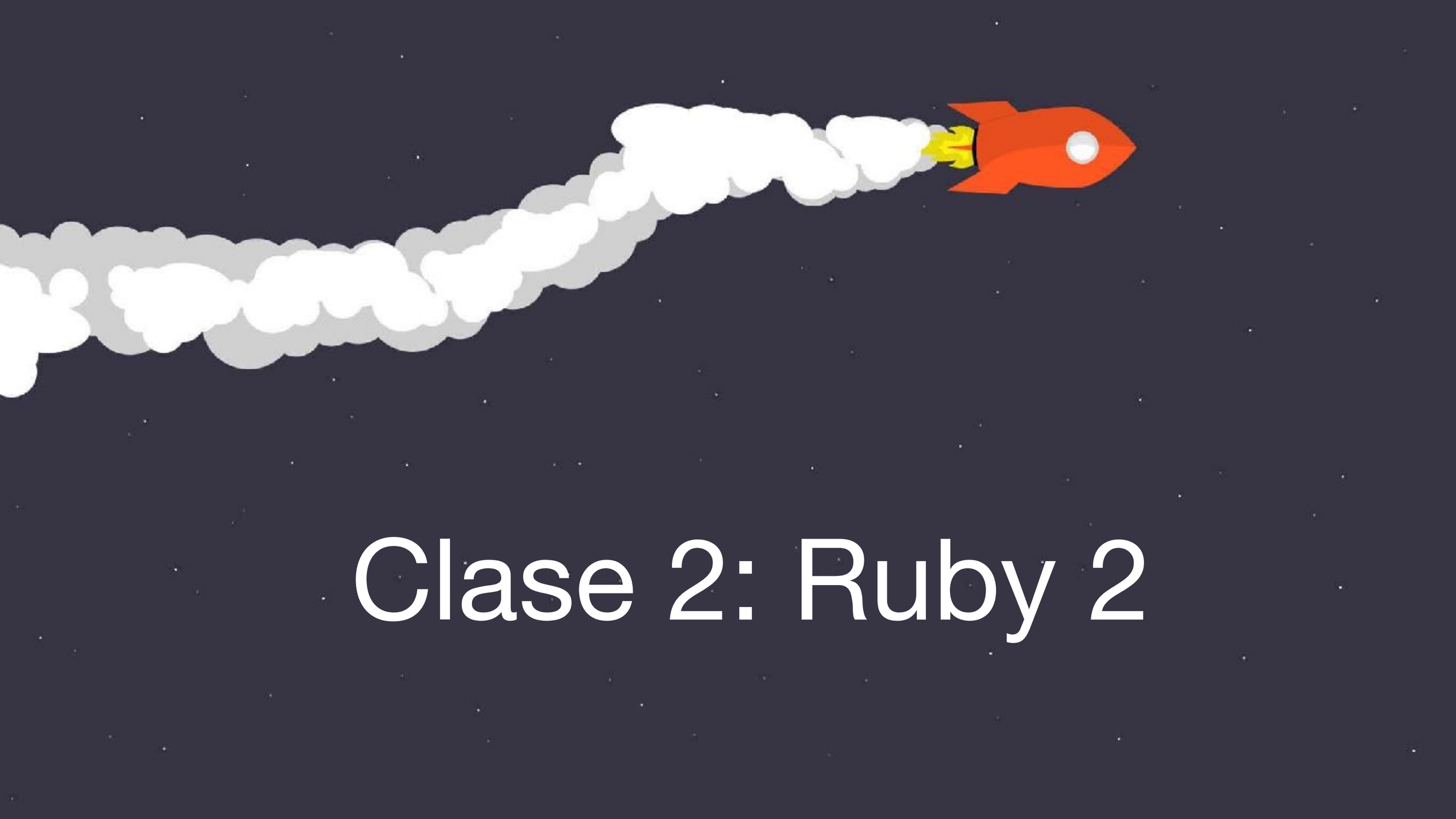


Clase 2: Ruby 2

Clase 2: Contenido

- Recapitular clase anterior: ¿Qué es Ruby?, IRB, strings, números, conversiones
- Más de Ruby
  - ➢ Variables: profundización
  - ➢ Métodos
  - ➢ If
- Mini tarea

Launchpad

Recapitulando la Clase 1

- ¿Qué es Ruby?
- IRB
- Strings
- Números
- Conversiones (.to_s, etc.)

Launchpad

Más sobre variables (1/2)

- Crear un archivo imperial_to_metric_height.rb
- Ingresar el siguiente código:

```
my_name = 'John Smith'

height_inches = 60

weight_pounds = 120

height_centimeters = height_inches * 2.54

weight_kilograms = weight_pounds * 0.453592

puts my_name + ' is ' + height_centimeters.to_s + ' cm and ' +
weight_kilograms.to_s + ' kg.'
```

Más sobre variables (2/2)

- Ejecuta el archivo en el Terminal
- Ahora haz el siguiente cambio y ejecuta el archivo

```
a = gets
puts a.to_i * 2.54
```

- La diferencia entre puts y gets es la siguiente:
  - Puts: Escribe caracteres en la pantalla (output)
  - Gets: Solicita caracteres desde la pantalla (input)

Launchpad

Ejercicio #1

Instrucciones:

- Edita height_to_centimeters.rb para que entregue el siguiente output en el Terminal

60 inches = 152.4 centimeters

Launchpad

Ejercicio #2

Instrucciones:

- Edita imperial_to_metric_height.rb para que le pregunte el nombre al usuario y tu output sea

Juan

is 152.4 cm and 54.431039999999996 kg.

- Evitar el salto de línea generado por gets utilizando .chomp

Launchpad

Métodos (1/2)

- Como se ha visto en esta clase, los strings pueden ser fácilmente alterados a través de lo que se conoce como métodos

- Los métodos, también conocidos como funciones en otros lenguajes de programación, son utilizados para empaquetar uno o más instrucciones repetibles en una unidad

- Todos estos son ejemplos de métodos:

puts

gets

.to_i

.to_s

.to_f

.chomp

Launchpad

Métodos (2/2)

- Es posible crear métodos utilizando la sintaxis def (inicio) y end (término). En imperial_to_metric_height.rb se vería:

Definición del método

```
def convert_inches_to_centimeters(number)
  height_centimeters = number * 2.54
  return height_centimeters
end

puts "What is your name?"

my_name = gets

my_name = my_name.chomp

puts "What is your height in inches?"
```

```
height_inches = gets.chomp.to_i

puts "What is your weight in pounds?"

weight_pounds = gets.chomp.to_f

height_centimeters =
convert_inches_to_centimeters(height_inches)

weight_kilograms = weight_pounds * 0.453592

puts my_name + ' is ' + height_centimeters.to_s + ' cm and ' +
weight_kilograms.to_s + ' kg.'
```

Llamada al método

Argumento ()

Launchpad

Ejercicio #3

Instrucciones:

- Crea un método que convierte el peso de pounds a kilogramos

Launchpad

Ejercicio #4

Instrucciones:

- Crea un programa que recibe un string y devuelve como output el string revertido

```
$ ruby reverse.rb Hello World
dlroW olleH
```

Launchpad

Ejercicio #5

Instrucciones:

- Escribe un programa que solicita un string y luego devuelve el largo del string

```
$ ruby string_length.rb
Enter a string: Four score and seven years
Your string is 26 characters long.
```

Launchpad

Control de flujo: “If”

- El “if” nos permite crear diferentes caminos para nuestro programa en base a condiciones que deben ser medidas en el minuto

- Ejemplo:

```
todays_temperature = 80

if todays_temperature > 50

puts “I’m going hiking!”

end
```

Launchpad

Operadores boolean

- Las expresiones condicionales dependen de operadores para comparar dos o más cosas. En el ejemplo anterior, “> a” es un comparador

- Hay distintos tipos de comparadores:

== equals to

!= not equal to

> greater than

>= greater than equal to

< less than

<= less than equal to

Launchpad

Ejercicio #6

Instrucciones:

- Prueba con distintos operadores en lugar de ”> a” en el programa anterior y ve qué ocurre

- Escribe un método going hiking que toma una variable, “temp” y retorna “Let’s go hiking” si temp>=50. Agrega otro método que retorne “That’s way too cold for hiking!” si es que la temperatura es bajo 50

Más sobre if

- Este es un ejemplo de respuesta del ejercicio #6 (parte 2)

```
temp = <value of your choice>

def going_hiking(temp)

  if temp >= 50

    puts "#{temp} degrees is perfect for hiking!"

      end

    if temp < 50

    puts "#{temp} degrees is WAY too cold for hiking!"

      end

end

puts going_hiking(temp)
```

Nota: “Interpolation” permite poner el valor de la variable dentro del string

Launchpad

Tarea #1 (1/2)

Instrucciones:

- Al dividir números en Ruby podríamos querer incluir el dividendo. Escribe un programa que pide dos integers, divide el primero por el segundo y retorna el resultado incluyendo el dividendo (no aceptar números no integers ni 0)

```
$ ruby divider.rb
    Enter an Integer: 6
    Enter an Integer: 4
    You said to calculate: 6 / 4
    The answer is 1 with a remainder of 2
```

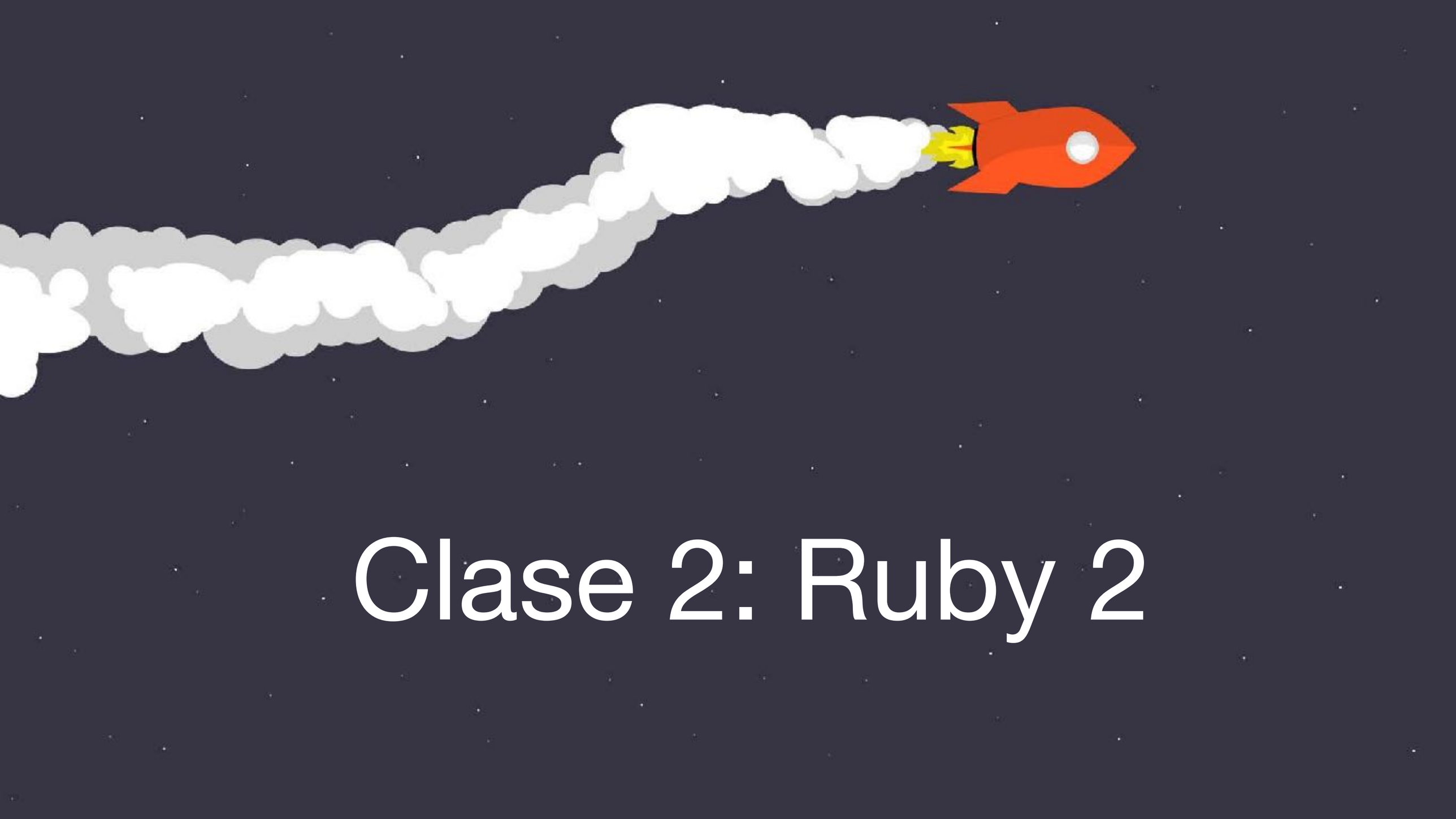
Launchpad

Tarea #1 (2/2)

Instrucciones:

- En el ejercicio going_hiking ve qué pasa si agregas múltiples “if”
- Agrega un if que devuelve “It is 23 degrees!” si es que temp es exactamente 23
- Haz que el programa sea interactivo y pregunte al usuario por la temperatura

Launchpad



Clase 2: Ruby 2