

Arrays

Devpoint Labs - Dave Jungst / Jake Sorce

Arrays - Recap

- `arr = Array.new() => []`
- `arr = Array.new(5) => [nil, nil, nil, nil, nil]`
- `arr = []`
- `arr = ['apple', 'orange', 'pear']`
- `arr = [1, 'a', Array.new, { }, (1 + 2)]`
- Arrays can contain any object because?
 - Everything in ruby is an Object
- Arrays are 0 based
 - `arr = ['A', 'B', 'C'] => arr[0] => 'A', arr[1] => 'B', arr[2] => 'C', arr[3] => nil`

Iterating arrays

```
['A', 'B', 'C'].each do |letter|
```

```
  puts letter
```

```
end
```

```
=>
```

```
'A'
```

```
'B'
```

```
'C'
```

```
['A', 'B', 'C'].each_with_index do |letter, i|
```

```
  puts "#{i} => #{letter}"
```

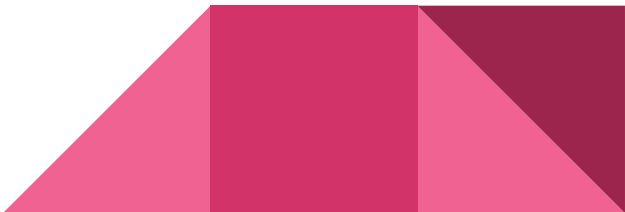
```
end
```

```
=>
```

```
'0 => A'
```

```
'1 => B'
```

```
'2 => C'
```



Modifying Arrays

```
arr = ['A', 'B', 'C']
```

Adding an item on to the end of an array

```
arr << 'D' => ['A', 'B', 'C', 'D']
```

Adding an item at the beginning of an array

```
['B', 'C', 'D'].unshift('A') => ['A', 'B', 'C', 'D']
```

Removing an item off the beginning of an array

```
['A', 'B', 'C'].shift => 'A' ( arr => ['B', 'C'] )
```

Removing an item from the end of an array

```
arr.pop => 'C' ( arr => ['A', 'B'] )
```

Inserting into an array

```
arr.insert(1, 'Z') => ['A', 'Z', 'B', 'C']
```

Move all items left

```
[1,2,3].rotate => [2, 3, 1]
```

Reverse array

```
[1,2,3].reverse => [3,2,1]
```

Sort array

```
['B', 'Z', 'A'].sort => ['A', 'B', 'Z']
```

```
arr.delete_at(1) => ['A', 'C']
```

```
arr.clear => [ ]
```



More array methods

`['A', 'B', 'C'].sample => 'C'`

`['A', 'B', 'C'].join(',') => 'A,B,C'`

`['A', ['B', 'C'], 'D'].flatten => ['A', 'B', 'C', 'D']`

`[1, 2, 1, 3, 4, 1].uniq => [1, 2, 3, 4]`

`[1,2,3].map { |item| item += 1 } => [2, 3, 4]`

`[1,2,3].map(&:to_s) => ['1', '2', '3'] (OR... [1,2,3].map { |item| item.to_s })`

`[1,2,3].shuffle => [2, 1, 3]`



MOAR!!!!

```
arr = [1,2,3]
```

```
arr.count => 3
```

```
arr.any? => true
```

```
b = arr.clone
```

```
arr.max => 3
```

```
arr.min => 1
```

```
arr.drop(1) => [1,2]  a.drop(2) => [3], drop(3) => []
```

```
['A', nil, 'B', 'C', nil].compact => ['A', 'B', 'C']
```



nil?, empty?, any?, none?

	nil?	empty?	any?	none?
nil	true	ERROR	ERROR	ERROR
[]	false	true	false	true
[nil]	false	false	false	false
['A']	false	false	true	false