




Объекты коллекции в Ruby

Array, Hash, Range, Set



Инициализация массивов

```
1  new_array = [1, 3, 4, 6]
2  empty_array = []
3
4  p new_array, empty_array, empty_array == [] #=> [1, 3, 4, 6], [], true
5  p Array.new, Array.new(5, 0) #=> [], [0, 0, 0, 0, 0]
6  p Array.new(5) { |i| i * 2 } #=> [0, 2, 4, 6, 8]
7
8  p Array.[](1, 2, 3), Array[1, 2, 3], Array(1..3) #=> [1, 2, 3]
9
10 puts new_array[0], new_array[-1], new_array.first, new_array.last #=> 1, 6
11
12 p new_array.index(4), new_array[10] #=> 2, nil
13
```

Методы для работы с массивами

```
14 empty_array << 4
15 empty_array.push(5)
16 empty_array.unshift(3)
17 empty_array[4] = 7
18
19 p empty_array #=> [3, 4, 5, nil, 7]
20
21 empty_array.delete_at(3)
22 empty_array.delete(3)
23
24 p empty_array #=> [4, 5, 7]
25
26 empty_array.pop; empty_array.shift
27
28 p empty_array #=> [5]
```

Hash. Инициализация

```
1 empty_hash = Hash.new
2 empty_hash_1 = Hash.new { |hash, key| hash[key] = 1 }
3 empty_hash_a = Hash.new('a')
4
5 p empty_hash, empty_hash_1, empty_hash_a #=> {}
6
7 p empty_hash[:val], empty_hash_1[2], empty_hash_a[[1, 2]] #=> nil, 1 , 'a'
8
9 new_hash = { a: 2, v: 4 }
10 p new_hash == { :a => 2, :v => 4 } #=> true
11
12 p Hash[a: 2] #=> { a: 2 }
13
14 new_hash[:a] = 3
15 p new_hash[:a], new_hash.fetch(:g, 5) #=> 3, 5
```

Hash. Основные методы

```
17 p new_hash.delete(:v) #=> 4
18 p new_hash #=> { a: 3 }
19
20 p new_hash.keys, new_hash.values #=> [:a], [3]
21
22 new_hash.each { |k, v| puts "#{k} => #{v}" } #=> 'a => 3'
23 new_hash.each_key { |k| puts k } #=> 'a'
24 new_hash.each_value { |v| puts v } #=> 3
25
26 p new_hash.has_key?(:a), new_hash.key?(:a), new_hash.include?(:a), new_hash.member?(:a) #=> true
27 p new_hash.has_key?('a') #=> false
28
29 p new_hash.value?(3), new_hash.has_value?(3) #=> true
30
```

Range. Инициализация

```
1 letter_range = ('a'..'z')
2 num_range = (1...20)
3
4 p letter_range.to_a #=> ['a', 'b', 'c', ..., 'z']
5 p num_range.to_a #=> [1, 2, 3, 4, ..., 19]
6
7 p num_range.step(2).to_a #=> [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
8
9 p num_range.include?(9), num_range.include?(21) #=> true, false
10 p num_range.cover?(1..5), num_range.cover?(3) #=> true, true
11
12 p letter_range.include?('word'), letter_range.cover?('word') #=> false, true
13
```

Создание Range класса

```
14 class YearMonth
15   include Comparable
16
17   attr_reader :year, :month
18
19   def initialize(year, month)
20     raise ArgumentError unless year > 0
21     raise ArgumentError unless month >= 1 && month <= 12
22
23     @year = year
24     @month = month
25   end
26
27   def next
28     month == 12 ? YearMonth.new(year + 1, 1) : YearMonth.new(year, month + 1)
29   end
30
31   alias_method :succ, :next
32
33   def <=>(other)
34     (year <=> other.year).nonzero? || month <=> other.month
35   end
36
37   def to_s
38     "#{year}-#{month}"
39   end
40
41   def inspect
42     to_s
43   end
44 end
45
46 year_month_1 = YearMonth.new(2021, 11)
47 year_month_2 = YearMonth.new(2022, 6)
48
49 p (year_month_1..year_month_2).to_a #=> [2021-11, 2021-12, 2022-1, 2022-2, 2022-3, 2022-4, 2022-5, 2022-6]
50 p year_month_2 > year_month_1 #=> true
```

- для перечисления нужен метод succ;
- для сравнения нужна подключение модуля Comparable.

Множество Set

```
1  require 'set'
2
3  set = Set.new
4  set << 5
5  set << 2
6  set << 5
7
8  p set #=> #<Set: {5, 2}>
9
10 p set.include?(5) #=> true
11
12 # p set[0] #=> undefined method `[]'
13
14 p Set.new(1..10) >= Set.new(2..4) #=> true
15 p Set.new(1..10) <= Set.new(1..20) #=> true
16
17 sorted_set = SortedSet.new
18 sorted_set << 5
19 sorted_set << 2
20 sorted_set << 1
21
22 p sorted_set #=> #<SortedSet: {1, 2, 5}> # was removed from ruby 3.0
```

- поиск элемента происходит быстрее, чем в массиве;
- прямого выбора элемента по индексу нет;
- Класс SortedSet был исключен из стандартных библиотек Ruby.

Полезные ссылки

<https://www.freecodecamp.org/news/common-array-methods-in-ruby/> – Основные методы для работы с массивами;

https://www.tutorialspoint.com/ruby/ruby_hashes.htm – Описание хешей;

<https://www.rubyguides.com/2016/06/ruby-ranges-how-do-they-work/> – Описание range коллекции;

<https://www.rubyguides.com/2018/08/ruby-set-class/> – Статья про множества.

Конец! Спасибо!