

Whats up in
RUBY 2.0

DOMAS BITVINSKAS
domasbitvinskas.com
@Nedomas

27.days.ago

ruby can almost play poker legally

PR

.....
Syntax sparkle

Evolved monkeys

PERFORMANCE boooooost

“100 hundred procent compatibl” - Matz

UTF-8 default encoding

```
♥ = 'love'
```

```
puts ♥
```

```
class Integer
  def dalinasi_iš?(skaičius)
    skaičius != 0 ? self % skaičius == 0 : zero?
  end
end

puts 6.dalinasi_iš?(2)
```

kwargs

```
# ruby 1.9
```

```
def shout(opts={})  
  loud = opts[:loud] || 'Hello'  
  quiet = opts[:quiet] || 'World'  
  
  p "#{loud} #{quiet}!"  
end
```

```
shout loud: "Old", quiet: "School"
```

```
# ruby 2.0
```

```
def shout(loud: 'Hello', quiet: 'World')  
  p "#{loud} #{quiet}!"  
end
```

```
shout loud: "New", quiet: "School"
```

PRO-TIP

```
%i(an array of symbols)
```

```
=> [:an, :array, :of, :symbols]
```

```
%I("#{100-50} shades of grey)
```

```
=> [:"50", :shades, :of, :grey]
```


Module#prepend

```
module Logger
  def log(msg)
    p msg
  end
end
```

```
class Logic
  include Logger
  def log(msg)
    p msg.reverse
  end
end
```

Module#prepend

```
module Logger
  def log(msg)
    p msg
  end
end
```

```
class Logic
  prepend Logger
  def log(msg)
    p msg.reverse
  end
end
```

```
Logic.new.log("Some message")
# "Some message"
```

```
Logic.ancestors
# include => [Logic, Logger, Object, Kernel, BasicObject]
# prepend => [Logger, Logic, Object, Kernel, BasicObject]
```

ENUMERATORS

.....

for the lazy

```
even_numbers = (1..10000).select(&:even?).map(&:to_s)
```

```
even_numbers = (1..10000).inject([]) do |acc, number|  
  if number.even?  
    acc << number.to_s  
  else  
    acc  
  end  
end
```

ENUMERATORS

.....

for the lazy

```
even_numbers = (1..10000).lazy.select(&:even?).map(&:to_s)
```

```
even_numbers = (1..10000).inject([]) do |acc, number|  
  if number.even?  
    acc << number.to_s  
  else  
    acc  
  end  
end
```

REFINEMENTS

experimental

```
module Fun
  refine Fixnum do
    def +(_)
      super - rand(3)
    end
  end
end
using Fun
p 5 + 5
```

.....

MAŽI GRAŽŪS DALYKAI

.....

.to_h

```
Drummer = Struct.new(:name, :age, :band)
```

```
idol = Drummer.new("Carter Beauford", 55, "Dave Matthews Band")
```

```
p idol
```

```
# => #<struct Drummer name="Carter Beauford", age=55, band="Dave  
Matthews Band">
```

```
p idol.to_h
```

```
# => {:name=>"Carter Beauford", :age=>55, :band=>"Dave Matthews  
Band"}
```

ONIGMO



REGEX VARIKLIS

```
"this\nis\nvMr.\nrNeo".gsub( /\R/, ' ' )
```


Array Range#bsearch

```
p [1, 2, 3, 6, 9, 15].bsearch { |i| i > 4 }
```

```
p (1..1000000000000000).bsearch { |i| i > 9 }
```

dir

IMPROVE

.....

GARBAGE & STUFF!

RubyGems 2.0 & RDoc 4.0



AČIŮ