STRING CHEAT SHEET - JAVASCRIPT

Creating Strings

- Literal: let str = 'Hello, World!';
- Constructor: let str = new String('Hello, Testleaf!');

Basic Properties

length: Returns the length of the string.

```
str.length
```

Retrieving Parts of a String

charAt(index): Returns the character at the specified index.

```
str.charAt(0)      // 'H'
```

substring(startIndex, [endIndex]): Returns a part of the string between the start and end indexes.

```
str.substring(0, 5)          // 'Hello'
```

slice(startIndex, [endIndex]): Similar to `substring` but can accept negative indexes.

```
str.slice(-6)                // 'Testleaf!'
```

substr(startIndex, length): Returns a part of the string starting from the index for a given number of characters.

```
str.substr(0, 5)             // 'Hello'
```

Modifying Strings

concat(string2, string3, ..., stringN): Combines the text of several strings and returns a new string.

```
str.concat(' How are you?')
```

trim(): Removes whitespace from both ends of a string.

```
str.trim()
```

toUpperCase(), toLowerCase(): Returns the string in upper or lower case.

```
str.toUpperCase()
```

replace(searchFor, replaceWith): Replaces the specified segment of the string.

```
str.replace('World', 'Mars')
```

Splitting a String

split([separator, [limit]]): Splits a String object into an array of strings by separating the string into substrings.

```
str.split(',')                // ['Hello', ' Testleaf!']
```

Searching in a String

indexOf(subString, [fromIndex]), lastIndexOf(subString,[fromIndex]): 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.

```
str.indexOf('l')              // 2
```

includes(subString, [position]): Determines whether one string may be found within another string.

```
str.includes('Testleaf')      // true
```

startsWith(searchString, [position]), endsWith(searchString, [position]): Determines whether a string begins with or ends with the specified string.

```
str.startsWith('Hello')       // true
str.endsWith('!')             // true
```

Miscellaneous

repeat(count): Constructs and returns a new string which contains the specified number of copies of the string on which it was called, concatenated together.

```
str.repeat(2)
```

match(regexp): Used to match a regular expression against a string.

```
str.match(/[A-Z]/g)
```

search(regexp): Executes a search for a match between a regular expression and this String object.

```
str.search(/[A-Z]/)
```

Template Literals

Backticks (\` \`): Allow embedded expressions and multi-line strings.

```
let name = 'World';
let greeting = `Hello, ${name}!`;
```