Intro to Ruby Class: Array Examples

1. Accessing (first & last) elements of an array

```
irb(main):074:0> a = [1,2,3,4,5,6,7,8]
=> [1, 2, 3, 4, 5, 6, 7, 8]
irb(main):075:0> a.first
=> 1
irb(main):076:0> a.last
=> 8
```

2. Add an element to the front of an array

```
irb(main):044:0> a=[1,2,3,4,5,6,7,8]
=> [1, 2, 3, 4, 5, 6, 7, 8]
irb(main):045:0> a.unshift(0)
=> [0, 1, 2, 3, 4, 5, 6, 7, 8]
```

3. Remove duplicates from an array.

```
2.1.2 :011 > a=[1,1,3,2,4,6,6,5,6,4,5,6]
 => [1, 1, 3, 2, 4, 6, 6, 5, 6, 4, 5, 6]
2.1.2 :020 > a.uniq
 => [1, 3, 2, 4, 6, 5]
```

4. Remove nil elements from an array

```
irb(main):026:0> a = ["t","h","i","s",nil,"i","s", nil, "a", nil, "t", "e", "s", "t"]
=> ["t", "h", "i", "s", nil, "i", "s", nil, "a", nil, "t", "e", "s", "t"]

irb(main):028:0> a.delete(nil)
=> nil
irb(main):029:0> a
=> ["t", "h", "i", "s", "i", "s", "a", "t", "e", "s", “t"]

irb(main):023:0>  a = ["t","h","i","s",nil,"i","s", nil, "a", nil, "t", "e",
"s", "t"]
=> ["t", "h", "i", "s", nil, "i", "s", nil, "a", nil, "t", "e", "s", "t"]
irb(main):024:0> a.compact
=> ["t", "h", "i", "s", "i", "s", "a", "t", "e", "s", “t"]
```

5. Add elements to the end of an array.

```
irb(main):069:0* a = [1, 2, 3, 4, 5, 6, 7, 8]
=> [1, 2, 3, 4, 5, 6, 7, 8]
irb(main):070:0> a + [9, 10]
=> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

irb(main):027:0>  a = [1, 2, 3, 4, 5, 6, 7, 8]
=> [1, 2, 3, 4, 5, 6, 7, 8]
```

```
irb(main):028:0> a << 9
=> [1, 2, 3, 4, 5, 6, 7, 8, 9]
irb(main):029:0> a << 10
=> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

6. Flatten an array

```
s=[[1,2,3],[4,5,6],[7,8,9]]
s.flatten
[1,2,3,4,5,6,7,8,9]
```

7. Is an array empty?

```
irb(main):012:0> a=[1,2,3,4,5,6,7,8].empty?
=> false
irb(main):013:0> b=[].empty?
=> true
irb(main):014:0> a
=> false
irb(main):015:0> b
=> true
```

8. Task: Take items from the end of an array.
Change the array [1, 2, 3, 4, 5, 6, 7, 8] to [1, 2, 3, 4 ,5]

Solution:

```
array = [1, 2, 3, 4, 5, 6, 7, 8]
array.delete_if {|i| i > 5 }
```

Deletes anything in the array that is higher than 5.

Output [1, 2, 3, 4, 5]

Another solution:

```
irb(main):033:0> a = [1, 2, 3, 4, 5, 6, 7, 8]
=> [1, 2, 3, 4, 5, 6, 7, 8]
irb(main):034:0> a.pop(3)
=> [6, 7, 8]
irb(main):035:0> a
=> [1, 2, 3, 4, 5]
```

9. Find the size of the array [1,2,3,4,5,6,7,8]

```
irb(main):050:0> a=[1,2,3,4,5,6,7,8]
=> [1, 2, 3, 4, 5, 6, 7, 8]
irb(main):051:0> a.count
```

=> 8

10. Change each element of an array.

```
myarray= [2,4,6,8,10]
myarray.map{|x| x*x}
[4,16,36,64, 100]
```

11. What is the difference between each, each_index, and each_with_index?

each: item's value is passed into the code block for each item in the array

each_index: item's index is passed into the code block for each item in the array

each_with_index: code block runs with both the index and the value passed into the block for each item in the array

12. Remove an item from inside an array. Change the array [1,2,3,4,5,6,7,8] to [1,2,3,6,7,8].

```
irb(main):065:0> a.delete_if {|number| [4,5].include?(number)}
=> [1, 2, 3, 6, 7, 8]
```

13. Change the array [1,2,3,4,5,6,7,8] to be [4,5,6,7,8]

```
arr=[1,2,3,4,5,6,7,8]
arr.delete_if{|x|x<4}
```

Another solution:

```
 arr=[1,2,3,4,5,6,7,8]
=> [1, 2, 3, 4, 5, 6, 7, 8]
irb(main):053:0> arr[3..-1]
=> [4, 5, 6, 7, 8]
```