Dynamic Object Properties

Seaching by Object Properties

Given this array of vegetables:

```
const veggies = ['Cucumber', 'Tomato', 'Kale'];
```

1. Use the **replace()** method and array items to log the expected message to the console:

```
let msg1 = 'Have you ever ordered the Sandwich?';
console.log(msg1); // Have you ever ordered the Cucumber Sandwich?
```

2. Use the replace() method and array items to log the expected message to the console:

```
let msg2 = 'The "T" in "BLT" stands for "Terminator."';
console.log(msg2); // The "T" in "BLT" stands for "Tomato."
```

3. Use the replace() method and array items to log the expected message to the console:

```
let msg3 = 'Bunny goes to Yale University.';
console.log(msg3); // Bunny goes to Kale University.
```

Given this array of phrases:

```
const phrases = ["A real customer", "A breeze", "Hangin' with the
kids", "Snoopy is a dog"];
```

4. Using string methods and concatenation, make "cool" the next to the last word, no matter which phrase is chosen. Expected output is one of the following:

- a real cool customer
- a cool breeze
- hangin' with the cool kids
- Snoopy is a cool dog

Since we can't predict what the random phrase will be, we cannot use the **replace()** method on a hard-coded last word of the string. So, where does cool need to go? It needs to go in at the last space location. What string method returns the last space of a phrase: **lastIndexOf(" ")** test this on **phrases[0]**, which is the first item in phrases Save the last index of the the space to a variable:

```
    let lastSpaceIndex;
```

then use the **slice()** method twice: once to get everything up to (but not including) the first space; use the **slice()** method again to get everything from the last space to the end of the string; concatenate the two slices together--with " cool " in between.

```
    let coolPhrase;
    console.log("Cool phrase test:", coolPhrase);
```

test the result with a random number in the range of the array

```
    let r;
    console.log("Random cool phrase:", coolPhrase);
```

Given this array of fruits and vegetables:

```
    const words = ['Avocado', 'Broccoli', 'Cauliflower', 'Dragonfruit',
'Eggplant'];
```

5. Use the includes() method and if-else logic to check if the word starts with a consonant or a vowel

- If the word is "Avocado", log: Avocado starts with "A", which is a vowel.
- If the word is "Broccoli", log: Broccoli starts with "B", which is a consonant.

```
    let word;
    let firstChar;
    console.log('firstChar', firstChar);
    let vowels;
    if(vowels) {
    }
```

REFACTOR: Since the if-else console logs differ only in one word: vowel / consonant, better would be to have a variable "consonant" that the if statements sets to "vowel" and then concat that "consonant" or "vowel" vvariable into the output:
Try it again:

```
    let consonantOrVowel;
    if(vowels) {
    }
```

Given this file name:

```
    let fileName = "Mets-Lead-Off-Game-By-Hitting-Three-Straight-
Homers.html";
```

6. Turn the file name into a news headline, with all words capitalized:

Expected result:
Mets Lead Off Game By Hitting Three Straight Homers

```
    let wordsArr;
    console.log(wordsArr); // ['Mets', 'Lead', 'Off', 'Game', 'By',
'Hitting' .. etc.]
    let headline;
```

Remove the file extension, but don't assume '.html' -- detect the dot and remove everything from the dot to the end of the string

```
    let indexOfDot;
```

Use slice() method to get everything from the beginning of the string to to dot (but not includding the dot)

See: Lab 04.04 Solution