**CONTROL FLOW IN RUBY**

**How It Works**

You may have noticed that the kinds of programs we’ve written so far in Ruby aren’t very flexible. Sure, they can take user input, but they always produce the same result based on that input; they don’t change their behavior in reaction to the **environment** (the collection of all variables and their values that exist in the program at a given time).

**Control flow** gives us the flexibility we’re looking for. We can select different outcomes depending on information the user types, the result of a computation, or the value returned by another part of the program.

**Instructions**

**1.**

Check out the code in the editor. There’s some new syntax there, but we’ll bet you can guess what it does. Click Run to see the program in action! (Go ahead and give Ruby an integer—that is, a positive or negative number with no decimal bit.)

Checkpoint 2 Passed

Hint

Note: Be sure to give input for the terminal. When it expects input but never receives it an error will be displayed after some time. This prevents it from running indefinitely.

**script.rb**

print "Integer please: "

user\_num = Integer(gets.chomp)

if user\_num < 0

  puts "You picked a negative integer!"

elsif user\_num > 0

  puts "You picked a positive integer!"

else

  puts "You picked zero!"

end

**If**

Ruby’s if statement takes an **expression**, which is just a fancy word for something that has a value that evaluates to either true or false. If that expression is true, Ruby executes the block of code that follows the if. If it’s not true (that is, false), Ruby doesn’t execute that block of code: it skips it and goes on to the next thing.

Here’s an example of an if statement in action:

if 1 < 2  
  print "I'm getting printed because one is less than two!"  
end

Ruby doesn’t care about **whitespace** (spaces and blank lines), so the indentation of the print statement isn’t *necessary*. However, it’s a convention that Rubyists (Ruby enthusiasts) follow, so it’s good to get in the habit now. The block of code following an if should be indented two spaces.

When you’re done with your if, you have to tell Ruby by typing end.

**Instructions**

**1.**

Write your own if statement in the editor. It can take any expression you want (even just true!), but it should evaluate to true. When it does, it should print a string of your choice to the console (using print or puts).

**script.rb**

if "four".length > "two".length

  puts "The word four is longer than the word two"

  end