

Ruby master - Feature #4147

Array#sample

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

Status: Feedback	
Priority: Normal	
Assignee:	
Target version:	
Description =begin Array#sample omikujiji_box = [:name => " ", :weight => 1000}, { :name => " ", :weight => 100}, { :name => " ", :weight => 10}, { :name => " ", :weight => 1}] omikujiji = omikujiji_box.sample { v v[:weight] } puts omikujiji[:name] =end	
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 m < 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
[]
2010121816:20 Yoji Ojima redmine@ruby-lang.org:
[]
```

```
API
each
[]
omikuj_box = %w(0 1 2)
omikuj_weight = [1000, 1100, 1110, 1111]
omikuj = omikuj_box.sample(weight: omikuj_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年12月19日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

```
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([], {})
```

RandomSample(?)

RandomSample
RandomChoice

sample O(n) sum

--
Yusuke Endoh mame@tsg.ne.jp

=end

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

=begin
RandomSample

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] RandomSample  
RandomChoice matz  
(Random) Random
```

RandomSample trunk
RandomChoice

RandomSample
RandomChoice

RandomSample matz new feature
RandomChoice

RandomSample matz new
feature RandomSample
(Random 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 R http://stat.ethz.ch/R-manual/R-devel/library/base/html/sample.html
Ruby R Ruby

Ruby take_while

code block

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:

code block

code block

- [ruby-dev:41918]
• mrkn Math/Random
o mrkn Array#sample

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

=end

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

=begin

sample O(n)

Walker's alias method

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

=begin
code block

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

=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 }
```

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

```
=begin
```


ランダムに抽出した要素の数を O(log n) に制限する
API は O(n) のランダムに抽出した要素の数を制限する
ランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する Ruby のランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する
ランダムに抽出した要素の数を制限する
ランダムに抽出した要素の数を制限する
ランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する R
ランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する

--
Yusuke Endoh mame@tsg.ne.jp

=end

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

=begin

ランダムに抽出した要素の数を制限する
ランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する
ランダムに抽出した要素の数を制限する

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

ランダムに抽出した要素の数を制限する

=end

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

=begin

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

- ランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する
ランダムに抽出した要素の数を制限する

x = f(rand(n))

ランダムに抽出した要素の数を制限する

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

ランダムに抽出した要素の数を制限する
(call ランダムに抽出した要素の数を制限する)

f, n = Random.new._rand_int(rand(n)) { |v| rand }
f, n = Random.new._rand_int(rand(n)) { |v| rand }

...

ランダムに抽出した要素の数を制限する
ランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する

ランダムに抽出した要素の数を制限する Array#sample

=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)
    • (generator)

==

optional hash Array#choice

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

==

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:

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

()
--
[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)

```
% ./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

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
```


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` (Enumerator)
- Enumerator
- `Array#sample`

- `Array`

- `weight`
- `Array#sample` (with `weight`)

`Array#each_choice([opt])`

- API

- `yield`
- Enumerator
- `Array#sample`

- `Array`

- `weight`
- Walker's alias method

Examples

- `array.sample` #=> shuffled array
- `array.sample(4)` #=> shuffle 4 items
- `array.sample(weight: [1000, 100, 10, 1])` #=> weighted sample (with `weight`)
- `array.sample(4, weight: [1000, 100, 10, 1])` #=> weighted sample (with `weight` and count) (e.g., "a", "b", "c", "d")
- `array.sample(4, weight: proc {|x| ... })` #=> weighted sample (with `weight` as a proc)
- `array.choice` #=> single choice
- `array.choice(4)` #=> choice of 4 items
- `array.choice(weight: [1000, 100, 10, 1])` #=> weighted choice (with `weight`)
- `array.choice(4, weight: [1000, 100, 10, 1])` #=> weighted choice (with `weight` and count) (e.g., "a", "b", "c", "d")
- `array.each_sample.first` #=> sample one item
- `array.each_sample.to_a` #=> shuffle all items
- `array.each_sample(weight: [1000, 100, 10, 1]).take(4)` #=> weighted sample (with `weight`)
- `array.each_choice.first` #=> choice one item
- `array.each_choice.to_a` #=> choice of 4 items
- `array.each_choice(weight: [1000, 100, 10, 1]).take(4)` #=> weighted choice (with `weight`)

--
Yusuke Endoh mame@tsg.ne.jp

=end

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

=begin
2010122218:01 Yoji Ojima redmine@ruby-lang.org:

```

array.sample
array.sample(4)

```

`array.sample`

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

- Status changed from Assigned to Feedback