

Hello Everyone!

プレゼンがうまくいくかどうかは
最初の30秒で決まる

Ruby Development at Shopify

ShopifyでRubyの開発

Aaron Patterson



shopify

Senior Staff Engineer

Ruby Infrastructure Team

Thank you!!

Ruby Development at Shopify

ShopifyでRubyの使い方

ShopifyでRubyの開発

110010111001101011001011

How I started with Ruby

How Shopify uses Ruby

Shopify's Development of Ruby

過去

現在

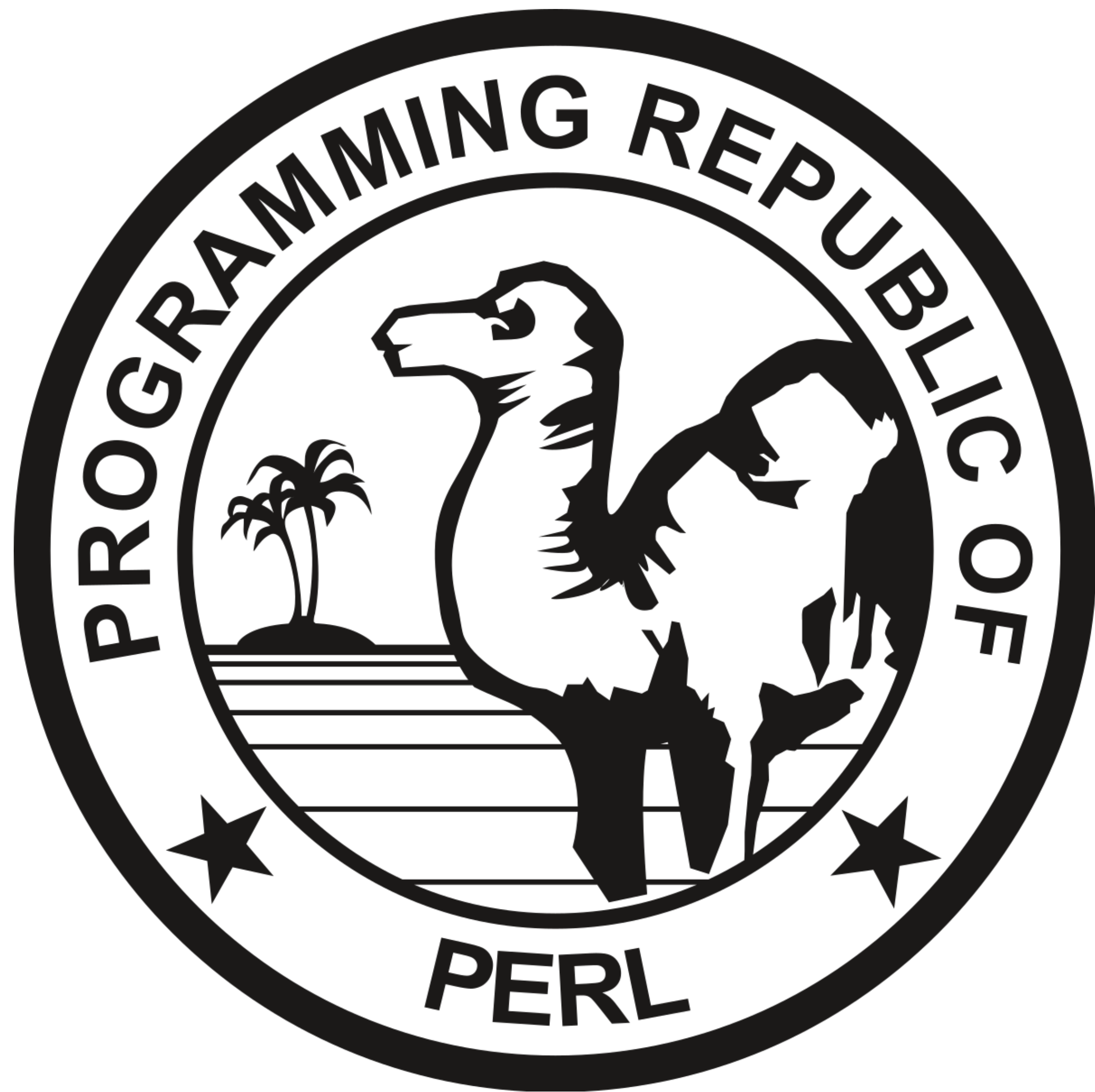
未來

Ruby is great to learn!

Ruby is great to use!

How I started with Ruby

I started programming in 1999



Typical Perl Program

Something like what I would have written

Person Class

```
#!/usr/bin/perl

package Person;

sub new {
    my $class = shift;
    my $self = {
        _firstName => shift,
        _lastName  => shift,
    };
    bless $self, $class;
    return $self;
}

sub setFirstName {
    my ( $self, $firstName ) = @_;
    $self->{_firstName} = $firstName;
    return $self->{_firstName};
}

sub getFirstName {
    my( $self ) = @_;
    return $self->{_firstName};
}

1;
```


Using The Class

```
use Person;

$person = new Person("Aaron", "Patterson");
$firstName = $person->getFirstName();
```

What type is \$self?

Why 1?



Language Rules

2001: I learned Java

Typical Java Program

Java 1.4 at the time

Person Class

```
public class Person {  
    private String firstName;  
    private String lastName;  
  
    public Person(String firstName, String lastName) {  
        this.firstName = firstName;  
        this.lastName = lastName;  
    }  
  
    public String getFirstName() {  
        return firstName;  
    }  
  
    public void setFirstName(String firstName) {  
        this.firstName = firstName;  
    }  
}
```

Using the Class

```
class Main {  
    public static void main(String[] args) {  
  
        Person person = new Person("Aaron", "Patterson");  
        person.setFirstName("Testing");  
  
        System.out.println(person.getFirstName());  
    }  
}
```

Typical Java Program

Java 1.4 at the time

Person Class

```
public class Person {  
    private String firstName;  
    private String lastName;  
  
    public Person(String firstName, String lastName) {  
        this.firstName = firstName;  
        this.lastName = lastName;  
    }  
  
    public String getFirstName() {  
        return firstName;  
    }  
  
    public void setFirstName(String firstName) {  
        this.firstName = firstName;  
    }  
}
```

Using the Class

```
class Main {  
    public static void main(String[] args) {  
  
        Person person = new Person("Aaron", "Patterson");  
  
        person.setFirstName("Testing");  
  
        System.out.println(person.getFirstName());  
    }  
}
```


Typical Java Program

Java 1.4 at the time

Person Class

```
public class Person {
    private String firstName;
    private String lastName;

    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
}
```

Using the Class

```
class Main {
    public static void main(String[] args) {

        Person person = new Person("Aaron", "Patterson");

        person.setFirstName("Testing");

        System.out.println(person.getFirstName());
    }
}
```

Boilerplate Code

List Iteration Java Program

```
import java.util.ArrayList;
import java.util.Iterator;

class Main {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();

        list.add(new Integer(0));
        list.add(new Integer(1));
        list.add(new Integer(2));
        list.add(new Integer(3));

        Iterator iter = list.iterator();
        while (iter.hasNext()) {
            Integer item = (Integer)iter.next();
            System.out.println(item + 1);
        }
    }
}
```

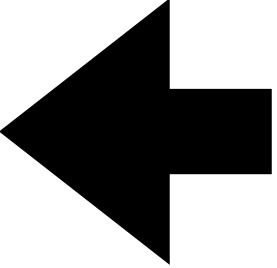
List Iteration Java Program

```
import java.util.ArrayList;
import java.util.Iterator;

class Main {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();

        list.add(new Integer(0));
        list.add(new Integer(1));
        list.add(new Integer(2));
        list.add(new Integer(3));

        Iterator iter = list.iterator();
        while (iter.hasNext()) {
            Integer item = (Integer)iter.next();
            System.out.println(item + 1);
        }
    }
}
```



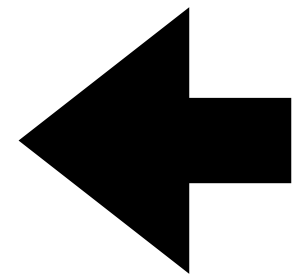
List Iteration Java Program

```
import java.util.ArrayList;
import java.util.Iterator;

class Main {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();

        list.add(new Integer(0));
        list.add(new Integer(1));
        list.add(new Integer(2));
        list.add(new Integer(3));

        Iterator iter = list.iterator();
        while (iter.hasNext()) {
            Integer item = (Integer)iter.next();
            System.out.println(item + 1);
        }
    }
}
```



Book Keeping

2004: Discover Ruby

Ruby で日本語を

Typical Ruby Program

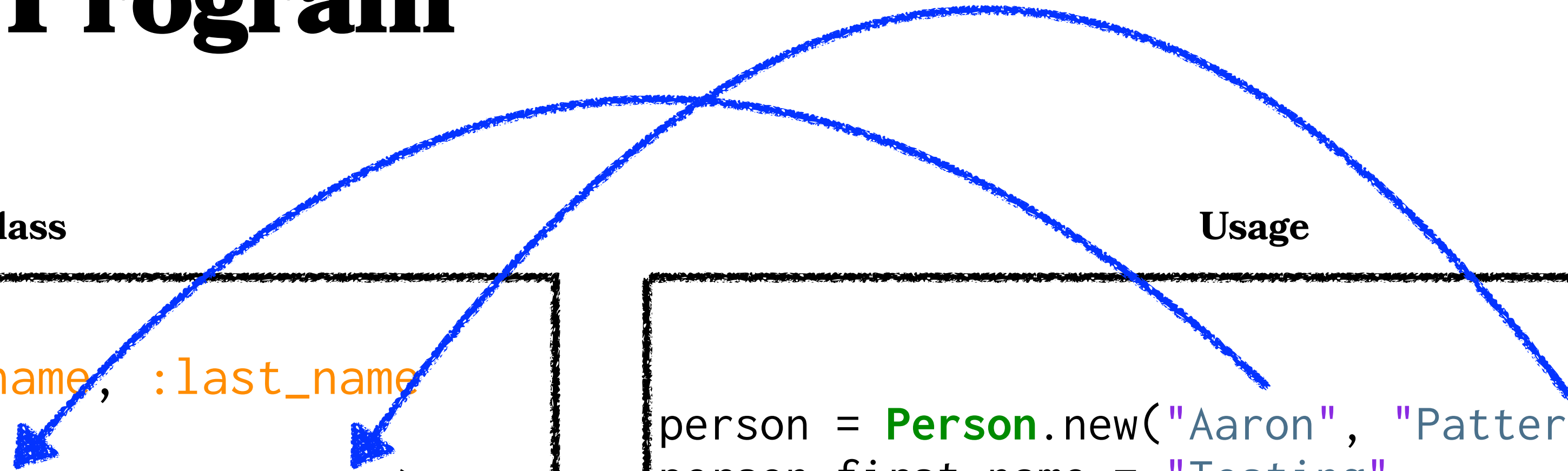
Person Class

```
class Person
  attr_accessor :first_name, :last_name
  def initialize(first_name, last_name)
    @first_name = first_name
    @last_name = last_name
  end
end
```

Usage

```
person = Person.new("Aaron", "Patterson")
person.first_name = "Testing"

puts person.first_name
```



Why learn Ruby?

Simple Language Rules

No Boilerplate Code

No Book Keeping

Focus on Solutions

Easier than Japanese

2006: First Ruby Job

2020: Starting at Shopify

Top Ruby Companies

<https://toprubycompanies.info>

The screenshot shows the homepage of toprubycompanies.info. The navigation bar includes 'TOP RUBY COMPANIES', 'Subsidiaries', 'New Consultancies', 'Coming Soon', 'Startups', and a '+ Add Your Company' button. The main section features the heading 'Explore Top Ruby Companies Around the World' and a sub-headline 'Discover top Ruby companies worldwide and see the impact of the programming language that focuses on developer happiness.' Below this are three statistics: 48 Public Companies, \$251 Billion Total Market Cap, and \$62.7 Billion Total TTM Revenue. A circular diagram of logos is on the right. A link 'Acquisitions & Subsidiaries' is provided for those wondering why GitHub or Heroku are not listed. The bottom section, 'WHAT'S NEW Ruby Consultancies That Build Great Software', describes consultancies that use Ruby and includes a link to learn more.

TOP RUBY COMPANIES Subsidiaries **New** Consultancies Coming Soon Startups + Add Your Company

Explore Top Ruby Companies Around the World

Discover top Ruby companies worldwide and see the impact of the programming language that focuses on developer happiness.

48 Public Companies **\$251 Billion** Total Market Cap **\$62.7 Billion** Total TTM Revenue


















Wondering why GitHub or Heroku are not even here? Have a look at [Acquisitions & Subsidiaries](#)

WHAT'S NEW Ruby Consultancies That Build Great Software

Consultancies that use Ruby help startups and enterprises build great software. From the home country of the language to even 🇺🇦 Ukraine, Ruby Consultancies are everywhere.

[Learn more about Ruby Consultancies](#)

Public Ruby Companies

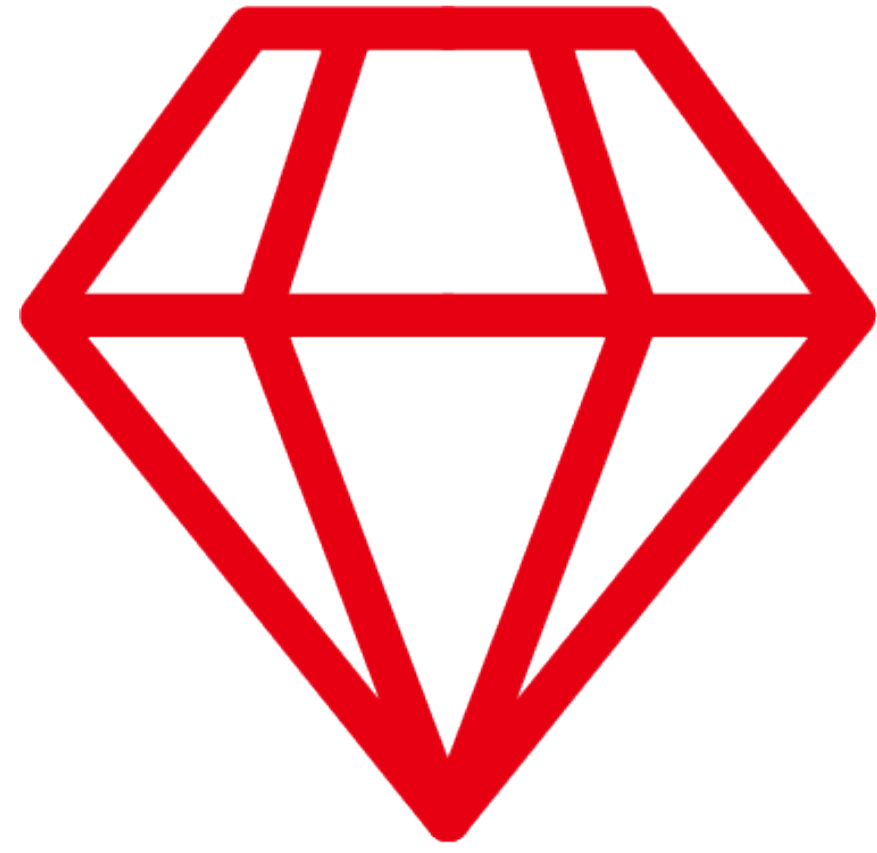
	Name	Exchange	HQ	Market Cap
# 1	 Airbnb airbnb.com	NASDAQ ABNB	 San Francisco, CA	\$73.7 B
# 2	 Shopify shopify.com			\$43.4 B
# 3	 Square block.xyz	NYSE SQ	 San Francisco, CA	\$36.2 B
# 4	 Coinbase coinbase.com	NASDAQ COIN	 Wilmington, DE	\$16.2 B
# 5	 Zendesk zendesk.com	NYSE ZEN	 San Francisco, CA	\$9.45 B
# 6	 Procore procore.com	NYSE PCOR	 Carpinteria, CA	\$7.45 B
# 7	 GitLab about.gitlab.com	NASDAQ GTLB	 San Francisco, CA	\$7.30 B
# 8	 On Running on-running.com	NYSE ONON	 Zurich, Switzerland	\$5.65 B
# 9	 Doximity doximity.com	NYSE DOCS	 San Francisco, CA	\$5.09 B



Why learn Ruby?



 **Technical Section** 



YJIT

I want to write Ruby



x86_64 Only

<http://github.com/tenderlove/tenderjit>

How to write a JIT compiler?

JIT Compiler Ingredients

Assembler

アセンブラ

Executable Memory

実行出来るメモリー

AArch64 Gem

<https://github.com/tenderlove/aarch64>

Machine Code: Just Bytes

機械語：バイトだけ

AArch64 Example

JIT a function that returns 0xF00DCAFE

JIT Code

```
require "aarch64"
require "jit_buffer"

# create a JIT buffer
jit_buffer = JITBuffer.new 4096

asm = AArch64::Assembler.new

# Make some instructions
asm.pretty do
  asm.movz x0, 0xCAFE
  asm.movk x0, 0xF00D, lsl(16)
  asm.ret
end

# Write the instructions to a JIT buffer
jit_buffer.writeable!
asm.write_to jit_buffer
jit_buffer.executable!

# Execute the JIT buffer
p jit_buffer.to_function([], -Fiddle::TYPE_INT).call.to_s(16) #
=> f00dcafe
```

Terminal Output

```
$ gel exec ruby test.rb
"f00dcafe"
```

AArch64 Example

JIT a function that returns 0xF00DCAFE

JIT Code

```
asm = AArch64::Assembler.new

# Make some instructions
asm.pretty do
  asm.movz x0, 0xCAFE
  asm.movk x0, 0xF00D, lsl(16)
  asm.ret
end
```

Terminal Output

```
$ gel exec ruby test.rb
"f00dcafe"
```

AArch64 Example

JIT a function that returns 0xF00DCAFE

JIT Code

```
require "aarch64"
require "jit_buffer"

# create a JIT buffer
jit_buffer = JITBuffer.new 4096

asm = AArch64::Assembler.new

# Make some instructions
asm.pretty do
  asm.movz x0, 0xCAFE
  asm.movk x0, 0xF00D, lsl(16)
  asm.ret
end

# Write the instructions to a JIT buffer
jit_buffer.writeable!
asm.write_to jit_buffer
jit_buffer.executable!

# Execute the JIT buffer
p jit_buffer.to_function([], -Fiddle::TYPE_INT).call.to_s(16) #
=> f00dcafe
```

Terminal Output

```
$ gel exec ruby test.rb
"f00dcafe"
```

AArch64 Example

JIT a function that returns 0xF00DCAFE

JIT Code

```
require "aarch64"
require "jit_buffer"

# create a JIT buffer
jit_buffer = JITBuffer.new 4096

asm = AArch64::Assembler.new

# Make some instructions
asm.pretty do
  asm.movz x0, 0xCAFE
  asm.movk x0, 0xF00D, lsl(16)
  asm.ret
end

# Write the instructions to a JIT buffer
jit_buffer.writeable!
asm.write_to jit_buffer
jit_buffer.executable!

# Execute the JIT buffer
p jit_buffer.to_function([], -Fiddle::TYPE_INT).call.to_s(16) #
=> f00dcafe
```



JITBuffer.new

Terminal Output

```
$ gel exec ruby test.rb
"f00dcafe"
```


JITBuffer: Executable Memory

Linux / macOS

Any Platform

Test Code Execution

Write some bytes and ensure it can execute them

```
# create a JIT buffer  
jit_buffer = JITBuffer.new 4096
```

メモリーを割
る当てる

```
# Write the instructions to a JIT buffer  
jit_buffer.writeable!  
jit_buffer.write asm.to_binary  
jit_buffer.executable!
```

コードを
書き込む

```
# Execute the JIT buffer  
p jit_buffer.to_function([],  
  -Fiddle::TYPE_INT).call
```

実行する

Test on x86_64 *and* ARM

Test Code Execution

Add a check for the current platform

```
# create a JIT buffer
jit_buffer = JITBuffer.new 4096

# Write the instructions to a JIT buffer
jit_buffer.writeable!

if arm64?
  jit_buffer.write arm64_bytes
else
  jit_buffer.write x86_bytes
end

jit_buffer.executable!
```

Test Code Execution

Add a check for the current platform

```
# create a JIT buffer
jit_buffer = JITBuffer.new 4096

# Write the instructions to a JIT buffer
jit_buffer.write instructions

if arm64?
  jit_buffer.write arm64_bytes
else
  jit_buffer.write x86_bytes
end

jit_buffer.executable!
```



How to determine the platform?

if 文って複雑

if 文って格好悪い

CPUに聞いたらどう？

Machine Code is Just Bytes

Bytes Valid for
x86_64 and ARM64?

YES!

一番最高のアイデア

x86 Machine Code

格好いい Code (Intel Syntax)

```
.foo:  
mov rax, 0x2b  
ret  
jmp foo
```

関数から返す

"foo" ラベルまで
ジャンプ????

x86 Machine Code Bytes

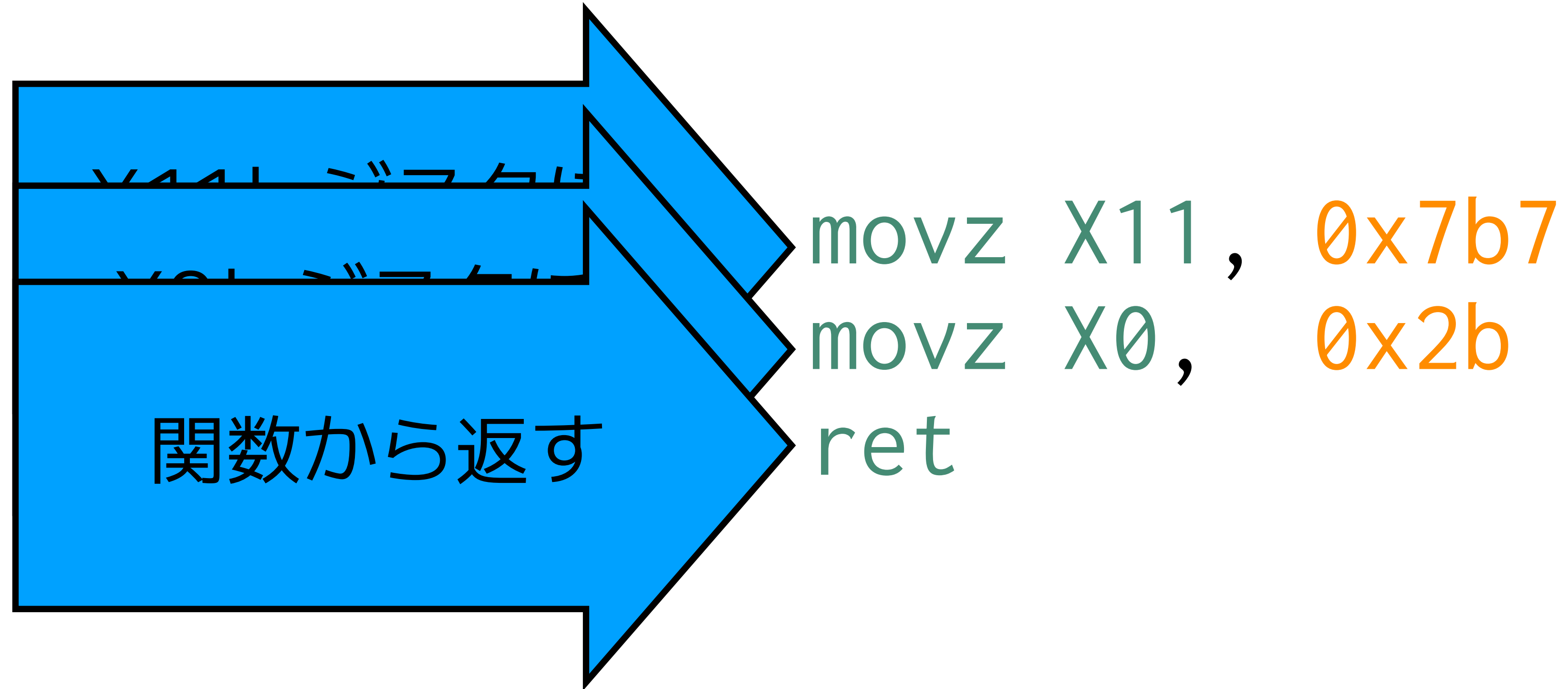
Bytes are in base 16

48	C7	C0	2B	00	00	00	mov rax, 0x2b
C3							ret
EB	F6						jmp foo

[0x48, 0xc7, 0xc0, 0x2b, 0x00, 0x00, 0x00, 0xc3, 0xeb, 0xf6]

ARM64 Machine Code

超かっこいい Code



ARM64 Machine Code Bytes

Bytes are in base 16

EB	F6	80	D2				movz x11, 0x7b7
60	05	80	D2				movz x0, 0x2b
C0	03	5F	D6				ret

Machine Code for x86 and ARM64

x86_64 Machine Code

48	C7	C0	2B	00	00	00	mov rax, 0x2b
C3							ret
EB	F6						jmp foo

ARM64 Machine Code

EB	F6	80	D2				movz x11, 0x7b7
60	05	80	D2				movz x0, 0x2b
C0	03	5F	D6				ret

Machine Code for x86 and ARM64

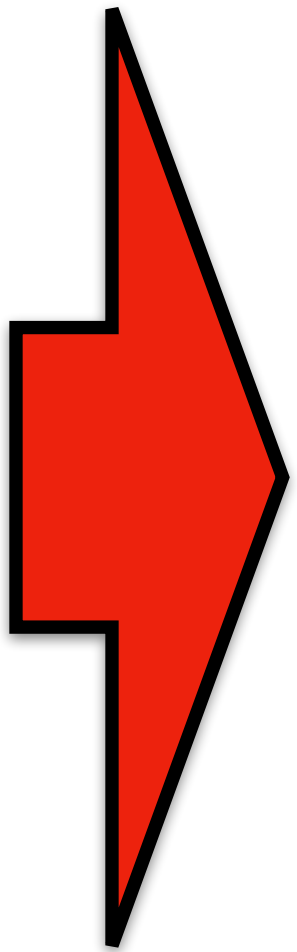
x86_64 Machine Code

48	C7	C0	2B	00	00	00	mov rax, 0x2b
C3							ret
EB	F6						jmp foo

ARM64 Machine Code

EB	F6	80	D2				movz x11, 0x7b7
60	05	80	D2				movz x0, 0x2b
C0	03	5F	D6				ret

Machine Code for x86 and ARM64



48	C7	C0	2B	00	00	00	mov rax, 0x2b
C3							ret
EB	F6						jmp foo

EB	F6	80	D2				movz x11, 0x7b7
60	05	80	D2				movz x0, 0x2b
C0	03	5F	D6				ret

Machine Code for x86 and ARM64

48	C7	C0	2B	00	00	00	mov rax, 0x2b
C3							ret
EB	F6	80	D2				jmp foo
60	05	80	D2				movz x0, 0x2b
C0	03	5F	D6				ret

Machine Code for x86 and ARM64

48	C7	C0	2B	00	00	00	mov rax, 0x2b
C3							ret
EB	F6	80	D2				movz x11, 0x7b7
60	05	80	D2				movz x0, 0x2b
C0	03	5F	D6				ret

Both Return 0x2B

JITBuffer Test Case

VCP (Very Cool Program)

```
jit = JITBuffer.new 4096
```

```
bytes = [0x48, 0xc7, 0xc0, 0x2b, 0x00, 0x00, 0x00, # x86_64 mov rax, 0x2b  
         0xc3, # x86_64 ret  
         0xeb, 0xf6, # x86 jmp  
         0x80, 0xd2, # ARM movz X11, 0x7b7  
         0x60, 0x05, 0x80, 0xd2, # ARM movz X0, #0x2b  
         0xc0, 0x03, 0x5f, 0xd6] # ARM ret
```

```
jit.writeable!
```

```
jit.write bytes.pack
```

```
jit.executable!
```

```
func = Fiddle::Function.new(jit.to_i + 8, :TYPE_INT)  
assert_equal 0x2b, func.call
```

Offset 8

Always
0x2b!!

Start at
offset 8

クロスプラットフォーム!!!!!!*

***x86_64 と ARM64 だけ**

If 文なし

完全に無複雑なコード

So Simple!

My Favorite Ruby Program

That I have written

```
jit = JITBuffer.new 4096

bytes = [0x48, 0xc7, 0xc0, 0x2b, 0x00, 0x00, 0x00, # x86_64 mov rax, 0x2b
         0xc3,                                     # x86_64 ret
         0xeb, 0xf6,                               # x86 jmp
         0x80, 0xd2,                               # ARM movz X11, 0x7b7
         0x60, 0x05, 0x80, 0xd2,                  # ARM movz X0, #0x2b
         0xc0, 0x03, 0x5f, 0xd6]                 # ARM ret

jit.writeable!
jit.write bytes.pack("C*")
jit.executable!

func = Fiddle::Function.new(jit.to_i + 8, [], Fiddle::TYPE_INT)
assert_equal 0x2b, func.call
```

Why learn Ruby?

Ruby is fun!

Ruby at Scale



shopify

10,000 Employees!



shopify

Type	Public
Traded as	TSX: SHOP (Class A) NYSE: SHOP S&P/TSX 60 component
ISIN	CA82509L1076
Industry	E-commerce
Founded	2006; 16 years ago Ottawa, Ontario, Canada
Founders	Tobias Lütke Daniel Weinand Scott Lake
Headquarters	Ottawa, Ontario, Canada
Area served	Worldwide
Services	Online shopping
Revenue	▲ US\$4.61 billion (2021) ^[1]
Net income	▲ US\$2.91 billion (2021) ^[1]
Total assets	▲ US\$13.34 billion (2021) ^[1]
Total equity	▲ US\$11.13 billion (2021) ^[1]
Number of employees	10,000+ ^[2]
Website	shopify.com

Thousands of Engineers

Shopify Core

Shopify Core

Code Statistics

Summary

Name	Lines	Lines of Code (LOC)	Classes (C)	Methods (M)	M/C	LOC/M
app	2170911	1751432	31857	128785	4	11
config	34706	30126	41	210	5	141
lib	93182	72187	1043	5359	5	11
test	4223476	3465671	26142	253154	9	11
Total	6521700	5318943	59072	387482	6	11

Code LOC: 5319416

Test LOC: 0

Code to Test Ratio: 1:0.0

1,751,432 lines of
application code

3,465,671 lines of
test code

5,318,943 lines of code

5,318,943 lines of Ruby code

Over 5 mil lines of code
Thousands of developers

How do we manage?

Build Infrastructure

**Thousands of Devs vs.
Millions of Lines**

Sorbet



Sorbet Example

```
class A
  extend T::Sig

  sig {
    def bar(x)
      x.to_s
    end
  end
end
```

The diagram illustrates the type annotations in the code. A large black arrow points from the `sig` block to the left. A black arrow labeled "Takes Integer" points from the `Integer` parameter to the left. A black arrow labeled "Returns String" points from the `String` return type to the right.

Book keeping

Optional

**~60% of Methods Have Types
in Shopify Core**

**The more you do,
the better it gets**

76% say: "Sorbet **caught an error**
in my code"

75% say: "Sorbet signatures **give me confidence** when I perform a code change"

Grows with you!

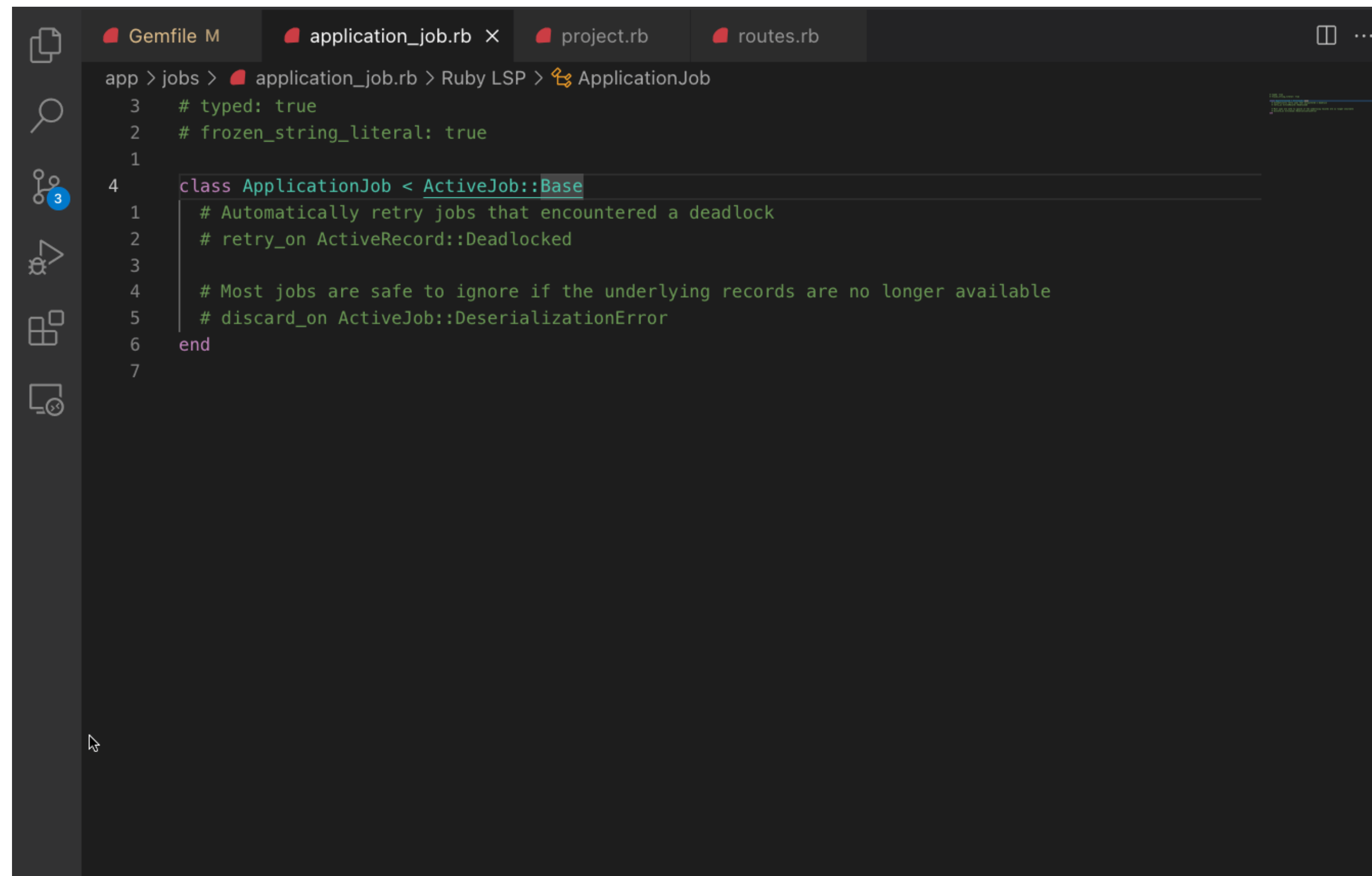
Ruby LSP

<https://github.com/Shopify/ruby-lsp>

Fix Indentation

```
example.rb U ●
example.rb > Person > initialize
6 require "pathname"
5
4 class Person
3   Layout/IndentationConsistency: Inconsistent
2   indentation
1   detected. RuboCop(Layout/IndentationConsistency)
11  View Problem Quick Fix... (⌘.)
1   @age = T.let(0, Integer)
2   end
3
4   sig { void }
5   def age!
6     @age += 1
7   end
8 end
9
10 person = Person.new
11 person.age!
12
```

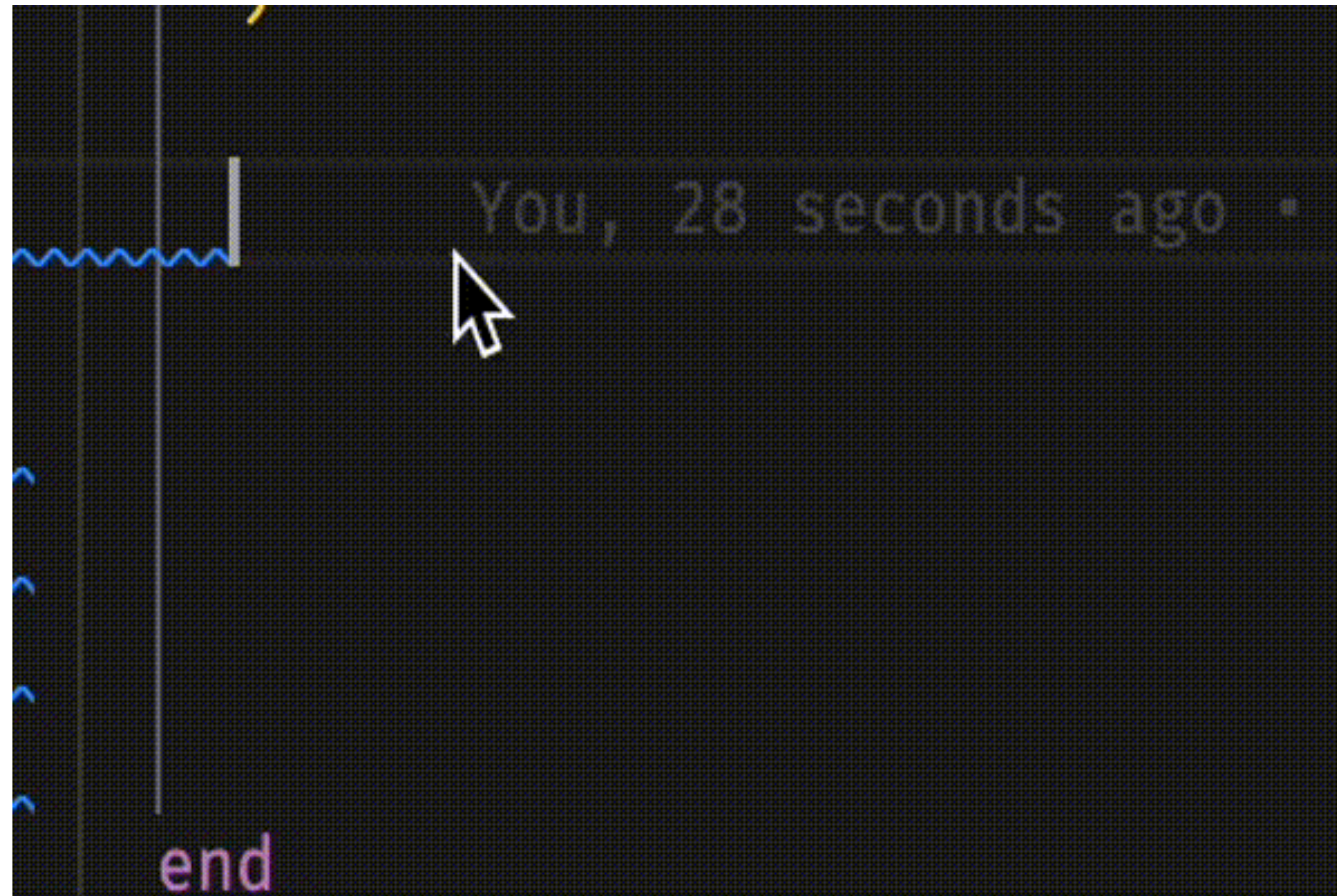
Show Documentation



The screenshot shows a code editor with several tabs: 'Gemfile M', 'application_job.rb X', 'project.rb', and 'routes.rb'. The current file is 'application_job.rb' and the cursor is on line 4, which contains the class definition 'class ApplicationJob < ActiveRecord::Base'. A tooltip is displayed over this line, showing the class name 'ApplicationJob' and a link to its documentation. The code in the background is as follows:

```
app > jobs > application_job.rb > Ruby LSP > ApplicationJob
3 # typed: true
2 # frozen_string_literal: true
1
4 class ApplicationJob < ActiveRecord::Base
1 # Automatically retry jobs that encountered a deadlock
2 # retry_on ActiveRecord::Deadlocked
3
4 # Most jobs are safe to ignore if the underlying records are no longer available
5 # discard_on ActiveRecord::DeserializationError
6 end
7
```

Code Completion



Why use Ruby?

It's easy to get started

Ruby scales *with your* business

Ruby in the Future

ErrorHighlight
^~~~~~

ErrorHighlight



Rails Integration

Thank you @mame!!!

NoMethodError in ApplicationController#hello

undefined method `gsub' for "Hello world":String

Did you mean?

gsub
gsub!

Extracted source (around line #4):

```
2 def hello
3   str = "Hello world"
4   str.gsub("foo", "bar")
5 end
6 end
```

NoMethodError in ApplicationController#hello

undefined method `[]' for nil:NilClass

Extracted source (around line #4):

```
2 def hello
3   json = { articles: { author: { name: "Yusuke" } } }
4   json[:article][:author][:name]
5 end
6 end
```

Syntax Suggest

Missing Keywords

"do" is missing

Unmatched `end`, missing keyword (`do`, `def`, `if`, etc.) ?

```
1 class Dog
2   def speak
> 3     @sounds.each |sound|
> 4     end
5   end
6 end
```

Editor Support

YARP

Yet Another Ruby Parser

Developer Ergonomics

Fast

Easy to Use

Get the job done.

Ruby

Thank you!