

Section Review

Learn to Code with Ruby

Array Iteration with `each` and `each_index`

- The `each` method iterates over every sequential array element. It accepts a block.
- Ruby passes each array element to the block as a block variable. We can then do whatever we'd like with the value.
- To get access to the index, use the `each_with_index` method. Ruby passes the index position as the second block value.

The map and collect Methods

- The **map/collect** methods return a new array consisting of the results of performing a consistent operation on all elements in an array.
- Use these methods to create a new array with the same length as the original one., Think of it as "mapping one value to another".

Filtering Array Elements

- We can manually construct a new array consisting of a subset of elements from an original array. Combine the **each** method with an **if** statement.
- The more idiomatic approach is to use methods like **select** or **reject**. The methods accept a block that should return a Boolean.
- The **select** method will keep all elements for which the block returns true.
- The **reject** method will discard all elements for which the block returns true.
- The **partition** method returns an array of two arrays. The first array holds the elements that satisfy the block condition; the second array holds the elements that fail the block condition.

Other Array Methods

- The **find/detect** method locates the first element that satisfies the block's condition.
- The **any?/all?** methods verify if any or all of the elements satisfy the block's condition.
- The **max** and **min** methods return the largest and smallest values within an array.
- The **index/find_index** methods find the first index position of a given value in the array. They return **nil** if the element does not exist.
- The **include?** method checks for inclusion within an array. The method also exists on other Ruby objects like strings.

Unlimited Method Arguments/Splat Arguments

- Use the `*` syntax to enable a parameter to accept an unlimited number of arguments. When the method is invoked, Ruby will capture the arguments in an array.
- We sometimes call this a **sponge** or **splat** argument because it "absorbs" all its values. A method can only have one sponge argument.