Tutorials: **Resources:**   **Forums:**

Ruby-on-Rails

Become a member
of a new community of
IT evaluators.

eWEEK **labs** REAL IT PROS
REAL REVIEWS
REAL WORLD

Dev Articles Forums

ADO.NET

Apache

ASP

ASP.NET

C#

C++

ColdFusion

COM/COM+

Delphi-Kylix

Design Usability

Development Cycles

DHTML

RUBY-ON-RAILS   RSS

Comparing and Manipulating Strings in Ruby

By: O'Reilly Media

Rating: ★★★★☆ / 3

2008-08-14

• Search For More Articles!
• Disclaimer
• Author Terms

Table of Contents:

Rate this Article: Poor		Best

Click Here to download

CLOSE ✕

Seo hosting - Seo server

Reseller hosting

Dedicated server

Links From PR9 Sites

Comparing Strings

Sometimes you need to test two strings to see if they are the same or not. You can do that with the == method. For example, you might want to test a string before printing something:

```
print "What was the question again?" if question == ""
```

Also, here are two versions of the opening paragraph of Abraham Lincoln's Gettysburg Address, one from the so-called Hay manuscript, the other from the Nicolay (see *http://www.loc.gov/exhibits/gadd/gadrft.html*):

```
hay = "Four score and seven years ago our fathers brought forth, upon this continent,
a new nation, conceived in Liberty, and dedicated to the proposition that all men are
created equal."

nicolay = "Four score and seven years ago our fathers brought forth, upon this
continent, a new nation, conceived in liberty, and dedicated to the proposition
```

```
"all men are created equal""
```

The strings are only slightly different (for example, *Liberty* is capitalized in the Hay version). Let's compare these strings:

```
hay == nicolay # => false
```

The result is `false`, because they must match exactly. (We'll let the historians figure out how to match them up.) You could also apply the `eql?` method and get the same results, though `eql?` and `==` are slightly different:

- `==` returns true if two objects are `String`s, false otherwise.
- `eql?` returns true if two strings are equal in length and content, false otherwise.

Here `eql?` returns false:

```
hay.eql? nicolay # => false
```

Yet another way to compare strings is with the `<=>` method, commonly called the *spaceship operator*. It compares the character code values of the strings, returning `-1` (less than), `0` (equals), or `1` (greater than), depending on the comparison, which is case-sensitive:

```
"a" <=> "a" # =>0
"a" <=> 97.chr # => 0
"a" <=> "b" # => -1
"a" <=> "`" # => 1
```

A case-insensitive comparison is possible with `casecmp`, which has the same possible results as `<=>` (`-1,0,1`) but doesn't care about case:

```
"a" <=> "A" # => 1
"a".casecmp "A" # => 0
"ferlin husky".casecmp "Ferlin Husky" # => 0
"Ferlin Husky".casecmp "Lefty Frizzell" # => -1
```
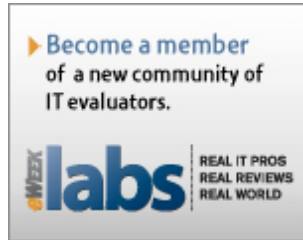
Next: Manipulating Strings >>

More Ruby-on-Rails Articles

More By O'Reilly Media

Comments On: Comparing and Manipulating Strings in Ruby

· This article is an excerpt from the book "Learning Ruby," published by O'Reilly. We...

>>> **Post your comment now!**

Buy this book…

This article is excerpted from chapter four of *Learning Ruby*, written by Michael Fitzgerald (O'Reilly; ISBN: 0596529864). Check it out today at your favorite bookstore. Buy this book now.

RUBY-ON-RAILS ARTICLES

 - Application Deployment with Capistrano
 - Deploying an Application to the Server
 - Secure Application Deployment with Ruby on R…
 - Installing the Database for Application Depl…
 - Application Deployment
 - Recording Acceptance Tests
 - Commands for Acceptance Testing
 - Acceptance Testing
 - Checkout and Order Processing: the User Side
 - Checkout and Order Processing: the Administr…
 - Online Order Processing: Using PayPal
 - Order and Payment Handling for an Ecommerce …
 - Checkout and Order Processing for an Ecommer…
 - Getting Text and Currency Working for Multip…
 - Translating a Site for Multiple Language Sup…

Find More Ruby-on-Rails Articles

1

Developer Tutorials:

Developer Resources:

Developer Forums:

IT Professionals:

IT Marketing:

Open Source / Microsoft: