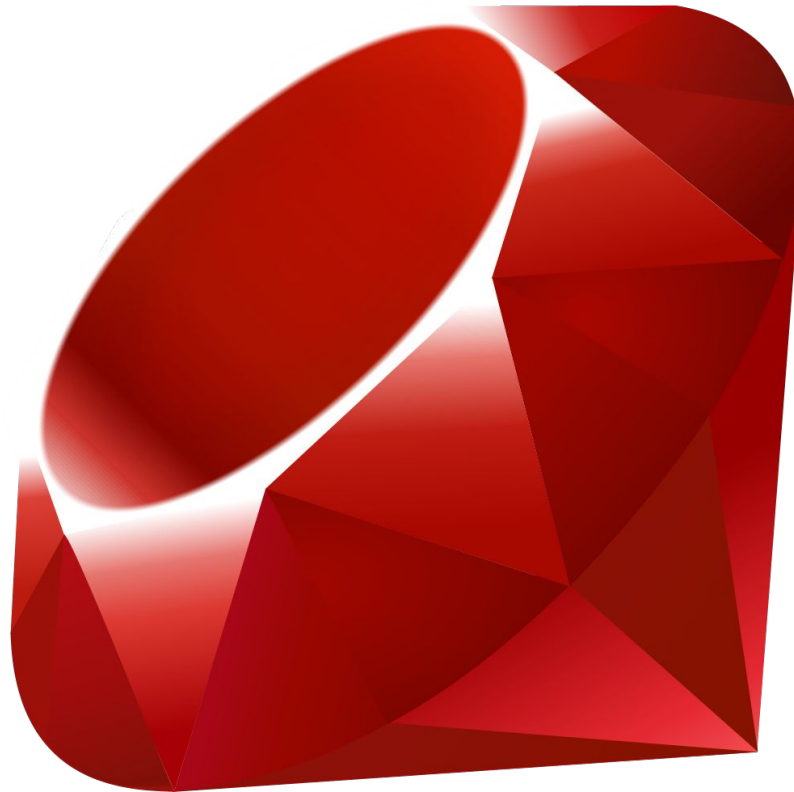




sheCodesNow

Introduction to Ruby

Presented by: Allison Northrop | agnorthrop@gmail.com | @AllisonNorthrop



What is Ruby

- Dynamic (executes many behaviors, not static), reflective (modify at runtime), object-oriented (everything is an object)
- Created by Yukihiro “Matz” Matsumoto
- Created with people in mind (very readable and fast to type!)
- Developed in the mid 1990's
- Open source
- Often used for rapid development (a lot of startups are fans)



Quotes by Matz

- “Often people, especially computer engineers, focus on the machines. They think, ‘By doing this, the machine will run faster. By doing this, the machine will run more effectively. By doing this, the machine will something something something.’ They are focusing on machines. But in fact we need to focus on humans, on how humans care about doing programming or operating the application of the machines.”
- “Trying to make Ruby natural, not simple.”
- “Ruby is simple in appearance but is very complex inside, just like our human body.”



Ruby in Real Life

Airbnb

Ask.fm

Bleacher Report

Bloomberg

Couchsurfing

Github

Goodreads

Groupon

Hulu

Amazon



GROUPON®

hulu

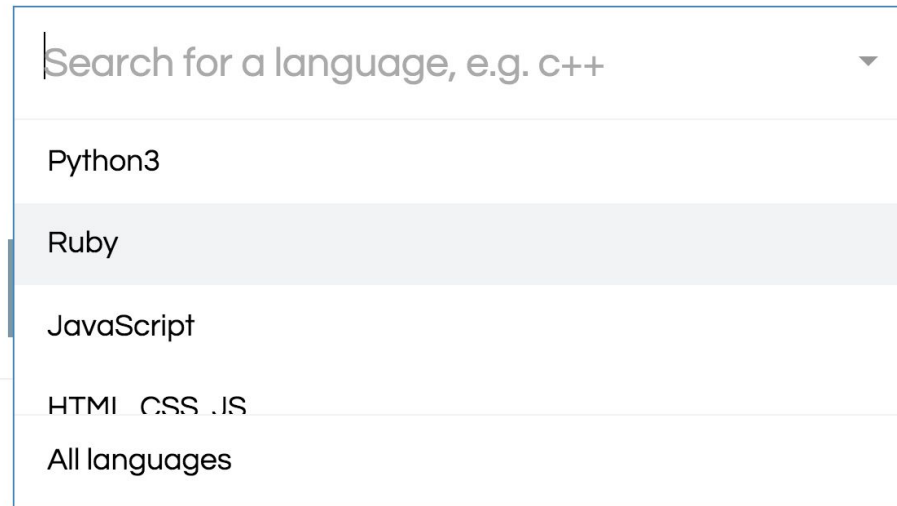


airbnb



Setup the Editor

1. Open your favorite browser (Chrome, Firefox, Safari etc)
2. Type repl.it in address bar and hit enter
3. On the page, select Ruby



A screenshot of a language selection dropdown menu. The search bar at the top contains the text "Search for a language, e.g. c++". Below the search bar, a list of languages is displayed: "Python3", "Ruby", "JavaScript", "HTML CSS JS", and "All languages". The "Ruby" option is highlighted with a light gray background.

Search for a language, e.g. c++
Python3
Ruby
JavaScript
HTML CSS JS
All languages



Hello World!

- print does what it sounds like it does -- It prints out to the terminal (output)

```
1  #1  
2  print "Hello World"
```



Hello World!

- print does what it sounds like it does -- It prints out to the terminal (output)

Try it!

```
1  #1  
2  print "Hello World"
```



Hello World!

- print does what it sounds like it does -- It prints out to the terminal (output)

```
1 #1  
2 print "Hello World"
```

This # is a comment. Anything after a # on a line won't be executed by the computer

Anything in quotes in ruby is considered a string (think words)



Variable declaration

`x = 8.543`

`y = 6`

`bananas = "Monkeys are going bananas"`

`hip_hip = "Hooray!"`

`numbers = [1, 2, 3, 4, 9, 2, 1]`



Variable declaration

`x = 8.543` ← A number with a decimal is called a **float**

`y = 6`

`bananas = "Monkeys are going bananas"`

`hip_hip = "Hooray!"`

`numbers = [1, 2, 3, 4, 9, 2, 1]`



Variable declaration

`x = 8.543` ← A number with a decimal is called a **float**

`y = 6` ← A number without a decimal is called a **integer or fixnum**

`bananas = "Monkeys are going bananas"`

`hip_hip = "Hooray!"`

`numbers = [1, 2, 3, 4, 9, 2, 1]`



Variable declaration

`x = 8.543` ← A number with a decimal is called a **float**

`y = 6` ← A number without a decimal is called a **integer or fixnum**

`bananas = "Monkeys are going bananas"`

`hip_hip = "Hooray!"`

`numbers = [1, 2, 3, 4, 9, 2, 1]` ← This is a ruby **array**. It's a way of storing multiple pieces of information. An array can have floats, integers, strings or a combination of the three



Assign “Hello World” to a variable

Variable

```
4 #2 assign a variable so we can re-use this phrase without typing it out every time  
5 hi_world = "Hello World!"
```



Now print the “Hello World” variable

Try it!

```
4  #2 assign a variable so we can re-use this phrase without typing it out every time  
5  hi_world = "Hello World!"
```



Print vs Puts

```
7  print hi_world  
8  puts hi_world
```



Print vs Puts

Try it!

```
7  print hi_world  
8  puts hi_world
```



String Concatenation

- Joining multiple strings together.
- Side note: The + sign is used to join strings together or add floats or integers.

```
11  #3 string concatenation  
12  print hi_world + " " + "My name is Allison"
```



String Concatenation

- Joining multiple strings together.
- Side note: The + sign is used to join strings together or add floats or integers.

Try it!

```
11 #3 string concatenation  
12 print hi_world + " " + "My name is Allison"
```



Strings Concatenation

- What else could we do to this to make it better/cleaner?

```
15 hi_world = "Hello World!"  
16 name = "Allison"  
17  
18 print hi_world + " " + "My name is" + name
```



Strings Concatenation

- What else could we do to this to make it better/cleaner?

Try it!

```
15 hi_world = "Hello World!"  
16 name = "Allison"  
17  
18 print hi_world + " " + "My name is" + name
```



Ruby command: gets.chomp

- gets.chomp **assigns** whatever the user **inputs** into the terminal to a variable

```
21  #4 assigning a variable to gets.chomp
22  hi_world = "Hello World! My name is "
23  print "What is your name?"
24  name = gets.chomp
25  print hi_world + name
```



Ruby command: gets.chomp

- gets.chomp **assigns** whatever the user **inputs** into the terminal to a variable

Try it!

```
21  #4 assigning a variable to gets.chomp
22  hi_world = "Hello World! My name is "
23  print "What is your name?"
24  name = gets.chomp
25  print hi_world + name
```



If statement

- If statements **executes** (runs) the code only if the condition/statement is **True**
- In Ruby, they are always ended by an **end** statement

```
28  #5 if statement
29  if 5 > 4
30    puts "5 is greater than 4"
31  end
```

Condition that uses an operator to return a boolean value (True or false)



If, else statement

- If else statements execute the code in the if block, if the condition is true. Otherwise, it executes the else statement
- Another way to put it is: It checks the if statement condition first and ignores the else statement, if the if statement is true. Otherwise it does the else statement.

```
33  #6 if else statement
34  if 6 == 9
35  |   puts "Six is equal to 9?! What sort of world is this?"
36  else
37  |   puts "-- very funny, you know that six is not equal to nine"
38  end
```



If, elsif, else

What will this code print out?

```
40  #7 if elsif else statement
41  if 8 < 4
42  |   puts "Eight is less than four"
43  elsif 9 > 4
44  |   puts "Nine is greater than four"
45  else
46  |   puts "Numbers do not compute"
```



Stolen from Wikipedia!

```
if condition then
    --statements
elseif condition then
    -- more statements
elseif condition then
    -- more statements;
...
else
    -- other statements;
end if;
```



Conditionals using user input

- How could we use `gets.chomp` to create a multiple choice question and answer program?
- Here's something that can get you started:

Try it!

```
50 print "What is larger? The moon, the sun or earth?"
51 puts "a. moon"
52 puts "b. sun"
53 puts "c. earth"
54 answer = gets.chomp
55
56 if answer == |
```



```
1 puts "What is larger? The earth's moon, the sun or the earth?"
2 puts "a. moon"
3 puts "b. sun"
4 puts "c. earth"
5 answer = gets.chomp
6
7 if answer == "a"
8   puts "Yikes, no."
9 elsif answer == "b"
10  puts "Phew, I'm glad you got that right."
11 elsif answer == "c"
12  puts "Umm no."
13 else
14  puts "I don't understand :("
15 end
16 |
```



Edge Cases

- An Edge Case is when something happens that might not initially be expected (example bad data, a user leaving a field blank on a form etc)
- Can you think of edge cases in our multiple choice program?



Resources

<http://shecodesnow.com/>

<https://www.ruby-lang.org/en/>

<https://github.com/Ada-Developers-Academy/textbook-curriculum/tree/master/01-ruby-fundamentals>

<http://ruby-for-beginners.rubymonstas.org/index.html>

<https://www.codewars.com/?language=ruby>

Thank You!

Allison Northrop | agnorthrop@gmail.com |
@AllisonNorthrop

<https://www.meetup.com/She-Codes-Now-Seattle/>

We have monthly workshops!