Data Structures

Strings, Symbols, Arrays, Hashes, Sets, Structs

J. Scott Johnson
AppData, LLC
scott@appdata.com

String

- my_string = “foo”
- That’s all folks!

Interpolation

- my_name = “Scott”
- puts “my_name” <== gives ‘my_name’
- puts “#{my_name” <== gives ‘Scott’
- double quoted strings can interpolated
- single quoted strings cannot
- single quoted strings are faster (tiny bit)

Integer

- A lowly number
- Again that's all folks

Symbol

- lower overhead string
- denoted by :
- highly internal to ruby

Array

- collection of things represented by [a,b]
- things might be strings, integers, db objects, anything
- normally named as a plural i.e. rows
- rows.each do |row|
- # do something to or with the row
- end

Array 2

- names = ["scott", "alex", "max", "shelley"]
- names.each do |name|
- puts name
- end

Array 3

- names.sort.each do |name|
- puts name
- end
- or the 1 liner:
- names.sort.each {|name| puts name }

Array 4

- between do and end or { } is a “block”
- the |name| means a variable that only exists **local** to that block
- local = doesn’t exist outside the block
- If you want it to escape outside the block then define it before you use it

Array 5

- ctr = 0
- names.sort.each {|name| puts name; ctr = ctr + 1 }
- puts ctr

Hash

- Key Value Pair
- Also called associative array
- my_hash = {}
- my_hash[:apples] = “red”
- my_hash[:grapes] = “green”

Set

- Merge of an array and a hash where things can exist only once
- Very useful for preventing duplicates

- set = Set.new
- set << “bar”
- set << “bar”
- Will now exist only once in the set

Struct

- Here things get fun -- build something custom
- my_obj = Struct.new(:name =>
  “bar”, :method => “foo”)
- my_obj is really a hash but it now responds to .name and .foo
- “psuedo object”
- hash like flexibility with object like properties

Putting it Together

- Array as a collection of ANYTHING
- Example: an array of structs