Dealing with Errors

Paolo Perrotta
FREELANCE DEVELOPER
@nusco

Terser Exception Management

A Little Extra: Advanced Error Management

The Basic *rescue*

```
begin
    login()
rescue
    log("Login error")
end
```

Rescuing a Specific Exception

```
begin
    login()
rescue UnknownUsernameError
    log("Unknown username")
end
```

Getting the Exception as a Variable

```
begin
    login()
rescue UnknownUsernameError => e
    log(e)
end
```

Re-raising the Exception

```
begin
    login()
rescue UnknownUsernameError => e
    log(e)
    raise e
end
```

else Clauses

```
begin
    login()
rescue UnknownUsernameError => e
    log(e)
    raise e
else
    puts "Welcome back!"
end
```

ensure Clauses

```
begin
    login()
rescue UnknownUsernameError => e
    log(e)
    raise e
else
    puts "Welcome back!"
ensure
    log("User login")
end
```

Use *raise/rescue* to manage exceptions, not *throw/catch*.

What We Did

```ruby
TEXT_FILE = "romeo-juliet.txt"

# Load the words from a file
def words_from_file(text_file)
    File.read(text_file).downcase.gsub(/[^a-z]/, " ").split
rescue
    puts "Give me #{text_file} and let's get the party started!"
    exit
end

# Load the list of words in the text
words = words_from_file(TEXT_FILE)

# Create a dictionary of word counts
word_count = {}
words.each do |w|
    word_count[w] = 0 unless word_count[w]
    word_count[w] += 1
end

```