

Pre-Ada Live

In classroom conversation for day 1

Rename yourself in Zoom with

- Preferred Name
- Pronouns

Hello

Introduce yourself!

- Name
- Pronouns
- What you were doing before Ada **OR** something interesting about yourself

About Me - Chris McAnally

He/Him

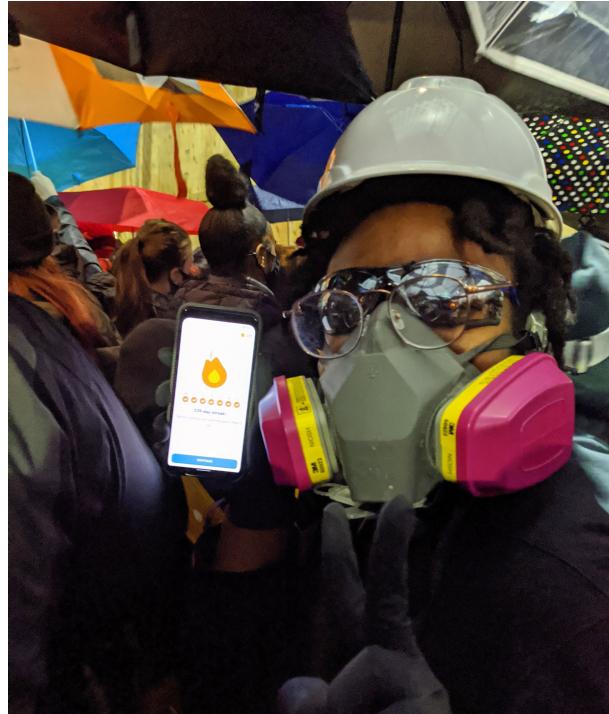
- Pronounced: Kris (rhymes with kiss)
 - I teach at Ada CS Fundamentals and pretty much anything needed
 - Former IT Administrator/Teacher
 - Cyclist & Gamer
 - I lived overseas for about 10 years
 - 大家好！



About Me - Ariana Bray

She/Her

- Pronounced: Ah-Ree-Ah-Nuh
(rhymes with nothing)
 - I'm a Developer Programs Engineer at Google
 - Formerly Chris' favorite student
 - I'm learning how to roller skate!
 - At the moment I type this, I have a 301-day streak on Duolingo



Learning Goals

Today we will

- Get more familiar with Ada, instructors and classmates
- Gain some practice using VS Code
- Work on learning to write more readable code
- Review Ruby data types & assignment statements
- Practice writing & talking about Ruby code

Agenda

1. Welcomes (already done)
2. Learning Goals
3. Review Unit 2 Exercises
4. Getting Familiar with our editor
5. Ruby Style Guide
6. Ruby data types & assignment statements
7. Debrief

Peer code review: Unit 2 Exercises

Pointers for code review:

- Ruby style guides: snake_case, spaces, comments, indentation
- Are all minimal requirements followed and working?
- Are any additional, optional requirements coded and working?
- User experience
 - Are the prompts useful, educational, and sufficient for the user of the program who is unaware of the requirements?
 - What is the user enters information in an incorrect format? (case insensitive input, detailed error messages, prompt for re-entry)

Workflow

Goal

- Be comfortable writing and executing programs on your machine

Tools needed

- VS Code, or another text editor
- terminal, or another terminal emulator such as iTerm2
- ruby
 - What version do you have? Run `ruby --version` in a terminal
- irb

Turn This

A screenshot of the repl.it web interface. At the top, there's a navigation bar with icons for user profile, save, run, share, and new repl. Below the bar, the main area has a sidebar on the left with 'Files' and other icons. A file named 'main.rb' is selected. The code editor shows the following Ruby code:

```
1 puts("Hello, repl.it!")
```

To the right of the code editor is a terminal window displaying the output of running the code:

```
ruby 2.5.5p157 (2019-03-15 revision 67260) [x86_64-linux]
> hello, repl.it!
=> nil
>
```

Into This

A screenshot of a terminal window titled "e4yc@M-C02X5D19JG5J: ~/code/ada/js1/day1 (zsh)". The terminal shows the following command sequence:

```
cd code/ada/js1/day1
ls
hello.rb
ruby hello.rb
Hello, VS Code!
```

The terminal is running in a dark-themed environment. On the left, there is a vertical sidebar with various icons, likely from a code editor like VS Code. At the bottom of the terminal window, there is a status bar with the following information:

V master*+ 0 0 ▲ 0 Ln1, Col 24 Spaces: 4 UTF-8 LF Ruby

Lets Try VS Code & Terminal

On your computer

- Start up Terminal
- Create a folder called if you do not already have it with:
 -
- Change into that folder with:
 -
- Then create a JSL folder
 -
 -
- Then create a day1 folder and move into that folder.
 -
 -
- Create a ruby file and open it in VS Code:
 -
 -

Lets Create This File

The image shows a terminal window with a light gray background. In the top left corner, there is a red icon followed by the text "hello.rb" and a red "x". To the right of this is a large, semi-transparent gray rectangular box. Below the window title, the path "Users > e4yc > code > ada > jsl > day1 >" is displayed, followed by another red icon and the text "hello.rb". Inside the large gray box, two lines of code are shown: "1 puts("Hello, Ada!")" and "2". Below the terminal window, the output "Hello, Ada!" is displayed in a black box with a purple border.

```
hello.rb ×  
Users > e4yc > code > ada > jsl > day1 > hello.rb  
1 puts("Hello, Ada!")  
2  
Hello, Ada!
```

Ruby Style

Goal

- To write Ruby code that is easy to understand and maintain.
- Tools Needed
 - [Ruby Style guide https://github.com/rubocop-hq/ruby-style-guide](https://github.com/rubocop-hq/ruby-style-guide)
- Mention of Rubocop

These Two Programs Do The Same Thing

What would you do to improve'em?

```
1
2
3 puts('What time of day is it?')
4 x = gets.chomp# get the user input
5 if x == "morning"
6   puts("Good morning!")
7 elsif x == "afternoon"
8   puts("Afternoon")
9 elsif x == "evening"
10  puts 'Evenin\''
11 else
12   puts "I see, thank you."
13
14 end|
```

```
1
2
3 puts('What time of day is it?')
4
5 timeofday = gets.chomp()# get the user input
6
7 if (timeofday == "morning")
8   puts("Good morning!" )
9
10 elsif (timeofday == "afternoon")
11   puts "Afternoon"
12 elsif (timeofday == "evening")
13   puts "Evenin\""
14 else
15   puts "I see, thank you."
16
17 end|
```

A Good Code Style Sample

```
1  # Asks the User what time of day it is, and then greets
2  # them appropriately, responding graciously if you have
3  # no idea what they just said.
4
5  puts "What time of day is it?"
6
7  time_of_day = gets.chomp
8
9  ↵ if time_of_day == "morning"
10   | puts "Good morning!"
11  ↵ elsif time_of_day == "afternoon"
12   | puts "afternoon"
13  ↵ elsif time_of_day == "evening"
14   | puts "evening"
15  ↵ else
16   | puts "I see, think you."
17  end
```

Ruby Data Types

Data Type	Description	Example
String	Anything surrounded by single/double quotes	"hello", 'hello'
Integer	A number with no decimal	1, 0, -7
Float	A "real" value (decimal)	0.5, -2.75
Array	An ordered list of values	['hey', 3, 7]
Symbol	A named object	:apple
Hash	A set of key-value pairs	{ A: 2, B: 3, C: 27 }
Range	A set of values with a beginning & end	(2..6) or (2...7)

Question How do you convert from one to another?

Useful tools, irb & the method.

Lets do something!

Goal:

- Practice writing a program which uses keyboard input.

Exercise:

- Create a program which calculates the square footage of a room given the user enters the dimensions of the room
- Use

Example Solution

```
1 puts "That's a nice room you're standing in."
2 print "How long is it, in feet? "
3 length = gets.chomp.to_i
4
5 print "How wide is it, in feet? "
6 width = gets.chomp.to_i
7
8 area = length * width
9 puts "Only #{area} square feet? That's Seattle for you"
10
```

Example Solution

```
1 puts "That's a nice room you're standing in."
2 print "How long is it, in feet? "
3 length = gets.chomp.to_i
4
5 print "How wide is it, in feet? "
6 width = gets.chomp.to_i
7
8 area = length * width
9 puts "Only #{area} square feet? That's Seattle for you"
10
```

```
~/code/ada/jsl/day1 ➤ ruby area.rb
That's a nice room you're standing in.
How long is it, in feet? 8
How wide is it, in feet? 12
Only 96 square feet? That's Seattle for you
```

Strings Practice

- Create a program that makes only one variable (a string) and prints it out to the user.
 - The string must span multiple lines when printed
 - It should include a contraction (don't, can't, I'm, etc..)
 - It must also include a user-supplied string within it

Strings Practice

- Create a program that makes only one variable (a string) and prints it out to the user.
 - The string must span multiple lines when printed
 - It should include a contraction (don't, can't, I'm, etc..)
 - It must also include a user-supplied string within it

```
the cart before the horse
~/code/ada/jsl/day1 ➔ ruby cart.rb
What should never go before a horse? apples
Don't put

apples before the horse
```

Is there another solution?

```
1  print 'What should never go before a horse? '
2  noun = gets.chomp
3
4  sentence = "Don't put"
5  sentence += "\n\n"
6  sentence += noun
7  sentence += " before the horse"
8
9  puts sentence
10
```

Another Solution

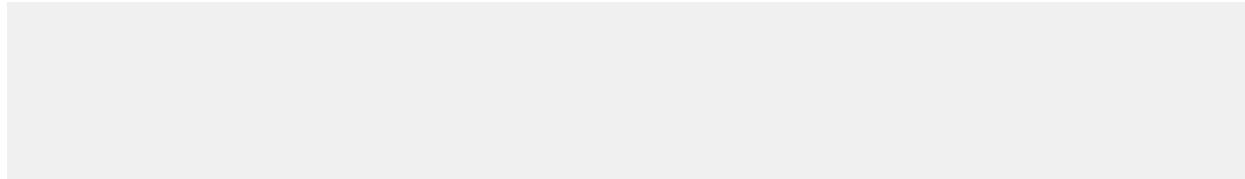
```
1 print 'What should never go before a horse? '
2 noun = gets.chomp
3
4 sentence = "Don't put\n\n#{noun} before the horse"
5 puts sentence
6
```

Assignment - When you give a set a variable to a value

```
x = 5  
y = x  
y = y * 2  
x += 3 # Equivalent to x = x + 3  
puts x # what does this print?
```

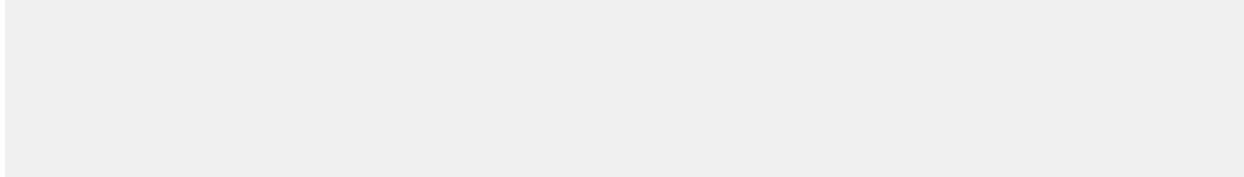
Assignment - When you give a set a variable to a value

Output



Why???

Output



Why???

Pass by reference...

Debrief - Breakout Rooms

- What new things did you learn today?
- What topics are you still struggling with?
- What did you especially enjoy about today's class?
- What can be improved for future classes?

Give us feedback

[Give us feedback](#)