

Programming for Everybody

4. Arrays & Hashes

Arrays

a **collection** of Ruby data that stores a list of values (called **elements**) in a single variable

arrays can contain: **numbers** (in any order, repeated or not), **strings**, **booleans**, **symbols** and even... **other arrays!** :)
(arrays of arrays are called *multidimensional arrays*)

arrays are defined by specifying values between square brackets [], separated by commas

```
my_array = ["Bob", "Joe", "Zack"]
```

Arrays (cont.)

each element in the array has what's called an **index** - > the first element is at index **0**, the next is at index 1, the following is at index 2, etc.

we can **access** elements of an array by putting the index within **square brackets**

```
array = [5, 7, 9, 2, 0]  
array[1]
```

(returns "7", since "7" is at index 1)

Arrays (cont.)

we can **add** elements to an array

`.push(new element)` or `<< new element`

we can **delete** elements from an array

`.delete_at(index)`

Hashes

a **collection** of Ruby data that stores a list of **key-value pairs** in a single variable

we can use **any** Ruby object as a key or value

values are assigned to keys using =>

```
hash_name = {  
  key1 => value1,  
  key2 => value2,  
  key3 => value3  
}
```

Hashes (cont.)

creating a new hash:

```
my_hash = {  
  "cat" => "Garfield",  
  "dog" => "Snoopy"  
}
```

or

```
my_hash = Hash.new  
my_hash["cat"] = "Garfield"  
my_hash["dog"] = "Snoopy"
```

Hashes (cont.)

we can **access** a specific key-value pair like so:

```
my_hash = {  
  "cat" => "Garfield",  
  "dog" => "Snoopy"  
}
```

```
puts my_hash["cat"]
```

(will print out "Garfield")

Hashes (cont.)

we can **add** key-value pairs to an hash

```
my_hash = {  
  "cat" => "Garfield",  
  "dog" => "Snoopy"  
}  
my_hash["mouse"] = "Mickey"
```

we can **delete** key-value pairs from a hash

```
my_hash.delete("dog")
```


Iterating... again!

we can loop over an array or a hash, in which case we say we're **iterating** over them

Iterating... again!

1. Iterating over an Array

```
my_array = ["Bob", "Joe", "Zack"]
```

```
my_array.each do | names |  
  puts names  
end
```

or

```
my_array.each { | names | puts names }
```

(both will print out Bob, Joe, Zack)

Iterating... again!

2. Iterating over a Multidimensional Array

```
my_array = [["Bob", "Joe", "Zack"], ["Zoe", "Nina", "Chloe"]]
```

```
my_array.each do | sub_array |  
  sub_array.each do | names |  
    puts names  
  end  
end
```

(prints out Bob, Joe, Zack, Zoe, Nina, Chloe)

Iterating... again!

3. Iterating over a Hash

we need **two placeholders** to represent each key/value pair:

```
students_grades = {  
  "Zack" => 7,  
  "Zoe" => 10  
}
```

```
students_grades.each do | student, grade |  
  puts "#{student}: #{grade}"  
end
```

(prints out Zack: 7, Zoe: 10)

Thank you! :)