**PUTTING THE FORM IN FORMATTER**

**What You'll Be Building**

This project will help you create a small program that will read a user’s input and correct his or her capitalization. Users can provide an almost infinite range of input, so it makes our lives easier as programmers to make their input standard before doing anything with it.

**Instructions**

Check out the code in the editor—we’ve added some new things that we’ll be teaching you. Can you guess what it does? Click Run to find out!

Answer each question, and hit enter (return on some keyboards).

**script.rb**

print "What's your first name? "

first\_name = gets.chomp

first\_name.capitalize!

print "What's your last name? "

last\_name = gets.chomp

last\_name.capitalize!

print "What city are you from? "

city = gets.chomp

city.capitalize!

print "What state or province are you from? "

state = gets.chomp

state.upcase!

puts "Your name is #{first\_name} #{last\_name} and you're from #{city}, #{state}!"

**Prompting the User**

First, let’s write the code we’re already familiar with. In order to get input from the user, we’ll first need to print a prompt on the screen.

**Instructions**

**1.**

print the question "What's your first name?" to the screen. Feel free to peek back at the first exercise if you need a syntax reminder.

**script.rb**

print "What's' your first name? "

**Getting Input**

Good! Now let’s try something new. You may have noticed this weird little line of code repeated in our example:

variable\_name = gets.chomp

gets is the Ruby method that *gets* input from the user. When getting input, Ruby automatically adds a blank line (or **newline**) after each bit of input; chomp removes that extra line. (Your program will work fine without chomp, but you’ll get extra blank lines everywhere.)

**Instructions**

**1.**

Declare a variable first\_name and set it equal to gets.chomp.

This checkpoint may seem like it’s running forever, but the terminal to the right is actually waiting for input because of gets.chomp. Answer the question in the terminal and press “Enter” or “Return” to finish checking your work.

Checkpoint 2 Passed

Hint

After running your code you’ll see What is your first name? printed. You can select the terminal with your cursor to type your name and then press Enter.

**script.rb**

print "What's' your first name? "

first\_name = gets.chomp

**Repeat for More Input**

All right! Now we need to repeat what we’ve done for last\_name, city, and state.

**Instructions**

**1.**

Add print prompts, variables, and gets.chomps for the user’s last name, city, and state/province. Use last\_name as the variable for the user’s last name, city for their city, and state for their state or province. (Hint: prompt the user to provide an abbreviation for their state or province, such as “NY” for New York. This will naturally lead us to use .upcase later!)

After pressing **Run**, type a response in the console and hit enter for each prompt.

**script.rb**

print "What's your first name? "

first\_name = gets.chomp

print "What's your last name? "

last\_name = gets.chomp

print "What city are you from? "

city = gets.chomp

print "What state or province are you from? "

state = gets.chomp

**Printing the Output**

As you may have noticed, Ruby’s not returning any feedback to us. We want to be able to see our string formatting in action! For this, we’ll need one more new piece of syntax.

If you define a variable monkey that’s equal to the string "Curious George", and then a string that says "I took #{monkey} to the zoo", Ruby will do something called *string interpolation* and replace the #{monkey} bit with the value of monkey. Therefore, it will print "I took Curious George to the zoo".

We can do the same thing here. For example:

first\_name = "Kevin"  
puts "Your name is #{first\_name}!"

The code above will print Your name is Kevin!.

**Instructions**

**1.**

Let’s use this syntax to print out the values of first\_name, last\_name, city, and state using #{} syntax.

Press the **Run** button to test your code. Don’t forget to enter your answers in the console!

Checkpoint 2 Passed

Hint

Remember, your string interpolation always has to be contained within double quotes, like so:

n = "bacon"  
puts "#{n}"  
puts "I love #{n}!" # Prints: I love bacon!

**script.rb**

print "What's your first name? "

first\_name = gets.chomp

puts "#{first\_name}"

print "What's your last name? "

last\_name = gets.chomp

puts "#{last\_name}"

print "What city are you from? "

city = gets.chomp

puts "#{city}"

print "What state or province are you from? "

state = gets.chomp

puts "#{state}"

**Formatting with String Methods**

Great! Now we’ve got our output, but as you can see, we haven’t used string methods to properly capitalize everything yet.

print "This is my question?"  
answer = gets.chomp  
answer2 = answer.capitalize   
answer.capitalize!

1. First we introduce one new method, capitalize, here. It capitalizes the first letter of a string and makes the rest of the letters lower case. We assign the result to answer2
2. The next line might look a little strange, we don’t assign the result of capitalize to a variable. Instead you might notice the ! at the end of capitalize. This modifies the value contained within the variable answer itself. The next time you use the variable answer you will get the results of answer.capitalize

**Instructions**

**1.**

After each variable assignment: first\_name, last\_name, and city add the .capitalize! method

For state use the .upcase! method

Press **Run** to test your code. Don’t forget to enter your answers in the console!

**script.rb**

print "What's your first name? "

first\_name = gets.chomp

first\_name.capitalize!

puts "#{first\_name}"

print "What's your last name? "

last\_name = gets.chomp

last\_name.capitalize!

puts "#{last\_name}"

print "What city are you from? "

city = gets.chomp

city.capitalize!

puts "#{city}"

print "What state or province are you from? "

state = gets.chomp

state.upcase!

puts "#{state}"