

INTRO TO RUBY

stella lee 

1. download ruby

windows: rubyinstaller.org

mac: ruby-lang.org (installed by default)

2. download a text editor

sublime text: sublimetext.com



about me



developer @ shopify



Shopify powers ambitious entrepreneurs all over the world

600,000

BUSINESSES POWERED BY SHOPIFY

1,000,000+

ACTIVE USERS

\$55B+

SOLD ON SHOPIFY

I am seriously happy about running our store on Shopify. Powerful features and easy to set up, running an online store shouldn't be this much fun!



Tina Roth Eisenberg

Designer, Founder of Tattly

Shopify has helped me grow my business amazingly. It's been a pleasure because everything has been easier than I ever imagined.



Corrine Anestopoulos

Founder, BIKO Jewelry

The best thing about Shopify is that it's made me not have to worry about anything that has to do with ecommerce.



Chris Tsang

Founder, Mindzai Toys



Glossier.

COLOURPOP



KYLIE



adidas®



overview

1. introduction to ruby
2. data types && variables
3. methods && objects
4. conditionals && logic
5. loops
6. arrays
7. object-oriented programming

introduction to ruby

high level
interpreted
object-oriented
easy to use



ruby irl



GROUPON®



ACTIVITY

let's do things!



ACTIVITY

programming is hard.

1. ask your buddy
2. ask google
3. ask me

ACTIVITY

programming is hard.

1. ask your buddy
2. ask google
3. ask me

🤔 bored? look for this emoji for new challenges

📚 want to know more? look for this emoji for resources

ACTIVITY

- 1. open up IRB (Interactive Ruby)**
 - Mac: open Terminal > type `irb` > hit enter
 - Windows: Start Menu > Ruby > Interactive Ruby
- 2. type “Hello World” # => “Hello World”**
- 3. type `print “Hello World”` # Hello World=> nil**

data types && variables

my_number = 10	# => 10
my_boolean = true	# => true
my_string = "I'm learning Ruby!"	# => "I'm learning Ruby!"
my_nil = nil	# => nil
print my_number	# 10=> nil
print my_boolean	# true=> nil
print my_string	# I'm learning Ruby!=> nil
print my_nil	# => nil

- 🤔 what happens when you set a boolean with a capital letter? a string without quotes?
- 🤔 what happens when you call a variable that you haven't defined?

data types && variables

```
my_num = 10          # => 10  
my_num = 100         # => 100  
my_num = 10 * 3 - 4 # => 26
```

🤔 what happens when you try to set my_num as a boolean or a string?

comments

```
# anything after this hash will not execute
def hi
  "Hello, World!" # you can use comments to explain your code
end
```

🤔 what happens when you put a comment between quotations?

methods && objects

```
def learning_ruby?  
  true  
end
```

```
def greeting(name)  
  "Hello, #{name}!"  
end
```

```
def divide(dividend, divisor)  
  dividend / divisor  
end
```

learning_ruby?	# => true
greeting("Beyonce")	# => "Hello, Beyonce!"
divide(10, 5)	# => 2

🤔 what happens when you call a method before it is defined?

methods && objects

(almost) everything in ruby is an object

methods && objects

“I am a string”.class

=> String

“I am also a string”.class

=> String

methods && objects

“I am a string”.class

=> String

“I am also a string”.class

=> String

“I’m learning Ruby at Flawless Hacks”.length

=> 35

“Flawless Hacks”.reverse

=> “skcaH sselwalF”

“Ruby”.upcase

=> “RUBY”

“cool”.downcase

=> “cool”

“puppies”.nil?

=> false

🤔 why didn’t the word “cool” change in the last example?

🤔 what happens if you call .length on something that is not a string?

ACTIVITY

1. open up **Sublime Text** sublimetext.com
2. type the following in a new file:

```
print "Hello, there!"
```

3. save the file on your desktop and name it `hello.rb`

🤔 `puts` can also be used to print to the screen. how is it different from `print`?

ACTIVITY

1. open up **Sublime Text** sublimetext.com
2. type the following in a new file:

```
print "Hello, there!"
```

3. save the file on your desktop and name it `hello.rb`
4. open a new tab in your console (you should be in your user directory by default)
5. `cd Desktop` to navigate to your Desktop
6. `run hello.rb` to run your program



resources: [Don't Fear the Command Line](#) && [Codecademy Learn the Command Line](#)

ACTIVITY

1. open up **Sublime Text** sublimetext.com
2. type the following in a new file:

```
print "Hello, there!"
```

3. save the file on your desktop and name it `hello.rb`
4. open a new tab in your console (you should be in your user directory by default)
5. `cd Desktop` to navigate to your Desktop
6. `ruby hello.rb` to run your program



resources: [Don't Fear the Command Line](#) && [Codecademy Learn the Command Line](#)

ACTIVITY

challenge: make your program say hello to you!

instructions:

1. make your program print a prompt asking for your name
2. use the `gets` method to get user input from the console
3. save this value to a variable
4. print the greeting with your name to the screen
5. `ruby hello.rb` try running your program to test it!

ACTIVITY

challenge: make your program say hello to you!

instructions:

1. make your program print a prompt asking for your name
2. use the `gets` method to get user input from the console
3. save this value to a variable
4. print the greeting with your name to the screen
5. `ruby hello.rb` try running your program to test it!

```
print "What's your name? "
```

ACTIVITY

challenge: make your program say hello to you!

instructions:

1. make your program print a prompt asking for your name
2. use the `gets` method to get user input from the console
3. save this value to a variable
4. print the greeting with your name to the screen
5. `ruby hello.rb` try running your program to test it!

```
print "What's your name? "
name = gets
```

ACTIVITY

challenge: make your program say hello to you!

instructions:

1. make your program print a prompt asking for your name
2. use the `gets` method to get user input from the console
3. save this value to a variable
4. print the greeting with your name to the screen
5. `ruby hello.rb` try running your program to test it!

```
print "What's your name? "
name = gets
print "Sup, " + name + "?"
```

conditionals && logic

```
true      # => true  
false     # => false
```

conditionals && logic

true	# => true
false	# => false
1 < 1	# => false
1 >= 1	# => true
1 == 1	# => true
1 != 1	# => false

conditionals && logic

true	# => true
false	# => false
1 < 1	# => false
1 >= 1	# => true
1 == 1	# => true
1 != 1	# => false
(1 < 1) true	# => true
(1 < 1) && true	# => false

conditionals && logic

```
print "Enter an integer: "
user_input = Integer(gets.chomp)

if user_input < 0
  puts "Why so negative?"
end
```

🤔 what happens if you put a non-boolean value as the if statement expression?

conditionals && logic

```
print "Enter an integer: "
user_input = Integer(gets.chomp)
```

```
if user_input < 0
  puts "Why so negative?"
end
```

🤔 what happens if you put a non-boolean value as the if statement expression?

conditionals && logic

```
print "Enter an integer: "
user_input = Integer(gets.chomp)
```

```
if user_input < 0
  puts "Why so negative?"
elsif user_input > 0
  puts "Way to be positive!"
end
```

🤔 what happens if the user_input is 0?

conditionals && logic

```
print "Enter an integer: "
user_input = Integer(gets.chomp)
```

```
if user_input < 0
  puts "Why so negative?"
elsif user_input > 0
  puts "Way to be positive!"
else
  puts "Zero is cool."
end
```

🤔 what if there are more than 3 cases?

ACTIVITY

challenge: make your program yell at you if you don't enter valid input

instructions:

1. make your program check to see if the user entered something
2. if the user entered some characters, print the greeting as usual
3. if the user hasn't entered anything, print an error to the screen

hint: use the `gets.chomp` method to get user input from the console

🤔 why can't we use just `gets`?



ACTIVITY

challenge: make your program yell at you if you don't enter valid input

instructions:

- 1. make your program check to see if the user entered something**
- 2. if the user entered some characters, print the greeting as usual**
- 3. if the user hasn't entered anything, print an error to the screen**

```
print "What's your name? "
name = gets.chomp
if name.length == 0
  puts "That's not a name, dummy!"
else
  print "Sup, " + name + "?"
end
```

hint: use the `gets.chomp` method to get user input from the console

🤔 why can't we use just `gets`?

ACTIVITY

challenge: make your program ask you for your name again if you don't enter valid input

loops

```
i = 0
while i < 5      # don't run this code!
    print i      # if you do run this code
end              # press `ctrl + c`
```

loops

```
i = 0
while i < 5
    print i
    i = i+1
end                # 01234=> nil
```

loops

```
i = 0
while i < 5
    print i
    i = i+1
end
```

```
for num in 0..4
    print num
end
```

these will both print 01234
.. is used for inclusive range
... is used for exclusive range
if you use 0...4, you'll print 0123

ACTIVITY

challenge: make your program ask you for your name again if you don't enter valid input

instructions:

1. make your program check to see if the user entered something
2. if the user entered some characters, print the greeting as usual
3. if the user hasn't entered anything, print an error to the screen and ask for their name again

ACTIVITY

challenge: make your program ask you for your name again if you don't enter valid input

```
print "What's your name? "
name = gets.chomp

while name.length == 0
  puts "That's not a name, dummy!"
  print "What's your name? "
  name = gets.chomp
end

print "Sup, " + name + "?"
```

ACTIVITY

challenge: make your program ask you for your name again if you don't enter valid input

```
print "What's your name? "
name = gets.chomp

while name.length == 0
  puts "That's not a name, dummy!"
  print "What's your name? "          # redundant code 😞
  name = gets.chomp
end

print "Sup, " + name + "?"
```

ACTIVITY

challenge: make your program ask you for your name again if you don't enter valid input

```
name = ''  
initial_ask = true  
  
while name.length == 0  
  puts "That's not a name, dummy!" if !initial_ask  
  print "What's your name? "  
  name = gets.chomp  
  initial_ask = false  
end  
  
print "Sup, " + name + "?"
```

non-redundant code 😊

🤔 why can't we use nil instead of the empty string when we initialize name?

ACTIVITY

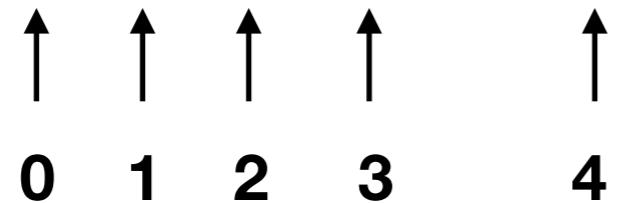
challenge: make your program use a new greeting every time you run it

arrays

```
empty_array = []
numbers = [1, 99, -2]
random = [8, 'hi', nil, true, numbers]      # => [8, "hi", nil, true, [1, 2]]
```

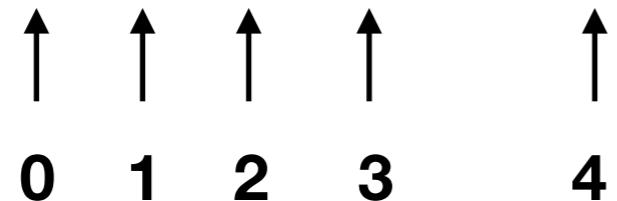
arrays

random = [8, 'hi', nil, true, numbers]



arrays

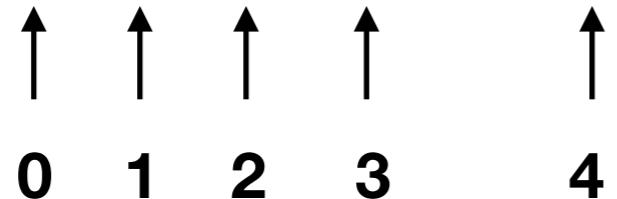
random = [8, ‘hi’, nil, true, numbers]



random[1] # => “hi”
random[4] # => [1, 2]

arrays

```
random = [8, 'hi', nil, true, numbers]
```



```
random.each { |elem| print elem }      # 8hitrue[1, 2]=> [8, "hi", nil, true, [1, 2]]
```

ACTIVITY

challenge: make your program use a new greeting every time you run it

hint: use the `.sample` Array method to choose a random element



ACTIVITY

challenge: make your program use a new greeting every time you run it

instructions:

- 1. create an array with different greetings**
- 2. choose a random array element and use that greeting instead of the standard one**

hint: use the `.sample` Array method to choose a random element



ACTIVITY

challenge: make your program use a new greeting every time you run it

```
name = ''  
initial_ask = true  
greetings = ["Hi ", "Sup ", "How you doin "]  
  
while name.length == 0  
  puts "That's not a name, dummy!" if !initial_ask  
  print "What's your name? "  
  name = gets.chomp  
  initial_ask = false  
end  
  
print greetings.sample + name + "?"
```

oop

“Flawless”.length



oop

```
class Person
end
```

```
Person.new
```

oop

```
class Person  
  def initialize  
  end  
end
```



oop

```
class Person
  def initialize(name, favourite_animal = nil)
    @name = name
    @favourite_animal = favourite_animal
  end

  def name
    @name
  end

  def greet
    "Hello, #{@name}!"
  end
end
```

oop

```
me = Person.new("Stella", "dog")
```

```
me.name
```

```
me.favourite_animal
```

```
me.greet
```

```
me = Person.new("Stella")
```

```
# => #<Person:0x007fc66c80fef0 @name="Stella", @favourite_animal="dog">
# => #<Person:0x007fc66c9187c0 @name="Stella", @favourite_animal=nil>
# => "Stella"
# NoMethodError: undefined method `favourite_animal' for #<Person:0x007fc66c80fef0>
# => "Hello, Stella"
```

oop

```
class Creature
  def initialize(name)
    @name = name
  end
end
```

```
class Dragon < Creature; end
```

ACTIVITY

challenges: make your program...

- **use proper punctuation for each greeting**
- **take multiple names and greet each person in a different way**
- **run a bunch of small methods instead of one big code block**
- **object-oriented!**



slack: @Stella Lee
email: stella.lee@shopify.com
linkedin: stellarxo