

Ruby master - Feature #4147

Array#sample

12/10/2010 10:06 PM - oj (Yoji Ojima)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	
Description	
<pre>=begin Array#sample 1000 omikuj_i_box = [{:name => " ", :weight => 1000}, {:name => " ", :weight => 100}, {:name => " ", :weight => 10}, {:name => " ", :weight => 1}] omikuj_i = omikuj_i_box.sample { v v[:weight] } puts omikuj_i[:name] =end</pre>	
Related issues:	
Related to Ruby master - Feature #4247: New features for Array#sample, Array#... Assigned	

History

#1 - 12/10/2010 10:22 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Feedback

```
=begin
Ruby
=end
```

#2 - 12/10/2010 10:45 PM - sorah (Sorah Fukumori)

```
=begin
Ruby
```

```
class Array
def sample(n=1)
if block_given?
self.size.times.map do |i|
Array.new(yield(self[i]), self[i])
end.flatten.sample(n)
else
n == 1 ? self[rand(self.size)] : Array.new(n){ self[rand(self.size)] }
end
end
end
```

```
# --
```

```
def a
[1,2,3].sample { |v| v*100 }
end
```

```
def b
[1,2,3].sample { |v| v*100 }
end
ary = 100.times.map{a}
bry = 100.times.map{b}
```

```

result_a = {}

ary.each do |i|
  result_a[i] ||= 0
  result_a[i] += 1
end

result_b = {}

bry.each do |i|
  result_b[i] ||= 0
  result_b[i] += 1
end

# a [*10] b {i => }

p result_a #=> {2=>31, 3=>56, 1=>13}
p result_b #=> {2=>37, 3=>45, 1=>18}

# -- sora_h
=end

```

#3 - 12/10/2010 10:47 PM - sorah (Sorah Fukumori)

```

=begin
[]

# -- []

def a
  [1,2,3].sample { |v| v*100 }
end

def b
  [1,2,3].sample { |v| v }
end
ary = 100.times.map{a}
bry = 100.times.map{b}

result_a = {}

ary.each do |i|
  result_a[i] ||= 0
  result_a[i] += 1
end

result_b = {}

bry.each do |i|
  result_b[i] ||= 0
  result_b[i] += 1
end

# a [*10] b {i => }

p result_a #=> {3=>56, 1=>13, 2=>31}
p result_b #=> {2=>34, 3=>46, 1=>20}

# -- sora_h
=end

```

#4 - 12/10/2010 10:58 PM - oj (Yoji Ojima)

```

=begin
[]

class Array
  alias :org_sample :sample

  def sample
    if block_given?
      prob_map = collect {|v| yield(v) }
      rn = rand * prob_map.inject(0) {|r, v| r += v }
      prob_map.each_with_index do |v, i|
        rn -= v
      end
    end
  end
end

```

```
return self[j] if rn < 0
end
else
org_sample
end
end
end
=end
```

#5 - 12/10/2010 11:00 PM - oj (Yoji Ojima)

```
=begin
[]
=end
```

#6 - 12/18/2010 04:20 PM - oj (Yoji Ojima)

```
=begin
[]
=end
```

#7 - 12/19/2010 01:37 AM - shyouhei (Shyouhei Urabe)

- Status changed from Feedback to Assigned
- Assignee set to shyouhei (Shyouhei Urabe)

```
=begin
[]
[]
=end
```

#8 - 12/19/2010 10:31 AM - hasari (Hiro Asari)

```
=begin
[]
[]
[]gem
=end
```

#9 - 12/19/2010 11:15 AM - oj (Yoji Ojima)

```
=begin
[]
[]
=end
```

#10 - 12/19/2010 11:25 AM - mame (Yusuke Endoh)

```
=begin
[]
2010年12月18日16:20 Yoji Ojima redmine@ruby-lang.org:
```

[]

```
API
each
[]
```

```
omikuji_box = %w( [] [] [] )
omikuji_weight = [1000, 1100, 1110, 1111]
omikuji = omikuji_box.sample(weight: omikuji_weight)
```

```
[]
each []
[]
```

~~~~~  
~~~~~  
~~~~~  
~~~~~ API ~~~~~  
~~~~~

--  
Yusuke Endoh [mame@tsq.ne.jp](mailto:mame@tsq.ne.jp)

=end

**#11 - 12/19/2010 11:44 AM - hasari (Hiro Asari)**

=begin  
On Dec 18, 2010, at 9:15 PM, Yoji Ojima wrote:

~~~~~

~~~~~  
~~~~~

~~~~~  
~~~~~

1.9 Math.gamma Math.lgamma JRuby
~~~~~

=end

**#12 - 12/19/2010 11:48 AM - hasari (Hiro Asari)**

=begin  
~~~~~  
=end

#13 - 12/19/2010 12:41 PM - oj (Yoji Ojima)

=begin
~~~~~ each ~~~~~

~~~~~

~~~~~ Ruby ~~~~~  
=end

**#14 - 12/19/2010 01:33 PM - mame (Yusuke Endoh)**

=begin  
~~~~~  
2010 12 19 12:41 Yoji Ojima redmine@ruby-lang.org:

~~~~~ each ~~~~~

~~~~~

~~~~~ Ruby ~~~~~

~~~~~ Ruby ~~~~~  
~~~~~  
~~~~~

~~~~~ Asari ~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~

~~~~~ new feature ~~~~~ matz  
~~~~~  
~~~~~ matz ~~~~~

matz
matz
matz

--
Yusuke Endoh mame@tsg.ne.jp

=end

#15 - 12/19/2010 02:03 PM - mame (Yusuke Endoh)

=begin
2010/12/19 13:33 Yusuke ENDOH mame@tsg.ne.jp:

new feature
matz
matz

matz

[ruby-dev:42791]
matz
()

--
Yusuke Endoh mame@tsg.ne.jp

=end

#16 - 12/19/2010 02:06 PM - tarui (Masaya Tarui)

=begin
item

sample item

[ruby-dev:41918] [Feature #3647] Array#sample(n, replace=false)
matz

- #3647
- 1)
- 2)
- 3)

--
(Masaya TARUI)
No Tool, No Life.

=end

#17 - 12/19/2010 02:33 PM - shyouhei (Shyouhei Urabe)

=begin
(2010/12/19 14:02), Yusuke ENDOH wrote:

[ruby-dev:42791]
matz
()

trunk

=end

#18 - 12/19/2010 02:36 PM - oj (Yoji Ojima)

=begin

- #3647
- 1)
- 2) (-1)
- 3)

4) 0

=end

#19 - 12/19/2010 03:20 PM - oj (Yoji Ojima)

=begin

R <http://stat.ethz.ch/R-manual/R-devel/library/base/html/sample.html>

R Ruby

=end

#20 - 12/19/2010 08:13 PM - mame (Yusuke Endoh)

=begin

20101219 14:06 Masaya TARUI tarui@prx.jp:

- #3647
- 1)
- 2) (-1)
- 3)

4) 0 ()

R <http://stat.ethz.ch/R-manual/R-devel/library/base/html/sample.html>

sample(, replace = false, = nil)

Matlab <http://www.mathworks.com/help/toolbox/stats/randsample.html>

randsample(, replace = false, = nil)

Mathematica

<http://reference.wolfram.com/mathematica/ref/RandomSample.html>
<http://reference.wolfram.com/mathematica/ref/RandomChoice.html>

```
RandomSample[[], {}]  
RandomSample[[]->[], {}]  
RandomChoice[[], {}]  
RandomChoice[[]->[], {}]
```

RandomSample RandomChoice
-> RandomChoice

RandomSample
RandomChoice 4

RandomSample
Ruby shuffle
RandomChoice 1 (Ruby
sample) Ruby

Python
<http://docs.python.org/library/random.html>

```
choice([]) # 1  
shuffle([])  
sample([], {})
```

RandomChoice(?)

RandomSample
RandomChoice

sample O(n) sum

--
Yusuke Endoh mame@tsg.ne.jp

=end

#21 - 12/19/2010 08:18 PM - mame (Yusuke Endoh)

=begin
RandomChoice

2010年12月19日14:33 Urabe Shyouhei shyouhei@ruby-lang.org:

(2010/12/19 14:02), Yusuke ENDOH wrote:

```
[ruby-dev:42791] RandomChoice  
matz (Ruby) RandomChoice
```

RandomChoice trunk
RandomChoice

RandomChoice
RandomChoice

matz new feature
RandomChoice

matz new feature
RandomChoice (matz
RandomChoice)

--
Yusuke Endoh mame@tsg.ne.jp

=end

#22 - 12/19/2010 08:22 PM - mame (Yusuke Endoh)

=begin

2010/12/19 15:20 Yoji Ojima redmine@ruby-lang.org:

RubyのArray#sampleメソッドについて

Rのhttp://stat.ethz.ch/R-manual/R-devel/library/base/html/sample.html

RのArray#sampleメソッドはRubyのArray#sampleメソッドと同じ

RubyのArray#sampleメソッドは、配列からランダムに要素を抽出する。take_whileメソッド

Array#sampleメソッドは、配列からランダムに要素を抽出する。

--
Yusuke Endoh mame@tsg.ne.jp

=end

#23 - 12/19/2010 09:15 PM - yugui (Yuki Sonoda)

=begin
Yugui

2010/12/19 Masaya TARUI tarui@prx.jp:

RubyのArray#sampleメソッドは、配列からランダムに要素を抽出する。

Array#sampleメソッドは、配列からランダムに要素を抽出する。

Array#sampleメソッドは、配列からランダムに要素を抽出する。

- [ruby-dev:41918] RubyのArray#sampleメソッドは、配列からランダムに要素を抽出する。
• mrknのMath/RandomモジュールのArray#sampleメソッド

--
Yuki Sonoda (Yugui)
yugui@yugui.jp
http://yugui.jp

=end

#24 - 12/19/2010 10:45 PM - oj (Yoji Ojima)

=begin

sampleメソッドはO(n)の計算量

Walker's alias methodは、ランダムに要素を抽出するための高速なアルゴリズム

#25 - 12/19/2010 10:52 PM - matz (Yukihiro Matsumoto)

=begin

In message "Re: [ruby-dev:42806] Re: [Ruby 1.9-Feature#4147] Array#sample"

on Sun, 19 Dec 2010 20:17:57 +0900, Yusuke ENDOH mame@tsg.ne.jp writes:

```
|matz new feature |
```

```
matz new feature |
```

```
|matz new feature |
|matz new feature |
```

matz

=end

#26 - 12/20/2010 12:41 AM - mame (Yusuke Endoh)

```
=begin
|
```

2010121922:52 Yoji Ojima redmine@ruby-lang.org:

```
sample O(n)
```

```
Walker's alias method
```

Walker's alias method R (<http://svn.r-project.org/R/trunk/src/main/random.c>)

```
R sample 1 2
```

- <<
- sample O(n)

matz

```
Ruby gem
```

--
Yusuke Endoh mame@tsg.ne.jp

=end

#27 - 12/20/2010 01:26 AM - oj (Yoji Ojima)

```
=begin
```

```
Ruby gem
```

```
matz
```

```
(1..n).map { ary.sample }
matz
=end
```

#28 - 12/20/2010 12:43 PM - mame (Yusuke Endoh)

```
=begin
```


ランダムに抽出する O(log n) のアルゴリズム
API の O(n) のアルゴリズム
ランダムに抽出する

ランダムに抽出する Ruby のアルゴリズム

ランダムに抽出する

ランダムに抽出する

ランダムに抽出する
ランダムに抽出する
ランダムに抽出する
ランダムに抽出する

ランダムに抽出する R
ランダムに抽出する

ランダムに抽出する

--
Yusuke Endoh mame@tsg.ne.jp

=end

#31 - 12/20/2010 10:33 PM - oj (Yoji Ojima)

=begin

ランダムに抽出する
ランダムに抽出する

ランダムに抽出する

ランダムに抽出する
ランダムに抽出する

<http://code.activestate.com/recipes/576564-walkers-alias-method-for-random-objects-with-diffe/>

ランダムに抽出する

=end

#32 - 12/20/2010 10:56 PM - akr (Akira Tanaka)

=begin

2010年12月19日21:15 Yugui yugui@yugui.jp:

- ランダムに抽出する

ランダムに抽出する (ランダムに抽出する)
ランダムに抽出する

x = f(rand(n))

ランダムに抽出する f, n

f, n = Random.new.rand(0..n)

ランダムに抽出する
(call してランダムに抽出する)

f, n = Random.new.rand(0..n)(beg..end) {|v| }
f, n = Random.new.rand(0..n)(beg..end) {|v| }

...

ランダムに抽出する
ランダムに抽出する

ランダムに抽出する f, n

ランダムに抽出する Array#sample

- [ruby-dev:41918] 000

```
00000000000000000000000000000000000000000000000000000
Array#sample 00000000 (00000000000000000000000000000000) 0000
[ruby-dev:41918] 00000000000000000000000000000000000000000000000000000
[ruby-dev:41919] 00000000000000000000000000000000000000000000000000000
(00000000000000000000000000000000000000000000000000000)
```

```
00000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000
```

- mrkn000Math/Random000

```
Random 00000000000000000000000000000000000000000000000000000
Array#sample 00000000000000000000000000000000000000000000000000000
```

```
00000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000
```

```
--
[00 0][000 000][Tanaka Akira]
```

```
=end
```

#33 - 12/21/2010 01:33 AM - mame (Yusuke Endoh)

```
=begin
00000000
```

2010122022:33 Yoji Ojima redmine@ruby-lang.org:

```
00000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000
```

```
00000000000000000000000000000000000000000000000000000
```

```
00000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000
```

```
00000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000
http://code.activestate.com/recipes/576564-walkers-alias-method-for-random-objects-with-diffe/
00000000000000000000000000000000000000000000000000000
```

```
00000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000 API 00000000
00000000000000000000000000000000000000000000000000000
```

```
0000 sample 00000000000000000000000000000000000000000000000000000
0000000000 API 000000000000000000000000000000000000000000000000000000
sample 00000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000
0000000000
```

```
00000000 naruse 000 (IRC 0) 000000000000000000000000000000000000000000000000000000
Enumerator 00000000 API 000000000000000000000000000000000000000000000000000000
```

```
e = %(00 00 00 0).sample_each
e.each {|x| p x } #=> 0 00 00 00
```

```
e = %(00 00 00 0).sample_each(unique: false, weight: [1000, 100, 10, 1])
e.each {|x| p x } #=> 00 00 00 00 ...
```

```
00000000000000000000000000000000000000000000000000000
```

--
Yusuke Endoh mame@tsg.ne.jp

=end

#34 - 12/21/2010 12:36 PM - oj (Yoji Ojima)

```

=begin
  naruse (IRC )
  Enumerator API

  e = (%( p x ).sample_each
  e.each {|x| p x } #=>

  e = (%( p x ).sample_each(unique: false, weight: [1000, 100, 10, 1])
  e.each {|x| p x } #=>

  sample_each.each
=end

```

#35 - 12/21/2010 03:20 PM - naruse (Yui NARUSE)

```

=begin
  •
  • (block)
  • (

  ==

  optional hash Array#choice
  Array#choice 1.8.7
  sample/choice Mathematica (see also [ruby-dev:42805])

  ==

  ary
  ary hash
  block

  ==

  enumerator (generator )

  sample_each.each

  sample_each {|x| .. }
--
NARUSE, Yui
naruse@airemix.jp

=end

```

#36 - 12/21/2010 05:58 PM - oj (Yoji Ojima)

```

=begin
  optional hash Array#choice

```

1.8.7

sample_each.each

sample_each {|x| .. }

=end

#37 - 12/21/2010 09:35 PM - akr (Akira Tanaka)

=begin
2010-12-20 22:56 Tanaka Akira akr@fsij.org:

sample_each

Pavlos S. Efraimidis, Paul G. Spirakis
Weighted random sampling with a reservoir
Information Processing Letters
Volume 97, Issue 5 (16 March 2006)

sample_each

sample_each (sample)

--
[akr@fsij.org] [Tanaka Akira]

=end

#38 - 12/22/2010 12:34 AM - akr (Akira Tanaka)

=begin
2010-12-21 21:35 Tanaka Akira akr@fsij.org:

Pavlos S. Efraimidis, Paul G. Spirakis
Weighted random sampling with a reservoir
Information Processing Letters
Volume 97, Issue 5 (16 March 2006)

sample_each

sample_each (sample)

sample_each

sample_each

```
% ./ruby -I /home/akr/ruby/depq/lib -rdepq -e '
module Enumerable
def sample2(n)
Depq.nlargest(n, self) {|e| rand ** (1.0/(yield(e))) }
end
end
```

```
enum = (-20..20).to_a*10000
a = enum.sample2(20000) {|e| Math.exp(-(e/5.0)*2) }
h = a.categorize(:op=>+) {|e| [e,1] }
-10.upto(10) {|x| puts "" * (h[x]/30.0).to_i }
'
```

*

*

]: depq, categorize

--

[Tanaka Akira]

=end

#39 - 12/22/2010 01:08 PM - akr (Akira Tanaka)

=begin

2010122115:20 NARUSE, Yui naruse@airemix.jp:

==

[ruby-dev:42735]
[ruby-dev:42741]
[ruby-dev:42797]

--
[Tanaka Akira]

=end

#40 - 12/22/2010 03:23 PM - naruse (Yui NARUSE)

=begin

2010122117:59 Yoji Ojima redmine@ruby-lang.org:

sample_each.each

sample_each {|x| .. }

--
NARUSE, Yui
naruse@airemix.jp

=end

#41 - 12/22/2010 03:29 PM - naruse (Yui NARUSE)

=begin
2010/12/22 13:08 Tanaka Akira akr@fsij.org:

2010/12/21 15:20 NARUSE, Yui naruse@airemix.jp:

==
[redacted]

[redacted]

[ruby-dev:42735] [redacted]
[ruby-dev:42741] [redacted]
[ruby-dev:42797] [redacted]

[redacted]

IRC [redacted] RPG [redacted]

- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

[redacted]

--
NARUSE, Yui
naruse@airemix.jp

=end

#42 - 12/22/2010 06:01 PM - oj (Yoji Ojima)

=begin
[redacted]

[redacted]
[redacted]
=end

#43 - 12/23/2010 02:00 PM - mrkn (Kenta Murata)

=begin
[redacted]
yugui [redacted]

Math/Random [redacted]

On 2010/12/19, at 21:15, Yugui wrote:

2010/12/19 Masaya TARUI tarui@prx.jp:
[redacted]

[redacted]

Array#sample
Array#sample

- Array#sample
- [ruby-dev:41918] Array#sample

replace
choice

- mrkn Array#sample

Ojima Array#sample

Ojima n Array#sample

- Array#sample
- Array#sample_by

sort_by Array#sample

o

n Walker
Walker boost
CodeRepos <http://bit.ly/eAk1M1>

- Array#sample
- Array#sample
- Array#sample Array#sample

Random

- mrkn Math/Random

Math/Random

- Random
- boost random

:

```
gen = Math::Random::SFMT216091.new # Random.new  
gen() #  
gen.int(min, max) # [min, max]  
gen.real # [0, 1) (53-bit)
```

```
ndist = Math::Random::NormalDistribution.new(0, 1) #  
ndist(gen) #
```

Random

```
# TypedData  
# TypedData  
#
```

```
Walker  
Array#sample Array#sample
```

~~~~~

~~~~~ Walker ~~~~~  
~~~~~  
~~~~~ Array#sample ~~~~~  
~~~~~ :P

~~~~~

~~~~~ Walker ~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~

Array#sample ~~~~~
~~~~~  
~~~~~ GNU R ~ sample ~~~~~  
~~~~~ [ruby-dev:42817] ~~~~~ Array#sample ~~~~~

~~~~~ Array ~ Hash ~~~~~  
~~~~~  
~~~~~ Array#sample ~~~~~  
~~~~~

~~~~~ [ruby-dev:42829] ~~~~~  
~~~~~ sample\_each ~~~~~  
~~~~~

--

Kenta Murata
OpenPGP FP = 1D69 ADDE 081C 9CC2 2E54 98C1 CEFE 8AFB 6081 B062

~~~~~!!  
~~~~~ Ruby ~~~~~ <http://www.amazon.co.jp/dp/4798119881/mrkn-22>

E-mail: mrkn@mrkn.jp
twitter: <http://twitter.com/mrkn/>
blog: <http://d.hatena.ne.jp/mrkn/>

=end

#44 - 12/23/2010 02:56 PM - mame (Yusuke Endoh)

=begin
~~~~~

2010~12~23~14:00 Kenta Murata [muraken@gmail.com](mailto:muraken@gmail.com):

~~~~~ replace ~~~~~  
~~~~~ choice ~~~~~  
~~~~~

~~~~~ ruby-core ~~~~~  
~~~~~ sample/choice ~~~~~  
~~~~~ matz ~~~~~  
~~~~~ (replace ~~~~~  
~~~~~)

Ojima ~~~~~n~~~~~ Array#sample ~~~~~

- Array#sample ~~~~~
- Array#sample\_by ~~~~~

IRC ~~~~sample(weight: proc { ... }) ~~~~~  
~~~~~

~~~~~  
sort\_by ~~~~~  
~~~~~sample\_each ~~~~~

n Walker boost CodeRepos <http://bit.ly/eAk1M1>

mrkn

Walker

snip Array#sample

gem

[ruby-dev:42829] sample_each

Yusuke Endoh mame@tsg.ne.jp

=end

#45 - 12/23/2010 03:04 PM - mame (Yusuke Endoh)

=begin

API sample_each each_sample

Array#sample([size, [opt]])

- API
- size 1
- size size
- opt weight: proc or array random: Random instance

- - weight
- weight akr [ruby-dev:42844]

Array#choice([size, [opt]])

- API
- size 1
- size size
- opt Array#sample
•
- size (weight)
- Walker's alias method (size size)

Array#each_sample([opt])

- API

- `yield` 関数 (Enumerator 生成)
- Enumerator 生成
- `opt` `Array#sample` 関数

- 関数

- `weight` 引数 (Array#sample)
- `weight` 引数 `akr` (Array#sample)

`Array#each_choice([opt])`

- API

- `yield` 関数
- Enumerator 生成
- `opt` `Array#sample` 関数

- 関数

- `weight` 引数 (Array#sample)
- `weight` 引数 Walker's alias method (Array#sample)

例

- `%(1 2 3 4).sample #=> 1 2 3 4`
- `%(1 2 3 4).sample(4) #=> shuffle 1 2 3 4`
- `%(1 2 3 4).sample(weight: [1000, 100, 10, 1]) #=> 1 2 3 4 (Array#sample)`
- `%(1 2 3 4).sample(4, weight: [1000, 100, 10, 1]) #=> 1 2 3 4 (Array#sample)`

- `%(1 2 3 4).sample(4, weight: proc {|x| ... }) #=> 1 2 3 4 (Array#sample)`
- `%(1 2 3 4).choice #=> 1 2 3 4`
- `%(1 2 3 4).choice(4) #=> choice 4 1 2 3 4`
- `%(1 2 3 4).choice(weight: [1000, 100, 10, 1]) #=> 1 2 3 4 (Array#sample)`
- `%(1 2 3 4).choice(4, weight: [1000, 100, 10, 1]) #=> 1 2 3 4 (Array#sample)`
- `%(1 2 3 4).each_sample.first #=> sample 1 2 3 4`
- `%(1 2 3 4).each_sample.to_a #=> shuffle 1 2 3 4`
- `%(1 2 3 4).each_sample(weight: [1000, 100, 10, 1]).take(4) #=> 1 2 3 4 (Array#sample)`
- `%(1 2 3 4).each_choice.first #=> choice 1 2 3 4`
- `%(1 2 3 4).each_choice.to_a #=> choice(4) 1 2 3 4`
- `%(1 2 3 4).each_choice(weight: [1000, 100, 10, 1]).take(4) #=> 1 2 3 4 (Array#sample)`

--
Yusuke Endoh mame@tsg.ne.jp

=end

#46 - 12/23/2010 03:15 PM - akr (Akira Tanaka)

=begin
2010年12月22日 18:01 Yoji Ojima redmine@ruby-lang.org:

```

Array#sample
Array#sample(4)

```

Array#sample(weight: [1000, 100, 10, 1])

RPGRPG

<http://tkool.jp/>

<http://d.hatena.ne.jp/ktakaki/20111201/p1>

ping

#54 - 12/01/2011 10:51 AM - shyouhei (Shyouhei Urabe)

#55 - 07/14/2012 01:45 PM - akr (Akira Tanaka)

repeat:

#56 - 10/25/2012 11:53 PM - yhara (Yutaka HARA)

- Target version changed from 2.0.0 to 2.6

#57 - 10/01/2014 03:34 AM - gogotanaka (Kazuki Tanaka)

.

.

#58 - 10/01/2014 11:44 AM - naruse (Yui NARUSE)

- Assignee deleted (shyouhei (Shyouhei Urabe))

gogo tanaka wrote:

.

.

#59 - 12/04/2014 01:13 AM - shyouhei (Shyouhei Urabe)

Yui NARUSE wrote:

#1

#60 - 11/21/2016 04:59 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Assigned to Feedback