Lesson Weekend

Introduction to Programming (/introduction-to-programming) / JavaScript and Web Browsers (/introduction-to-programming/javascript-and-web-browsers) / String and Number Methods

Text Cheat sheet

Let's dive more into methods. JavaScript offers built-in methods for strings and numbers. Let's take a look at a few of these methods, including some we've learned about already. Note that the list below isn't comprehensive — there are many more methods out there that we'll take a look at soon!

String Methods

`toUpperCase`

Here's one we've learned already:

```
> "supercalifragilisticexpialidocious".toUpperCase();
"SUPERCALIFRAGILISTICEXPIALIDOCIOUS"
```

`toLowerCase`

Here's the opposite of `toUpperCase`, making a string all lowercase:

```
> "HOWDY, NEIGHBOR".toLowerCase();
"howdy, neighbor"
```

`concat`

Here's the other string method we've learned — a method so important that we recommend memorizing it (though you can always look up the documentation if you need to):

```
> const word = "foo";
undefined
> word.concat("bar");
"foobar"
```

`charAt`

Some string methods take numbers as arguments. Here's an example:

```
> "caterpillar".charAt(5);
"p"
```

Did you notice something funny about this example? What's the fifth character of the word "caterpillar"? Well, with the way we normally count, "c" is the first letter, and "r" is the fifth. But JavaScript says that the character at the 5th position is "p". That's because computers start counting at zero. So "c" is the zeroth letter, and "r" is the fourth. This is really important — and it's another thing that trips up beginners. However, it will soon become second nature.

Number Methods

`toString`

This method turns a number into a string.

```
> const myBirthday = 42;
> const myStringifiedBirthday = myBirthday.toString();
"42"
```

We can also call `toString` on a number instead of a variable:

```
> 42.34.toString();
"42.34"
> 42..toString();
"42"
```

Note that you need to add the decimal point after a whole number before you can call `toString` on it! So that means there are two decimal points.

`toFixed`

This method returns a string with only the number of decimal points that is specified in the argument. If you want 1 decimal point, and your number has 3, then `toFixed` will chop off the extra decimal points, leaving only 1. Note that this method is called on a number, but turns the number into a string.

```
> 42.222.toFixed(1);
'42.2'
> const myFavRealNumber = 3.14;
> myFavRealNumber.toFixed(0);
'3'
```

Chaining Methods

Methods can be **chained** like this:

```
> "foo".concat("bar").toUpperCase();
"FOOBAR"
```

We actually tried out chaining these exact string methods when we discussed the distinction between methods and functions. As we can see, when we chain methods together like this, we first concatenate the string and then uppercase that string.

Here's an example of chaining number methods:

```
> 4.3354.toFixed(2).toString();
"4.34"
```

In this example, we are first executing `toFixed` on the number, and then we're turning it onto a string with `toString`. **Take note**, it is redundant to call `toString` after `toFixed`, because `toFixed` returns a string anyways.

Concatenation or Arithmetic with the `+` Operator

The `+` operator lets us do addition if we're working with numbers, or concatenation if we're working with strings. Using the `+` with numbers should be familiar:

```
> 1 + 3
4
```

Let's take a look at using `+` with strings:

```
> "I love" + " " + "Epicodus.";
"I love Epicodus."
```

As we can see, we can just use the `+` sign to "add" strings together! This has the same effect as the `concat` method.

And we can do this with variables, too:

```
> const sentiment = "I love ";
undefined
> const animal1 = "cats";
undefined
> const animal2 = "dogs";
undefined
> const exclamation = "!";
> sentiment + animal1 + exclamation;
"I love cats!"
> sentiment + animal2 + exclamation;
"I love dogs!"
```

Or with variables holding numbers:

```
> const number1 = 1;
undefined
> const number2 = 99;
undefined
> number1 + number2;
100
```

Now, check out this more complex example, where we include a variable with a method called on it:

```
> const myBirthday = 42;
> const aboutMe = "I am " + myBirthday.toString() + " and I
was born in '79.";
> aboutMe;
"I am 42 and I was born in 1979."
```

Here's the same example written in a different way:

```
> "I am " + 42.toString() + " and I was born in '79.";
"I am 42 and I was born in 1979."
```

Previous (/introduction-to-programming/javascript-and-web-browsers/functions)
Next (/introduction-to-programming/javascript-and-web-browsers/practice-string-and-number-methods)

Lesson 14 of 75
Last updated March 24, 2023

disable dark mode

Epicodus (http://www.epicodus.com)