LESSON 4

app.rb

```ruby
cities = %w[Kabul Herat]
cities << "Balkh"
h = { name: "Mariam" }
```

Ruby

Arrays & Hashes

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

RUBY PROGRAMMING

LESSON 4

Arrays and Hashes

Creating, reading and changing arrays and hashes, and iterating over them

Arrays

arrays.rb

```ruby
cities = ["Kabul", "Herat", "Balkh"]
cities = %w[Kabul Herat Balkh]     # same
cities[0]          # "Kabul"
cities[-1]         # "Balkh"
cities[0, 2]       # ["Kabul", "Herat"]
cities.first; cities.last; cities.length
```

- An array is an ordered list that can hold any objects and grows as needed. Index from zero, …
- The percent w shortcut makes an array of words without quotes and commas.

01

Changing Arrays

change.rb

```ruby
nums = [5, 3, 8]
nums << 1               # [5, 3, 8, 1]
nums.pop                # 1
nums.sort               # [3, 5, 8]  (new array)
nums.sort!              # changes nums itself
nums.include?(8)        # true
[1, [2, 3]].flatten     # [1, 2, 3]
[1, 1, 2].uniq          # [1, 2]
```

- push, or the shovel operator, adds to the end. pop removes from the end, shift from the ...
- Methods ending with an exclamation mark, like sort bang, change the array itself instead of ...

02

Hashes

hashes.rb

```ruby
student = { name: "Mariam", city: "Kabul", score: 95 }
student[:name]              # "Mariam"
student[:age]               # nil
student.fetch(:age, 18)     # 18
student[:score] = 97
student.delete(:city)
student.key?(:name)         # true
```

- A hash links keys to values. With symbol keys, use the short form: name colon, then the value.
- Read with square brackets. A missing key returns nil, while fetch raises an error or uses a …

03

Iterating Hashes

count.rb

```ruby
counts = Hash.new(0)
text.split.each { |w| counts[w.downcase] += 1 }

counts.each do |word, n|
  puts "#{word}: #{n}"
end
counts.max_by { |_, n| n }      # most common word
```

- each on a hash gives you the key and the value. keys and values return arrays.
- A hash with a default value, Hash dot new of zero, makes counting easy.

04

Nested Data

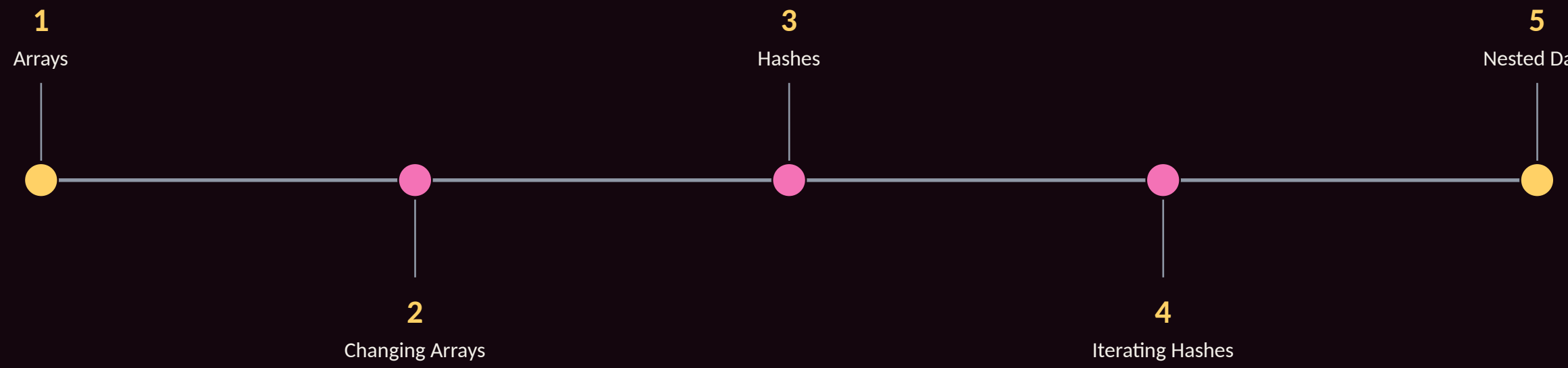
nested.rb

```ruby
students = [
  { name: "Mariam", scores: { math: 95, art: 88 } },
  { name: "Ahmad",  scores: { math: 72 } }
]
students[0][:scores][:math]          # 95
students[1].dig(:scores, :art)       # nil (no error)
students.each { |s| puts s[:name] }
```

- Arrays of hashes are how Ruby represents records, just like JSON. dig safely reaches deep …
- This shape is what you get from APIs, CSV files and databases.

05

Lesson Map


1
Arrays
2
Changing Arrays
3
Hashes
4
Iterating Hashes
5
Nested Data

Sources and Further Reading

arrays.rb

```ruby
cities = ["Kabul", "Herat", "Balkh"]
cities = %w[Kabul Herat Balkh]     # same
cities[0]          # "Kabul"
cities[-1]         # "Balkh"
cities[0, 2]       # ["Kabul", "Herat"]
cities.first; cities.last; cities.length
```

RUBY PROGRAMMING · 4

hashes.rb

```ruby
student = { name: "Mariam", city: "Kabul", score: 95 }
student[:name]             # "Mariam"
student[:age]              # nil
student.fetch(:age, 18)    # 18
student[:score] = 97
student.delete(:city)
student.key?(:name)        # true
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari