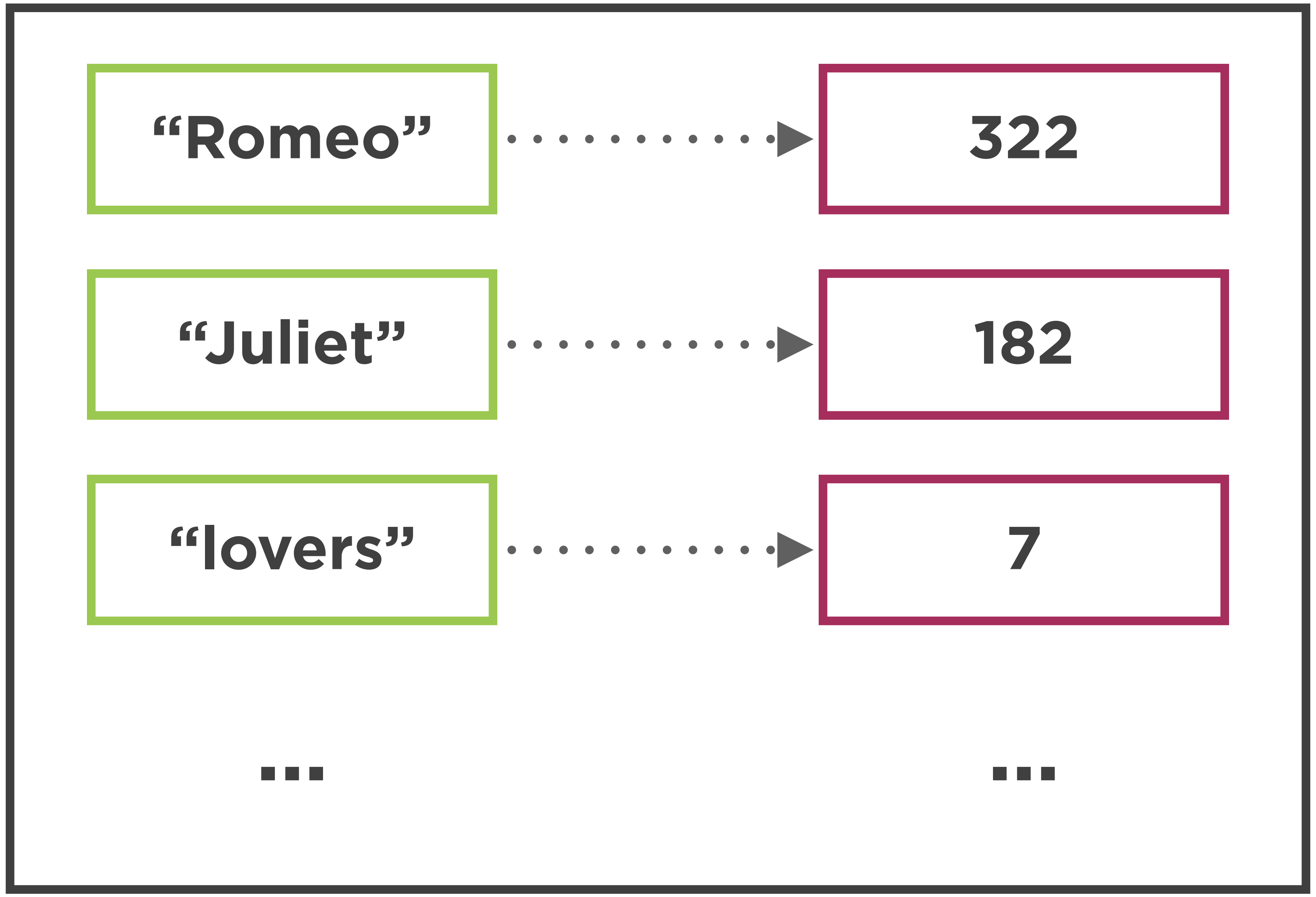
Writing Conditionals

Paolo Perrotta
FREELANCE DEVELOPER

@nusco

The *word_count* Hash



“Romeo” 322

“Juliet” 182

“lovers” 7

... ...

Cleaning Up the Conditional

Refactoring the Conditional

```
if word_count[word] == nil
    word_count[word] = 1
else
    word_count[word] += 1
end
```

Refactoring the Conditional

```
if word_count[word] == nil
    word_count[word] = 0
end

word_count[word] += 1
```

Refactoring the Conditional

```ruby
word_count[word] = 0 if word_count[word] == nil
word_count[word] += 1
```

In Ruby, only *false* and *nil* count as “false” in a conditional.

Refactoring the Conditional

```
word_count[word] = 0 if !word_count[word]
word_count[word] += 1
```

Refactoring the Conditional

```ruby
word_count[word] = 0 unless word_count[word]
word_count[word] += 1
```

A Little Extra: Multiple Cases

Chains of Conditions

```
if items == 1
    count = "one"
elsif items == 2
    count = "two"
elsif items == 3
    count = "three"
else
    count = "a lot"
end
```

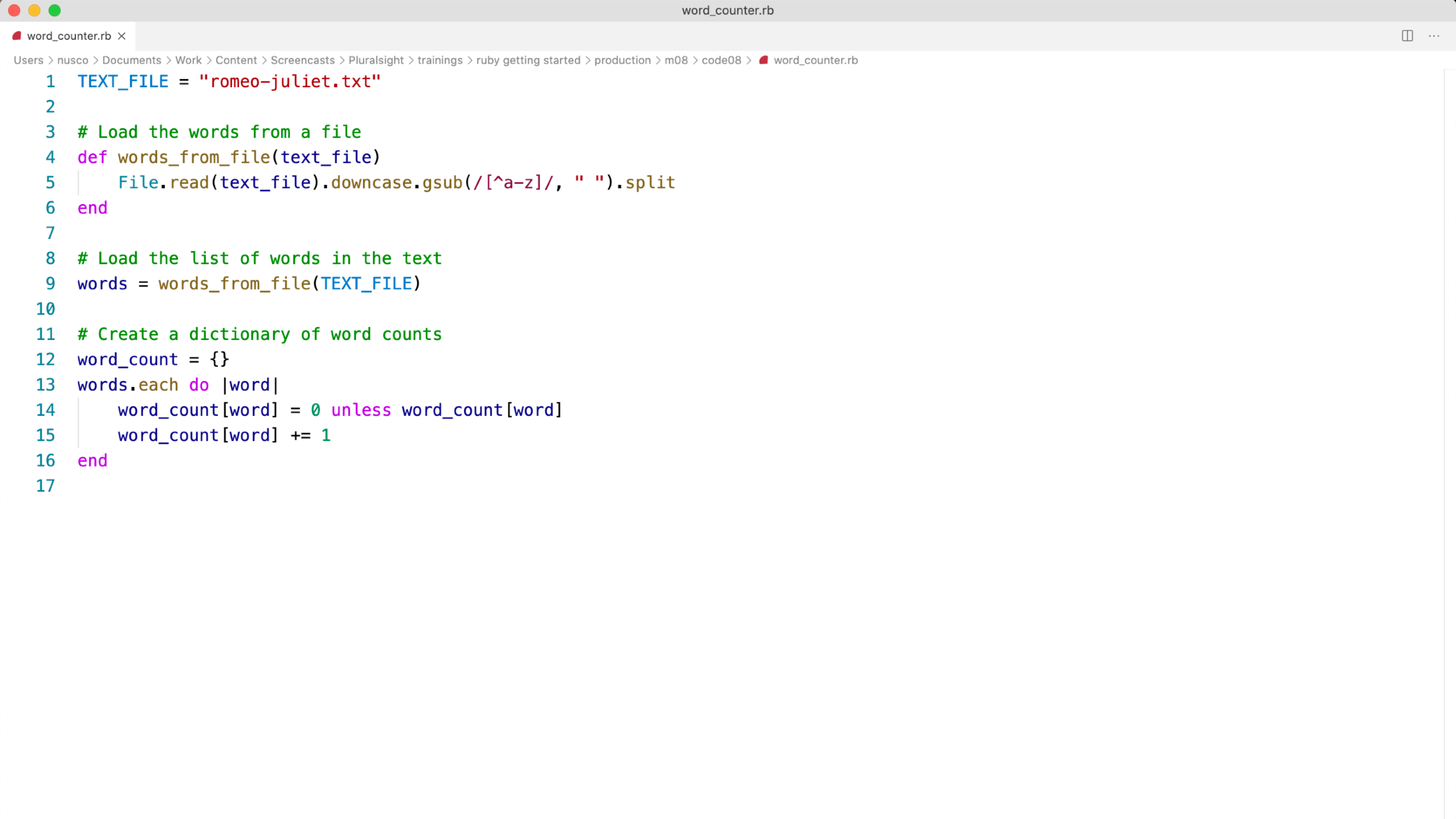
case Statements

```
case items
when 1
    count = "one"
when 2
    count = "two"
when 3
    count = "three"
else
    count = "a lot"
end
```

case and Ranges

```
case distance
when (1..3)
    category= "walking distance"
when (4..10)
    category = "short trip"
else
    category = "long trip"
end
```

What We Did


word_counter.rb
word_counter.rb
Users > nusco > Documents > Work > Content > Screencasts > Pluralsight > trainings > ruby getting started > production > m08 > code08 > word_counter.rb

```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
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 |word|
    word_count[word] = 0 unless word_count[word]
    word_count[word] += 1
end

```