CSE341 – Section 8
Arrays (and such), Blocks, Reflection

Sunjay Cauligi

February 28th, 2013

Yay it's my birthday
Arrays in Ruby

Arrays

- Has a lot of functionality you would expect
- Indexed by integers
- Dynamically typed
  - So array values can have mixed types

Properties of Ruby Arrays

- Very flexible in use
- Lots of supported operations
- Less efficient than arrays in other languages
Using Arrays

More than a List
Arrays in Ruby can be used for
- Lists
- Tuples
- Stacks (push/pop)
- Queues (push/shift/unshift)
- Sets (&/|/-, that is, intersection/union/subtraction)
- Much more

Incredibly Dynamic
Many operations allowed that would be errors in other languages
- Accessing/assigning out of bounds
- Replacing subsets with subsets of a different length
Some Ruby Data Structures

Arrays  Hashes  Ranges

Looping Over Arrays

Built-In Methods
- Many methods that apply iterative functions
  - `each`, `collect/map`, `inject/reduce`, `select/filter`, ...
- Typically take a block that is called for each element
  - More on blocks later

Loops in Ruby
- Traditional loops such as `for` and `while` do exist
- In Ruby, hardly ever used
- Most “loops” can be implemented using these iteration functions

Sunjay Cauligi  CSE341 – Section 8
Hashes in Ruby

Hashes
- Similar to arrays, but index can be any object
- Unordered, since no natural way to order keys

Using Hashes
- Many of the same methods as arrays
  - But, of course, many additional methods specific to hashes
- When used as a record, usually indexed with symbols
  - A little faster/more efficient than using string keys
- Methods with many possible arguments will sometimes take a hash instead
Ranges in Ruby

Ranges

- Similar to arrays of successive numbers
- Much more efficient to operate with

Using Ranges

- Can be iterated over in much the same way as arrays
  - Can sometimes even be used interchangeably (duck typing)
- Very useful for processing over a sequence of contiguous numbers
### Almost Closures

- Bits of code, similar to anonymous functions, that can passed to method calls
- Have lexical scope, like closures
- However, cannot be assigned to variables

### Blocks and Ruby

- Blocks are used extensively in Ruby code
- Very convenient, just like anonymous functions
- Ruby’s standard library has many methods that take blocks
Using Reflection

**In the REPL**
- Can determine the “type” (class) of an object by calling its `class` method.
- Can determine what methods it defines by calling its `methods` method.
- Useful for exploring the language while in the REPL.

**In Code**
- Can branch the program at runtime depending on the class of an object.
- Can even branch depending on what methods are defined by an object.
Exploratory Programming

- Write programs by fiddling around in the REPL and trying stuff out
- Don’t have to deal with write/compile/debug cycle
- Can interactively test portions of code