Lesson 04.04 - PROG

string methods

includes(), indexOf(), lastIndexOf(), toUpperCase(), toLowerCase()

slice(), replace(), replaceAll(), charAt(), split()

String are kind of like arrays of characters; both structures have:

- index position, with the first item (or character) at index 0
- length property, which returns the number of items (or characters)
- includes(X) method, which checks if X exists in the array or string
- slice(A,B) method, which copies from index A to B (not including B)
- indexOf(X) method, which gets the index of the first instance of X
- lastIndexOf(X) method gets the index of the last instance of X

Naturally, there are also many differences between strings and arrays:

- strings can only store one value at a time (pet = 'cat'), whereas arrays can store many values at a time (pets = ['cat', 'dog'])
- const strings cannot be changed, whereas the items of const arrays can be changed; the const part being "once an array, always an array"
- strings have methods not found in arrays, and vice-versa

string[index]

1. Declare a string, and then get the first and fourth characters:

```
let car = 'Corvette C3';
console.log(car[0]); // C
console.log(car[3]); // v
```

string.length

2. Get the number of characters in car; spaces count:

```
console.log(car.length); // 11
```

3. Get the last character of car, the same way you would get the last item of an array: length-1

```
console.log(car[car.length-1]); // 3
```

4. Check if this 'Vette is a "C2" or "C3" model:

```
console.log(car.includes('C2')); // false
console.log(car.includes('C3')); // true
```

5. Get the index of the first and last "C":

```
console.log(car.indexOf('C')); // 0
console.log(car.lastIndexOf('C')); // 9
```

slice(start_index, end_index)

Copy the make ("Corvette") and model ("C3") to their own variables. We will use the slice method for this.

- slice takes a start and end index as its arguments
- slice has negative indices: -1 is the last character, -2 is next to last
- slice copies and returns a substring from the start to end range
- the end index is exclusive, that is, not included in the substring
- the slice operation does not modify the original string in any way

6. To get the make, "Corvette", slice from index 0 to the index of the space:

```
let spaceIndex = car.indexOf(" ");
let make = car.slice(0, spaceIndex);
console.log(make); // Corvette
```

7. To get the model, "C3", slice from the next index after the space to the end. We are slicing to the end, so omit the second argument (the end index):

```
let model = car.slice(spaceIndex+1);
console.log(model); // C3
```

8. If we are sure that the model is two characters, we can slice backwards:

```
model = car.slice(-2);
console.log(model); // C3
```

Given this string promoting the car:

~~~
```js
let carPromo = "The 2023 Chevrolet Corvette furthers its reputation as a
high-value everyday supercar. The Corvette provides blistering
acceleration, phenomenal handling, a comfortable and well-trimmed cabin
~~~

```
and usable cargo space. Chevrolet reintroduces the high-performance
Corvette Z06 for the 2023 model year, features a new V8 engine with 670
horsepower. Chevrolet also adds a special appearance package that
celebrates the Corvette's 70th anniversary.";
```
```

9. Get the first half of the promo, so from index 0 to 1/2 the length:

```
let firstHalf = carPromo.slice(0, carPromo.length/2);
console.log('firstHalf:', firstHalf);
```

10. Sample a middle slice. Starting some distance before the middle index, and end an equal distance past the half-way point. The math is easier to read if we do this in steps, assigning each number to a variable:

```
let halfLen = carPromo.length / 2;
let distance = 18;
let middleText = carPromo.slice(halfLen - distance, halfLen +
distance);
console.log(middleText); // usable cargo space. Chevrolet reintroduces
```

getting the second sentence of the passage

Let's get the second sentence. This ivolves a little fancy slicing:

11. Get the start index of the second sentence. This is the index of the first period, plus one:

```
let sentence2StartIndex = carPromo.indexOf('.') + 1;
console.log('sentence2StartIndex:', sentence2StartIndex);
// indexOf() method for getting the second instance of a character
```

indexOf('.') gets the index of the first '.', but we want the second period. Pass in startIndex as a second argument. This tells it to start looking for the '.from the startIndex, not from the beginning of the string.

12. Get the index of the second sentence's period.

```
let period2Index = carPromo.indexOf('.', sentence2StartIndex);
console.log('period2Index:', period2Index);
```

13. Get the second sentence slice. Add one to the end to include the period:
```

```
let sentence2 = carPromo.slice(period2Index+1);
console.log('sentence2:', sentence2); // The Corvette provides
blistering acceleration, etc.
```

getting the last sentence of the passage

We need the index of the next-to-last period, as right after this is the starting point of the last sentence. Now, lastIndexOf('.') returns the index of the last period, so we want to tell the method to sample a carPromo that does not end in a period.

14. Copy carPromo to itself, minus the last character, that last '.':

```
carPromo = carPromo.slice(0,-1);
// The last period now belongs to the next-to-last sentence.
```

15. Get the index of the last period:

```
let lastPeriod = carPromo.lastIndexOf('.');
console.log('lastPeriod:', lastPeriod);
```

16. The last sentence starts two characters after the last period, so slice from there to the end. Since we are going to the end, there is no second argument, but we do need to put back the period at the end:

```
let lastSentence = carPromo.slice(lastPeriod + 2) + '.';
console.log('lastSentence:', lastSentence);
```

17. Take a slice of the middle 20 characters. The index of the mid-point is one-half the string length. So, start 10 characters before the mid-point and end 10 charcters after the mid-point:

```
let mid20Chars = carPromo2.slice(carPromo.length/2-10,
carPromo.length/2+10);
console.log('mid20Chars:', mid20Chars); // Chevy reintroduces
```

Now that we've seen how arrays and strings both have index, length, includes(), indexOf() and lastIndexOf(), let's get into the methods that are unique to strings.

toUpperCase(), toLowerCase()

- toUpperCase() returns an uppercase version of the string
- toLowerCase() returns a lowercase version of the string
- the original string is unchanged

18. Make uppercase and lowercase versions of car:

```
let carUC = car.toUpperCase();
console.log(carUC); // CORVETTE C3
let carLC = car.toLowerCase();
console.log(carLC); // corvette c3
```

replace(a,b), replaceAll(a,b)

- replace() is called on a string and takes two arguments: a substring to replace and its replacement.
- replace() operates on the first instance of the target substring.
- replaceAll() works like replace() but hits all instances of the target

19. Replace "Corv" with "'V". This gives us the car's nickname:

```
let nickname = make.replace("Corv", "'V");
console.log(nickname); // 'Vette
```

20. Use replaceAll() to globally change "Covette" to "'Vette". The method returns a new string, so save it to a variable, either itself or a new var:

```
carPromo = carPromo.replaceAll("Corvette", "'Vette");
console.log(carPromo); // 'Vette has replaced Corvette
```

Saving the string to itself vs. to a new variable:

- To modify the original string, save it to itself.
- To also keep the original string, save it to a new var, such as carPromo2.

21. Use replaceAll() again to globally change "Chevrolet" to "Chevy". Save carPromo to itself, so that both replacements accumulate in the same string:

```
carPromo = carPromo.replaceAll("Chevrolet", "Chevy");
console.log(carPromo); // 'Vette & Chevy have replaced Corvette &
Chevrolet
```

22. On second thought, we should have one full "Chevrolet Corvette" before reverting to nicknames. Use replace() to restore the first instance of each:

```
carPromo = carPromo.replace("'Vette'", "Corvette");
carPromo = carPromo.replace("Chevy", "Chevrolet");
```

charAt()

The charAt() method is called on a string:

- charAt(index) takes an index and returns the character found there.
- if the index argument exceeds the max index, it returns undefined.
- charAt() is another way of getting characters by index with []

23. Get the 100th character of the car promo by index and with charAt():

```
console.log(carPromo[99]); // v
console.log(carPromo.charAt(99)); // v
```

split()

- The split() method is called on a string and returns an array.
- It takes an argument (the delimiter) that specifies how the split is done.
- the split() method is often used in conjunction with join() to split a string into an array, perform some operations, and then join the array back into a string.

24. Declare a string (a movie quote about a Corvette C3):

```
let movieQuote = 'Start down low with a 350 cube, three and a quarter
horsepower, 4-speed, 4:10 gears, ten coats of competition orange,
hand-rubbed lacquer with a dual-plane manifold';
```

25. Call the split() method on the string, saving the result to an array:

```
const movieQuoteArr = movieQuote.split();
console.log(movieQuoteArr); // ['Start down low with a 350 cube, three
and a quarter horsepower, etc.]
```

Notice there's only one item in the array--the entire string.

26. Make each word an array item by passing in a space as delimiter:

```
const quoteWords = movieQuote.split(" ");
console.log(quoteWords); // ['Start', 'down', 'low', 'with' 'a',
'350', etc.]
```

27. Make each individual letter an array item with an empty string delimiter:

```
const quoteLetters = movieQuote.split("");
console.log(quoteLetters); // ['S', 't', 'a', 'r', 't', ' ', 'd', 'o', etc.]
```

28. Pass in a comma-space (", ") as the delimiter. This splits the string every few words, into phrases:

```
const quotePhrases = movieQuote.split(", ");
console.log(quotePhrases); // ['Start down low with a 350 cube', 'three and a quarter horsepower', etc.]
```

That's nice, but that last phrase should be two. Let's pop the last item, make two strings from it, and then push the two strings back into the array.

29. Pop the last item, and save it:

```
let lastPart = quotePhrases.pop();
```

30. Get part A by slicing from the beginning to "with":

```
let partA = lastPart.slice(0, lastPart.indexOf("with"));
console.log(partA); // hand-rubbed lacquer
```

31. Get part B by starting at indexOf("with") and slicing to the end. Add "with a".length, because we want to skip "with a ", which has a length of 6:

```
let partB = lastPart.slice(lastPart.indexOf("with") + "with a".length+1);
console.log(partB); // dual-plane manifold
```

32. Push the two parts into the quotePhrases array:

```
quotePhrases.push(partA, partB);
console.log(quotePhrases); // ['Start down low with a 350 cube', 'three and a quarter horsepower', '4-speed', '4:10 gears', 'ten coats of competition orange', 'hand-rubbed lacquer', 'dual-plane manifold']
```