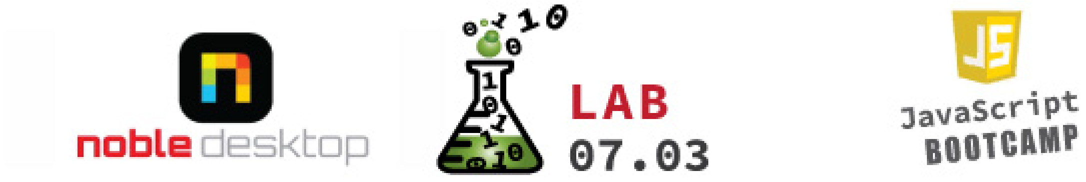

forEach

1. Given an array of numbers, use forEach to cube the integers while doubling the floats. Log each result to the console.
   HINT: If rounding off the number does not change its value, it's an integer

```
    const nums = [4, 3.14, 5, 7.98, 6, 17.75, 7, 55.55];
```

2. Refactor the above by making a new array of the answers, rather than console.logging them. Then log the resulting array.
   HINT: you need to declare a new, empty array above the forEach

```
    const numeros = [];
    // your code
    console.log(numeros);
```

3. Refactor the above using a ternary instead of an if-else.
   HINT: A ternary is one line, so the arrow function syntax can be very concise. In fact, the entire forEach can be written in one line on code.

```
    const numz = [];
    // your code
    console.log(numz);
```

filter-map-chaining vs. forEach

4. Given this array of fruits, use filter-into-map chaining to make jellybeans of fruits with 5 or fewer letters. The expected output is: 'apple jellybean', 'grape jellybean', 'kiwi jellybean', etc.

```
    const fruits = ['apple', 'apricot', 'banana', 'blueberry', 'cherry',
'grape', 'kiwi', 'lime', 'lemon', 'mango', 'orange', 'pineapple',
'strawberry'];
```

5. Do another filter-into-map chaining to make rollups of fruits that begin with a consonant (so, no 'apple', 'apricot' or 'orange', which begin with a vowel).

HINT: It is more efficient to check to see if a word does NOT start with one of 5 vowels, as opposed to seeing if it starts with one of 21 consonants.

6. Use **forEach** with conditional logic to make the jellybeans or rollups, subject to the same conditions:

- 5 or fewer letters for jellybeans
- can't start with a vowel for roll-ups.
  HINT: start with two new empty arrays, one for each category of output

find() and findIndex() methods

7. Use the find() method to find the first animal that *starts* with a vowel.

```
    const animals = ['bunny', 'cat', 'dog', 'elephant', 'frog', 'giraffe',
'hedgehog', 'iguana', 'lynx', 'manatee'];
```

8. Use the findIndex() method to find the index of the first animal that *ends* in a vowel.

9. Use the find() method to find the last number that is equal to twice the number before it. In other words, find the "last double".
   HINT: Use all three available callback arguments: **e**, **i** and **a**.

- a[i+1] is the next item. if a[i+1] / 2 equals e, the current item, it is is doubled
- and so, the current item is not the last double,
- BUT if the a[i+1] / 2 does NOT equal e, then the current item IS the "last "double"
- The answer is 32, since that is the last double

```
    const numz = [8, 16, 32, 75, 100, 256];
```

10. Use **findIndex()** to find the **index** of the first number that has an integer for its square root.
    HINT: If the square root of an integer is rounded off, the integer does not change, so, if the square root is rounded off and it is still equal to the square root itself, then the square root is an integer. The target is 16, since that is the first number that has an integer for a square root. The expected answer is the index, so 1 is the correct answer

END Lab 07.03
SEE Lab 07.03 Solution