Infinum Student Academy

Ruby On Rails Course

Infinum

Design and software development agency

Clients

Awards

Deloitte.

SOMO BORAC

MIXX 2014

AWWWARDS

WEBSI 2013 SPLETNI PRVAKI

SAMSUNG

ERSTE BANK

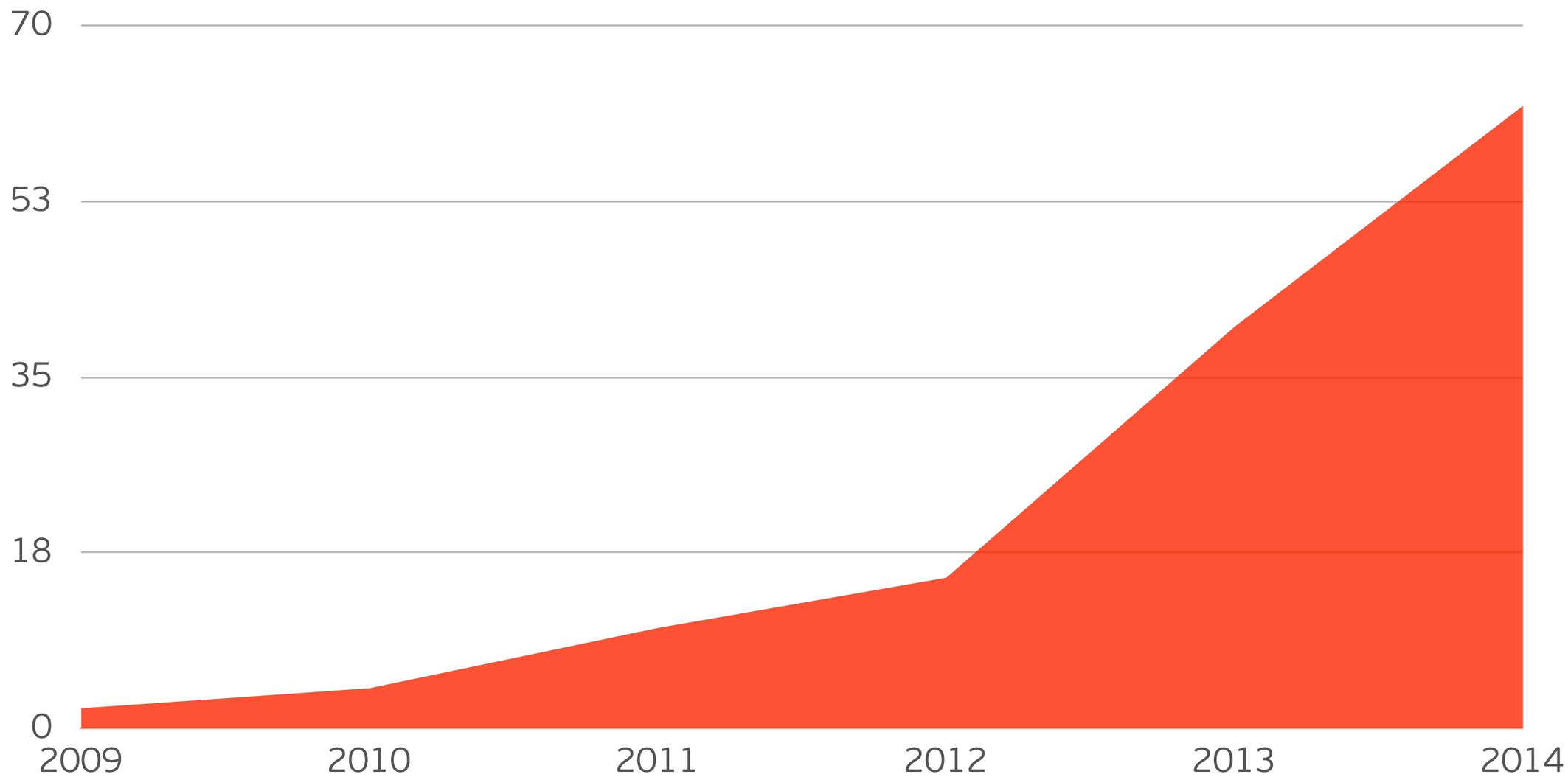
VIDI' 12 WEB TOP 100

vip

T··Com···

10 years ago / First year of university (FER)

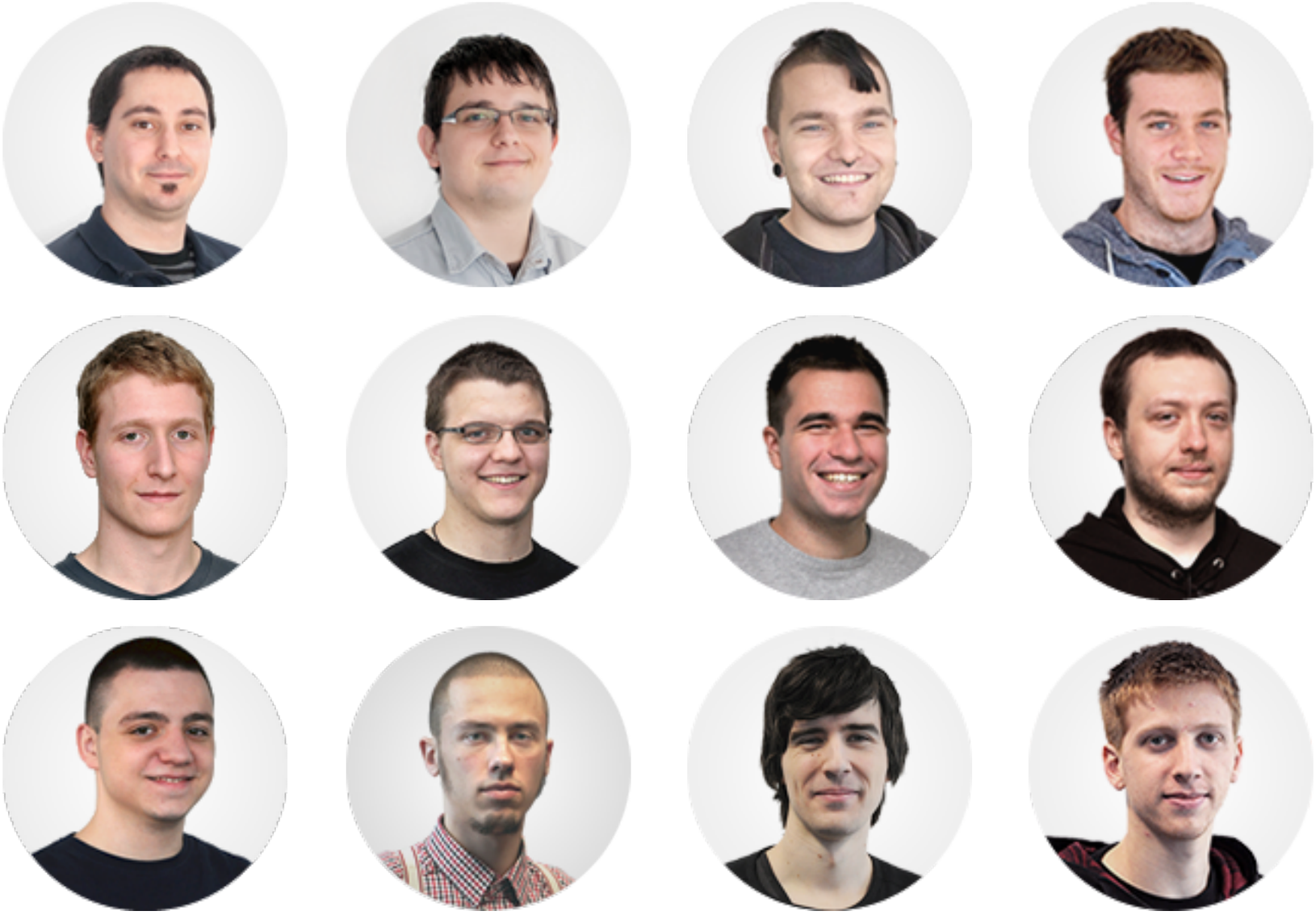
People



Teams

Android Team	iOS Team
Design Team	Javascript Team
WebSites Team	WebApps Team
Business Support Team	Management

Web Applications Team



Ruby On Rails Course

Curriculum, Calendar, Lecturers, Course Goals

Ruby On Rails Lecturers

Stjepan Hadjic

Web Applications
Team Lead

Gabrijel Skoro

Web Sites
Team Lead

Damir Svrtan

Web Applications
Team Lead

Jan Varljen

Productive.io
Technical Lead

2015

JULY

M	T	W	T	F	S	S
29	30	1 1st lecture	2	3 2nd lecture	4	5
6 3rd lecture	7	8	9 4th lecture	10	11	12
13 5th lecture	14	15	16	17 6th lecture	18	19
20 7th lecture	21	22	23 8th lecture	24	25	26

RoR Course Curriculum

4 weeks, 8 lectures

- Informative lecture + Intro to Ruby
- Web Architecture / Intro to Rails
- Database Persistance Layer & Forms
- Presentation Layer + User Authentication
- Advanced Active Record
- Active Support, Action Mailer, Image Uploading
- Creating an API
- Testing

RoR Course Goal

Learn how to build a web application with Ruby On Rails

Build your own reddit

MY SUBREDDITS FRONT - ALL - RANDOM | FUNNY - PICS - TODAYILEARNED - ASKREDDIT - WORLDNEWS - POLITICS - TECHNOLOGY - ATHEISM - SC

reddit RUBY hot new rising controversial top gilded promoted

1 Best resources for testing? (self.ruby)
11 points submitted 8 hours ago by drew_lee 6 comments share save hide report pocket

2 Gkv: Git As A Key/Value Store (github.com)
5 points submitted 8 hours ago by _devbob 1 comment share save hide report pocket

3 [HELP] How do I package a green_shoes application to a Mac/Windows app? (self.ruby)
1 point submitted 20 minutes ago by chipcrazy comment share save hide report pocket

4 Creating easy, readable attributes with ActiveRecord enums (justinweiss.com)
1 point submitted an hour ago by jakubgarfield comment share save hide report pocket

5 Is that what we're doing? Sharing our gems? Gem to create an API reference spec / (github.com)
3 points submitted 8 hours ago by rickcarlino 1 comment share save hide report pocket

6 Mashing together DRb and Net::SSH? (self.ruby)
9 points submitted 15 hours ago * by Funktapus 2 comments share save hide report pocket

Homework

- 8 homeworks in total
- All homework must be submitted within one week
- Pull Request Flow with Github

The Rules of the Game

- Total maximum of one absence through 8 lectures
- All homework must be submitted

Communication & Links

Github:

- https://github.com/InfinumAcademy
- https://github.com/InfinumAcademy/rubyonrails-materijali

Emails:

- ror.academy@infinum.hr
- damir.svrtan@infinum.hr, stjepan.hadjic@infinum.hr, jan.varljen@infinum.hr, gabrijel.skoro@infinum.hr

Slack:

- https://infinumacademy.slack.com

How to Slack

Channels

- #general
- #ruby-on-rails-tecaj
- privatni kanali

Tools

- Editor
- Terminal

Ruby

The Programming Language

Ruby

- invented in mid 1990s
- gained extreme popularity with Rails in mid 2000s

Very simple

Hello World in Java

```
public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("Hello, World");
    }

}
```

Hello World in Ruby

```
puts 'Hello World'
```

Extremely readable

```ruby
5.times { puts 'Hello World' }
```

amount_of_cash_in_dollars = 1000000000

amount of cash in dollars = 1 000 000 000

Interpreted, not compiled

```
>> touch hello_world.rb
>> echo 'puts "hello world"' >> hello_world.rb
>> ruby hello_world.rb
hello world
```

Dynamically typed

Java is statically typed

```
Integer lowBarrier = -8;
double division = -100;
float divisionee = -90;
char divisioner = 'a';
String message = "Hello";
```

Ruby is dynamically typed

```
low_barrier = -8
divisionee = -90.0
message = ‘Hello’
```

Highly Object oriented

Everything is an object

Get absolute value in Java

```
Integer a = -8;
Math.abs(a) => 8
```

Get absolute value in Ruby

```
-8.abs()
```

In Ruby parenthesis are optional

```
-8.abs()
```

```
-8.abs
```

More methods ...

```
0.zero? # => true
2.zero? # => false

0.nonzero? # => false
2.nonzero? # => true

0.2.floor # => 0
0.2.ceil # => 1

1.next #=> 2
1.pred #=> 0

1.next.pred #=> 1
```

Comes with a REPL

IRB/PRY

Standard Types

But actually they're all instances of classes

Numeric & Float

```
8.odd?
7.even?

7.123.round
```

String

```
"Hello World".size
"Hello World".count(‘o’)
"Hello World".empty?

“”.empty?

"Hello World".downcase #=> 'hello world'
"Hello World".reverse #=> 'dlroW olleH'
```

Array

```
[1, 2, 3].include?(1) #=> true
[1, 2, 3].include?(10) #=> false

[1, 3, 2].sort #=> [1, 2, 3]
[1, 2, 3].reverse #=> [3, 2, 1]

[1, 2, 3].sample #=> 3
```

Ranges

```
(1..5).include?(4) #=> true
(1..5).to_a #=> [1, 2, 3, 4, 5]
```

Hash

```
grades = {
 'Math' => 2,
 'Programming' => 4,
 'English' => 5
}

grades['Math'] #=> 2
grades['Programming'] #=> 4

grades.keys
#=> ["Math", "Programming", "English"]
```

Symbols

```
grades = {
 :Math => 2,
 :Programming => 4,
 :English => 5
}

grades[:Math] #=> 2
grades[:Programming] #=> 4

grades.keys
#=> [:Math, :Programming, :English]
```

Nil

```
nil

nil.nil? # => true

‘Hello World’.nil? # => false
```

Flow Control

IF statement

```
math_grade = 2
chem_grade = 5

if math_grade > chem_grade
 puts "I'm better at math"
elsif math_grade < chem_grade
 puts "I'm better at chemistry"
else
 puts "I'm equally good/bad at chemistry and math"
end
```

Trailing IF statement

```
math_grade = 1

if math_grade != 1
 puts “I’m passing math"
end
```

Trailing IF statement

```
puts “I’m passing math” if math_grade != 1
```

Unless statement

```
puts “I’m passing math” unless math_grade == 1
```

Enumerable

And how we aren’t using for loops

For loop in Ruby

```
for i in (0..5)
    puts i
end
```

Each method

```
(1..5).each { |i| puts i }
```

What are blocks?

```
(1..5).each { |i| puts i }
```

```
(1..5).each do |i|
  puts i
end
```

Enumerable methods

```
[1, 2, 3, 4, 5]
 .select { |number| number.odd? } # [1, 3, 5]
 .map    { |number| number * 2 } # [2, 6, 10]
 .reject { |number| number > 7 } # [2, 6]
```

Other Enumerable methods

```
.select
.find -> .detect
.map -> .collect
.reject
.any?
.include?
.reverse
.partition
.one?
```

Defining methods

Defining methods & calling them

```
def hello_world
 puts 'Hello World'
end

hello_world
```

Defining methods with arguments

```
def puts_upcased(string)
  puts string.upcase
end

puts_upcased('Hello World')
#=> "HELLO WORLD"
```

Optional parenthesis when defining and calling

```
def puts_upcased string
  puts string.upcase
end

puts_upcased 'Hello World'
#=> "HELLO WORLD"
```

Enough with these ‘void’ methods

Defining methods & calling them

```
def url_friendly(string)
  string.downcase.gsub(‘ ‘,’-‘)
end

friendly_url = url_friendly(‘Hello World’)

puts friendly_url
#=> ‘hello-world’
```

Implicit return

These two are the same

```
def url_friendly(string)
  string.downcase.gsub(‘ ‘,’-‘)
end
```

```
def url_friendly(string)
  return string.downcase.gsub(‘ ‘,’-‘)
end
```

Ruby Classes

User class

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

  def name
    @name
  end
end
```

```
user = User.new('Damir')

puts user.name #=> 'Damir'
```

Writing a class with a getter and setter

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

  def name
    @name
  end

  def name=(name)
    @name = name
  end
end
```

Using a setter

```
user = User.new('Damir')

puts user.name #=> 'Damir'

user.name = 'Ivan'

puts user.name #=> 'Ivan'
```

Let’s make this User class more real

Add a phone_number and age to the User class.

Writing a with a getter and setter

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

  def name
    @name
  end

  def name=(value)
    @name = value
  end

  def age
    @age
  end

  def age=(value)
    @age = value
  end

  def phone_number
    @phone_number
  end

  def phone_number=(value)
    @phone_number = value
  end
end
```

DRY up the getters

```ruby
class User
  def name
    @name
  end
end
```

```ruby
class User
 attr_reader :name
end
```

DRY up the getters

```
class User

  attr_reader :name, :phone_number, :age

  def initialize(name, phone_number, age)
    @name = name
    @phone_number = phone_number
    @age = age
  end

  def name=(value)
    @name = value
  end

  def age=(value)
    @age = value
  end

  def phone_number=(value)
    @phone_number = value
  end
end
```

DRY up the setters

```ruby
class User
  def name=(value)
    @name = value
  end
end
```

```ruby
class User
 attr_writer :name
end
```

DRY up the setters

```
class User

  attr_reader :name, :phone_number, :age
  attr_writer :name, :phone_number, :age

  def initialize(name, phone_number, age)
    @name = name
    @phone_number = phone_number
    @age = age
  end
end
```

DRY up the getters & setters even more

```
class User
 attr_reader :name
 attr_writer :name
end
```

```
class User
 attr_accessor :name
end
```

DRY up the setters & getters

```
class User
  attr_accessor :name, :phone_number, :age
  def initialize(name, phone_number, age)
    @name = name
    @phone_number = phone_number
    @age = age
  end
end
```

Before

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

  def name
    @name
  end

  def name=(value)
    @name = value
  end

  def age
    @age
  end

  def age=(value)
    @age = value
  end

  def phone_number
    @phone_number
  end

  def phone_number=(value)
    @phone_number = value
  end
end
```

After

```
class User

  attr_accessor :name, :phone_number, :age

  def initialize(name, phone_number, age)
    @name = name
    @phone_number = phone_number
    @age = age
  end
end
```

Homework

How to do your homework

- Clone your repo:
- academy-#{ime}-#{prezime}
- Solve 6 assignments
- Make sure all the tests are passing
- Commit your solution
- Push to a branch called HW1
- Create a pull request

Email me your Github profiles:

damir.svrtan@infinum.hr