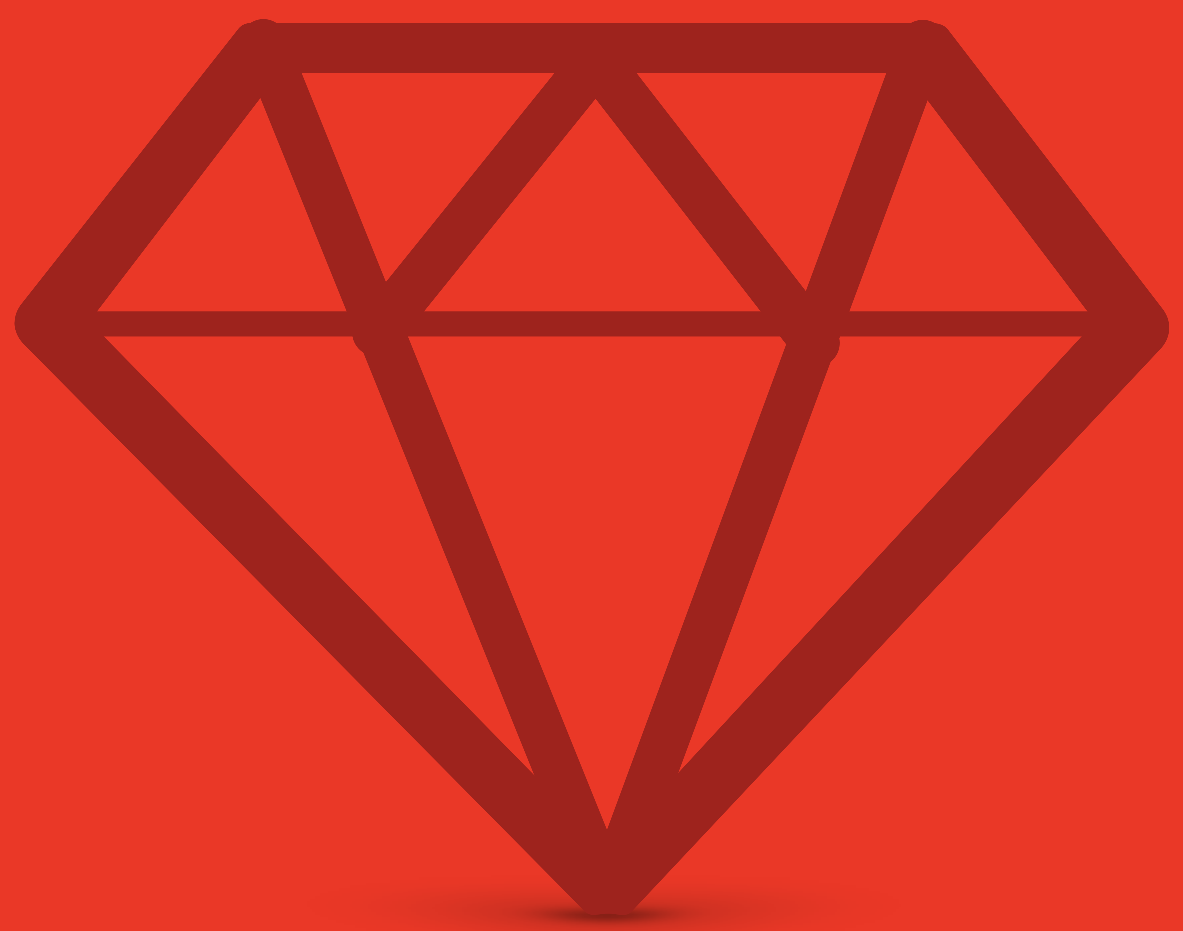
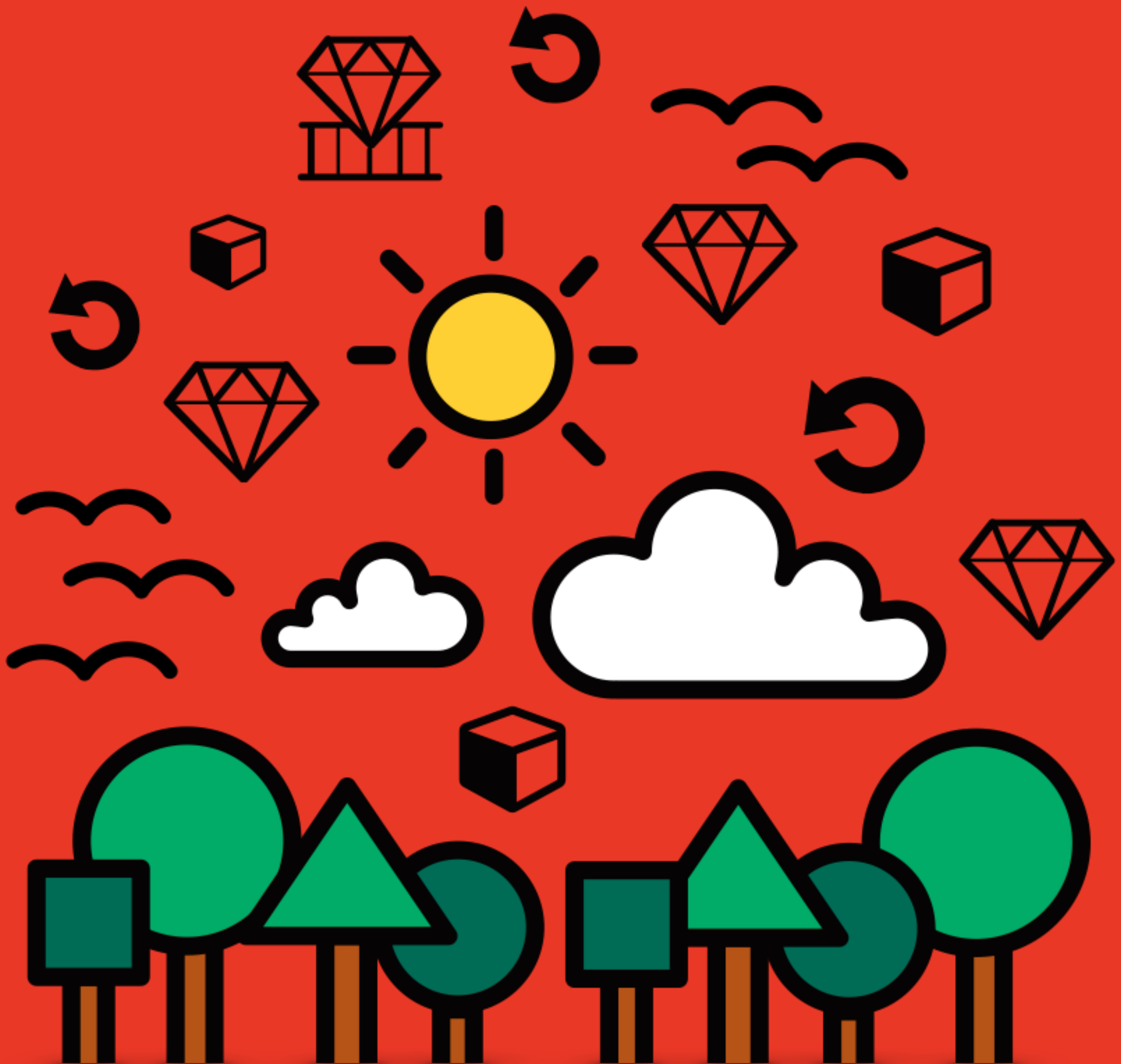
Basic Programming





by **Reindert-Jan Ekker**
and **Joost Elfering**

Programming Ruby: The Basics

A basic set of things which you can combine to build anything

Overview

- What is **Ruby**
- **Interactive Ruby Shell**
- **Values and Variables**
- **Operators and Comparisons**
- **Methods**

Ruby

The Programming Language

What?

- Multi-purpose Programming Language
- Multiparadigmatic
- Japanese

Why?

- Popular
- Easy to learn
- Very Productive
- Dynamic

But First

An Example

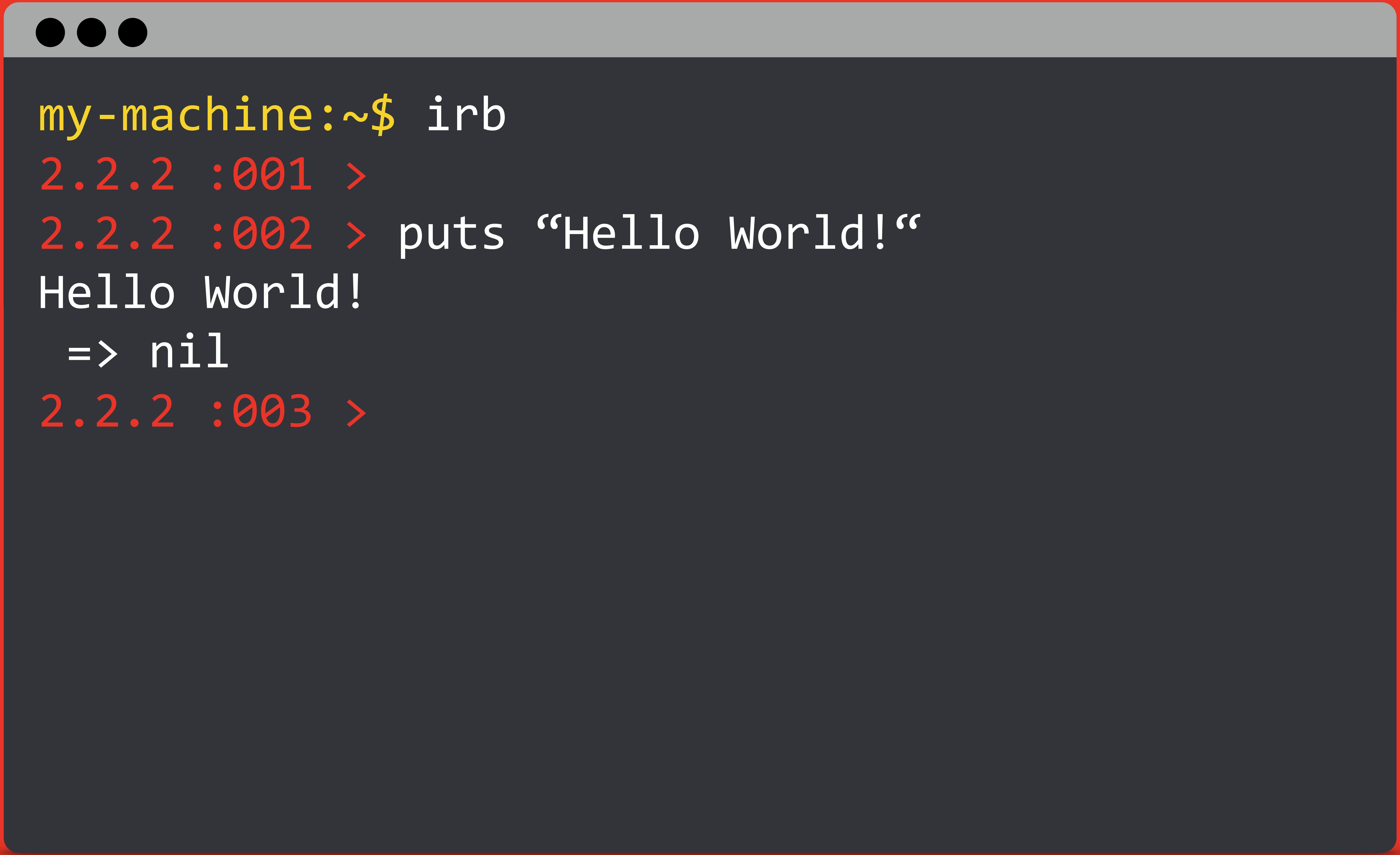
>_ Showcase: a little guessing game

IRB

Interactive Ruby Shell

>_

IRB

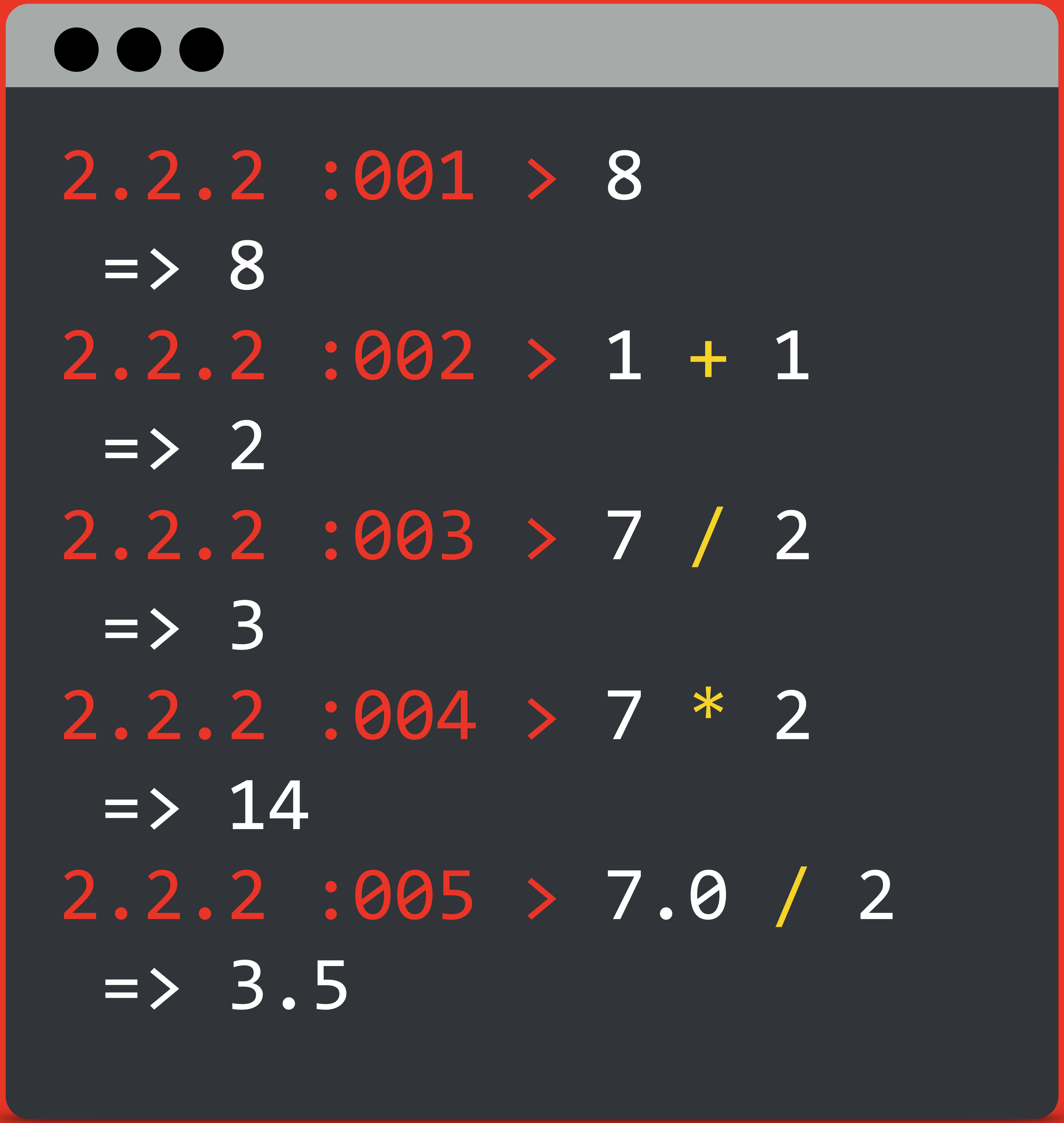


```
my-machine:~$ irb
2.2.2 :001 >
2.2.2 :002 > puts “Hello World!“
Hello World!
 => nil
2.2.2 :003 >
```

Working with Numbers

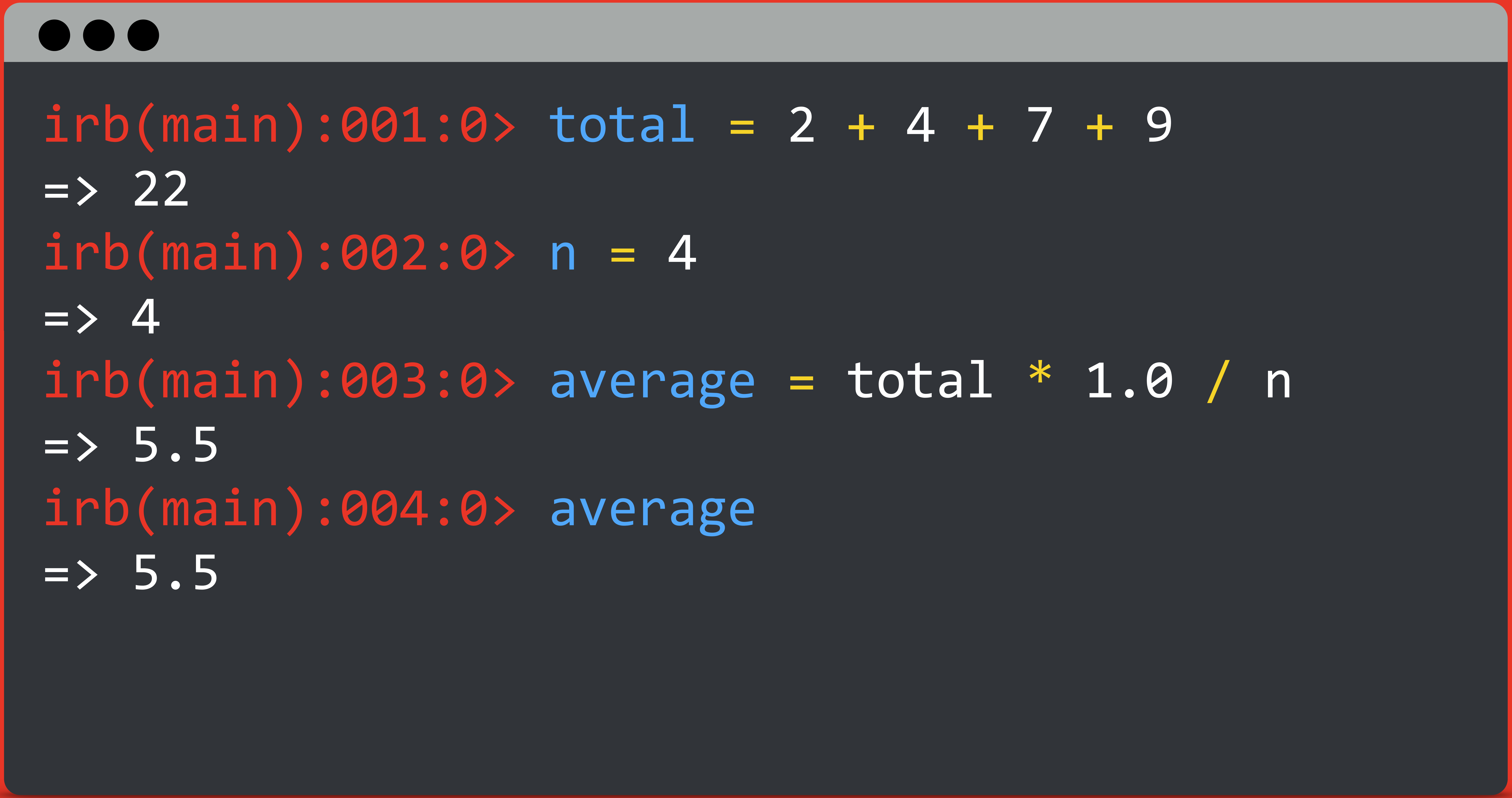
A Simple Introduction to Values, Operators and Expressions

Building Expressions



```
2.2.2 :001 > 8
 => 8
2.2.2 :002 > 1 + 1
 => 2
2.2.2 :003 > 7 / 2
 => 3
2.2.2 :004 > 7 * 2
 => 14
2.2.2 :005 > 7.0 / 2
 => 3.5
```

Variables: Naming Values

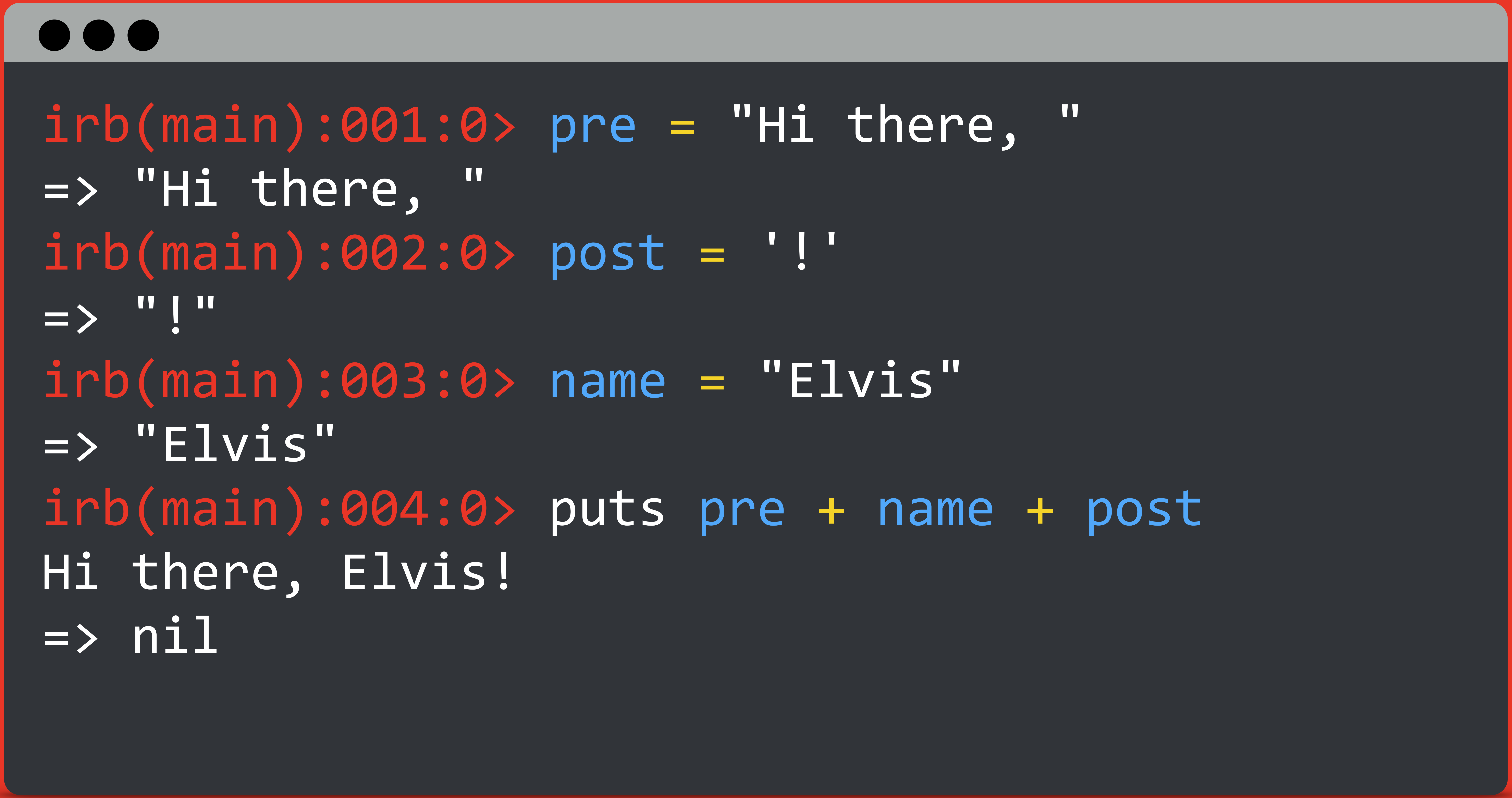


```
irb(main):001:0> total = 2 + 4 + 7 + 9
=> 22
irb(main):002:0> n = 4
=> 4
irb(main):003:0> average = total * 1.0 / n
=> 5.5
irb(main):004:0> average
=> 5.5
```

Working with Text

Strings, as we programmers call them

Strings go between quotes



```
irb(main):001:0> pre = "Hi there, "
=> "Hi there, "
irb(main):002:0> post = '!'
=> "!"
irb(main):003:0> name = "Elvis"
=> "Elvis"
irb(main):004:0> puts pre + name + post
Hi there, Elvis!
=> nil
```

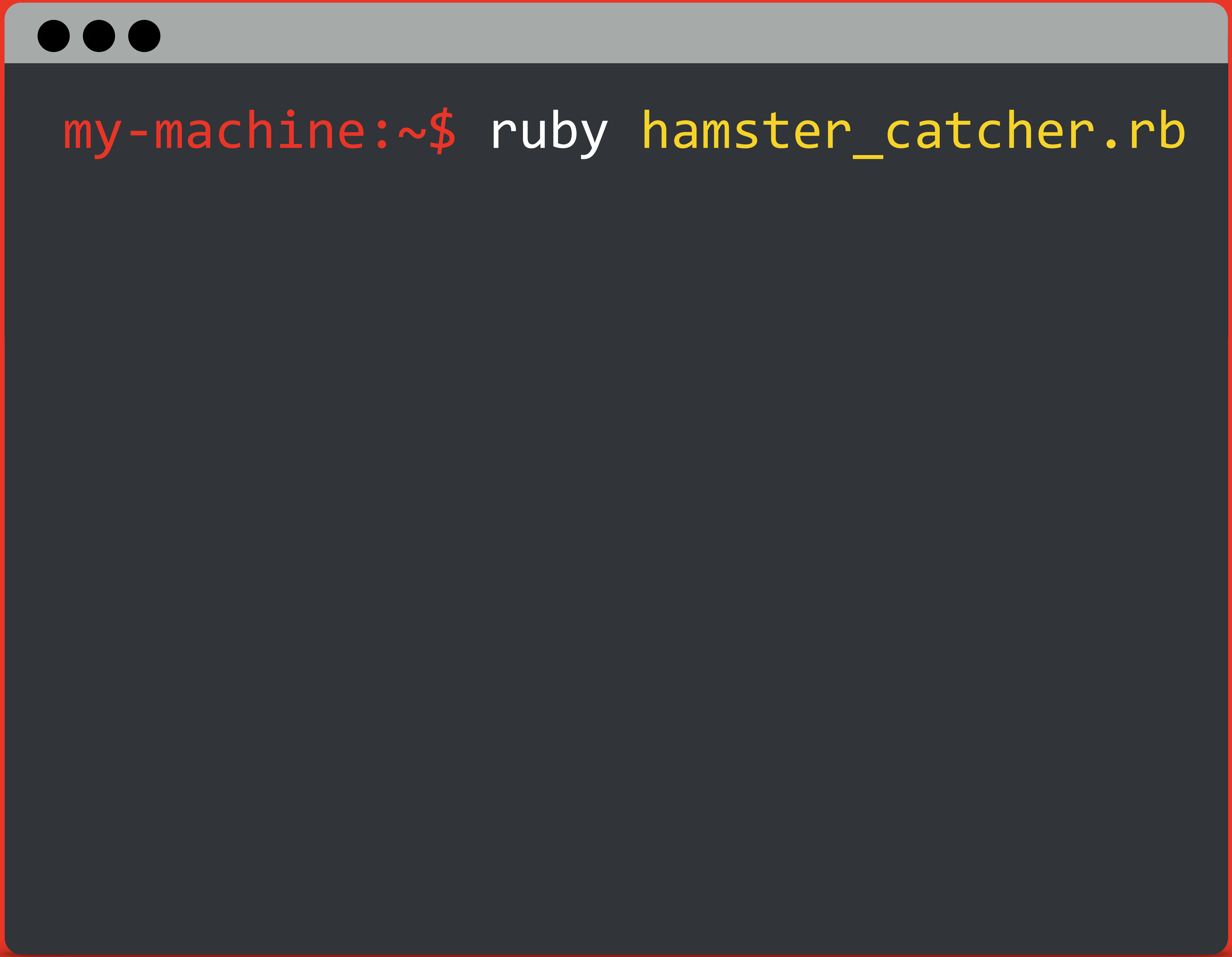
Input/output

- Print a value with puts
- Read a value with gets.chomp

Ruby Files

```
>_
```

Ruby Files



```
my-machine:~$ ruby hamster_catcher.rb
```

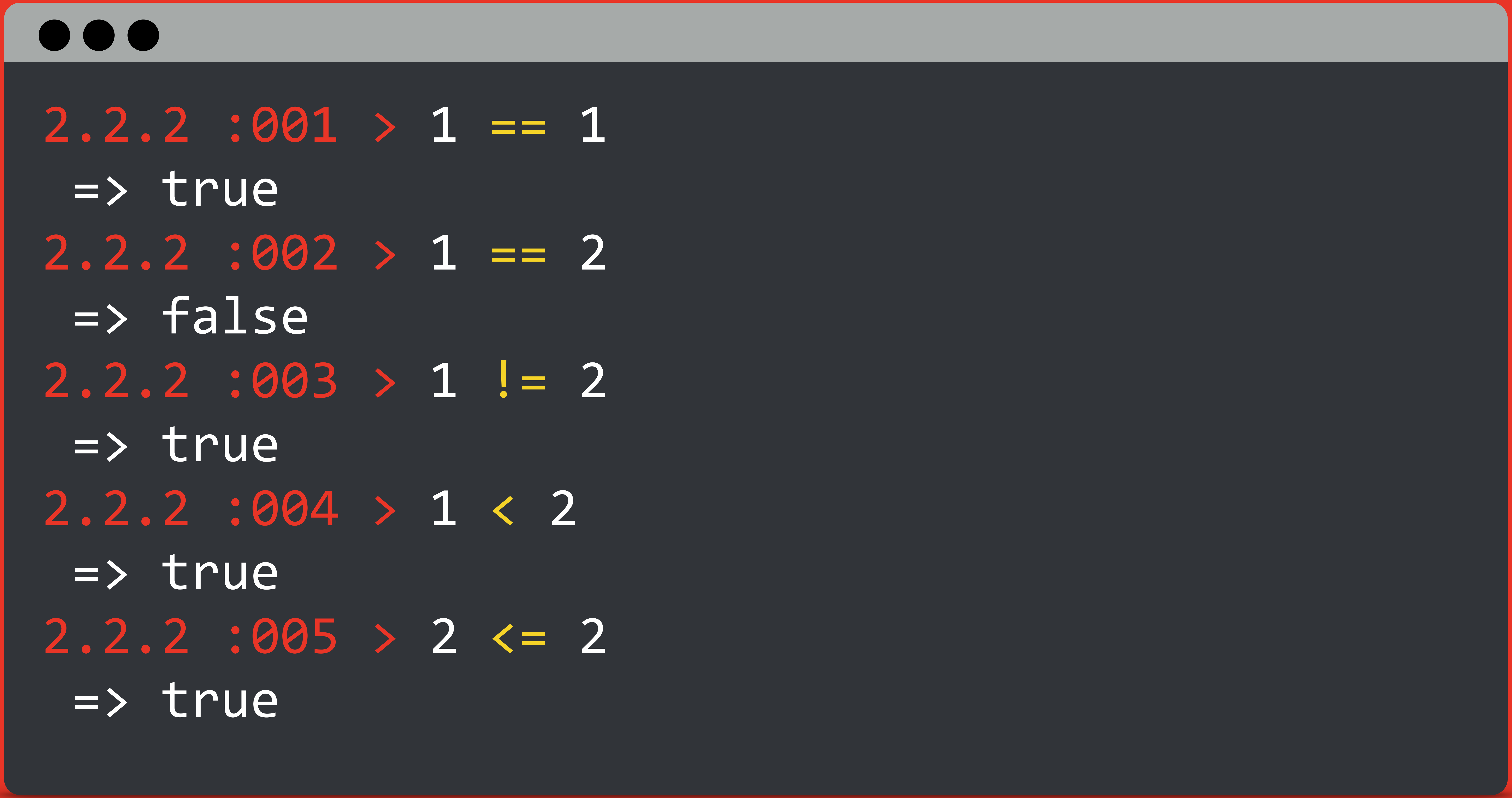
If/Else

>_

Conditional Execution

```
if test
    # do something
else
    # do something else
end
```

Operators: comparison



```
2.2.2 :001 > 1 == 1
 => true
2.2.2 :002 > 1 == 2
 => false
2.2.2 :003 > 1 != 2
 => true
2.2.2 :004 > 1 < 2
 => true
2.2.2 :005 > 2 <= 2
 => true
```

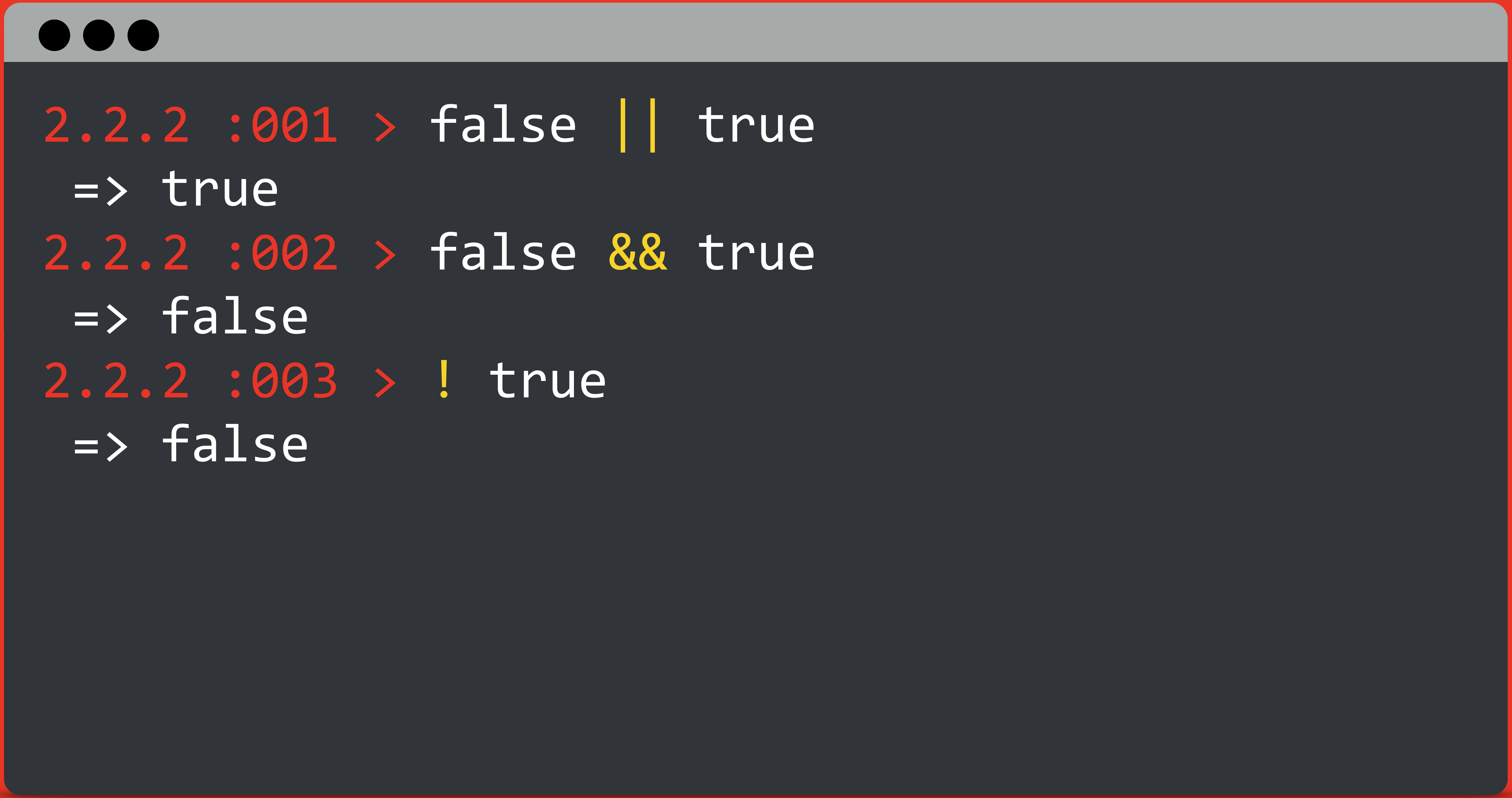
Getting Rid of Nested Ifs

```
if gender == "m"
    puts "meneer"
else
    if gender == "v"
        puts "mevrouw"
    else
        puts "??"
    end
end
```

Getting Rid of Nested Ifs

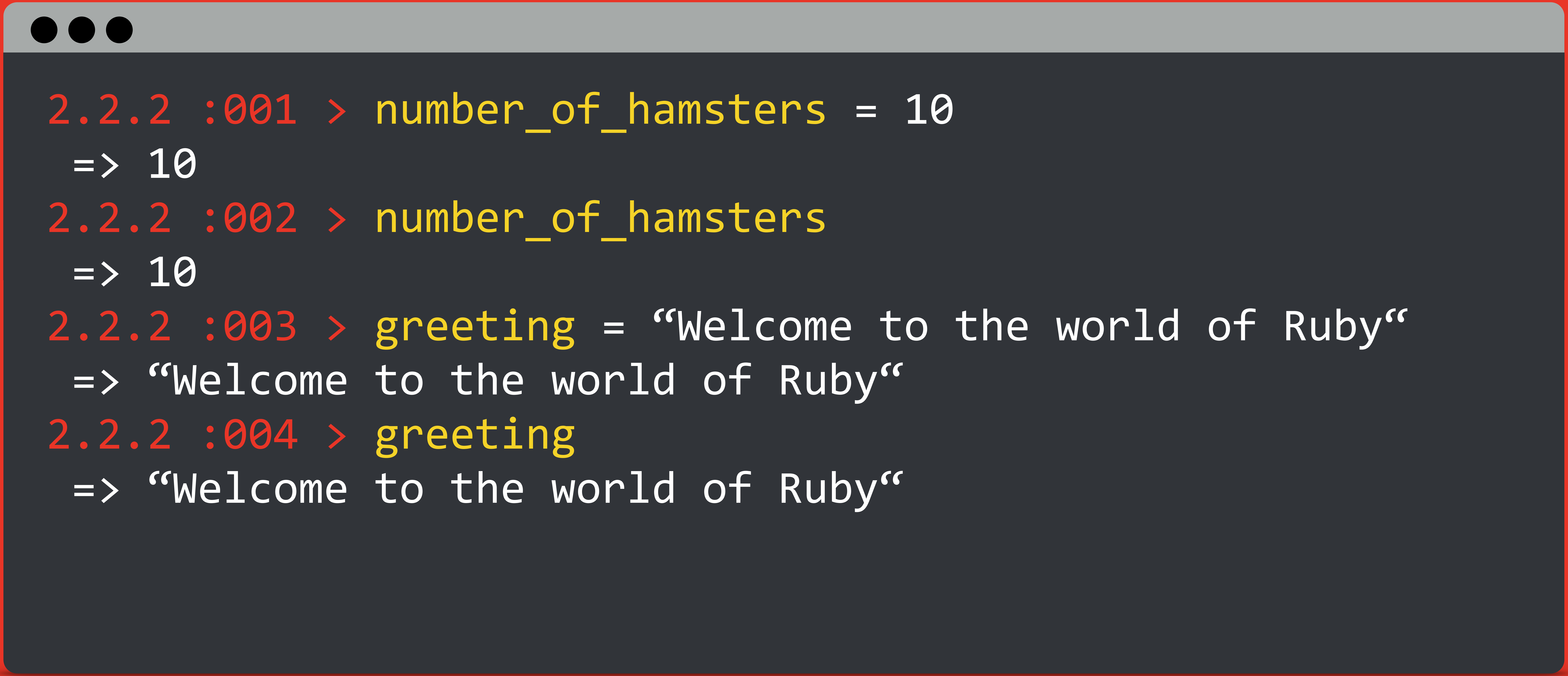
```
if gender == "m"
    puts "meneer"
elsif gender == "v"
    puts "mevrouw"
else
    puts "??"
end
```

Operators: logic



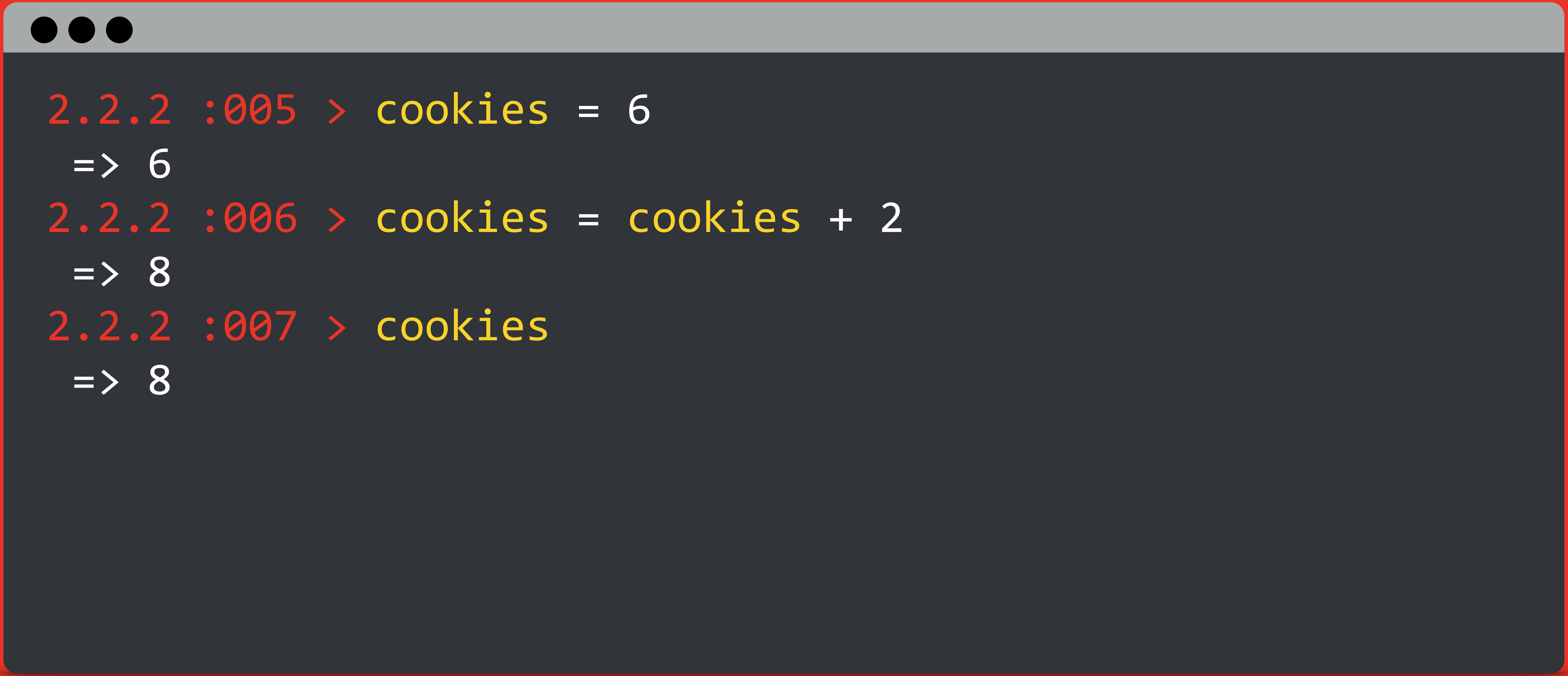
```
2.2.2 :001 > false || true
 => true
2.2.2 :002 > false && true
 => false
2.2.2 :003 > ! true
 => false
```

Operators: assignment



```
2.2.2 :001 > number_of_hamsters = 10
 => 10
2.2.2 :002 > number_of_hamsters
 => 10
2.2.2 :003 > greeting = “Welcome to the world of Ruby“
 => “Welcome to the world of Ruby“
2.2.2 :004 > greeting
 => “Welcome to the world of Ruby“
```

Operators: Updating

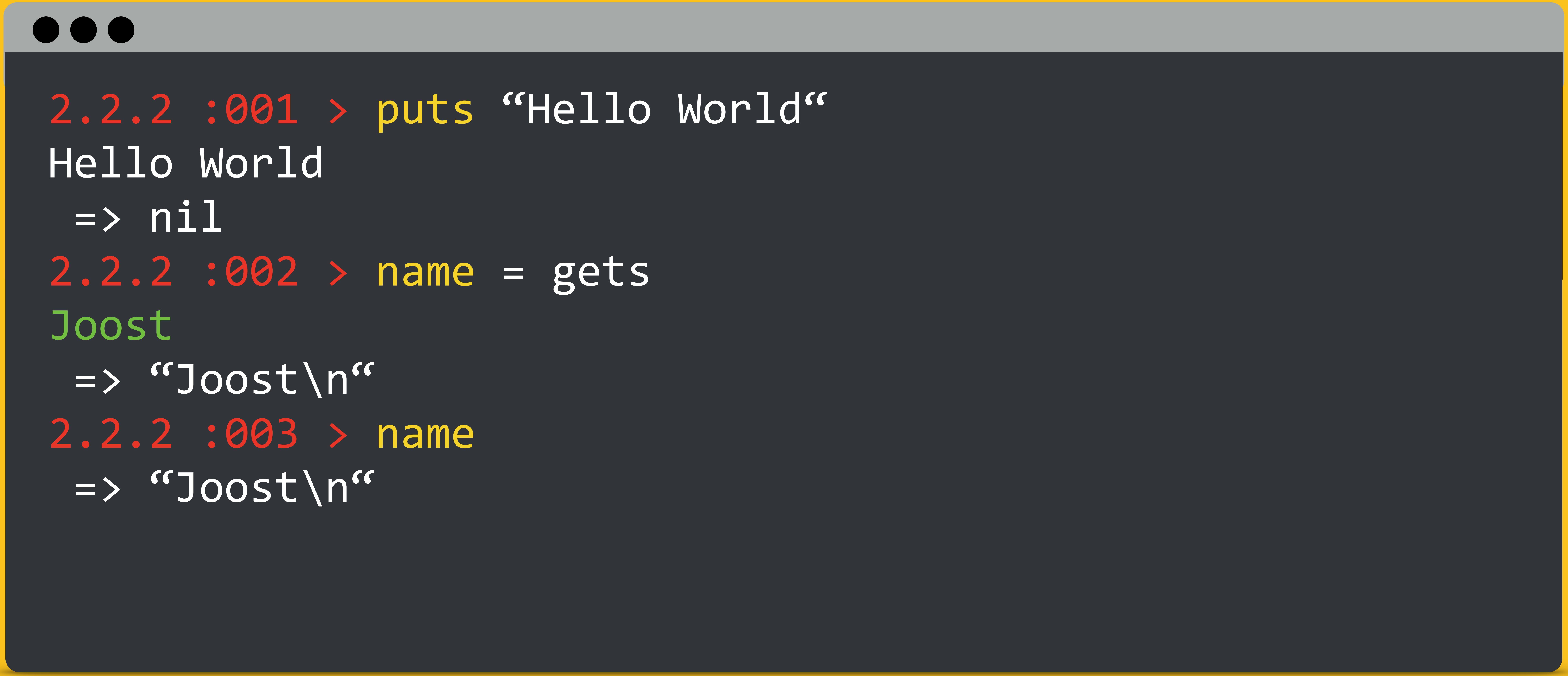


```
2.2.2 :005 > cookies = 6
 => 6
2.2.2 :006 > cookies = cookies + 2
 => 8
2.2.2 :007 > cookies
 => 8
```

Methods

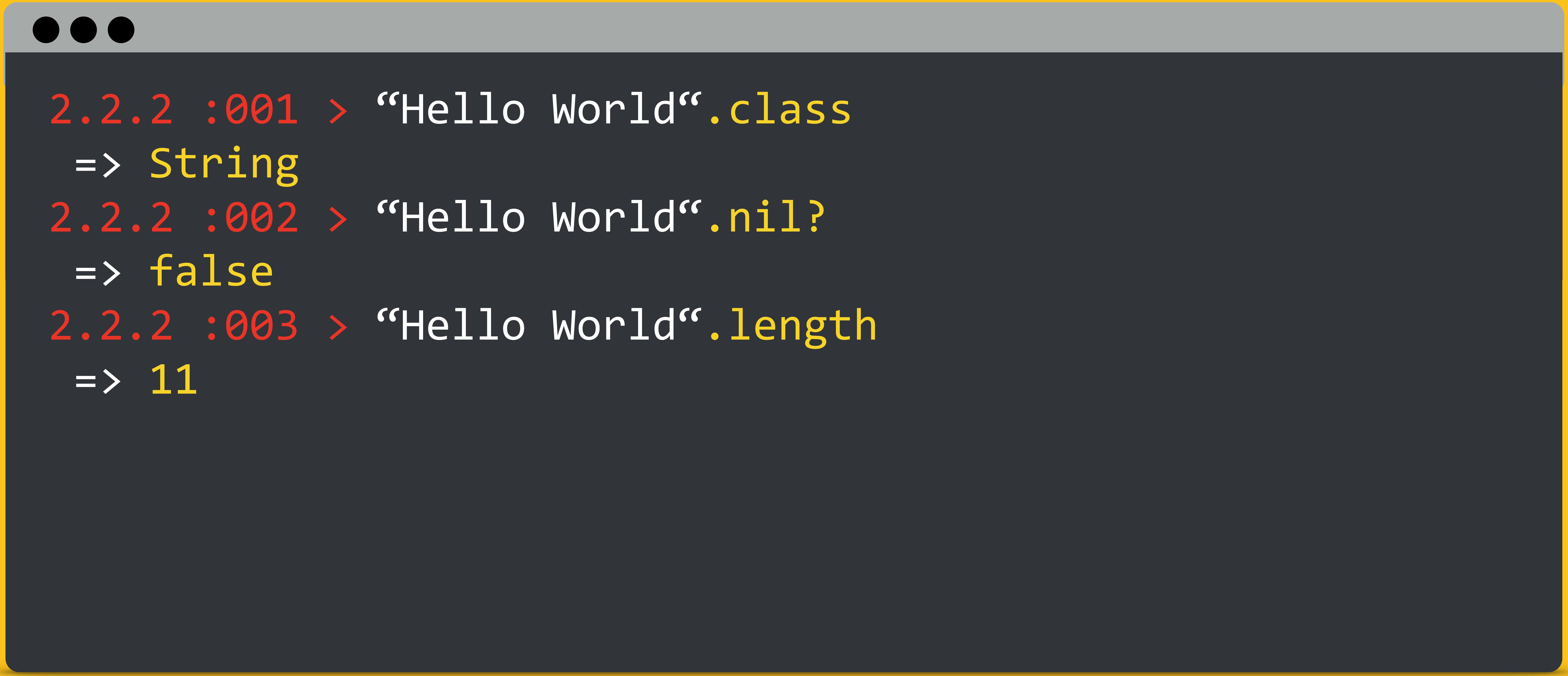
>_

Methods: Global



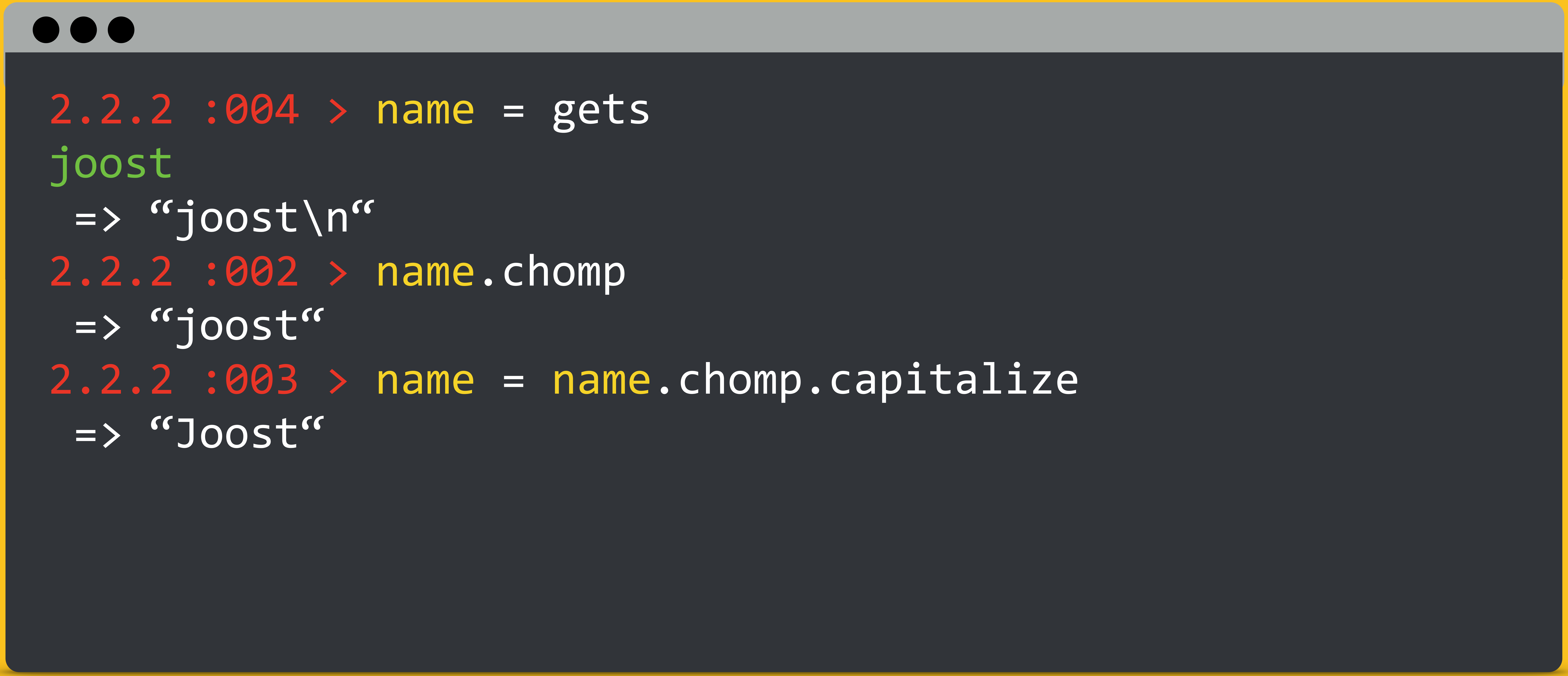
```
2.2.2 :001 > puts “Hello World“
Hello World
 => nil
2.2.2 :002 > name = gets
Joost
 => “Joost\n“
2.2.2 :003 > name
 => “Joost\n“
```

Methods: Object



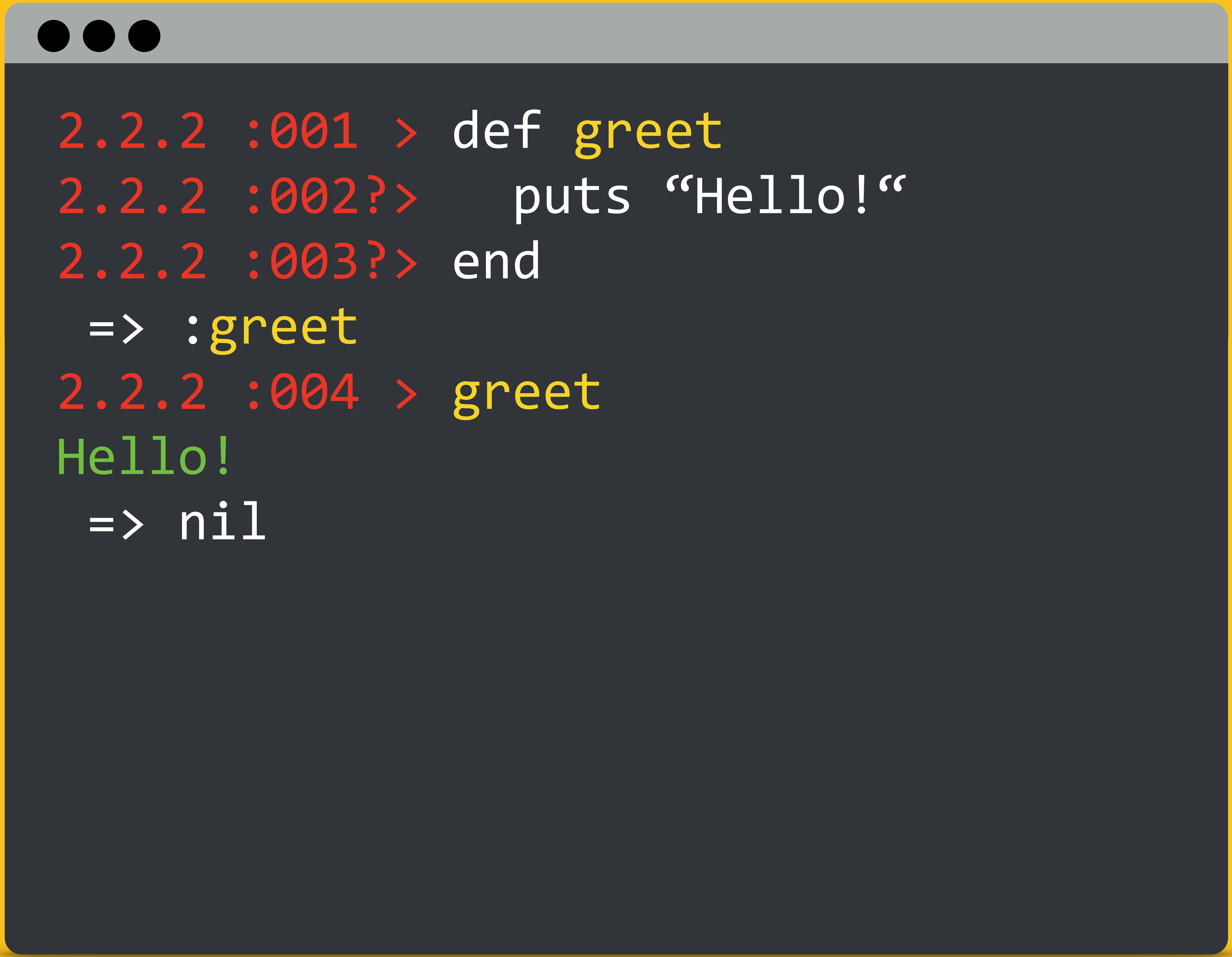
```
2.2.2 :001 > “Hello World“.class
 => String
2.2.2 :002 > “Hello World“.nil?
 => false
2.2.2 :003 > “Hello World“.length
 => 11
```

Methods: Object



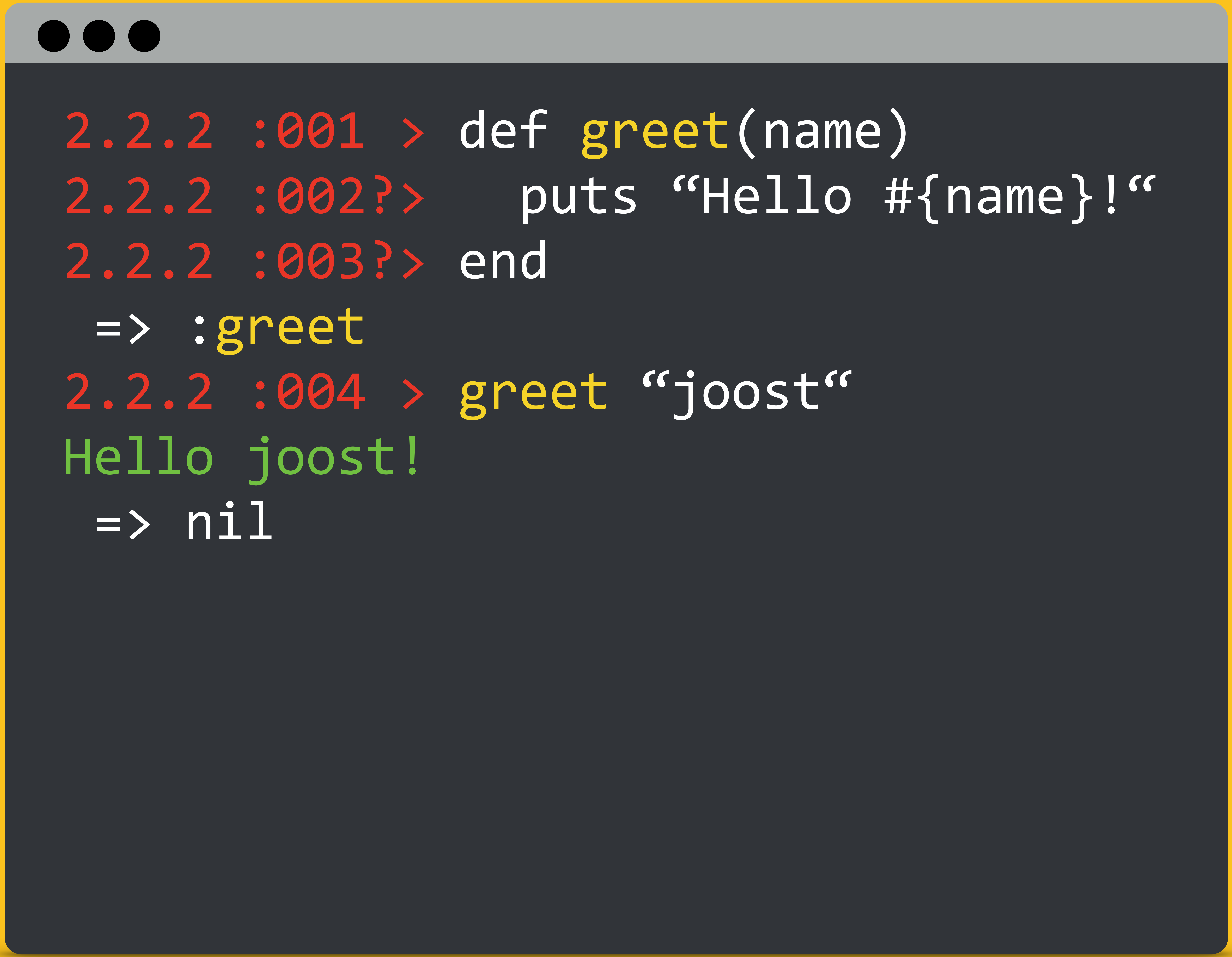
```
2.2.2 :004 > name = gets
joost
 => “joost\n“
2.2.2 :002 > name.chomp
 => “joost“
2.2.2 :003 > name = name.chomp.capitalize
 => “Joost“
```

Methods: custom



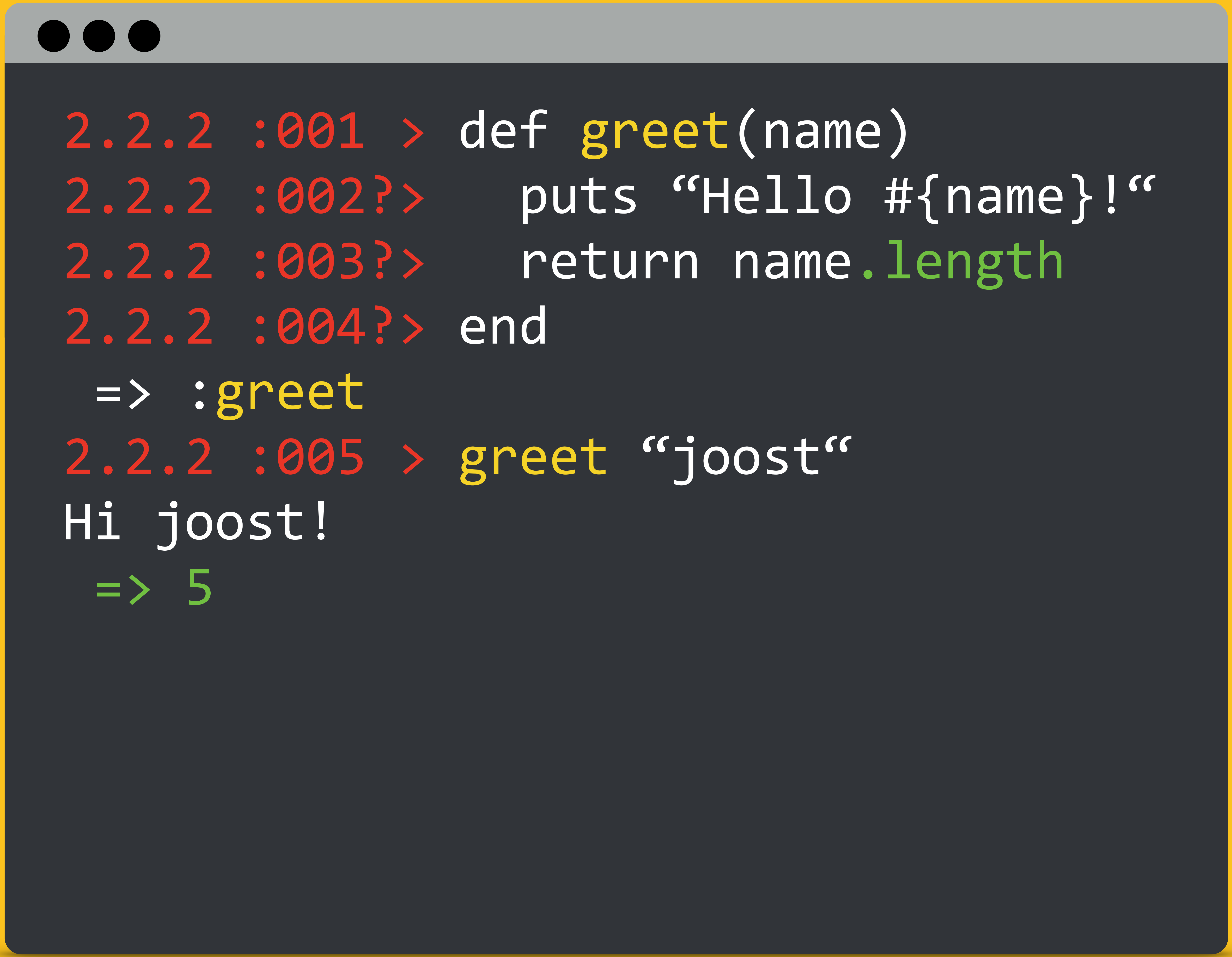
```
2.2.2 :001 > def greet
2.2.2 :002?>   puts “Hello!“
2.2.2 :003?> end
 => :greet
2.2.2 :004 > greet
Hello!
 => nil
```

Methods: arguments



```
2.2.2 :001 > def greet(name)
2.2.2 :002?>   puts “Hello #{name}!“
2.2.2 :003?> end
 => :greet
2.2.2 :004 > greet “joost“
Hello joost!
 => nil
```

Methods: return



```
2.2.2 :001 > def greet(name)
2.2.2 :002?>   puts “Hello #{name}!“
2.2.2 :003?>   return name.length
2.2.2 :004?> end
 => :greet
2.2.2 :005 > greet “joost“
Hi joost!
 => 5
```

Revisiting The Game

>_ Showcase: a little guessing game

Workshop

>_

Your Application

- A simple store
- List your products
- Allow the user to select a product
- Show the properties/prices for a product