codecademy

Looping with Ruby

Ruby Assignment Operators

Assignment operators in Ruby are used to assign or update values to variables. The most common assignment operator is `=` but others also exist, like `+=`, `-=`, `*=` and `/=`.

Ruby each Method

To iterate over an array in Ruby, use the `.each` method. It is preferred over a `for` loop as it is guaranteed to iterate through each element of an array.

Ruby “next” Keyword

In Ruby, the `next` keyword is used within a loop to pass over certain elements and skip to the following iteration. It is useful for omitting elements that you do not wish to have iterated. `next` is followed by an `if` statement which defines which elements are to be skipped.

```
for i in 1..10
  next if i % 2 == 0
  puts i
end

#In this example, the next keyword along with a shorthand if statement is used to skip over the even numbers in the sequence.

# Output:
# 1
# 3
# 5
# 7
# 9
```

Ruby while Loop

Putting a block of code in a `while` loop in Ruby will cause the code to repeatedly run the code as long as its condition is `true`.
If the block of code doesn’t have a way for the condition to be changed to `false`, the `while` loop will continue forever and cause an error.

Ruby times Method

To execute the same block of code a set a number of times in Ruby, use the `times` method.

```
5.times { puts ""Codecademy"" }

# Output:
# Codecademy
# Codecademy
# Codecademy
# Codecademy
# Codecademy
```

Ruby Range

In ruby, a sequence of integers can be demonstrated by a *range*. The range can be divided into an *inclusive range* where the last integer in the sequence is included and an *exclusive range* where the last integer is excluded.

Ruby loop

A `loop` method can be used to run a block of code repeatedly in Ruby. Either use curly braces (`{}`) or the `do / end` keyword combination to wrap the block the code that will be looped.

Ruby until Loop

Putting a block of code inside an `until` loop in Ruby will cause the code to run as long as its condition remains `false` . It's only when the condition becomes `true` that the loop stops.
If the block of code doesn't allow for a way for the condition to be changed to `true` then the loop will continue forever and it will cause an error.

Ruby for Loop

A block of code can be repeated a set amount of times with the `for` loop in Ruby.