

Programming for **Everybody**

1. Intro to ruby



Let's talk about

*"A dynamic, open source programming language with a focus on **simplicity** and **productivity**. It has an **elegant syntax** that is natural to read and easy to write."*

Yukihiro "Matz" Matsumoto



Data Types: String

Text (words or sentences)
Defined with single or double quotes “ “
Case-sensitive

```
"wagon".class      # => String  
"wagon".upcase     # => "WAGON"  
"wagon".capitalize # => "Wagon"
```

>_



String interpolation

You can inject Ruby code into a string using interpolation
It only works with double-quoted strings

```
'two: #{1 + 1}'  
"two: #{1 + 1}"
```

```
# => "two: #{1 + 1}"  
# => "two: 2"
```



Data Types: Numbers

integers (whole numbers) and **floats** (decimal numbers)
shouldn't be wrapped in “ “

```
4.class
```

```
1 + 2
```

```
2 * 4
```

```
# => Integer
```

```
# => 3
```

```
# => 6
```

```
3.14.class
```

```
1.23 + 2.1
```

```
# => Float
```

```
# => 3.33
```

Maths

With Ruby we can perform the following math operations:

- **Addition +**
- **Subtraction -**
- **Multiplication ***
- **Division /**
- **Exponentiation ****
- **Modulo %**

>_



Data Types: Boolean

true or false
shouldn't be wrapped in “ “

```
1 == 1           # => true  
"hello" == "joanna" # => false
```

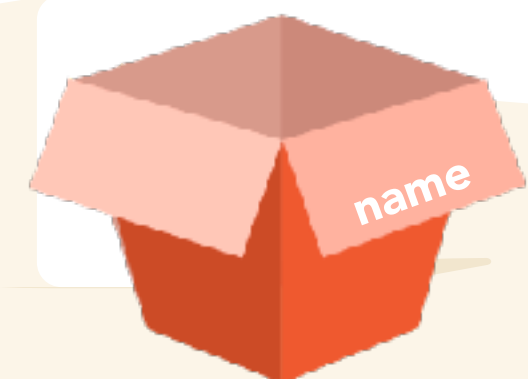


Variables

A **container** to which we assign a certain content
(string, number, boolean)

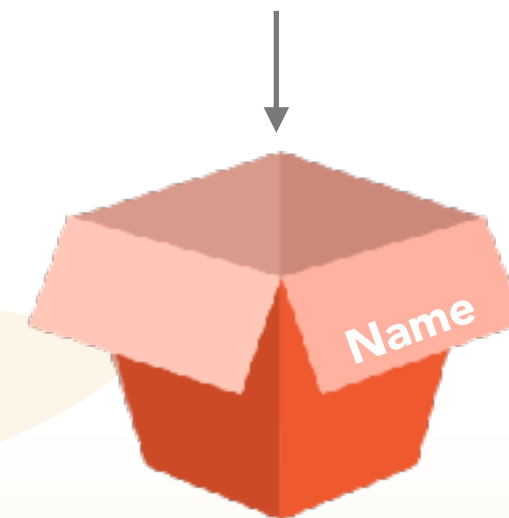
We set it and it remains there, ready for future use.
Variables can be overwritten and incremented

1. declaring the
variable

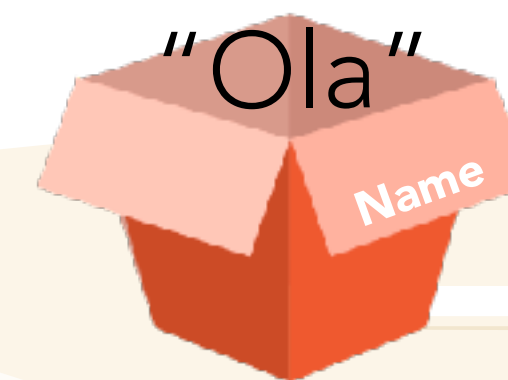


2. assigning a
value

"Ola"



3. Using the
variable



Variables

Naming a variable: always use **snake case** = small letters with words separated by an underscore (ex: my_name)

Assigning a value to a variable: we use the equals sign (ex: my_name = "Joanna")



Puts & Prints

Two commands we use to print info to the console

print

prints to the console whatever you give it

puts

prints to the console whatever you give it
and adds a new blank line below

The background features a stylized illustration of a laptop on the right and a terminal window on the left. The terminal window has a white background with a yellow prompt character '>_' and is set against a light yellow, wavy background that resembles a city skyline or abstract shapes.

>_

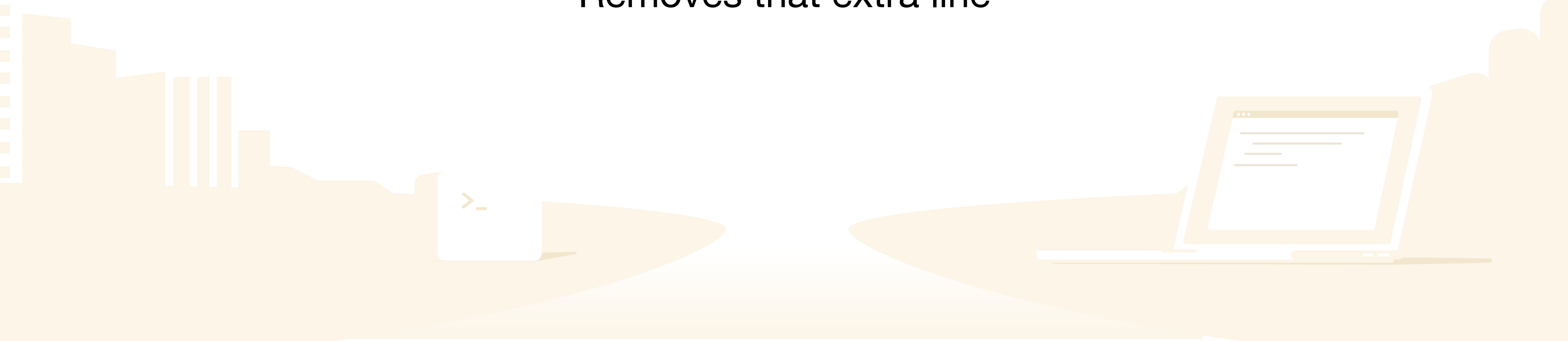
Gets and chomp

gets

Gets input from the user and adds a blank line after each input

chomp

Removes that extra line



Build-in methods

“Special behaviour” Ruby objects have

We *call* methods with a dot and the method name
(ex: `'hello'.upcase`)

The exclamation point after the method name changes the
content of the original variable for good
(ex: `'hello'.upcase!`)



Build-in methods

<https://ruby-doc.org/core/String.html>

<https://ruby-doc.org/core/Integer.html>

<https://ruby-doc.org/core/Float.html>



Commenting in Ruby

SINGLE LINE COMMENTS

#I'm a single line comment

MULTIPLE LINE COMMENTS

=begin

I'm a multiple line
comment

=end



Thank you!

