

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

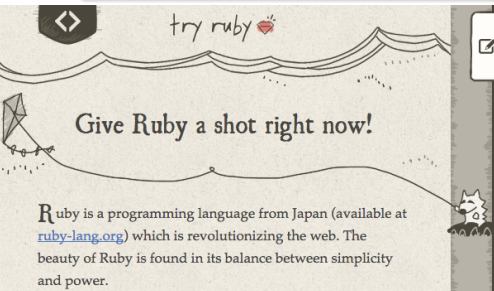
- skryptowy (interpretowany)
- zorientowany obiektowo
- dynamicznie typowany

Übung macht den Meister

Wypróbuj Ruby'ego bez instalacji



 tryruby.org



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?

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 - zmienna niepotrzebna

```
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
```

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
```

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

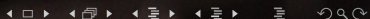
```
(1..4).map { |i| i ** 2 } # => [1,4,9,16]
```

Elegancja



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

TU OBARZEK HAL'A

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



Jeszcze bardziej pomocny

```
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
```

bloki (closures)



singleton classes



Gems for everything



+ Gemfile - so cool (bundle install)

Garbage Collection - Mark & Sweep



Wątki zielone



implementacje



* MRI - JRuby - różnice

Co nam dał?



- Technologie sieciowe - where it excels * lessCSS / SASS * HAML
- * CoffeeScript * Rails!!!

Z czym się tego nie je?



* GameDev * High-performance tasks (ale można integrować z modułami C++, ale nie ma np. bytecode'u - tylko AST) * Wątki - sztuczne (JRuby obsługuje kilka)

Zasady



CoC, DRY, MVC

CoC



MVC



TDD?



Jak zacząć?



Heroku EngineYard

PRUG



Big sites made with RoR



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

Rekrutujemy!



<https://netguru.co/career>