

SKRIPTSPRACHEN

* RUBY *

01 - VORBESPRECHUNG

NAUMANN
WINTERSEMESTER 201

SYSTEM(E/KOMPONENTEN)

Ruby

RDoc/ri

RubyGems

Ruby on Rails

RUBY

- ✱ Objektorientierte Skriptsprache
- ✱ Kern: Interpreter
- ✱ Aktuelle Version: 2.1.3
- ✱ Die Standardbibliothek umfasst mehr als 9000 Methoden.
- ✱ Programmentwicklung: beliebiger Editor (*emacs*, *vi*, ...) oder geeignete IDE (*Eclipse*, *Netbeans*, ...)

VERWENDUNG VON RUBY

```
$ ruby -v
```

```
ruby 1.9.2p180 (2011-02-18 revision 30909) [x86_64-darwin10.7.0]
```

```
$ ruby
```

```
puts "Hello, world!" ^D
```

```
Hello, World!
```

```
puts "Hello, Ruby Programmer"  
puts "It is now #{Time.now}"
```

meinruby.rb

```
$ ruby meinruby.rb
```

```
Hello, Ruby Programmer
```

```
It is now 2014-10-27 15:59:44 +0100
```

```
#!/usr/local/bin/ruby  
puts "Hello, Ruby Programmer"  
puts "It is now #{Time.now}"
```

meinruby.rb

```
$ ./meinruby.rb
```

```
Hello, Ruby Programmer
```

```
It is now 2014-10-27 15:59:56 +0100
```


RUBY INTERACTIVE

```
$ irb
```

```
irb(main):001:0> def sum(n1, n2)
```

```
irb(main):002:1> n1 + n2
```

```
irb(main):003:1> end
```

```
=> nil
```

```
irb(main):004:0> sum(3, 4)
```

```
=> 7
```

```
irb(main):005:0> sum("cat", "dog")
```

```
=> "catdog"
```

```
irb(main):006:0> exit
```

```
$
```


DOKUMENTATION

- ✱ Zur Dokumentation der in Bibliotheken enthaltenen Methoden wird in Ruby ein System namens *RDoc* verwendet.
- ✱ Die Dokumentation kann über verschiedene Webseiten (<http://www.ruby-doc.org> und <http://rubydoc.info>)
- ✱ oder durch das Kommandozeilen-Tool *ri* abgerufen werden.

HILFEN

```
$ ri assoc
```

More than one method matched your request. You can refine your search by asking for information on one of:

```
Array#assoc [Ruby 1.9.1]
```

```
Array#rassoc [Ruby 1.9.1]
```

```
Hash#assoc [Ruby 1.9.1]
```

```
Hash#rassoc [Ruby 1.9.1]
```

```
$ ri Array.assoc
```

```
----- Array#assoc
```

```
array.assoc(obj) -> an_array or nil
```

```
-----
```

Searches through an array whose elements are also arrays comparing obj with the first element of each contained array using obj.==. Returns the first contained array that matches (that is, the first associated array), or nil if no match is found. See also Array#rassoc.

```
:::
```


HILFEN

\$ ri GC

----- Class: GC

The GC module provides an interface to Ruby's mark and sweep garbage collection mechanism. Some of the underlying methods are also available via the ObjectSpace module.

Class methods:

count, disable, enable, malloc_allocated_size, malloc_allocations,
start, stress, stress=

Instance methods:

garbage_collect

RUBYGEMS

- ✻ RubyGems ist das Paketverwaltungssystem für Ruby und seit Version 1.9 Bestandteil der Standardbibliothek.
- ✻ Es erlaubt die einfache (Nach)Installation/Aktualisierung von Programmen und Bibliotheken.
- ✻ Verwendung:
*gem install **Name***
*gem uninstall **Name***
gem list --local
gem list --remote

RUBY ON RAILS

- ✻ Open-source *Web Application Framework*
- ✻ Basiert auf der „Model View Controller“-Architektur (MVC) und besteht aus fünf Modulen (Active Support, Active Record, Action Pack, Action Mailer, Active Resource).
- ✻ Aktuelle Version: 4.1.6

ARBEITSUMGEBUNG

Zum Einrichten einer lokalen Arbeitsumgebung sind in der Regel folgende Arbeitsschritte auszuführen:

- ✱ Laden und installieren einer stabilen (und halbwegs aktuellen) Rubyversion.

Quelle: <https://www.ruby-lang.org/de/>

- ✱ Zur Entwicklung von Programmen kann prinzipiell jeder beliebiger Editor verwendet werden.
Komfortabler sind Entwicklungsumgebungen (IDE) wie Eclipse, RubyMine oder NetBeans.

ARBEITSUMGEBUNG

- ✱ Die Einrichtung der IDE setzt in den drei genannten Fällen voraus, dass auf dem Computer eine möglichst aktuelle JRE (*Java Runtime Environment*) installiert ist.
- ✱ Bei der Verwendung von Eclipse oder NetBeans ist die Installation weiterer Software (Plugins DLTk für *Eclipse* bzw. Ruby und Rails Plugin für *Netbeans* - <http://plugins.netbeans.org/plugin/38549>).
- ✱ Anschließend muss in der IDE das installierte Ruby identifiziert werden.

SEMINARPLAN

- ✻ 28.10.14 - Vorbesprechung
- ✻ 04.11.14 - Erste Schritte
- ✻ 11.11.14 - Klassen, Objekte und Variablen
- ✻ 18.11.14 - Datenstrukturen, Blöcke und Iteratoren
- ✻ 25.11.14 - Module und Mixins
- ✻ 02.12.14 - Standardtypen und reguläre Ausdrücke
- ✻ 09.12.14 - Expressions
- ✻ 16.12.14 - Ein- und Ausgabe

SEMINARPLAN

- ✿ 06.01.15 - Exceptions
- ✿ 13.01.15 - Unit Testing
- ✿ 20.01.15 - Ruby und XML
- ✿ 27.11.15 - Ruby im Netzwerk
- ✿ 03.11.15 - Metaprogrammierung
- ✿ 10.02.15 - Klausur

RESOURCEN

- ✻ <http://www.ruby-lang.org/de/>
- ✻ <http://www.netbeans.org/downloads/index.html>
- ✻ <http://www.eclipse.org/>
- ✻ <http://www.ruby-doc.org>
- ✻ <http://rubydoc.info>

LITERATUR

- ✿ Programming Ruby 1.9 & 2.0: The Pragmatic Programmer's Guide. D. Thomas with Andy Hunt, 4. Auflage, The Pragmatic Bookshelf, 2013.
- ✿ The Ruby Programming Language. D. Flanagan und Y. Matsumoto, O'Reiley, 2008.
- ✿ The Ruby Cookbook. Recipes for Object Oriented Scripting. Lucas Carlson und Leonard Richardson, O'Reiley, 2006.
- ✿ Try Ruby! Interaktives Ruby-Tutorium, das den Einstieg sehr erleichtert:
<http://tryruby.hobix.com/>.
- ✿ Agile Web Development with Rails. Sam Ruby, Dave Thomas und David H. Hanson, 4. Auflage, The Pragmatic Bookshelf, 2011.