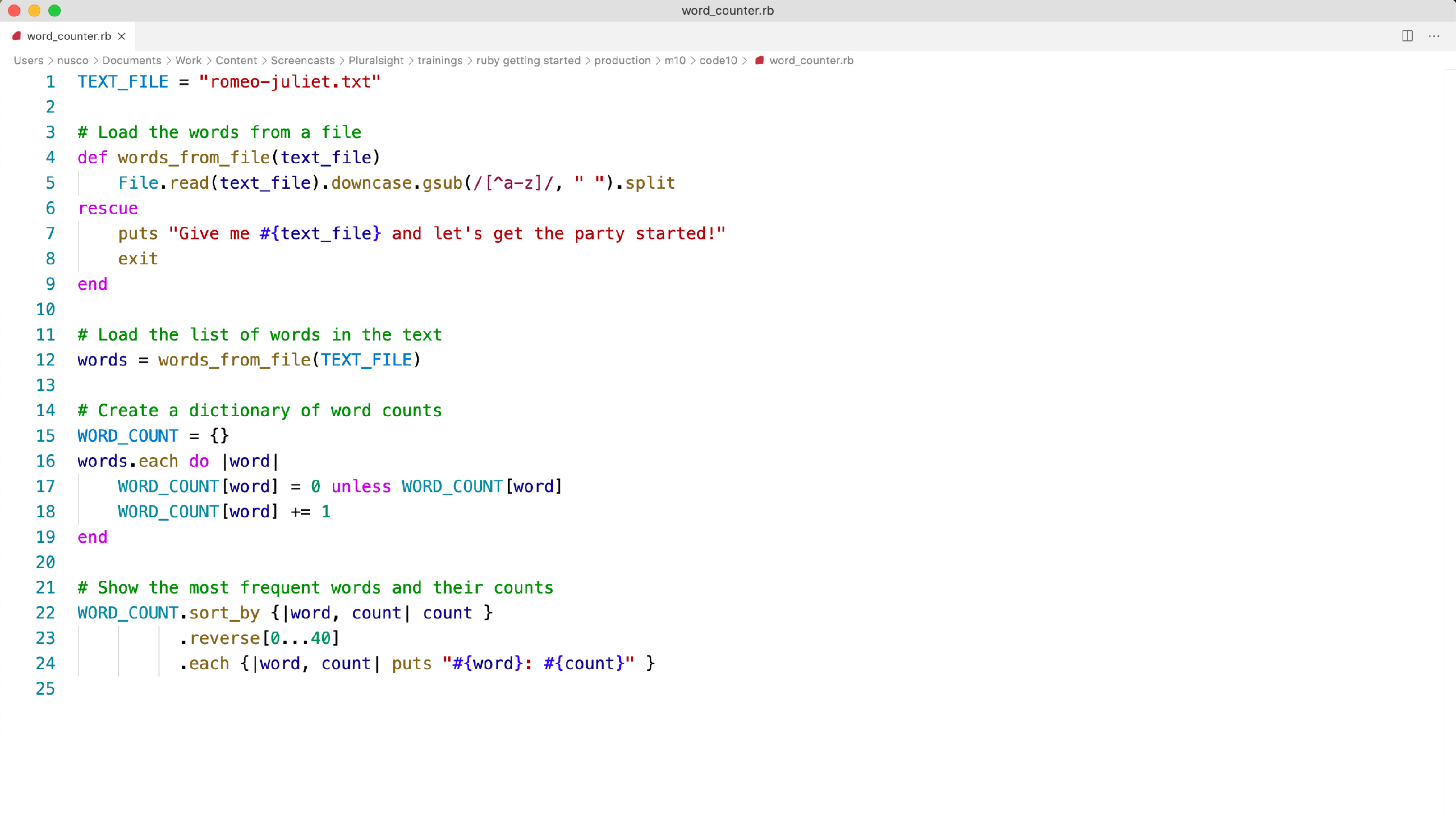
Wrapping Up the Program

Paolo Perrotta
FREELANCE DEVELOPER

@nusco


word_counter.rb
word_counter.rb
Users > nusco > Documents > Work > Content > Screencasts > Pluralsight > trainings > ruby getting started > production > m10 > code10 > word_counter.rb

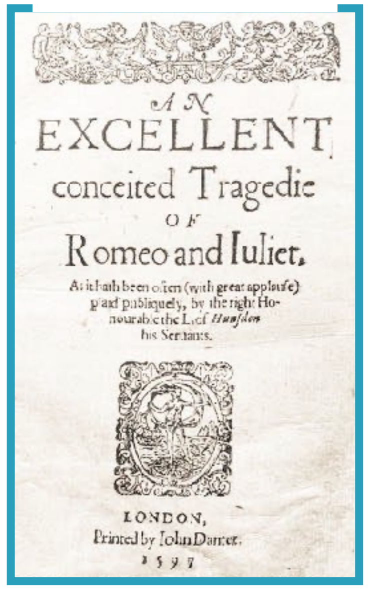
```ruby
TEXT_FILE = "romeo-juliet.txt"

# Load the words from a file
def words_from_file(text_file)
    File.read(text_file).downcase.gsub(/[^a-z]/, " ").split
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 |word|
    WORD_COUNT[word] = 0 unless WORD_COUNT[word]
    WORD_COUNT[word] += 1
end

# Show the most frequent words and their counts
WORD_COUNT.sort_by {|word, count| count }
          .reverse[0...40]
          .each {|word, count| puts "#{word}: #{count}" }

```

Exploring the Word Counts

Your Turn!

Our Plan, Updated

Romeo and Juliet

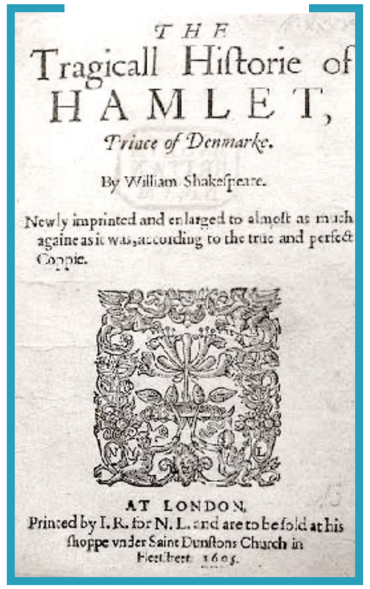


AN
EXCELLENT
conceited Tragedie
OF
Romeo and Iuliet.

As it hath been often (with great applause)
plaid publiquely, by the right Honourable the L. of Hunsdon
his Seruants.

LONDON,
Printed by Iohn Danter.
1597

Hamlet

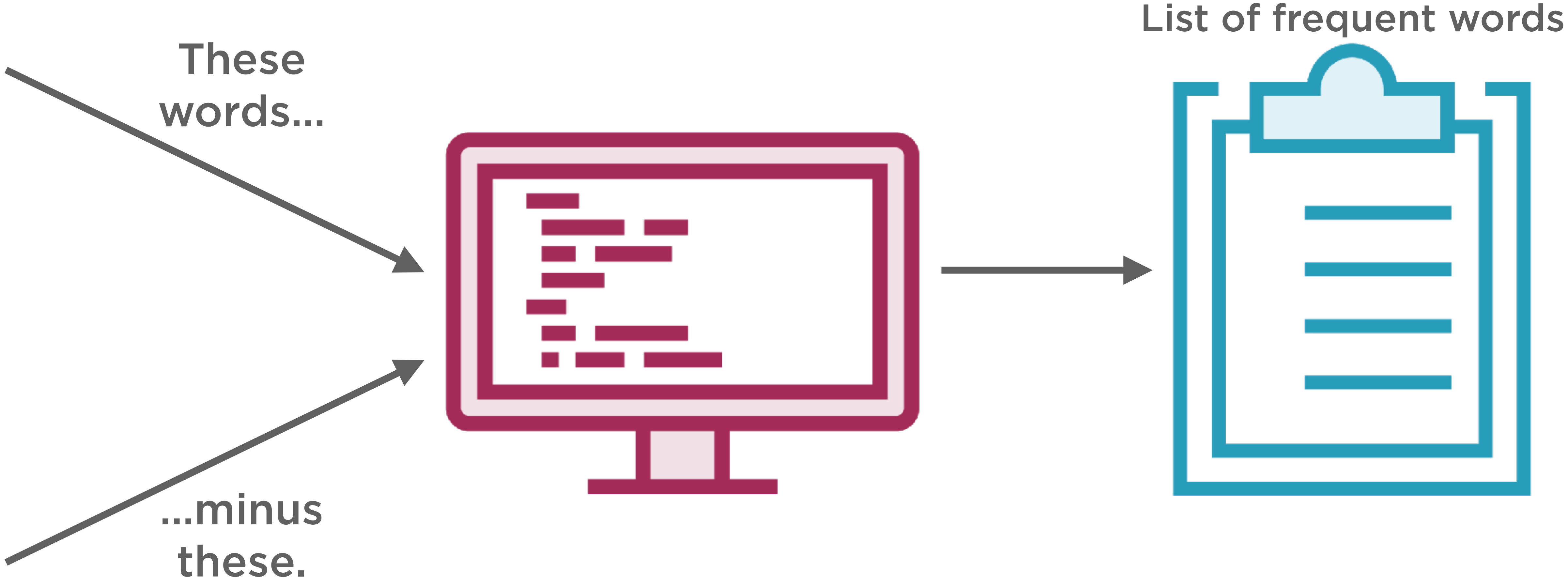


THE
Tragicall Hiſtorie of
HAMLET,
Prince of Denmarke.

By William Shakeſpeare.

Newly imprinted and enlarged to almoſt as much againe as it was, according to the true and perfect Coppie.

AT LONDON,
Printed by I. R. for N. L. and are to be ſold at his
ſhoppe vnder Saint Dunſtons Church in
Fleetſtreet. 1605.

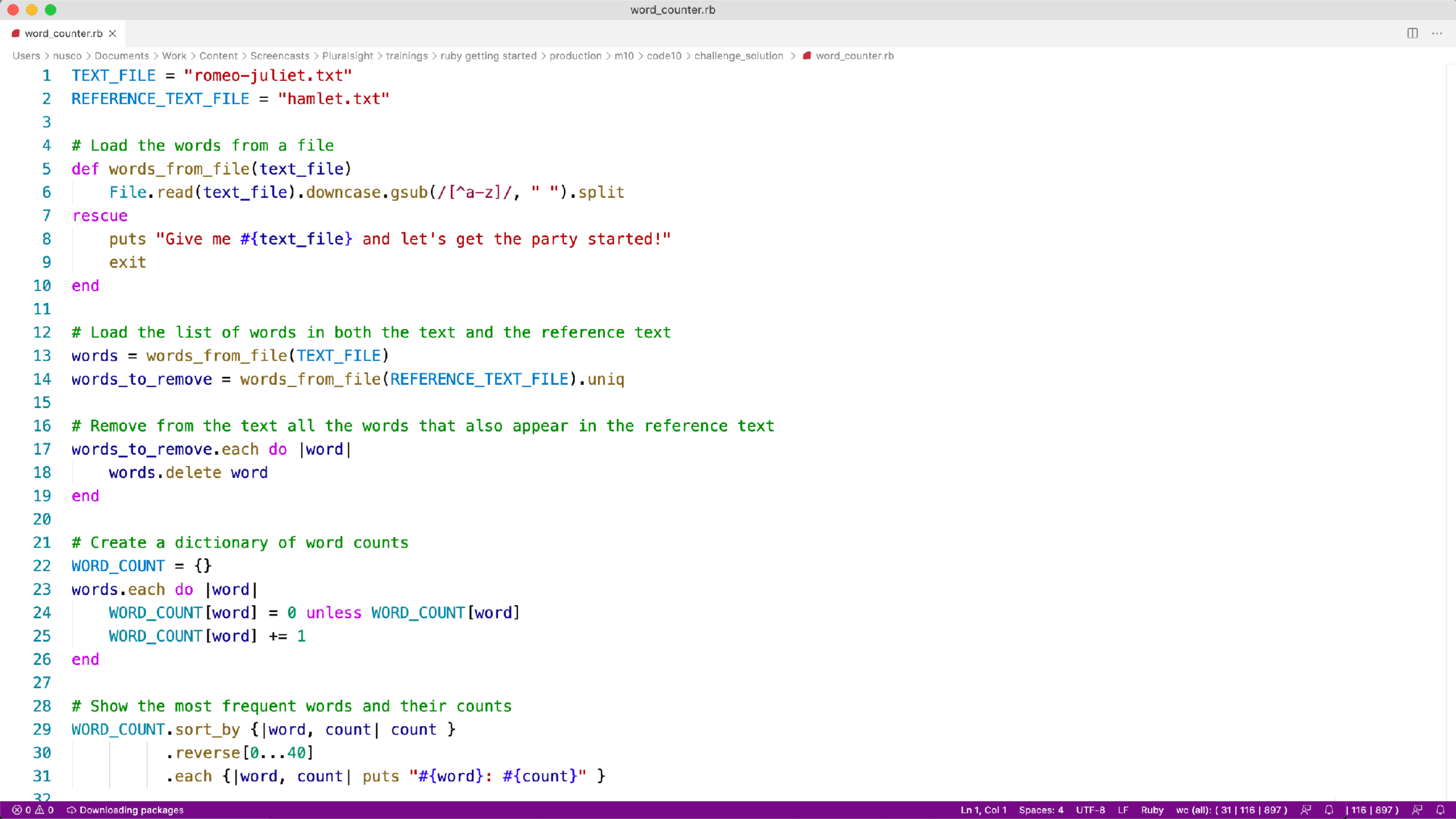


These words...

...minus these.

List of frequent words

A Solution to the Challenge


word_counter.rb
word_counter.rb ×
Users > nusco > Documents > Work > Content > Screencasts > Pluralsight > trainings > ruby getting started > production > m10 > code10 > challenge_solution > word_counter.rb
⊗ 0 ⚠ 0
Downloading packages
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Ruby
wc (all): (31 | 116 | 897)
| 116 | 897)

```ruby
TEXT_FILE = "romeo-juliet.txt"
REFERENCE_TEXT_FILE = "hamlet.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 both the text and the reference text
words = words_from_file(TEXT_FILE)
words_to_remove = words_from_file(REFERENCE_TEXT_FILE).uniq

# Remove from the text all the words that also appear in the reference text
words_to_remove.each do |word|
    words.delete word
end

# 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

# Show the most frequent words and their counts
WORD_COUNT.sort_by {|word, count| count }
          .reverse[0...40]
          .each {|word, count| puts "#{word}: #{count}" }
```

Wrapping up This Training

Overview

Running Ruby Code

Reading Input Data

Having Fun with Strings

Defining Functions

Juggling Collections

Understanding Iteration

Controlling Program Flow

Dealing with Errors

Wrapping Up the Program

```ruby
3.times do
    puts "Thank you!"
end
```

Paolo Perrotta
FREELANCE DEVELOPER

@nusco