

Programowanie w Ruby

Wykład 1

Marcin Młotkowski

3 października 2012

Plan wykładu

Sprawy organizacyjne

Wykład

- Źródła wiedzy

- Zaliczenia

O języku

- Historia i pochodzenie języka

- O języku

- Instrukcje złożone

- Procedury i funkcje

Informacje organizacyjne

Terminy konsultacji

poniedziałek 10:15–12:00, pok. 303

Strona wykładu

<http://www.ii.uni.wroc.pl/~marcinm/dyd/ruby/>

Plan wykładu

Sprawy organizacyjne

Wykład

Źródła wiedzy

Zaliczenia

O języku

Historia i pochodzenie języka

O języku

Instrukcje złożone

Procedury i funkcje

Cele wykładu

- ▶ Poznanie języka Ruby, jego składni i semantyki
- ▶ poznanie środowiska Ruby on Rails

Plan wykładu

1. Język Ruby: typy podstawowe, składania, semantyka
2. Typy wbudowane
3. refleksje (introspekcje)
4. wątki
5. środowisko graficzne
6. Ruby on Rails

Źródła internetowe

Oficjalna strona

<http://www.ruby-lang.org/pl/>

Źródła internetowe

Oficjalna strona

<http://www.ruby-lang.org/pl/>

Dokumentacja

<http://www.ruby-doc.org/>

Inne materiały internetowe

Patrz: strona wykładu

Książki do Ruby

-  L. Carlson, L. Richardson. Ruby. Receptury, Helion 2007
-  H. Fulton. Ruby. Tao programowania w 400 przykładach, Helion 2008
-  D. Flanagan, Y. Matsumoto. Ruby. Programowanie, Helion 2009
-  M.Fitzgerald. Ruby. Wprowadzenie, Helion 2007

Książki do Ruby on Rails



S. Holzner. Ruby on Rails, Helion 2008



B. Tate, L. Carlson, C. Hibbs. Ruby on Rails. Wprowadzenie, Helion 2009

Zasady prowadzenia pracowni

Pierwsza część semestru (ok. 10 tyg)

Po wykładzie będą ogłaszane listy krótkich zadań do zrobienia na najbliższą pracownię.

Zasady prowadzenia pracowni

Pierwsza część semestru (ok. 10 tyg)

Po wykładzie będą ogłaszane listy krótkich zadań do zrobienia na najbliższą pracownię.

Druga część semestru

Do wykonania będzie większy projekt, preferowane środowisko: Ruby on Rails

Zasady prowadzenia pracowni

Pierwsza część semestru (ok. 10 tyg)

Po wykładzie będą ogłaszane listy krótkich zadań do zrobienia na najbliższą pracownię.

Druga część semestru

Do wykonania będzie większy projekt, preferowane środowisko: Ruby on Rails

Zaliczenie semestru

Zdobycie przynajmniej połowy możliwych do zdobycia punktów.

Zasady prowadzenia pracowni

Pierwsza część semestru (ok. 10 tyg)

Po wykładzie będą ogłaszane listy krótkich zadań do zrobienia na najbliższą pracownię.

Druga część semestru

Do wykonania będzie większy projekt, preferowane środowisko: Ruby on Rails

Zaliczenie semestru

Zdobycie przynajmniej połowy możliwych do zdobycia punktów.

Egzamin

To jest kurs, więc egzaminu nie ma ;-)

Plan wykładu

Sprawy organizacyjne

Wykład

Źródła wiedzy

Zaliczenia

O języku

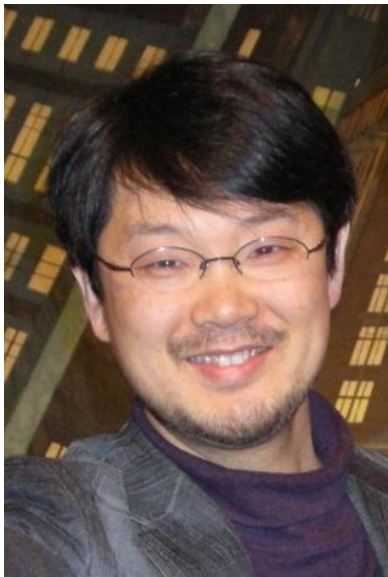
Historia i pochodzenie języka

O języku

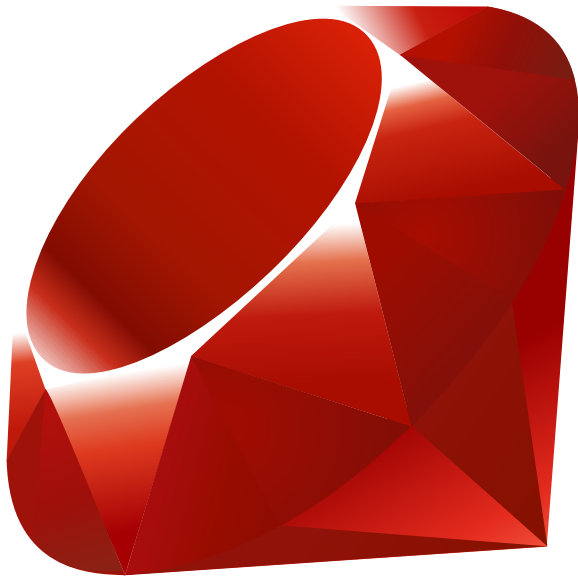
Instrukcje złożone

Procedury i funkcje

Yukihiro Matsumoto a.k.a Matz



Nazwa języka



Stan obecny języka

Aktualna wersja

1.9.3

Stan obecny języka

Aktualna wersja

1.9.3

Rozwój języka

Społeczność

Pochodzenie języka

- ▶ Trochę z Perla (\$, \$_)
- ▶ Trochę ze Smalltalka
- ▶ Trochę z Pythona

Co jest fajnego w Rubym

- ▶ programowanie strukturalne
- ▶ programowanie obiektowe
- ▶ programowanie funkcjonalne

Ogólnie o języku

- ▶ język skryptowy
- ▶ typowanie dynamiczne (duck typing)
- ▶ brak deklaracji typów
- ▶ sporo typów wbudowanych

Co jest fajnego w Ruby?

Skąd taka popularność?

Ruby on Rails!!!

Praca interaktywna

```
$ irb
irb(main):001:0> puts("A kuku")
A kuku
=> nil
irb(main):002:0> 2+2
=> 4
irb(main):003:0>
```


Wyrażenia w Ruby'm

Arytmetyka

- ▶ stałe: 1024, 3.1415
- ▶ wyrażenia: $(2 + 2) * 4$

Wyrażenia w Ruby'm

Arytmetyka

- ▶ stałe: 1024, 3.1415
- ▶ wyrażenia: $(2 + 2) * 4$

Napisy

'napis', "napis"

Składnia

Separacja instrukcji

- ▶ instrukcje są w kolejnych wierszach
- ▶ instrukcje można oddzielać średnikiem

Instrukcja warunkowa if

```
slowo = "dlugie"  
if "krotkie".length < slowo.length  
  puts 'Krótkie'  
else ; puts 'Długie' end
```

Instrukcja warunkowa case

```
zwierze = "hau"
case zwierze
  when 'hau', 'woof'
    puts 'pies'
  when 'miau'
    puts 'kot'
  else puts '???'
end
```

Jeszcze inna instrukcja warunkowa

```
puts "Zimno" if temperatura <= -20
```

```
puts "Pogoda" unless temp < 15 && temp > 20
```

Wyrażenia logiczne

- ▶ Fałsz: `nil` oraz `false`
- ▶ Operatory: `and`, `&&`, `or`, `||`, `not`, `!`
- ▶ `==`, `defined?`

Wyrażenia logiczne

- ▶ Fałsz: `nil` oraz `false`
- ▶ Operatory: `and`, `&&`, `or`, `||`, `not`, `!`
- ▶ `==`, `defined?`

Wartość wyrażenia logicznego

Wartością wyrażenia logicznego, gdy jest prawdziwe, jest ostatni operand.

Wyrażenia logiczne

- ▶ Fałsz: `nil` oraz `false`
- ▶ Operatory: `and`, `&&`, `or`, `||`, `not`, `!`
- ▶ `==`, `defined?`

Wartość wyrażenia logicznego

Wartością wyrażenia logicznego, gdy jest prawdziwe, jest ostatni operand.

Przykłady

<code>'nie'</code> and <code>"nie"</code>	<code>'nie'</code>
<code>false</code> and <code>99</code>	<code>false</code>

Pętlą while

```
a, b = 0, 1
while b < 10
    puts b
    a, b = b, a + b
end
```

Pętle for

```
for x in 1..10  
  puts x  
end
```

```
for el in [1,2,3,4,5,6,7,8,9,10]  
  sum = sum + el  
end
```

I jeszcze inne pętle

```
5.times do  
  puts "Hurra!\n"  
end
```

```
i += 1 while i < 100
```

Pętle, uzupełnienie

Instrukcja **break**

kończy pętlę

Pętle, uzupełnienie

Instrukcja **break**

kończy pętlę

Instrukcja **next**

przechodzi na koniec pętli

Pętle, uzupełnienie

Instrukcja **break**

kończy pętlę

Instrukcja **next**

przechodzi na koniec pętli

Instrukcja **redo**

powtarza pętlę od początku bez przeliczania warunku lub pobierana kolejnego elementu (w iteratorze)

Przykład

```
def suma(n)
  s=i=0
  for i in 1..n
    s += i
  end
  return s
end
```


Dygresja I

Kompilator wsadowy

```
$ ruby plik.rb
```

Dygresja I

Kompilator wsadowy

```
$ ruby plik.rb
```

Plik "wykonywalny"

```
#!/usr/bin/ruby  
def silnia(n)  
  if n == 0; 1  
  else  
    n * silnia(n-1)  
  end  
end
```

Dygresja II

Środowisko/edytor

vim, gedit, geany, notepad,

Dygresja II

Środowisko/edytor

vim, gedit, geany, notepad,

RubyMine firmy JetBrains

Produkt bezpłatny do celów edukacyjnych. Licencja: u mnie, przez maila.