Strings

methods

The String object is used to represent and manipulate a sequence of characters.

▼ What is the String constructor?

```
const string4 = new String("A String object");
```

▼ How can you access a character in a string with charAt() ?

```
return 'cat'.charAt(1) // returns "a"
```

▼ How can you access a character in a string by index ?

```
return 'cat'[1] // returns "a"
```

▼ How to compare strings?

If you wish to compare without regard to upper or lower case characters, use a function similar to this:

```
function isEqual(str1, str2)
{
    return str1.toUpperCase() === str2.toUpperCase()
} // isEqual
```

▼ What does String.prototype.concat() do?

The concat() method concatenates the string arguments to the calling string and returns a new string.

```
const str1 = 'Hello';
const str2 = 'World';

console.log(str1.concat(' ', str2));
// expected output: "Hello World"

console.log(str2.concat(', ', str1));
// expected output: "World, Hello"
```

▼ What does String.prototype.includes() do?

The includes() method performs a case-sensitive search to determine whether one string may be found within another string, returning true or false as appropriate.

```
const sentence = 'The quick brown fox jumps over the lazy dog.';

const word = 'fox';

console.log(`The word "${word}" ${sentence.includes(word) ? 'is' : 'is not'} in the sentence`);
// expected output: "The word "fox" is in the sentence"
```

▼ What does String.prototype.endsWith() do?

The endsWith() method determines whether a string ends with the characters of a specified string, returning true or false as appropriate.

```
const str1 = 'Cats are the best!';

console.log(str1.endsWith('best', 17));
// expected output: true

const str2 = 'Is this a question';

console.log(str2.endsWith('?'));
// expected output: false
```

▼ What does String.prototype.indexOf() do?

The indexOf() method returns the index within the calling String object of the first occurrence of the specified value, starting the search at fromIndex. Returns -1 if the value is not found.

```
const paragraph = 'The quick brown fox jumps over the lazy dog. If the dog barked, was it really lazy?';

const searchTerm = 'dog';
const indexOfFirst = paragraph.indexOf(searchTerm);

console.log(`The index of the first "${searchTerm}" from the beginning is ${indexOfFirst}`);
// expected output: "The index of the first "dog" from the beginning is 40"

console.log(`The index of the 2nd "${searchTerm}" is ${paragraph.indexOf(searchTerm, (indexOfFirst + 1))}`);
// expected output: "The index of the 2nd "dog" is 52"
```

▼ What does String.prototype.lastIndexOf() do?

The lastIndexOf() method returns the index within the calling String object of the last occurrence of the specified value, searching backwards from fromIndex. Returns -1 if the value is not found.

```
const paragraph = 'The quick brown fox jumps over the lazy dog. If the dog barked, was it really lazy?';

const searchTerm = 'dog';

console.log(`The index of the first "${searchTerm}" from the end is ${paragraph.lastIndexOf(searchTerm)}`);
// expected output: "The index of the first "dog" from the end is 52"
```

▼ What does String.prototype.match() do?

The match() method retrieves the result of matching a string against a regular expression.

```
const paragraph = 'The quick brown fox jumps over the lazy dog. It barked.';
const regex = /[A-Z]/g;
const found = paragraph.match(regex);

console.log(found);
// expected output: Array ["T", "I"]
```

▼ What does String.prototype.matchAll() do?

The matchAll() method returns an iterator of all results matching a string against a regular expression, including capturing groups.

```
const regexp = /t(e)(st(\d?))/g;
const str = 'test1test2';

const array = [...str.matchAll(regexp)];
```

```
console.log(array[0]);
// expected output: Array ["test1", "e", "st1", "1"]

console.log(array[1]);
// expected output: Array ["test2", "e", "st2", "2"]
```

▼ What does String.prototype.repeat() do?

The repeat() method constructs and returns a new string which contains the specified number of copies of the string on which it was called, concatenated together.

```
const chorus = 'Because I\'m happy. ';

console.log(`Chorus lyrics for "Happy": ${chorus.repeat(27)}`);

// expected output: "Chorus lyrics for "Happy": Because I'm happy. Because I'm happy. Because I'm happy. Because I'm happy. Because I'm
```

▼ What does String.prototype.replace() do?

The replace() method returns a new string with some or all matches of a pattern replaced by a replacement. The pattern can be a string or a RegExp, and the replacement can be a string or a function to be called for each match.

The original string is left unchanged.

```
const p = 'The quick brown fox jumps over the lazy dog. If the dog reacted, was it really lazy?';

console.log(p.replace('dog', 'monkey'));
// expected output: "The quick brown fox jumps over the lazy monkey. If the dog reacted, was it really lazy?"


const regex = /Dog/i;
console.log(p.replace(regex, 'ferret'));
// expected output: "The quick brown fox jumps over the lazy ferret. If the dog reacted, was it really lazy?"
```

▼ What does String.prototype.replaceAll() do?

The **`replaceAll()`** method returns a new string with all matches of a `pattern` replaced by a `replacement`. The `pattern` can be a string or a `RegExp`, and the `replacement` can be a string or a function to be called for each match.

The original string is left unchanged.

```
const p = 'The quick brown fox jumps over the lazy dog. If the dog reacted, was it really lazy?';

console.log(p.replaceAll('dog', 'monkey'));
// expected output: "The quick brown fox jumps over the lazy monkey. If the monkey reacted, was it really lazy?"


// global flag required when calling replaceAll with regex
const regex = /Dog/ig;
console.log(p.replaceAll(regex, 'ferret'));
// expected output: "The quick brown fox jumps over the lazy ferret. If the ferret reacted, was it really lazy?"
```

▼ What does String.prototype.search() do?

The search() method executes a search for a match between a regular expression and this String object.

```
const paragraph = 'The quick brown fox jumps over the lazy dog. If the dog barked, was it really lazy?';

// any character that is not a word character or whitespace
const regex = /[^\w\s]/g;

console.log(paragraph.search(regex));
// expected output: 43

console.log(paragraph[paragraph.search(regex)]);
// expected output: "."
```

▼ What does String.prototype.slice() do?

The slice() method extracts a section of a string and returns it as a new string, without modifying the original string.

```
const str = 'The quick brown fox jumps over the lazy dog.';

console.log(str.slice(31));
// expected output: "the lazy dog."

console.log(str.slice(4, 19));
// expected output: "quick brown fox"

console.log(str.slice(-4));
// expected output: "dog."

console.log(str.slice(-9, -5));
// expected output: "lazy"
```

▼ What does String.prototype.startsWith() do?

The startsWith() method determines whether a string begins with the characters of a specified string, returning true or false as appropriate.

```
const str1 = 'Saturday night plans';

console.log(str1.startsWith('Sat'));
// expected output: true

console.log(str1.startsWith('Sat', 3));
// expected output: false
```

▼ What does String.prototype.substring() do?

The substring() method returns the part of the string between the start and end indexes, or to the end of the string.

```
const str = 'Mozilla';

console.log(str.substring(1, 3));
// expected output: "oz"

console.log(str.substring(2));
// expected output: "zilla"
```

▼ What does String.prototype.toLowerCase() do?

```
const sentence = 'The quick brown fox jumps over the lazy dog.';

console.log(sentence.toLowerCase());
// expected output: "the quick brown fox jumps over the lazy dog."
```

▼ What does String.prototype.toUpperCase() do?

```
const sentence = 'The quick brown fox jumps over the lazy dog.';

console.log(sentence.toUpperCase());
// expected output: "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG."
```

▼ What does String.prototype.trim() do?

The trim() method removes whitespace from both ends of a string. Whitespace in this context is all the whitespace characters (space, tab, no-break space, etc.) and all the line terminator characters (LF, CR, etc.).

```
const greeting = '   Hello world!   ';

console.log(greeting);
// expected output: "   Hello world!   ";

console.log(greeting.trim());
// expected output: "Hello world!";
```

▼ What does String.prototype.trimStart() do?

The trimStart() method removes whitespace from the beginning of a string. trimLeft() is an alias of this method.

```
const greeting = '   Hello world!   ';

console.log(greeting);
// expected output: "   Hello world!   ";

console.log(greeting.trimStart());
// expected output: "Hello world!   ";
```

▼ What does String.prototype.trimEnd() do?

The trimEnd() method removes whitespace from the end of a string. trimRight() is an alias of this method.

```
const greeting = '   Hello world!   ';

console.log(greeting);
// expected output: "   Hello world!   ";

console.log(greeting.trimEnd());
// expected output: "   Hello world!";
```