Arrays and Objects

Sprint 4

Perform logical comparison operations.

- Arrays can be created using array literal syntax
- All arrays have the `length` property:
  - It returns the number of elements in an array
  - It can be used to access the last element in an array:

```
const emptyArray = [];
const numbers = [1, 2, 3];
numbers[numbers.length - 3];  // 3
```

- Array elements can be accessed or modified by their index using the brackets operator:

```
const numbers = [1, 2, 3];
numbers[0];                    // 1
numbers[0] = "a";
console.log(numbers);          // ["a", 2, 3]
```

Iterating Over Arrays

For example, using a `for` loop to iterate through an array:

```
const numbers = [1, 2, 3];
for (let i = 0; i < numbers.length; i++) {
  console.log(numbers[i]);
}

// logs 1, 2, and 3 to the console on separate lines
```

Analogous example of using a `for..of` loop:

```
const numbers = [1, 2, 3];
for (let number of numbers) {
  console.log(number);
}

// logs 1, 2, and 3 to the console on separate lines
```

Adding and Removing Array Elements

The `push()` method:

- Adds an element to the end of an array
- Returns the length of the array

```
const letters = ["a", "b", "c"];
letters.push("d");              // 4
console.log(letters);           // ["a", "b", "c", "d"]
```

The `pop()` method:

- Removes the last element of an array
- Returns the removed element

```
const letters = ["a", "b", "c"];
letters.pop();                  // "c"
console.log(letters);           // ["a", "b"]
```

Working with Objects

Data structures that store unsorted collections of key:value pairs.

Creating an object with object literal syntax:

- separate keys and values with a colon (`:`)
- keys must be given valid JavaScript variable names or be enclosed in quotes
- values can be any JavaScript value

```
const user = {
  name: "Sample User",
  "email address": "sample@example.com"
}
```

You can access and change keys using either:

- dot syntax (`.`), if the key has a valid JavaScript name
- brackets syntax (`[ ]`), if it does not

```
user.name;                              // "Sample User"
user.name = "New Name";
user.name;                              // "New Name"

user["email address"];                  // "kebab"
user["email address"] = "foo@bar.com";
user["email address"];                 // "foo@bar.com"
```