

Learning Ruby - 2

Ruby Arrays and Lists

Ruby Arrays are Cool!

- We've already seen a Ruby array – **song_lines**
- Ruby arrays **shrink** and **grow** dynamically - no more *annoying array index errors*.
- Ruby array elements can contain **any other object** (think about the implications of this!)
- Ruby arrays can be set with a list of items

Playing with Ruby Arrays

```
my_a = [ 'one', 'two', 'three' ]  
puts my_a[1]
```

```
other_a = ["four", "five", "six"]  
puts other_a[0]
```

```
my_a[3] = other_a
```

```
puts my_a[3][2]  
other_a[3] = "seven"  
my_a
```

Fun with Indices

```
puts other_a[-1]
```

```
another_a = other_a[1..2]
```

```
another_a
```

```
another_a.values_at(0, 2)
```

```
my_a[5] = "this is the end"
```

```
my_a.values_at(1, 3, 5).sort
```

Ruby Array Methods

```
puts another_a.length  
puts other_a.length  
puts my_a.length
```

```
Array.new  
Array.new(100)  
Array.new(10, 'x')
```

```
other_a.sort  
other_a.delete("five")  
other_a
```

```
my_a
```

Working with "each" Element

```
my_a.each { |element| puts element if element.length > 4 }
```

```
my_a.each { |element| puts element if element.length > 3 }
```

```
my_a.each { |element| puts element if element.length > 2 }
```

```
my_a.flatten  
my_a.flatten!
```

```
my_a.each { |element| puts element if element.length > 4 }
```

Ruby Array Stacks

```
last = my_a.pop  
first = my_a.shift
```

```
my_a
```

```
my_a.unshift( last )  
my_a.push( first )
```

```
my_a
```

More ... Ruby So Far

- Ruby arrays are cool!
- The array methods are cool!
- The array iterators are cool!
- Wave "bye bye" to implementing scary linked lists!!!!
- OK, all together now: ***Ruby arrays are cool!***