IRON
HACK

# Confronting Errors

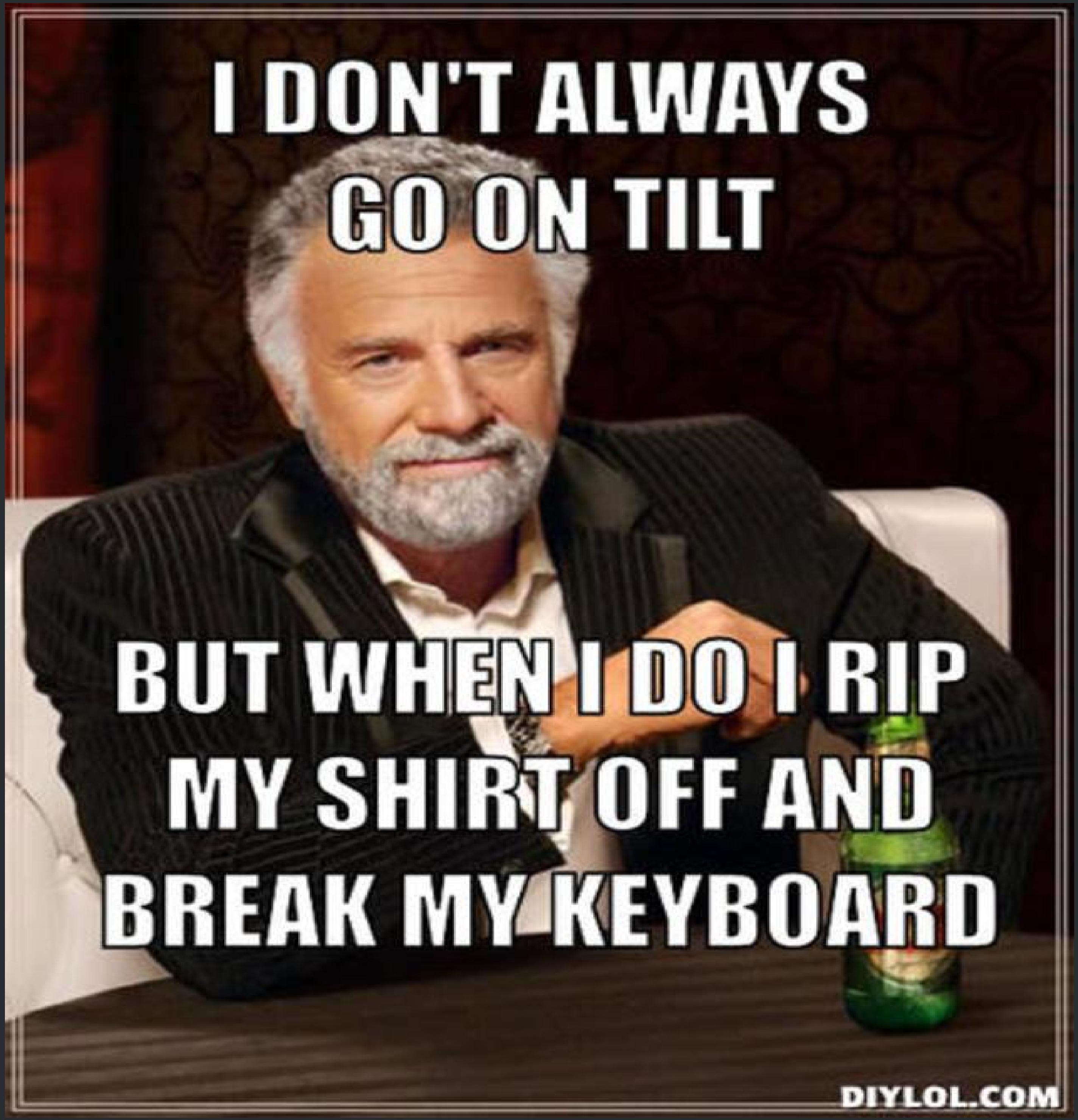
# Coding is hard

Sometimes we make mistakes.

Sometimes these mistakes cause errors

# So how do we fix these errors?

# Don’t go on tilt



Believe it or not errors are here to help us

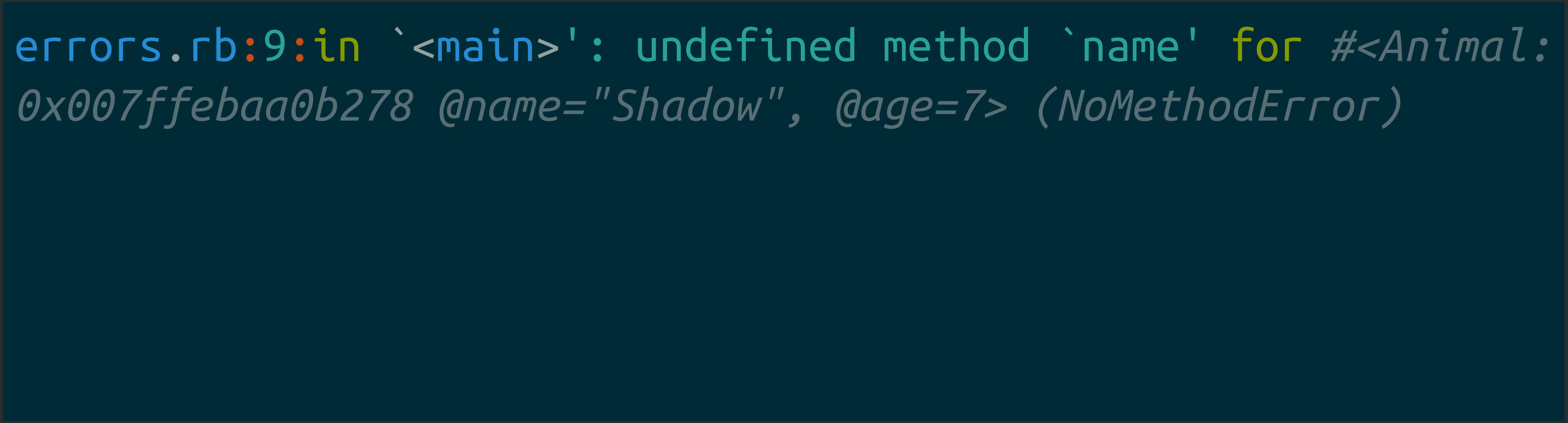
We should let the errors guide us to the correct answer

# Lets take a look at a couple of common errors and how to solve them

# No method error

```ruby
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end
end

dog = Animal.new("Shadow", 7)
puts dog.name
```

# No method error



```
errors.rb:9:in `<main>': undefined method `name' for #<Animal:
0x007ffebaa0b278 @name="Shadow", @age=7> (NoMethodError)
```

# Oh no, what to do?

# Look at the error first

# No method error

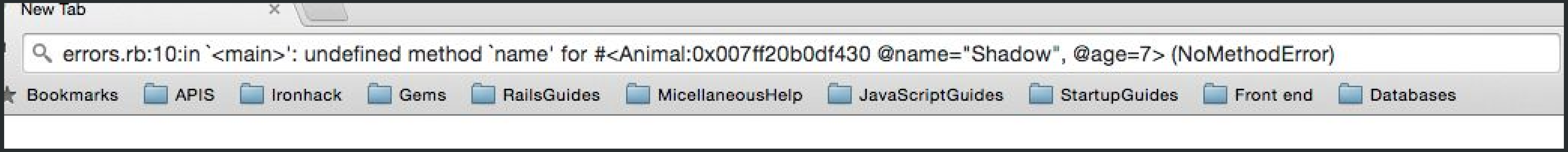
```ruby
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end
end

dog = Animal.new("Shadow", 7)
puts dog.name
```

Ruby thinks we’re trying to call a “name” method on our dog

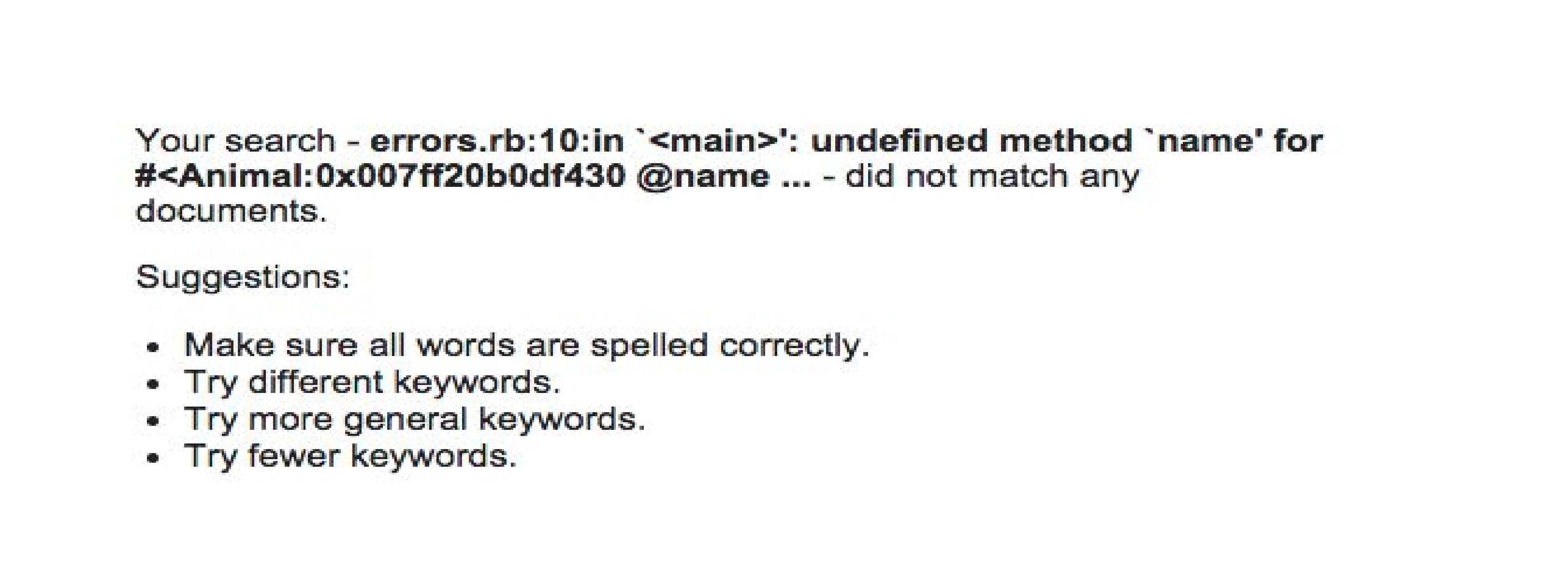
But our dog doesn’t have a name method!

There's a couple fixes for this, but how do we find them?

Let’s consult our old friend google.



```
errors.rb:10:in `<main>': undefined method `name' for #<Animal:0x007ff20b0df430 @name="Shadow", @age=7> (NoMethodError)
```



Your search - **errors.rb:10:in `<main>': undefined method `name' for #<Animal:0x007ff20b0df430 @name ...** - did not match any documents.

Suggestions:

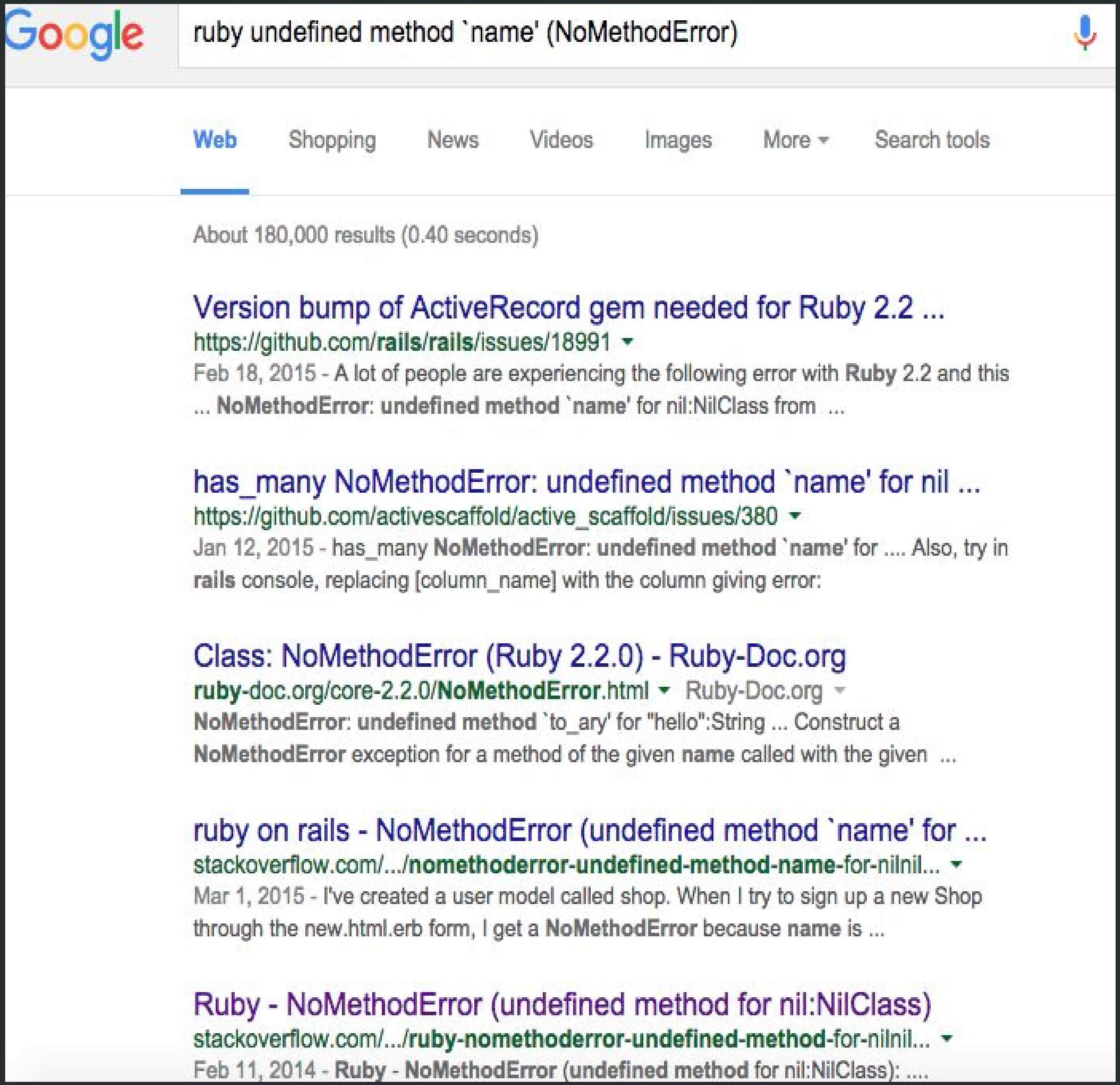
- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.

???

We have to remove information specific to our application from the error

Class names, method names, file names etc

What we really need is the error, and just the error



ruby undefined method `name' (NoMethodError)

About 180,000 results (0.40 seconds)

Version bump of ActiveRecord gem needed for Ruby 2.2 ...
https://github.com/rails/rails/issues/18991
Feb 18, 2015 - A lot of people are experiencing the following error with **Ruby** 2.2 and this ... **NoMethodError**: **undefined method** `**name**' for nil:NilClass from ...

has_many NoMethodError: undefined method `name' for nil ...
https://github.com/activescaffold/active_scaffold/issues/380
Jan 12, 2015 - has_many **NoMethodError**: **undefined method** `**name**' for .... Also, try in **rails** console, replacing [column_name] with the column giving error:

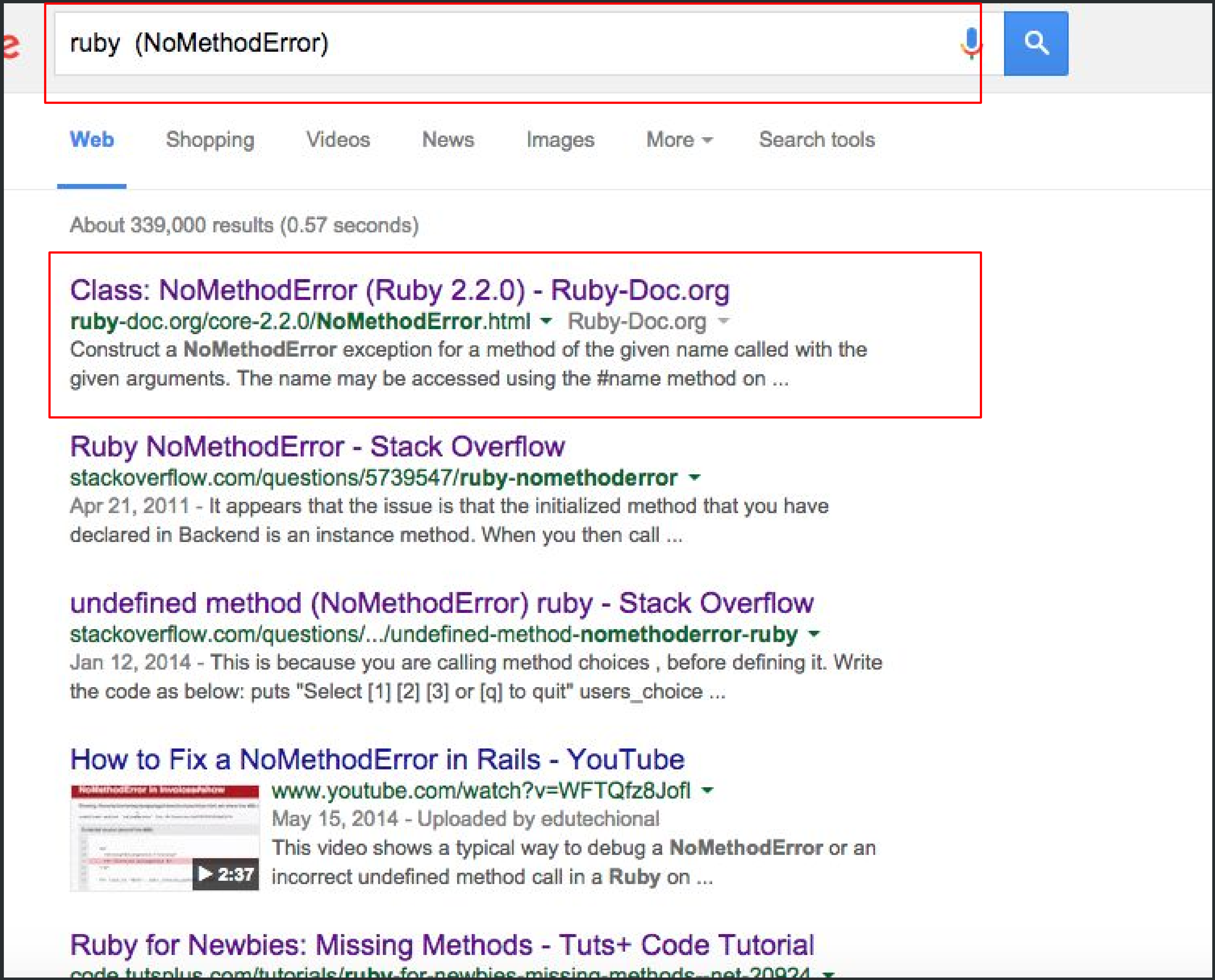
Class: NoMethodError (Ruby 2.2.0) - Ruby-Doc.org
**ruby**-doc.org/core-2.2.0/**NoMethodError**.html Ruby-Doc.org
**NoMethodError**: **undefined method** `to_ary' for "hello":String ... Construct a **NoMethodError** exception for a method of the given **name** called with the given ...

ruby on rails - NoMethodError (undefined method `name' for ...
stackoverflow.com/.../**nomethoderror-undefined-method-name**-for-nilnil...
Mar 1, 2015 - I've created a user model called shop. When I try to sign up a new Shop through the new.html.erb form, I get a **NoMethodError** because **name** is ...

Ruby - NoMethodError (undefined method for nil:NilClass)
stackoverflow.com/.../**ruby-nomethoderror-undefined-method**-for-nilnil...
Feb 11, 2014 - **Ruby** - **NoMethodError** (**undefined method** for nil:NilClass): ....

Not quite what we want either, Ruby and Rails have quite a bit of crossover.

Let’s try refining our search even further.



ruby (NoMethodError)

About 339,000 results (0.57 seconds)

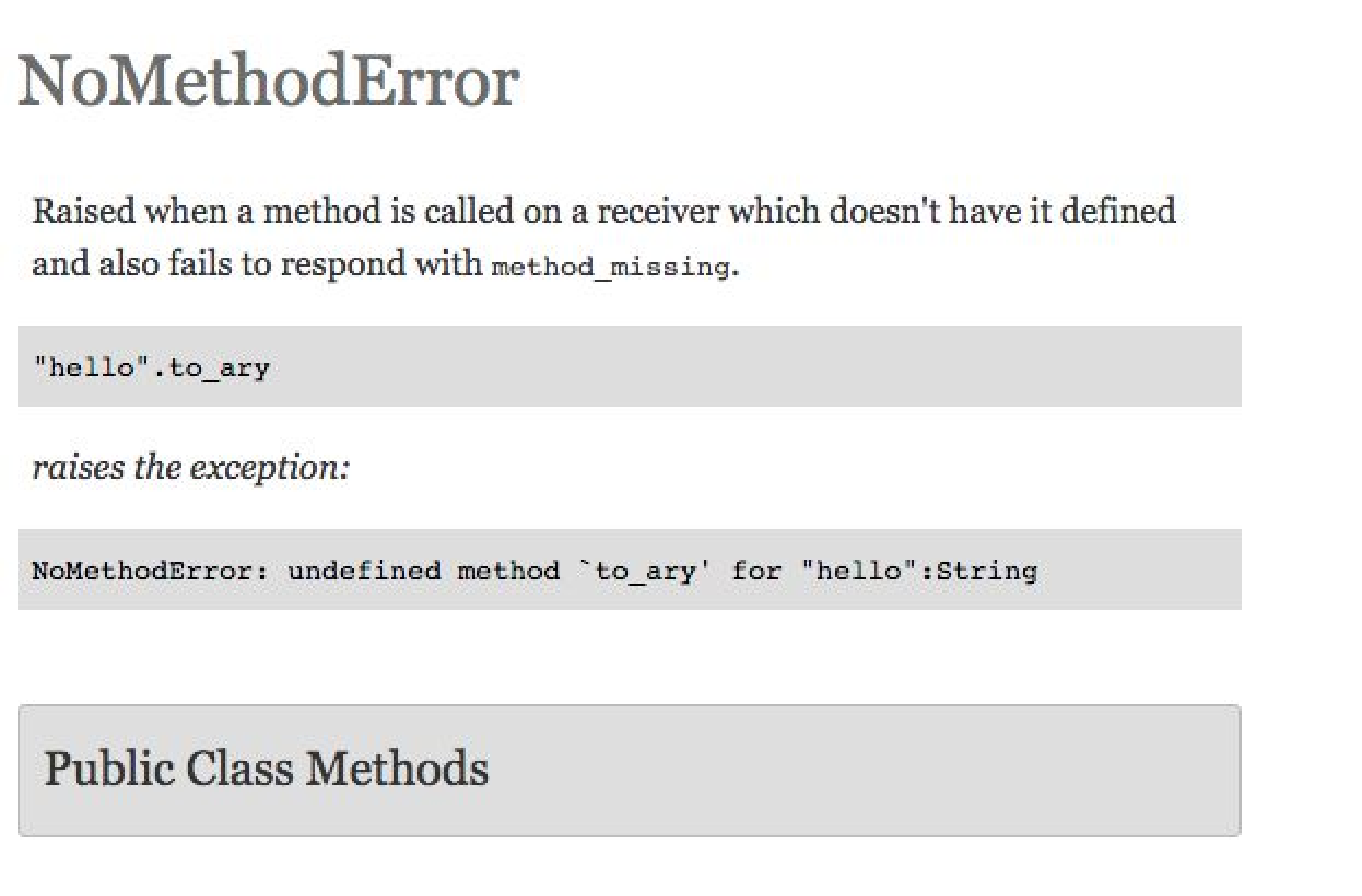
Class: NoMethodError (Ruby 2.2.0) - Ruby-Doc.org
ruby-doc.org/core-2.2.0/NoMethodError.html Ruby-Doc.org
Construct a NoMethodError exception for a method of the given name called with the given arguments. The name may be accessed using the #name method on ...

Ruby NoMethodError - Stack Overflow
stackoverflow.com/questions/5739547/ruby-nomethoderror
Apr 21, 2011 - It appears that the issue is that the initialized method that you have declared in Backend is an instance method. When you then call ...

undefined method (NoMethodError) ruby - Stack Overflow
stackoverflow.com/questions/.../undefined-method-nomethoderror-ruby
Jan 12, 2014 - This is because you are calling method choices , before defining it. Write the code as below: puts "Select [1] [2] [3] or [q] to quit" users_choice ...

How to Fix a NoMethodError in Rails - YouTube
www.youtube.com/watch?v=WFTQfz8Jofl
▶ 2:37
May 15, 2014 - Uploaded by edutechional
This video shows a typical way to debug a NoMethodError or an incorrect undefined method call in a Ruby on ...

Ruby for Newbies: Missing Methods - Tuts+ Code Tutorial
code.tutsplus.com/tutorials/ruby-for-newbies-missing-methods--net-20924



# NoMethodError

Raised when a method is called on a receiver which doesn't have it defined and also fails to respond with `method_missing`.

```
"hello".to_ary
```

*raises the exception:*

```
NoMethodError: undefined method `to_ary' for "hello":String
```

## Public Class Methods

There were also some helpful links to examples with our exact problem

Sometimes we just have to refine our search to get the right results

Now let's actually fix the problem.

We don’t have the method “name”, how can we add it?

# Option #1

```ruby
class Animal
   def initialize(name, age)
     @name = name
     @age = age
   end
   def name
     return @name
   end

end
dog = Animal.new("Shadow", 7)
puts dog.name
```

# Option #2

```
class Animal
  attr_reader :name
  def initialize(name, age)
    @name = name
    @age = age
  end
end

dog = Animal.new("Shadow", 7)
puts dog.name
```

# Step 1

Find the line of the error, Ruby gives this to you

# Step 2

Check your notes, and memory.
Have I had this error before?

# Step 3

If that fails, consult your friend google

Remember to only include the error, no specifics from your program.

# Step 4

Check the google results

Does someone have a similar problem to you?

Rarely will you have an exact match

Maybe someone has a User class with an undefined method age

Use your reasoning skills to draw a correlation

# Step 5

Attempt to fix the error, don’t copy and paste, apply

# Step 6

Repeat

# Other Common Errors

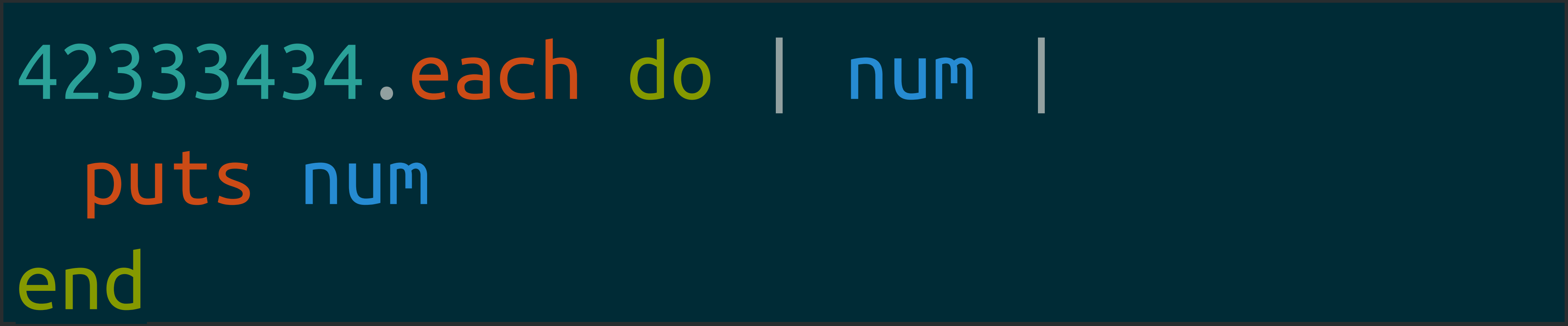
# Undefined Ruby Methods

Sometimes we call a method on the wrong kind of …something

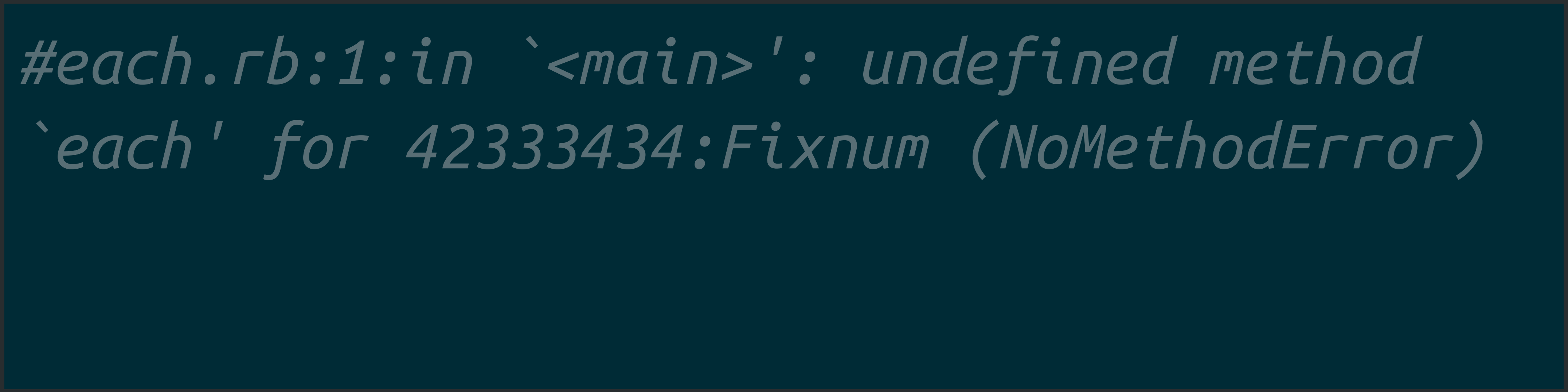
# Undefined Ruby Methods

Maybe I want to use a .each on a big number to get each of its numbers.

# Undefined Ruby Methods



```
42333434.each do | num |
  puts num
end
```

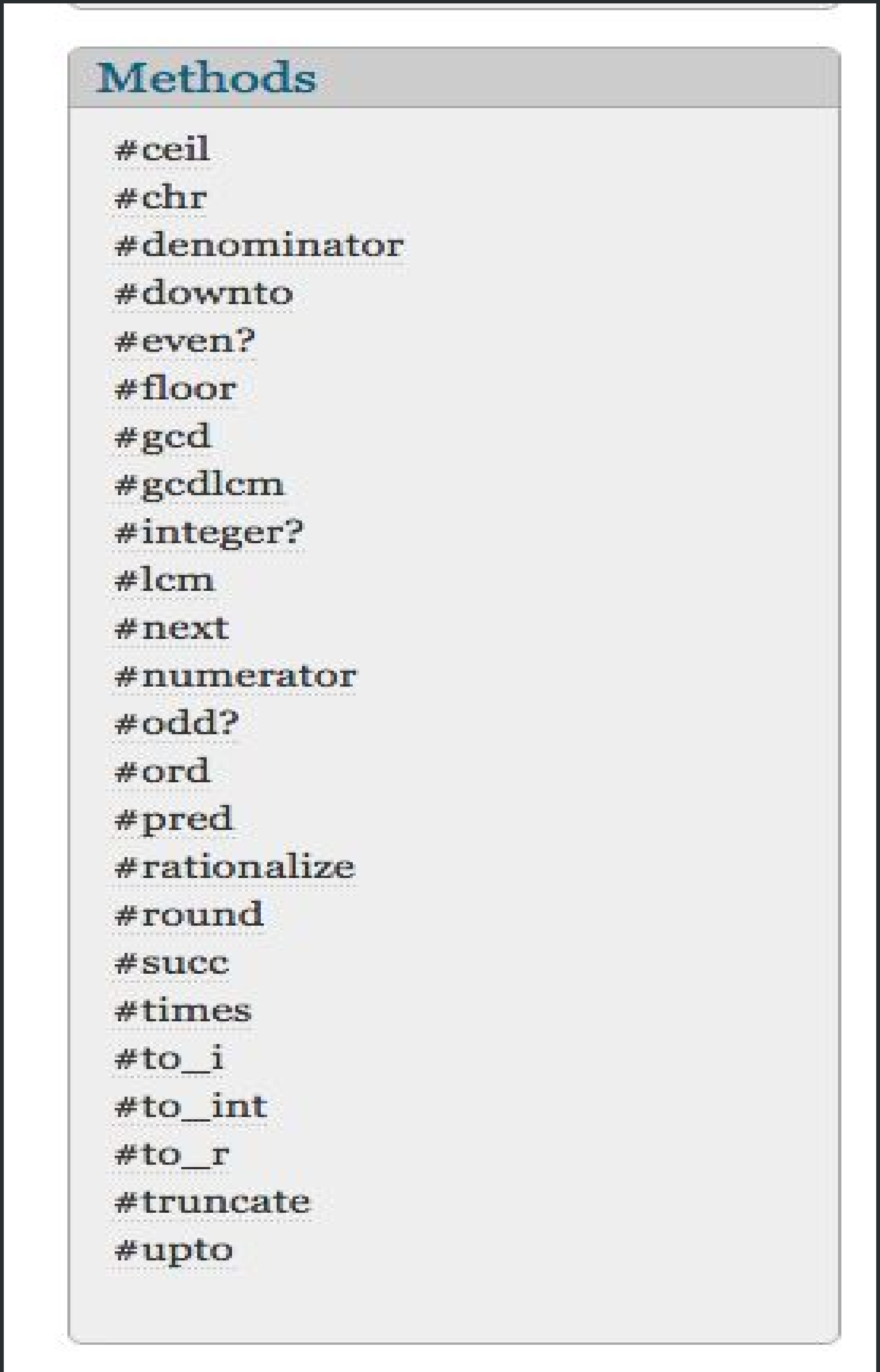


```
#each.rb:1:in `<main>': undefined method
`each' for 42333434:Fixnum (NoMethodError)
```

Looks familiar, huh?

# Where do we start?

We’re talking about an Fixnum so lets start in the Fixnum docs



### Methods

#ceil

#chr

#denominator

#downto

#even?

#floor

#gcd

#gcdlcm

#integer?

#lcm

#next

#numerator

#odd?

#ord

#pred

#rationalize

#round

#succ

#times

#to_i

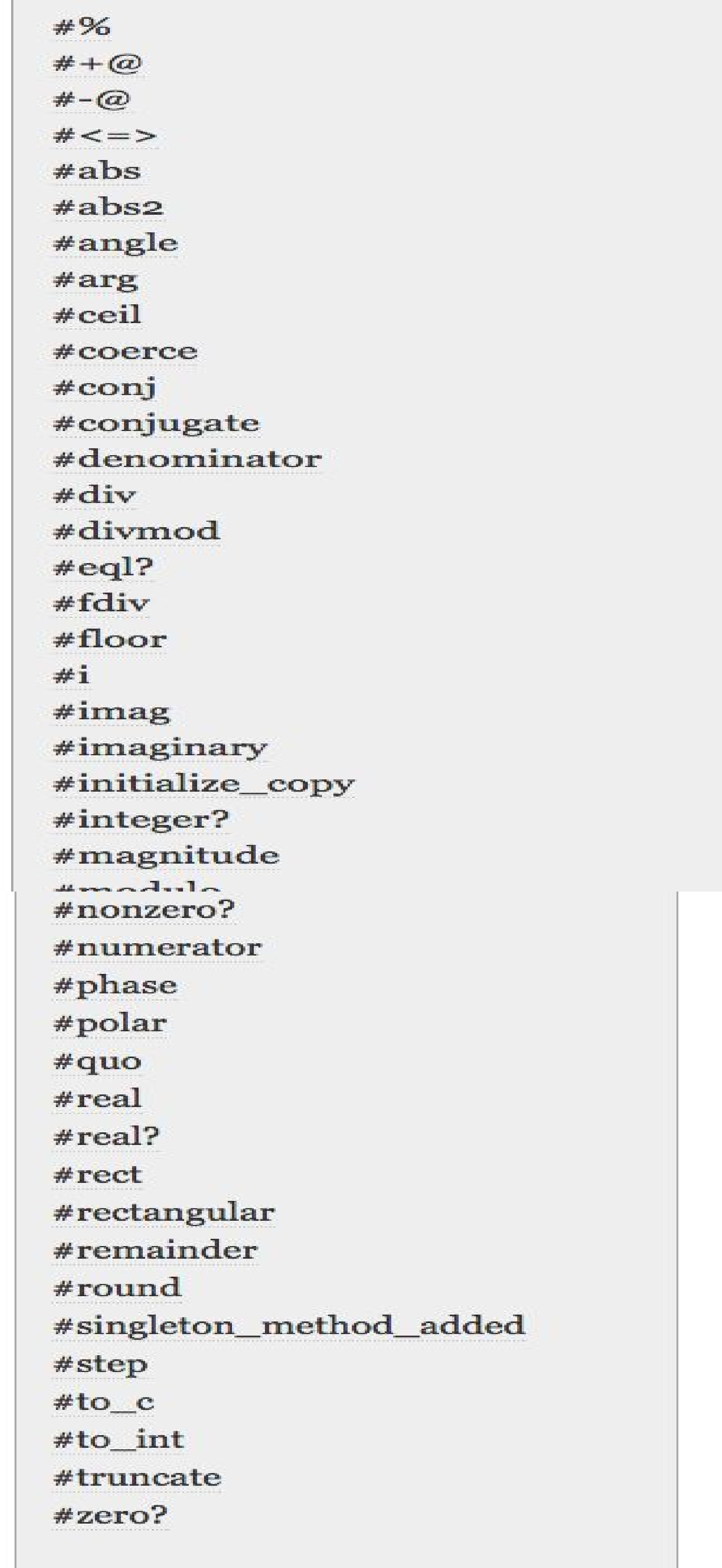
#to_int

#to_r

#truncate

#upto

Nope, let's try it's parent, numeric



```
#%
#+@
#-@
#<=>
#abs
#abs2
#angle
#arg
#ceil
#coerce
#conj
#conjugate
#denominator
#div
#divmod
#eql?
#fdiv
#floor
#i
#imag
#imaginary
#initialize_copy
#integer?
#magnitude
#modulo
#nonzero?
#numerator
#phase
#polar
#quo
#real
#real?
#rect
#rectangular
#remainder
#round
#singleton_method_added
#step
#to_c
#to_int
#truncate
#zero?
```

Nope, now we know why it’s an undefined method

Sorry, no cool tricks, we're just going to have to find a different solution

```
42333434.to_s.chars.each do | num |
  puts num
end
```

# Undefined Ruby Methods

```
undefined method [method_name] for
nil:NilClass (NoMethodError)
```

# Undefined Ruby Methods

It’s a common error to introduce nil values and try to call methods on them.

# Example

```
class Animal
   def initialize(name, age)
     @nama = name
     @age = age
   end
   def name
     return @name
   end

end
dog = Animal.new("Shadow", 7)
puts dog.name.upcase
```

# Undefined Local variable or method

Let’s do something different with my animal example.

# Undefined Local variable or method

Say I make a mistake with my dog variable.

# Undefined Local variable or method

```ruby
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end

  def name
    return @name
  end
end

dog = Animal.new("Shadow", 7)
puts dawg.name
```

# Undefined Local variable or method

```
#errors.rb:14:in `<main>':
undefined local variable or method
`dawg' for main:Object (NameError)
```

# Undefined Local variable or method

What does this mean?

# Undefined Local variable or method

The variable doesn’t exist. Check your syntax, check your spelling.

# Undefined Local variable or method

It could also mean you aren’t in the correct scope

# Undefined Local variable or method

```
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end
  def dog_years
    return age * 7
  end

  def name
    return @name
  end
end

dog = Animal.new("Shadow", 7)
puts dog.dog_years
```

# Undefined Local variable or method

```
errors.rb:8:in `dog_years': undefined local
variable or method `age' for #<Animal:
0x007fe19181b058 @name="Shadow", @age=7>
(NameError)
```

# Undefined Local variable or method

```
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end
  def dog_years
    return age * 7
  end

  def name
    return @name
  end
end

dog = Animal.new("Shadow", 7)
puts dog.dog_years
```

# Undefined Local variable or method

We don’t have access to “age” in our dog_years method

# Undefined Local variable or method

So Ruby throws us an error

# Undefined Local variable or method

…but we do have an @age in this method scope

# Undefined Local variable or method

```ruby
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end
  def dog_years
    return @age * 7
  end

  def name
    return @name
  end
end

dog = Animal.new("Shadow", 7)
puts dog.dog_years
```

# Adding the wrong types together

Let’s say I have an array of foods and I want to add another one to it

# Adding the wrong types together

```
array = ["Pizza", "Cookies", "Icecream"]

array += "Cake"

#adding.rb:3:in `<main>': no implicit conversion of
String into Array (TypeError)
```

# The problem?

In Ruby we can perform certain operations on objects of different type

This means that when we declare a variable, Ruby infers what type it is

After that, we can’t add something of another type to it

```
number = 1
number += "2"
#adding.rb:5:in `+': String can't be coerced
into Fixnum (TypeError)

string = “You can’t add numbers to me, go
ahead and try”
string += 2
#TypeError: no implicit conversion of Fixnum
into String
  from (irb):3:in `+'
  from (irb):3
```

```
number = 1
number += 2
# 3

string = “You can’t add numbers to
me, go ahead and try”
string += "!"
#“You can’t add numbers to me, go
ahead and try!”
```

# Remember, don’t get frustrated

# Find the line of the error.

Try to make sense of the error based on that

PRY is a great ally in this battle

And if all else fails, consult your friend google.

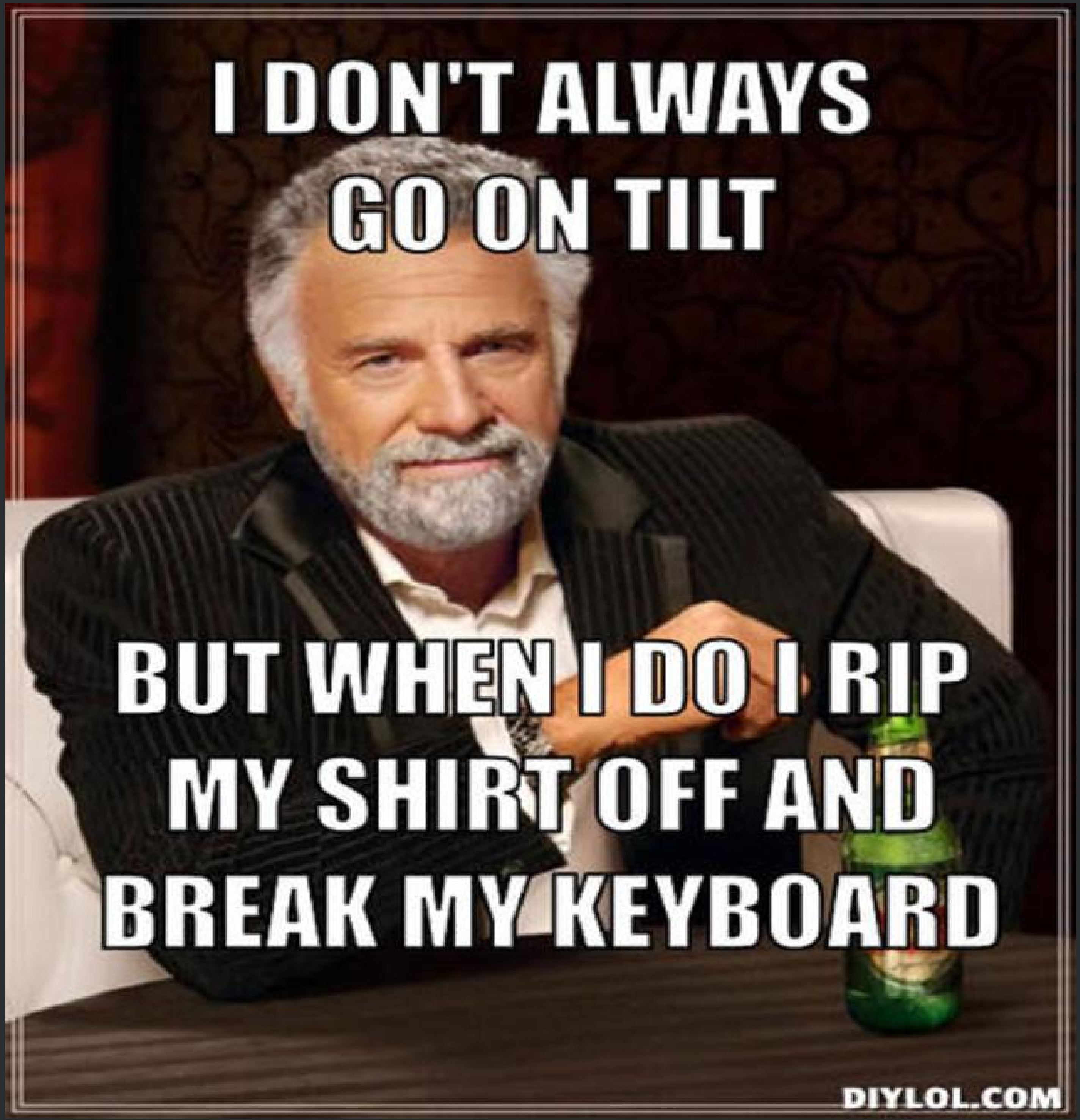
# Coding is hard

Sometimes we make mistakes.

Sometimes these mistakes cause errors

## So how do we fix these errors?

# Don’t go on tilt



Believe it or not errors are here to help us

We should let the errors guide us to the correct answer

Lets take a look at a couple of common errors and how to solve them

# No method error

```ruby
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end
end

dog = Animal.new("Shadow", 7)
puts dog.name
```

# No method error

```
errors.rb:9:in `<main>': undefined method `name' for #<Animal:
0x007ffebaa0b278 @name="Shadow", @age=7> (NoMethodError)
```

Oh no, what to do?

# Look at the error first

```
errors.rb:9:in `<main>': undefined method `name' for #<Animal:
0x007ffebaa0b278 @name="Shadow", @age=7> (NoMethodError)
```

# No method error

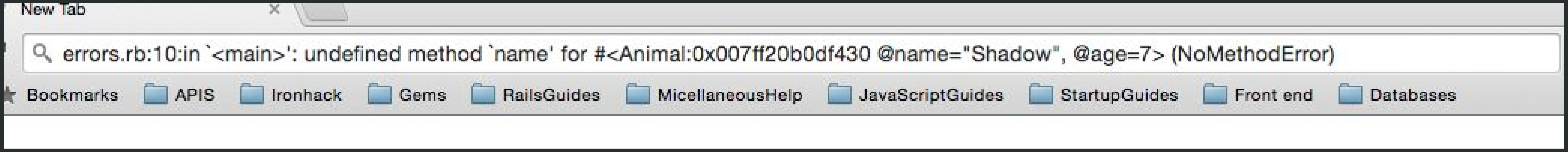
```
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end
end

dog = Animal.new("Shadow", 7)
puts dog.name
```

Ruby thinks we're trying to call a "name" method on our dog

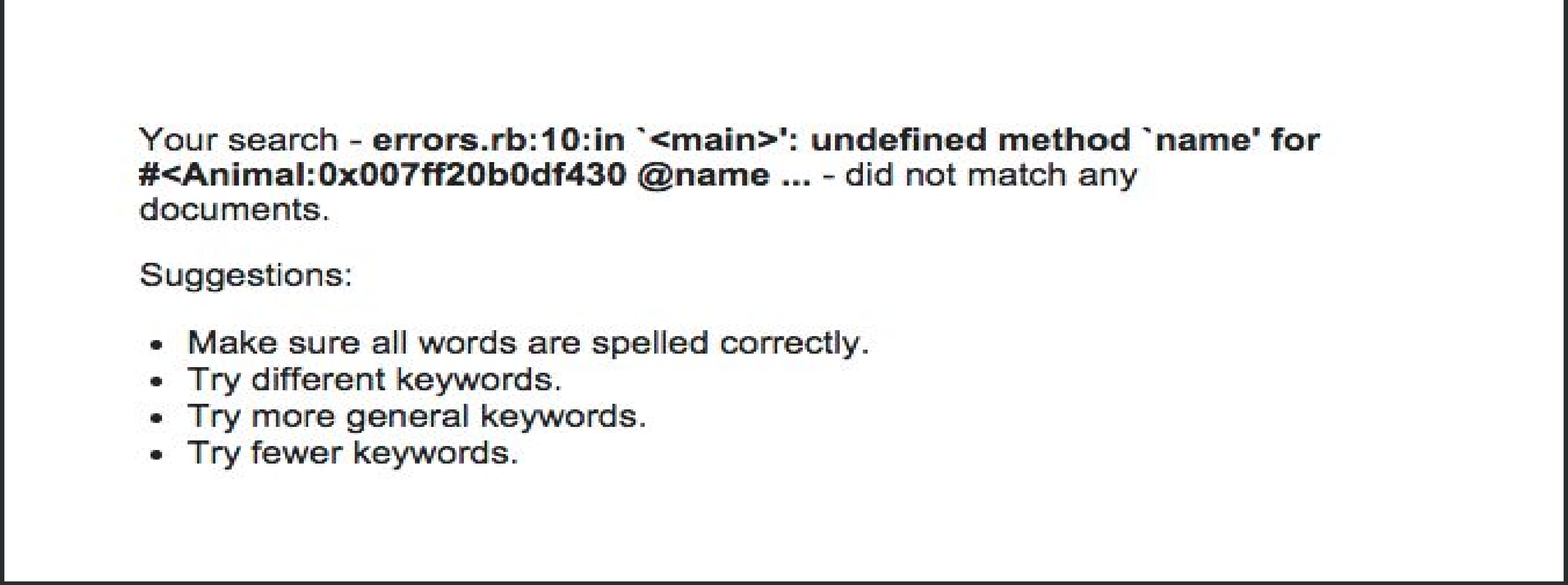
But our dog doesn’t have a name method!

There's a couple fixes for this, but how do we find them?

Let’s consult our old friend google.



```
errors.rb:10:in `<main>': undefined method `name' for #<Animal:0x007ff20b0df430 @name="Shadow", @age=7> (NoMethodError)
```



Your search - **errors.rb:10:in `<main>': undefined method `name' for #<Animal:0x007ff20b0df430 @name ...** - did not match any documents.

Suggestions:

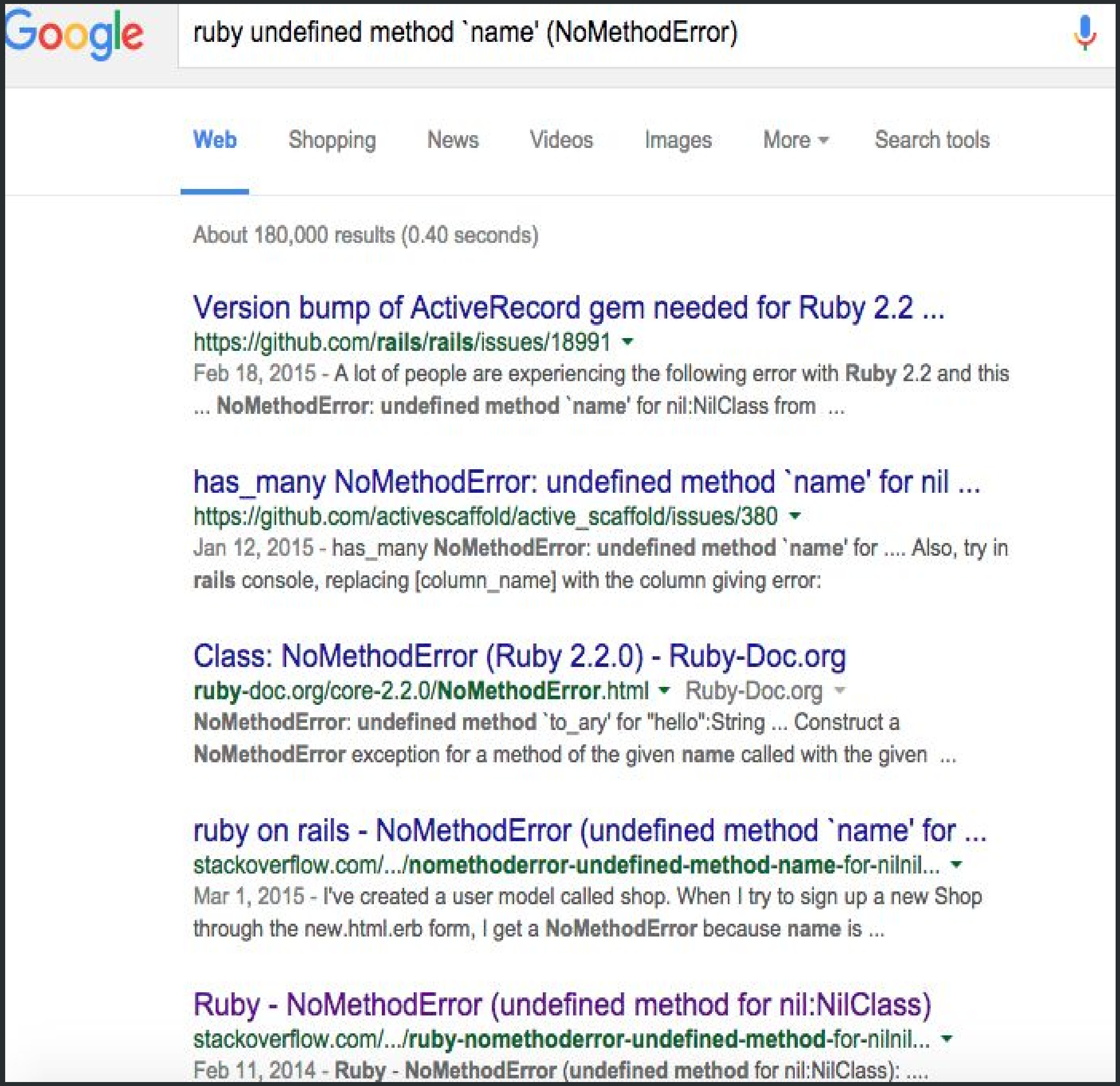
- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.

???

We have to remove information specific to our application from the error

Class names, method names, file names etc

What we really need is the error, and just the error



ruby undefined method `name' (NoMethodError)

About 180,000 results (0.40 seconds)

Version bump of ActiveRecord gem needed for Ruby 2.2 ...
https://github.com/rails/rails/issues/18991
Feb 18, 2015 - A lot of people are experiencing the following error with Ruby 2.2 and this ... NoMethodError: undefined method `name' for nil:NilClass from ...

has_many NoMethodError: undefined method `name' for nil ...
https://github.com/activescaffold/active_scaffold/issues/380
Jan 12, 2015 - has_many NoMethodError: undefined method `name' for .... Also, try in rails console, replacing [column_name] with the column giving error:

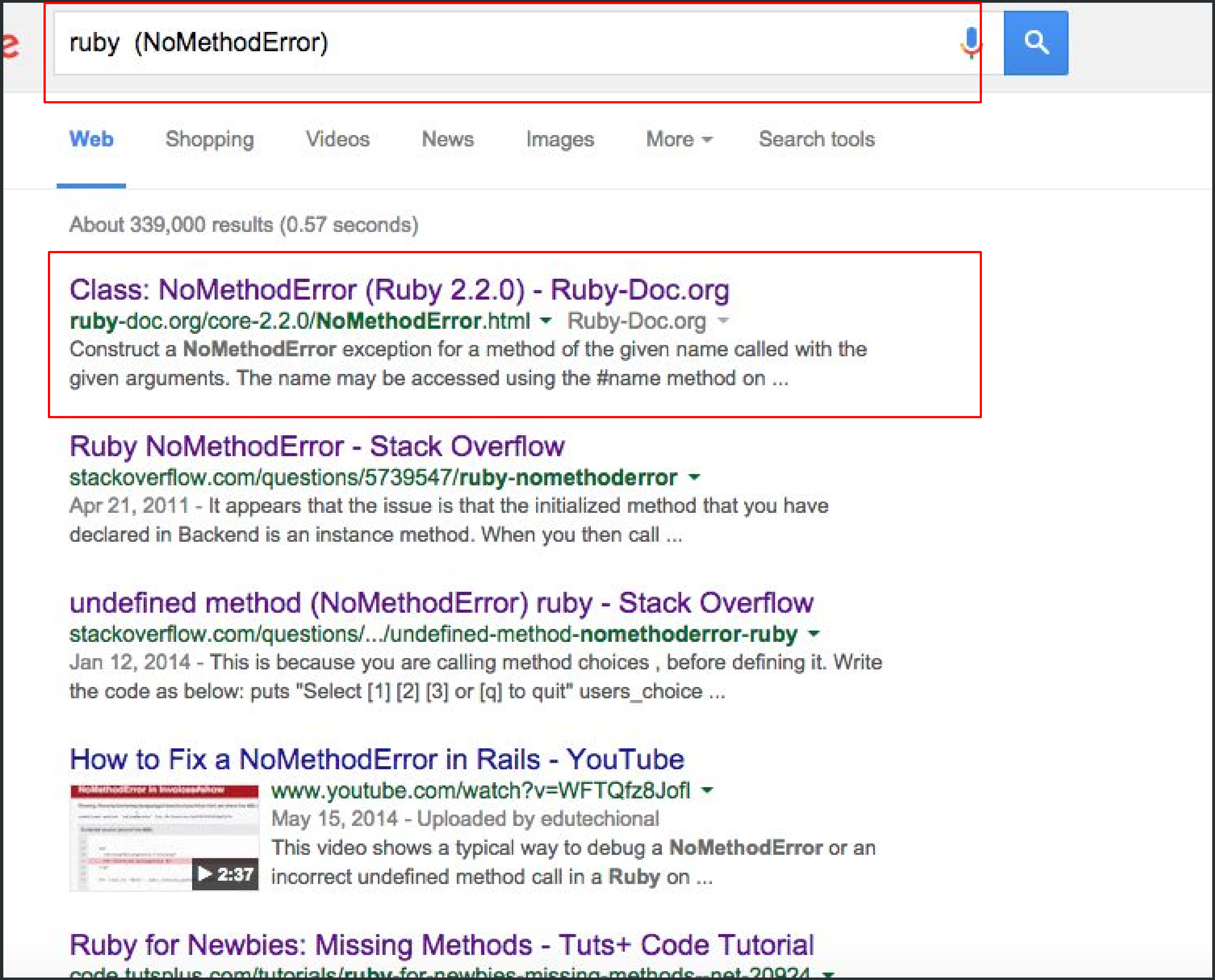
Class: NoMethodError (Ruby 2.2.0) - Ruby-Doc.org
ruby-doc.org/core-2.2.0/NoMethodError.html Ruby-Doc.org
NoMethodError: undefined method `to_ary' for "hello":String ... Construct a NoMethodError exception for a method of the given name called with the given ...

ruby on rails - NoMethodError (undefined method `name' for ...
stackoverflow.com/.../nomethoderror-undefined-method-name-for-nilnil...
Mar 1, 2015 - I've created a user model called shop. When I try to sign up a new Shop through the new.html.erb form, I get a NoMethodError because name is ...

Ruby - NoMethodError (undefined method for nil:NilClass)
stackoverflow.com/.../ruby-nomethoderror-undefined-method-for-nilnil...
Feb 11, 2014 - Ruby - NoMethodError (undefined method for nil:NilClass): ....

Not quite what we want either, Ruby and Rails have quite a bit of crossover.

Let’s try refining our search even further.



ruby (NoMethodError)

About 339,000 results (0.57 seconds)

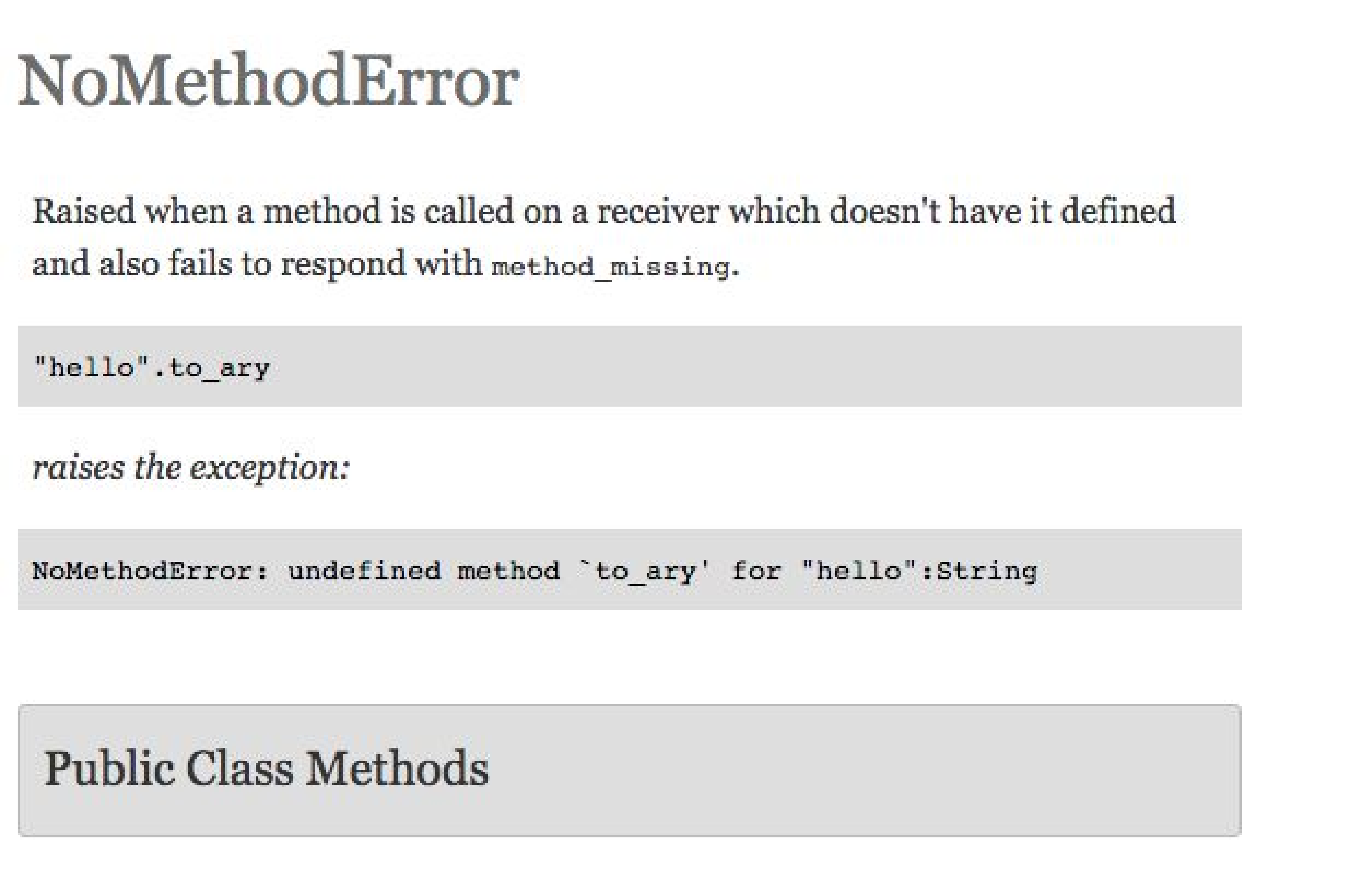
Class: NoMethodError (Ruby 2.2.0) - Ruby-Doc.org
ruby-doc.org/core-2.2.0/NoMethodError.html Ruby-Doc.org
Construct a **NoMethodError** exception for a method of the given name called with the given arguments. The name may be accessed using the #name method on ...

Ruby NoMethodError - Stack Overflow
stackoverflow.com/questions/5739547/**ruby-nomethoderror**
Apr 21, 2011 - It appears that the issue is that the initialized method that you have declared in Backend is an instance method. When you then call ...

undefined method (NoMethodError) ruby - Stack Overflow
stackoverflow.com/questions/.../undefined-method-**nomethoderror-ruby**
Jan 12, 2014 - This is because you are calling method choices , before defining it. Write the code as below: puts "Select [1] [2] [3] or [q] to quit" users_choice ...

How to Fix a NoMethodError in Rails - YouTube
www.youtube.com/watch?v=WFTQfz8Jofl
May 15, 2014 - Uploaded by edutechional
This video shows a typical way to debug a **NoMethodError** or an incorrect undefined method call in a **Ruby** on ...

Ruby for Newbies: Missing Methods - Tuts+ Code Tutorial
code.tutsplus.com/tutorials/ruby-for-newbies-missing-methods--net-20924



# NoMethodError

Raised when a method is called on a receiver which doesn't have it defined and also fails to respond with `method_missing`.

```
"hello".to_ary
```

*raises the exception:*

```
NoMethodError: undefined method `to_ary' for "hello":String
```

## Public Class Methods

There were also some helpful links to examples with our exact problem

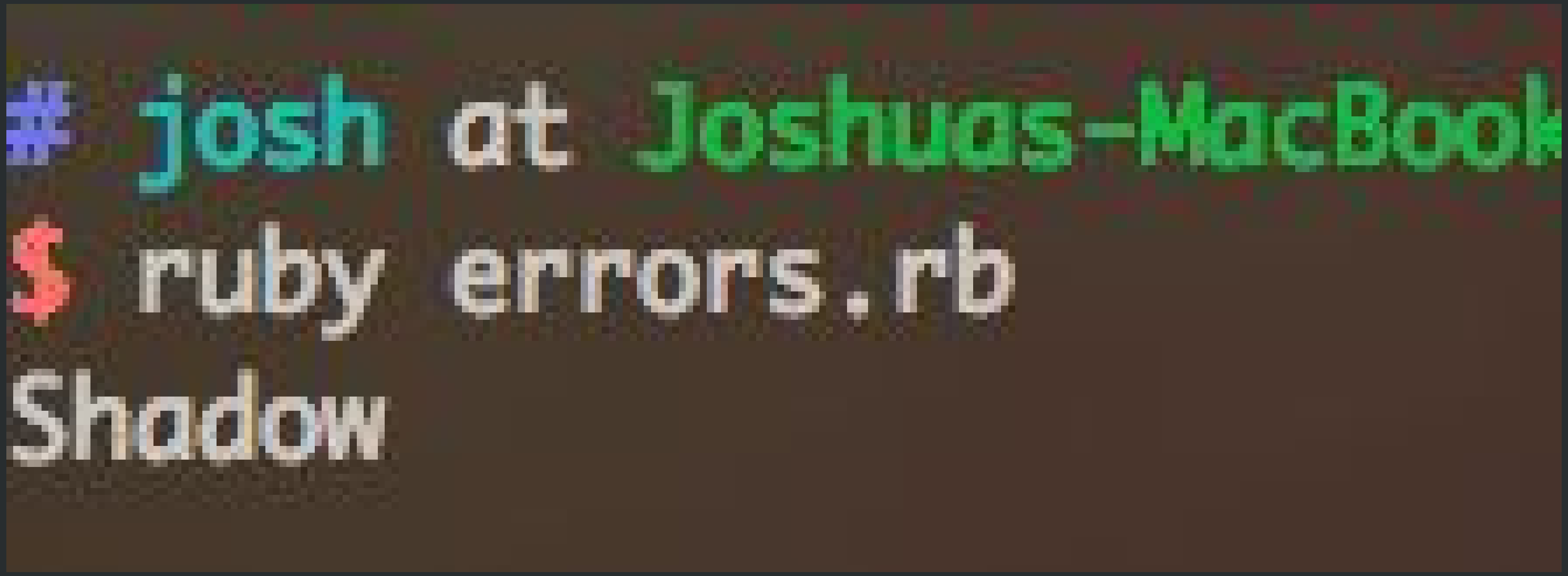
Sometimes we just have to refine our search to get the right results

Now let's actually fix the problem.

We don’t have the method “name”, how can we add it?

# Option #1

```ruby
class Animal
   def initialize(name, age)
     @name = name
     @age = age
   end
   def name
     return @name
   end

end
dog = Animal.new("Shadow", 7)
puts dog.name
```



```
# josh at Joshuas-MacBook
$ ruby errors.rb
Shadow
```

# Option #2

```ruby
class Animal
  attr_reader  :name
  def initialize(name, age)
    @name = name
    @age = age
  end
end

dog = Animal.new("Shadow", 7)
puts dog.name
```

# Step 1

Find the line of the error, Ruby gives this to you

# Step 2

Use that supercomputer called your brain, and your notes to hypothesize about why

# Step 3

If that fails, consult your friend google

Remember to only include the error, no specifics from your program.

# Step 4

Check the google results

Does someone have a similar problem to you?

Rarely will you have an exact match

Maybe someone has a “User” class with an undefined method “age”

Use your reasoning skills to draw a correlation

# Step 5

Attempt to fix the error, don't copy and paste, apply

# Step 6

Repeat

# Other Common Errors

# Undefined Ruby Methods

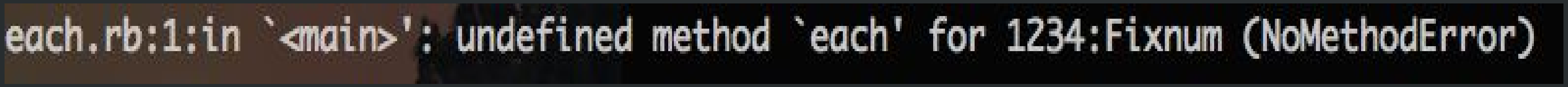
Sometimes we call a method on the wrong kind of …something

# Undefined Ruby Methods

Maybe I want to use a .each on a big number to get each of its numbers.

# Undefined Ruby Methods

```
42333434.each do | num |
  puts num
end
```

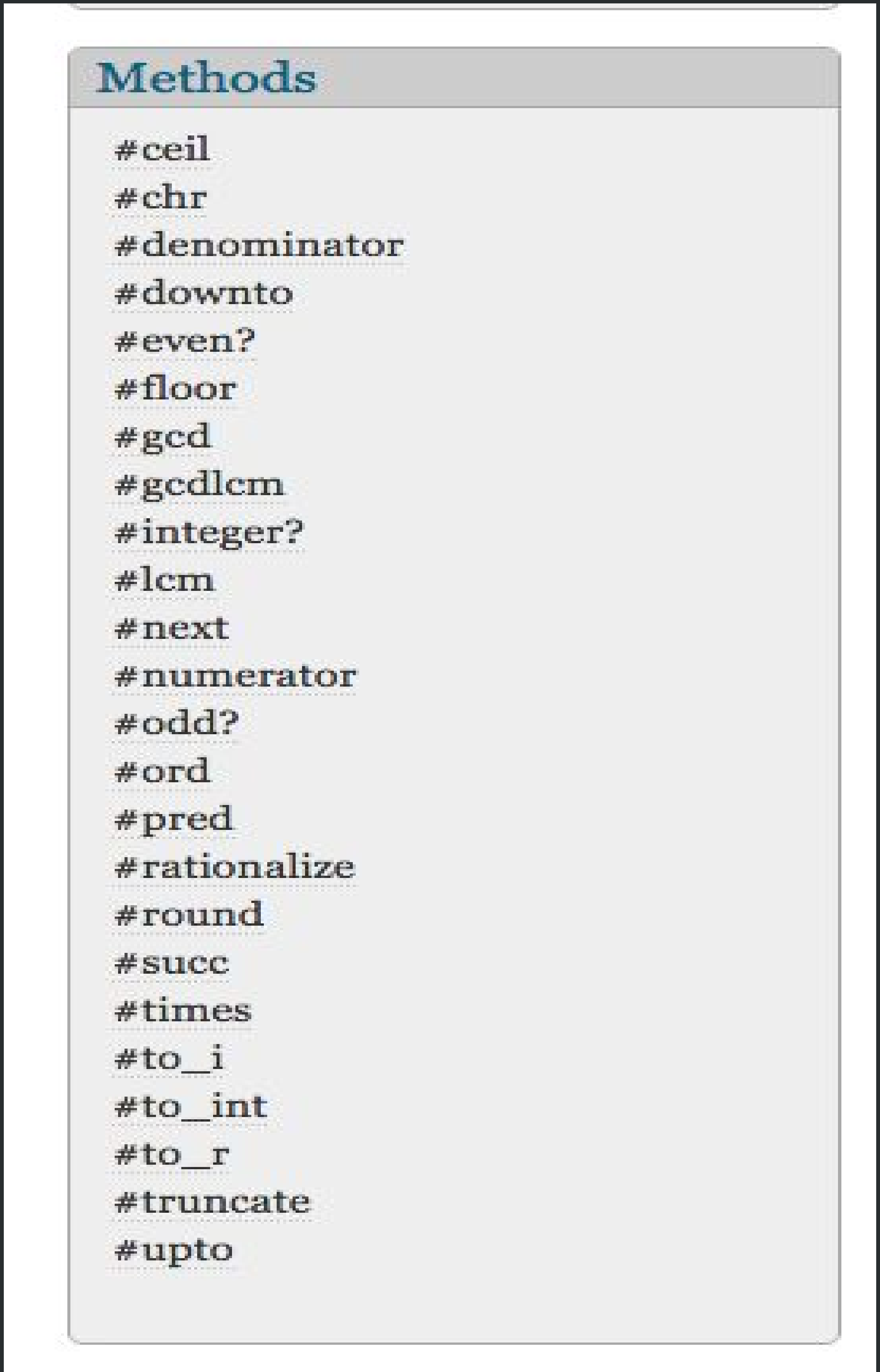


```
each.rb:1:in `<main>': undefined method `each' for 1234:Fixnum (NoMethodError)
```

Looks familiar, huh?

# Where do we start?

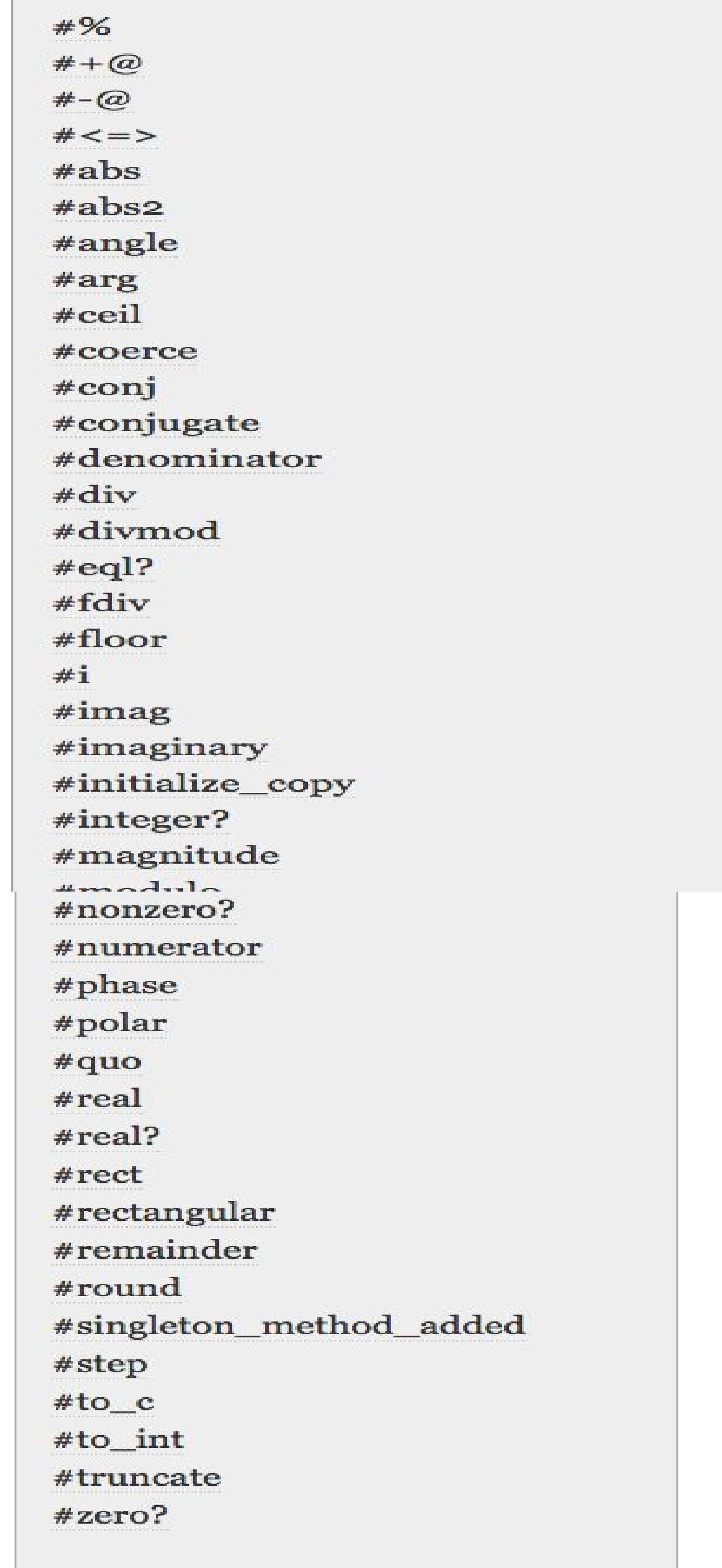
We’re talking about an Fixnum so lets start in the Fixnum docs



## Methods

- #ceil
- #chr
- #denominator
- #downto
- #even?
- #floor
- #gcd
- #gcdlcm
- #integer?
- #lcm
- #next
- #numerator
- #odd?
- #ord
- #pred
- #rationalize
- #round
- #succ
- #times
- #to_i
- #to_int
- #to_r
- #truncate
- #upto

Nope, let's try it's parent, numeric



```
#%
#+@
#-@
#<=>
#abs
#abs2
#angle
#arg
#ceil
#coerce
#conj
#conjugate
#denominator
#div
#divmod
#eql?
#fdiv
#floor
#i
#imag
#imaginary
#initialize_copy
#integer?
#magnitude
#modulo
#nonzero?
#numerator
#phase
#polar
#quo
#real
#real?
#rect
#rectangular
#remainder
#round
#singleton_method_added
#step
#to_c
#to_int
#truncate
#zero?
```

Nope, now we know why it's an undefined method

Sorry, no cool tricks, we're just going to have to find a different solution

```ruby
42333434.to_s.chars.each do | num |
  puts num
end
```

# Undefined Ruby Methods

```
undefined method [method_name] for
nil:NilClass (NoMethodError)
```

# Undefined Ruby Methods

It’s a common error to introduce nil values and try to call methods on them.

# Example

```
class Animal
   def initialize(name, age)
     @nama = name
     @age = age
   end
   def name
     return @name
   end

end
dog = Animal.new("Shadow", 7)
puts dog.name.upcase
```

# Undefined Local variable or method

Let's do something different with my animal example.

# Undefined Local variable or method

Say I make a mistake with my dog variable.

# Undefined Local variable or method

```
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end

  def name
    return @name
  end
end

dog = Animal.new("Shadow", 7)
puts dawg.name
```

# Undefined Local variable or method

```
#errors.rb:14:in `<main>':
undefined local variable or method
`dawg' for main:Object (NameError)
```

# Undefined Local variable or method

What does this mean?

# Undefined Local variable or method

The variable doesn’t exist. Check your syntax, check your spelling.

# Undefined Local variable or method

It could also mean you aren’t in the correct scope

# Undefined Local variable or method

```ruby
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end
  def dog_years
    return age * 7
  end

  def name
    return @name
  end
end

dog = Animal.new("Shadow", 7)
puts dog.dog_years
```

# Undefined Local variable or method

```
errors.rb:8:in `dog_years': undefined local
variable or method `age' for #<Animal:
0x007fe19181b058 @name="Shadow", @age=7>
(NameError)
```

# Undefined Local variable or method

```
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end
  def dog_years
    return age * 7
  end

  def name
    return @name
  end
end

dog = Animal.new("Shadow", 7)
puts dog.dog_years
```

# Undefined Local variable or method

We don’t have access to “age” in our dog_years method

# Undefined Local variable or method

So Ruby throws us an error

# Undefined Local variable or method

…but we do have an @age in this method scope

# Undefined Local variable or method

```ruby
class Animal
  def initialize(name, age)
    @name = name
    @age = age
  end
  def dog_years
    return @age * 7
  end

  def name
    return @name
  end
end

dog = Animal.new("Shadow", 7)
puts dog.dog_years
```

# Adding the wrong types together

Let’s say I have an array of foods and I want to add another one to it

# Adding the wrong types together

```
array = ["Pizza", "Cookies", "Icecream"]
```

```
array += "Cake"
```

```
#adding.rb:3:in `<main>': no implicit conversion of
String into Array (TypeError)
```

# The problem?

In Ruby we can perform certain operations on objects of different type

This means that when we declare a variable, Ruby infers what type it is

After that, we can’t add something of another type to it

```
number = 1
number += "2"
#adding.rb:5:in `+': String can't be coerced
into Fixnum (TypeError)

string = “You can’t add numbers to me, go
ahead and try”
string += 2
#TypeError: no implicit conversion of Fixnum
into String
  from (irb):3:in `+'
  from (irb):3
```

```
number = 1
number += 2
# 3

string = “You can’t add numbers to
me, go ahead and try”
string += "!"
#“You can’t add numbers to me, go
ahead and try!”
```

Remember, don’t get frustrated

Find the line of the error.

Try to make sense of the error based on that

PRY is a great ally in this battle

And if all else fails, consult your friend google.