

Ruby on Rails

z czym to się je?

Marek Waligórski

InfoMEET Poznań

31 stycznia 2015



Ruby on Rails = ?



Ruby

Język programowania

Rails

Framework MVC

Narzędzia

I kilka przydatnych bibliotek

Ruby

Ruby



Język programowania

- skryptowy (interpretowany)
- zorientowany obiektowo
- dynamicznie typowany

Ruby



Język programowania

- skryptowy (interpretowany)
- zorientowany obiektowo
- dynamicznie typowy

Übung macht den Meister

Wypróbuj Ruby'ego bez instalacji




 tryruby.org



try ruby 

Give Ruby a shot right now!

Ruby is a programming language from Japan (available at ruby-lang.org) which is revolutionizing the web. The beauty of Ruby is found in its balance between simplicity and power.



Interactive ruby ready.

```
> puts "Hello!"  
=> "Hello!"  
  
> RUBY_VERSION  
=> "1.9.3"  
  
> 
```

Koduj albo giń!



www.bloc.io/ruby-warrior

You see before yourself a long hallway with stairs at the end. There is nothing in the way.
Call warrior.walk! to walk forward in the Player 'play_turn' method.

NEW ABILITIES

♥ 20



```
1  
2 class Player  
3   def play_turn(warrior)  
4     # cool code goes here  
5   end  
6 end  
7
```

RUN!

Dlaczego warto poznać Ruby'ego?



Elegancja

Elegancja



Ruby brzmi jak prawdziwy język

```
exit unless "restaurant".include? "aura"

['toast', 'cheese', 'wine'].each { |food| print food.capitalize }

population = 12_000_000_000

(1..10).include? 5
```

Wszystko jest obiektem

Wszystko jest obiektem



```
5.class    # => Fixnum
5 + 3      # => 8
5.+(3)     # => 8
```

Nic nie jest wieczne

Nic nie jest wieczne



```
5.class    # => Fixnum
5 + 3      # => 8
5.+(3)     # => 8
```

Psikus!

```
class Fixnum
  def +(x)
    self - x
  end
end
```

```
5 + 3      # => 2
```

Język dynamiczny

Bardzo dynamiczny

Nic nie jest wieczne



Stałe

```
CONSTANT = 5
```

Nic nie jest wieczne



Stałe

```
CONSTANT = 5
```

Psikus!

```
CONSTANT = 5  
CONSTANT = 0  
# warning: already initialized constant CONSTANT  
puts CONSTANT  
# => 0
```

With great power comes great
responsibility.

Czy Ruby ma jakieś zalety?

Ma.

Elegancja



Pętla

```
for (int i=0; i<5; i++) {  
    printf("Hello!\n");  
}
```

Elegancko



Pętla

```
for (int i=0; i<5; i++) {  
    printf("Hello!\n");  
}
```


Nie twórzmy niepotrzebnych
zmiennych

Elegancja



Pętla w C++

```
for (int i=0; i<5; i++) {  
    printf("Hello!\n");  
}
```

Pętla w Ruby (bez zmiennej)

```
5.times { puts "Hello!" }
```

Zmienna grzecznie czeka na drugim
planie

Elegancja



Ruby nie narzuca ograniczeń

```
5.times { |i| puts i }
```

```
0  
1  
2  
3  
4
```

Jedna linijka

Elegancja



Podnieś do kwadratu wszystkie liczby w tablicy

```
tablica = []  
for i in [1,2,3,4]  
  tablica << i ** 2  
end  
tablica    # => [1,4,9,16]
```

Elegancko



Podnieś do kwadratu wszystkie liczby w tablicy

```
tablica = []  
for i in [1,2,3,4]  
  tablica << i ** 2  
end  
tablica # => [1,4,9,16]
```

Elegancja



Podnieś do kwadratu wszystkie liczby w tablicy

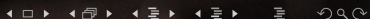
```
(1..4).map { |i| i ** 2 } # => [1,4,9,16]
```


Panie przodem



Sortowanie według ostatniej litery imienia?

```
["Stefan", "Tomek", "Agata", "Maja"].sort_by { |imie| imie[-1] }  
  
# => ["Maja", "Agata", "Tomek", "Stefan"]
```



Elegancja



Losowanie drużyn do turnieju

```
gracze = ["Ela", "Ola", "Agnieszka",  
          "Marek", "Wojtek", "Marcin"]
```

Elegancja



Losowanie drużyn do turnieju

```
gracze = ["Ela", "Ola", "Agnieszka",  
          "Marek", "Wojtek", "Marcin"]  
  
gracze.shuffle.each_slice(2).to_a  
  
# => [{"Agnieszka", "Marek"},  
#      {"Wojtek", "Ola"},  
#      {"Marcin", "Ela"}]
```

Duże liczby?



No problem!

```
7 ** 200
=> 1046183829131435717501889961181681
365981918855017023365995014008403512576
742426225177438261490936405029306524825
254631417406318034368359118815075426733
9816534637456120001
```

```
7.class
# => Fixnum
(7 ** 200).class
# => Bignum
```

Meta-programowanie

Metaprogramowanie



Sztuczne metody

```
class Auto
  def wypisz_predkosc
    puts "predkosc = #{predkosc}"
  end

  def wypisz_moc
    puts "moc = #{moc}"
  end

  private

  def predkosc
    "200 km/h"
  end

  def moc
    120
  end
end
```

Meta-programowanie



Sztuczne niemożliwy

```
class Auto
  def wypisz_predkosc
    puts "predkosc = #{predkosc}"
  end

  def wypisz_moc
    puts "moc = #{moc}"
  end

  private

  def predkosc
    "200 km/h"
  end

  def moc
    "120"
  end
end
```


REPEAT YOURSELF



YOU MUST NOT

```
obiekt.send(nazwa_metody)
```

Metaprogramowanie



Wywołanie wybranej metody

```
class Auto
  def wypisz(co)
    puts "#{co} = #{send(co)}"
  end

  private

  def predkosc
    "200 km/h"
  end

  def moc
    120
  end
end

a = Auto.new
a.wypisz('moc')
# moc = 120
```

Ruby
potrafi wywołać metodę

Ruby
potrafi wywołać metodę
która

Ruby
potrafi wywołać metodę
która

NIE ISTNIEJE

Metaprogramowanie

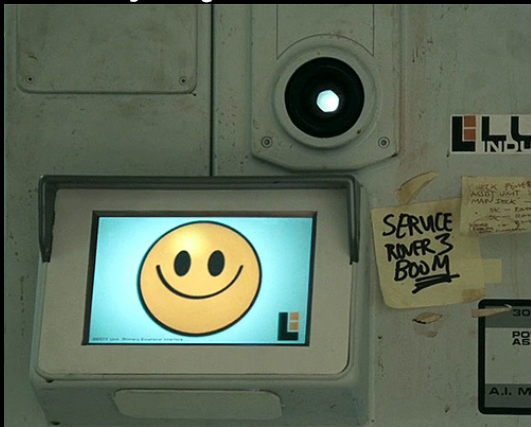


Wywołanie nieistniejącej metody

```
class A
  def hello
    puts "Hello, world!"
  end
end

a = A.new
a.hello
# "Hello, world!"
a.inna_metoda
# NoMethodError: undefined method 'inna_metoda' for a:A
```

Obiekty są samoświadome



Metaprogramowanie



Pomocny program

```
class A
  def hello
    puts "Hello, world!"
  end

  private

  def method_missing(name, *args)
    puts "Czy miales na mysli: #{self.methods.join(', ')}"
  end
end
```

Metaprogramowanie



Pomocny program

```
class A
  def hello
    puts "Hello, world!"
  end

  private

  def method_missing(name, *args)
    puts "Czy miales na mysli: #{self.methods.join(', ')}"
  end
end

a = A.new
a.inna_metoda

# Czy miales na mysli: hello, nil?, ==, =~, !~, eql?, hash, <=>, class,
  singleton_class, clone, dup, taint, tainted?, untaint, untrust, untrusted?,
  trust, freeze, frozen?, to_s, inspect, methods, singleton_methods,
  protected_methods, private_methods, public_methods, instance_variables,
  instance_variable_get, instance_variable_set, instance_variable_defined?,
  remove_instance_variable, instance_of?, kind_of?, is_a?, tap, send,
  public_send, respond_to?, extend, display, method, public_method,
  singleton_method, define_singleton_method, object_id, to_enum, enum_for,
  ==, equal?, !, !=, instance_eval, instance_exec, __send__, __id__
```

Metaprogramowanie



Trochę bardziej pomocny program

```
class A
  def hello
    puts "Hello, world!"
  end

  private

  def method_missing(name, *args)
    metody = self.methods - Object.methods
    puts "Czy miales na mysli: #{metody.join(', ')}"
  end
end

a = A.new
a.inna_metoda
# Czy miales na mysli: hello
```

To nie wszystko!



 **rubygems.org**



GEMS GUIDES CONTRIBUTE SIGN IN SIGN UP

Find, install, and publish
RubyGems.

Search Gems...



4,387,135,193

DOWNLOADS & COUNTING

Install RubyGems



95,447 gemów (29/01/2015)

Biblioteki



Gemfile

```
source 'https://rubygems.org'

gem 'rails', '3.2.21'
gem 'capistrano', '>= 2.15', require: false
gem 'netguru', :git => 'git://github.com/netguru/netguru.git'

gem 'rollbar', '1.4.0'
gem 'rails-backbone'
gem 'bson_ext'
gem 'carrierwave', '~> 0.9.0'
gem 'carrierwave-mongoid', require: 'carrierwave/mongoid'
gem 'colorize', require: false
gem 'dalli'
gem 'daemons'
gem 'decent_exposure'
gem 'draper'
...
```

Z czym się tego nie je?



- Gry komputerowe

Z czym się tego nie je?



- Gry komputerowe
 - Czyszczenie pamięci (mark & sweep)

Z czym się tego nie je?



- Gry komputerowe
 - Czyszczenie pamięci (mark & sweep)
- Zadania wymagające wysokiej wydajności

Z czym się tego nie je?



- Gry komputerowe
 - Czyszczenie pamięci (mark & sweep)
- Zadania wymagające wysokiej wydajności
 - kodu w Ruby'ego nie da się kompilować (w większości wersji)

Z czym się tego nie je?



- Gry komputerowe
 - Czyszczenie pamięci (mark & sweep)
- Zadania wymagające wysokiej wydajności
 - kodu w Ruby'ego nie da się kompilować (w większości wersji)
 - można integrować z modułami C++

Z czym się tego nie je?



- Gry komputerowe
 - Czyszczenie pamięci (mark & sweep)
- Zadania wymagające wysokiej wydajności
 - kodu w Ruby'ego nie da się kompilować (w większości wersji)
 - można integrować z modułami C++
- Wielowątkowość

Z czym się tego nie je?



- Gry komputerowe
 - Czyszczenie pamięci (mark & sweep)
- Zadania wymagające wysokiej wydajności
 - kodu w Ruby'ego nie da się kompilować (w większości wersji)
 - można integrować z modułami C++
- Wielowątkowość
 - Zielone wątki, GIL

Z czym się tego nie je?



- Gry komputerowe
 - Czyszczenie pamięci (mark & sweep)
- Zadania wymagające wysokiej wydajności
 - kodu w Ruby'ego nie da się kompilować (w większości wersji)
 - można integrować z modułami C++
- Wielowątkowość
 - Zielone wątki, GIL
- Grafika, wykresy, analiza danych

Z czym się tego nie je?



- Gry komputerowe
 - Czyszczenie pamięci (mark & sweep)
- Zadania wymagające wysokiej wydajności
 - kodu w Ruby'ego nie da się kompilować (w większości wersji)
 - można integrować z modułami C++
- Wielowątkowość
 - Zielone wątki, GIL
- Grafika, wykresy, analiza danych
 - Brak dobrych bibliotek

Oryginalna przyczyna popularności Ruby'ego?

Technologie sieciowe

Rails

Zasady



CoC, DRY, MVC

CoC



MVC



TDD?



Big sites made with RoR



Hulu, YellowPages, GitHub, Twitter (początkowo), Ask.fm

Co jeszcze dał nam Ruby?



- LessCSS / SASS
- HAML
- CoffeeScript
- Homebrew

Co dalej?



WZORY CHEMICZNE PROBOWKI I TAKIE TAM****



GO RAILS

powered by
One Month

Screencasts ▾

Setup Rails

Deploy Rails

Guides ▾

Forum

Sign Up

Login

Continue your Ruby on Rails education.

Video lessons so you can keep learning how to be a better developer.

Screencasts for Ruby on Rails Developers

Our screencasts are here to fill in the gaps, explain the confusing pieces, and give you an understanding you can't find elsewhere. There is a lot of "magic" that happens in Ruby on Rails and it's hard to wrap your head around so many different gems and concepts. You'll get to see my actual development environment, how I think about and design code, and learn how to become a confident developer.

Latest Screencasts



Fragment Caching And oEmbed

Pro Episode

Become a world-class Ruby on Rails developer

Monthly tips for developers
looking to become the best.





 railscasts.com

 Search Episodes[Browse Episodes](#)[RailsCasts Pro](#)[Notifications](#)[About](#)[Feedback](#)

Please read for an updated status on RailsCasts:

[Learn more](#) or [hide this](#)



EPISODE #417 - Jun 16, 2013 - 66 comments

Foundation

ZURB's Foundation is a front-end for quickly building applications and prototypes. It is similar to Twitter Bootstrap but uses Sass instead of LESS. Here you will learn the basics of the grid system, navigation, tooltips and more. (11 minutes)

[Watch Episode](#)[Read Episode](#)

EPISODE #416 - Jun 03, 2013 - 45 comments

Form Objects

Models have a tendency to become a complex mess as an application grows. In this episode you will learn a couple of techniques to extract form-behavior out into its own class. (18 minutes)

[Subscribe to Watch Episode](#)

View:  

Types

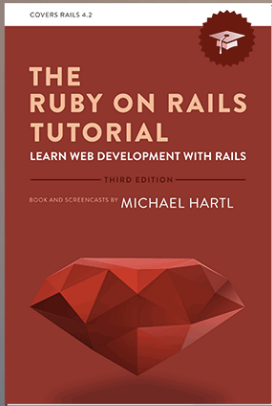
[Free Episodes](#)
[Pro Episodes](#)
[Revised Episodes](#)

Categories

[Active Record](#)
[Active Resource](#)
[Active Support](#)
[Administration](#)
[Ajax](#)
...



www.railstutorial.org



RUBY ON RAILS TUTORIAL (3RD ED.)

Learn Web Development with Rails

Michael Hartl

The Ruby on Rails Tutorial book and screencast series teach you how to develop and deploy real, industrial-strength web applications with Ruby on Rails, the open-source web framework that powers top websites such as Twitter, Hulu, GitHub, and the Yellow Pages. The Ruby on Rails Tutorial book is available for free online and is available for purchase as an ebook (PDF, EPUB, and MOBI formats). The companion screencast series includes 12 individual lessons, one for each chapter of the Ruby on Rails Tutorial book. All purchases also include a free copy of the Solutions Manual for Exercises, with solutions to every exercise in the book.

BUY NOW

FREE PREVIEWS

CONTACT AUTHOR

Share this book



Share

708



210

Darmowy serwer



 heroku.com



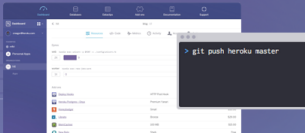
Features Pricing Add-ons Blog Documentation Support Contact Log in or [Sign up](#)

Focus on the app

[Sign up for free](#)

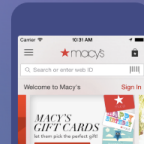
Powerful platform built
by developers for developers

Focus on building great apps →



Massively scalable, high
performance cloud platform

Engage customers with transformative apps →



Spotkania co miesiąc!



 prug.pl



Wykłady



Prelekcje



Spotkania z autorami różnych ciekawych bibliotek



Możliwość poznania poznających programistów Ruby on Rails

A jeszcze lepiej się poznacie, jeśli
zaaplikujecie do którejs z firm!****

Rekrutujemy!



 netguru.co



WE'RE HIRING

SERVICES PORTFOLIO TEAM OPEN SOURCE BLOG **CAREER** CONTACT

JOIN US
Junior RoR Developer

APPLY NOW

JOIN US
Poznań Ruby on Rails Developer

APPLY NOW

JOIN US
Remotely | Ruby on Rails Developer

APPLY NOW

JOIN US
Junior iOS Developer

APPLY NOW

JOIN US
Remotely | iOS Developer

APPLY NOW

JOIN US
Frontend Developer

APPLY NOW

Dziękuję za uwagę!

 github.com/marras/infomeet

