

Ruby: массивы

Информатика
10-11 классы

20 октября 2011 г.

Введение

- Одним из базовых типов переменных является *массив*. Ранее мы рассматривали переменные типа число (integer, float), немного строки (string).
- (Формально): Массив — ряд переменных, доступ к которым определён по индексу.
- (Упрощённо): Массив — группа переменных, пронумерованных начиная с нуля, хранящихся в единой переменной.
- Массивы позволяют упростить доступ к однотипным данным.
- Например, в виде массива легко хранить температуру за 30 последних дней.
- Массивы бывают двумерными — когда элементы массива сами являются массивами.

Табличная форма

Ключ	Значение
0	5.3
1	4.2
2	2.0
3	-0.8
4	1.5

- Рассмотрим массив, состоящий из 5 элементов (см. таблицу слева)
- В ruby такой массив записывается следующим образом:

Listing 1: Создание массива

```
arr = [5.3, 4.2, 2.0, -0.8, 1.5]
```

- где `arr` — название массива.
- Чтобы вывести на экран, например, элемент с ключом 3 достаточно написать `puts arr[3]`:

Создание массива

- Зададим массив, состоящий из 6 последовательных натуральных чисел.

Listing 2: Способы создания массива

```
arr = [1,2,3,4,5,6]
```

```
arr = Array.new
```

```
arr[0] = 1
```

```
arr[1] = 2
```

```
arr[] = 3
```

```
...
```

```
arr = (1..6).to_a
```

- 1 способ — обычный, 2 — ручной (обратите внимание, что ключ при добавлении элемента в конец указывать необязательно), 3 — через диапазон.

Методы

- Чтобы изменить массив, к нему нужно применить так называемый *метод*.
- *Метод* — это то, что меняет переменную по заданному правилу.
- Например, метод **sort** сортирует массив.
- Для разных типов переменных могут быть разные методы.
- Чтобы применить к переменной **var** метод **method** достаточно написать:

Listing 3: Применение метода

```
var . method
```

- Обратите внимание! Обычный метод не меняет саму переменную. Хотите изменить — присвойте результат другой переменной или этой же самой (`var = var.method`)

Методы работы с массивом

- Рассмотрим массив $arr = [6, 3, 5, 1, 2, 4]$.

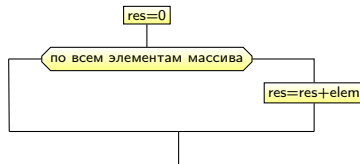
Метод	Описание	Результат
<code>arr.size</code>	количество элементов	6
<code>arr.min</code>	минимальный элемент	1
<code>arr.max</code>	максимальный элемент	6
<code>arr.sort</code>	сортировка	<code>[1, 2, 3, 4, 5, 6]</code>
<code>arr.reverse</code>	перевернуть	<code>[4, 2, 1, 5, 3, 6]</code>
<code>arr.sort.reverse</code>	отсортировать и перевернуть	<code>[6, 5, 4, 3, 2, 1]</code>
<code>arr.include?(6)</code>	есть ли в массиве элемент 6	<code>true</code>
<code>arr.empty?</code>	пуст ли массив	<code>false</code>
<code>arr.any?</code>	есть ли хоть один элемент	<code>true</code>
<code>arr.delete(6)</code>	удалить элемент 6*	<code>[3, 5, 1, 2, 4]</code>
<code>arr.delete_at(1)</code>	удалить элемент с ключом 1	<code>[6, 5, 1, 2, 4]</code>

Сумма

- Допустим дана средняя температура за 5 дней: `arr = [5.3, 2.1, 5.2, 1.8, -0.2]`.
- Как вычислить сумму?

Listing 4: Сумма

```
arr = [5.3, 2.1, 5.2, 1.8, -0.2]  
sum = arr.inject(0){ |res, elem| res+elem }
```



Разбор метода inject

Listing 5: Общий вид inject

```
res = arr.inject(start){|result, element| expression }
```

- start — чему изначально равна переменная result.
Например, если внутри inject вы будете умножать, то нельзя делать переменную равной нулю, так как ноль умножить на любое число — ноль.
- result — переменная, в которую записывается результат.
- element — текущий элемент массива (переменная меняется с каждой итерацией).
- expression — выражение для переменных result и element.
- **Обратите внимание!** На этом слайде названия переменных и выражение даны в виде схемы! Реальный пример — на предыдущем слайде.

Задания

- Напишите программу, вычисляющую с помощью `inject` произведение элементов массива (массив можно задать любой).
- Напишите программу, вычисляющую среднее арифметическое и среднее геометрическое элементов массива.

Поиск элементов

- В ряде задач нам нужно извлечь из массива определённые элементы.
- Допустим дана средняя температура за 5 дней: `arr = [5.3, 2.1, 1.2, -0.8, -0.2]`.
- Вычислим, сколько дней была отрицательная температура.

Listing 6: Метод `find_all`

```
arr = [5.3, 2.1, 1.2, -0.8, -0.2]
arr_neg = arr.find_all {|elem| (elem < 0) }
puts arr_neg.size

puts [5.3, 2.1, 1.2, -0.8, -0.2].
      find_all {|elem| (elem > 0) }.size
```

Разбор метода `find_all`

Listing 7: Общий вид `find_all`

```
arr_res = arr.find_all{ |element| condition }
```

- `element` — текущий (рассматриваемый) элемент массива (переменная меняется с каждой итерацией).
- `condition` — логическое выражение или несколько выражений, связанных логическими операциями (конъюнкция, дизъюнкция, отрицание).
- Метод извлекает из массива `arr` все элементы, удовлетворяющие условию `condition`, и записывает результат в массив `arr_res`.

Задания

- Дан массив из 10 целых чисел (любых). Вывести на экран все элементы массива, меньшие нуля и делящиеся на три.
- Дан массив из 10 целых чисел (любых). Вывести на экран сумму все чётных положительных элементов массива.

Изменение элементов массива

- Предположим, у нас есть массив цен на нефть в долларах США за последние 5 дней.
- Как перевести все цены в рубли, зная курс рубля по отношению к доллару?

Listing 8: Метод map

```
arr = [99.23, 101.42, 99.87, 96.49, 95.11]
usd_to_rub = 31.23
arr_in_rub = arr.map{|elem| elem*usd_to_rub}
puts arr_in_rub
```

Разбор метода map

Listing 9: Общий вид map

```
arr_res = arr.map{|element| expression }
```

- `element` — текущий (рассматриваемый) элемент массива (переменная меняется с каждой итерацией).
- `expression` — выражение, показывающее, как надо менять элемент массива.
- Метод проходит по всему массиву `arr` и меняет каждый элемент в соответствии с выражением `expression`.
Результат записывается в массив `arr_res`.

Задания

- Дан массив температуры за последние 10 дней (любые разумные числа) в градусах по Цельсию. Вывести на экран температуру на каждый день в градусах по Фаренгейту и по Кельвину.

References

- Все презентации доступны на <http://school.smirik.ru!>
- Вопросы, предложения, д/з: smirik@gmail.com