**Calculating length**

Since a single string can contain any number of characters, we may find it useful to count the number of characters in a string using .length:

console.log("ramen".length); // => 5

console.log("go home!".length); // => 8

console.log("".length); // => 0

**Indexing a string**

Strings consist of multiple characters. These characters are numbered by **indices** starting at 0. So in the string 'bootcamp', 'b' is at index 0, 'o' is at index 1, 'o' is at index 2, 't' is at index 3, and so on. We can look at particular characters of a string by using [] and specifying an index:

console.log("bootcamp"[0]); // => 'b'

console.log("bootcamp"[1]); // => 'o'

console.log("bootcamp"[10]); // => undefined

console.log("bootcamp"[-3]); // => undefined

### **Using indexOf**

We can also calculate the index of a given character within a string by using indexOf:

console.log("bagel".indexOf("b")); // => 0

console.log("bagel".indexOf("a")); // => 1

console.log("bagel".indexOf("l")); // => 4

console.log("bagel".indexOf("z")); // => -1

If we attempt to search for a character that is **not** present in a string, indexOf will return -1. This makes sense because we know that -1 is not a valid string index. The smallest index possible is 0!

If we search for a character that appears more than once in a string, indexOf will return the index of the first occurance of that character.

We can also use indexOf to search for a substring of characters. Under this circumstance, indexOf will return the index where the substring begins in the main string:

console.log("door hinge".indexOf("oor")); // => 1

console.log("door hinge".indexOf("hi")); // => 5

console.log("door hinge".indexOf("hint")); // => -1

## Concatenation

Concatenation is just a fancy word for joining strings together into a single string. To concatenate strings, we use the + operator: