Section Review

Learn to Code with Ruby

Arrays

- An **array** is an object for storing objects in order. The items inside an array are called **elements**.
- An **array** can store any kind of Ruby object: integers, strings, nil, and even other arrays.
- Ruby assigns each element an index position (an order in line). The index starts counting at 0.

Accessing and Overwriting Array Elements

- We can access an array element by its index position. Provide a pair of square brackets with the index position.
- We can provide two numbers inside square brackets (starting index, amount of elements to pull out).
- The **values_at** method extracts elements at non-consecutive index positions.
- The **first** and **last** methods can extract a number of elements from either the beginning or the end of the array.

Counting Values

- The **length** and **size** methods return the number of elements in the array.
- An array's length will be one greater than its final index position.
- The **count** method counts *how many times* a value occurs within the array.
- The **empty?** method returns true if the array is empty.

Equality Comparisons

- The **equality** (==) and **inequality** (!=) operators work for two arrays. Ruby will compare equality across all elements.
- The **spaceship operator** returns 0 if two values are equal, -1 if the value on the left-hand side is smaller, 1 if the value on the left-hand side is greater, and nil if the values are incomparable.

Mutating an Array

- The **push method** and **shovel operator (<<)** add one or more elements to the end of an array.
- The **insert** method adds one or more elements at a specified index position. Other elements are pushed down one slot.
- The **pop** method removes and returns a specified number of elements from the end of the array.
- The **shift** method removes and returns a specified number of elements from the beginning of the array.
- The **unshift** method adds one or more elements to the beginning of an array.