

Learning Ruby - 13

Grabbag III - The Return of the Grabbag

When Things Go Wrong

- This is our 13th set of Ruby slides ... something's bound to go wrong!
- Ruby can help with an integrated exception handling mechanism
- Just remember the sequence: **begin ... rescue ... ensure ... end**
- There's also **catch** and **throw**, but not as you know them

A Missing Example

Only Joking!

```
begin

  file = File.open( 'missing.txt', "r" )

  while data = file.gets
    puts data
  end # of while.

rescue SystemCallError

  puts "Problems with file: #{${!}} ... does it exist?"

ensure

  file.close unless file.nil?

end # of begin.
```

Not using \$!

```
begin
  eval some_code

rescue SyntaxError, NameError => sillymsg
  print "Your code does not compile: " + sillymsg

rescue StandardError, NameError => badsmg
  print "Your running code has errors: " + badsmg

end # of begin.
```

Catch That Throw!

```
if something
  if another_thing
    if something_else
      if one_last_chance
        if oh_no
          # Get me outta here!
          throw :phew
        end # of if.
      end # of if.
    end # of if.
  end # of if.
end # of if.

catch :phew
  # Code continues "normally" ...
```

More ... Ruby So Far

- When bad things can happen, be sure to protect your code with an appropriate exception handling mechanism
- Remember, it's simply:

begin ... rescue ... ensure ... end