` method can be used to either add or remove items from an array. It can also accomplish both tasks at the same time. Use `splice` to make the following changes to the final `cargoHold` array from exercise 2. Be sure to print the array after each step to confirm your updates.

   a. Insert the string `'keys'` at index 3 without replacing any other entries.
   b. Remove 'instruction manual' from the array. (Hint: `indexOf` is helpful to avoid manually counting an index).
   c. Replace the elements at indexes 2 - 4 with the items `'cat'`, `'fob'`, and `'string cheese'`.

---

4. Some methods---like **`splice`** and **`push`**---alter the original array, while others do not. Use the arrays

```
holdCabinet1 ['duct tape', 'gum', 3.14, false, 6.022e23]
```

and

```
holdCabinet2 ['orange drink', 'nerf toys', 'camera', 42, 'parsnip']
```

to explore the following methods: `concat`, `slice`, `reverse`, `sort`. Refer back to the chapter if you need to review the proper syntax for any of these methods.

a. Print the result of using `concat` on the two arrays. Does `concat` alter the original arrays? Verify this by printing `holdCabinet1` after using the method.
b. Print a `slice` of two elements from each array. Does `slice` alter the original arrays?
c. `reverse` the first array, and `sort` the second. What is the difference between these two methods? Do the methods alter the original arrays?

---

5. The `split` method converts a string into an array, while the `join` method does the opposite.

a. Try it! Given the string `str = 'In space, no one can hear you code.'`, see what happens when you print `str.split()` vs. `str.split('e')` vs. `str.split(' ')` vs. `str.split('')`. What is the purpose of the parameter inside the `()`?
b. Given the array `arr = ['B', 'n', 'n', 5]`, see what happens when you print `arr.join()` vs. `arr.join('a')` vs. `arr.join(' ')` vs. `arr.join('')`. What is the purpose of the parameter inside the `()`?
c. Do `split` or `join` change the original string/array?
d. The benefit, cadet, is that we can take a string with **delimiters** (like commas) and convert it into a modifiable array. Alphabetize these hold contents: "water,space suits,food,plasma sword,batteries", and then combine them into a new string.

---

6. Arrays can hold different data types, even other arrays! A **multi-dimensional array** is one with entries that are themselves arrays.

a. Define and initialize the following arrays, which hold the name, chemical symbol and mass for different elements:
   i. `element1 = ['hydrogen', 'H', 1.008]`
   ii. `element2 = ['helium', 'He', 4.003]`
   iii. `element26 = ['iron', 'Fe', 55.85]`
b. Define the array `table`, and use `push(arrayName)` to add each of the element arrays to it. Print `table` to see its structure.
c. Use bracket notation to examine the difference between printing `table[1]` and `table[1][1]`. Don't just nod your head! I want to HEAR you describe this difference. Go ahead, talk to your screen.
d. Using bracket notation and the `table` array, print the mass of element1, the name for element 2 and the symbol for element26.

---

7. For each of the exercises below, assume you are starting with the following people array.

   let people = ["Greg", "Mary", "Devon", "James"];

   1.Using a loop, iterate through this array and console.log all of the people.

   2.Write the command to remove "Greg" from the array.

   3.Write the command to remove "James" from the array.

   4.Write the command to add "Matt" to the front of the array.

   5.Write the command to add your name to the end of the array.

   6.Using a loop, iterate through this array and after console.log-ing "Mary", exit from the loop.

   7.Write the command to make a copy of the array using slice. The copy should NOT include "Mary" or "Matt".

   8.Write the command that gives the indexOf where "Mary" is located.

   9.Write the command that gives the indexOf where "Foo" is located (this should return -1).

   10.Redefine the people variable with the value you started with. Using the splice command, remove "Devon" from the array and add "Elizabeth" and "Artie". Your array should look like this when you are done ["Greg", "Mary", "Elizabeth", "Artie", "James"].

   11.Create a new variable called withBob and set it equal to the people array concatenated with the string of "Bob".