

 eBook Gratuit

APPRENEZ rorygems

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#rorygems

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec Rubygems.....	2
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
Installation sous Linux.....	2
Installer une version ruby précompilée et des gemmes ruby (Ubuntu).....	3
Changer de source de pierres précieuses.....	3
Chapitre 2: Anémone.....	4
Syntaxe.....	4
Paramètres.....	4
Remarques.....	4
Exemples.....	4
Analyse de site de base.....	4
Chapitre 3: Changer la source de gem dans Gemfile.....	5
Exemples.....	5
Utiliser le repo github pour une gemme spécifique dans Gemfile.....	5
Utiliser ref, branch, git pour gem spécifique dans gemfile.....	5
Chapitre 4: JSON.....	6
Introduction.....	6
Syntaxe.....	6
Paramètres.....	6
Exemples.....	6
Hash to JSON.....	6
JSON à Hash.....	6
JSON alternatif à Hash.....	6
Hash alternatif à JSON.....	7
Chapitre 5: Maruku.....	8
Introduction.....	8
Syntaxe.....	8

Paramètres.....	8
Exemples.....	8
Lecture du Markdown dans Maruku.....	8
Markdown to HTML.....	8
Crédits.....	10

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [rubygems](#)

It is an unofficial and free rubygems ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official rubygems.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec Rubygems

Remarques

Les gemmes peuvent être utilisées pour étendre ou modifier les fonctionnalités des applications Ruby. Généralement, ils sont utilisés pour distribuer des fonctionnalités réutilisables qui sont partagées avec d'autres Rubyists pour les utiliser dans leurs applications et bibliothèques. Certains gems fournissent des utilitaires de ligne de commande pour automatiser les tâches et accélérer votre travail.

Le logiciel RubyGems vous permet de télécharger, d'installer et d'utiliser facilement les progiciels Ruby sur votre système. Le progiciel est appelé une `gem` et contient une application ou une bibliothèque Ruby empaquetée.

Exemples

Installation ou configuration

Si vous n'avez pas installé *RubyGems*, il existe toujours l'approche pré-gem pour obtenir le logiciel, en le faisant manuellement:

- Télécharger depuis [RubyGems](#)
- Décompressez dans un répertoire et `cd`
- Installer avec: `ruby setup.rb` (*vous pourriez avoir besoin des privilèges admin / root*)

```
sudo ruby setup.rb
```

Pour plus de détails et d'autres options, voir:

```
ruby setup.rb --help
```

Installation sous Linux

Assurez-vous d'avoir ruby installé avant d'installer rubygems, puis:

Installer RubyGems en utilisant apt-get sur Ubuntu

```
sudo apt-get install rubygems
```

Installer RubyGems en utilisant yum

```
sudo yum install rubygems
```

Méthode d'installation manuelle

```
wget https://rubygems.org/rubygems/rubygems-2.6.6.tgz
tar xvf rubygems-2.6.6.tgz
cd rubygems-2.6.6
sudo ruby setup.rb
```

Installer une version ruby précompilée et des gemmes ruby (Ubuntu)

```
sudo apt-add-repository ppa:brightbox/ruby-ng
```

Appuyez sur `Enter` pour confirmer

```
sudo apt-get update
```

Ensuite, vous pouvez installer la version ruby de votre choix (le ppa prend en charge `ruby2.0` `ruby2.1` `ruby2.2` `ruby2.3` et les versions héritées `ruby1.8` `ruby1.9.1`) N'oubliez pas d'inclure le paquet respectif `-dev` pour votre version. Sinon, le développement d'extensions natives telles que `Nokogiri` ou `mysql2` échouera.

```
sudo apt-get install ruby2.3 ruby2.3-dev ruby-switch
```

```
sudo ruby-switch set ruby2.3
```

```
ruby -v
```

```
>> ruby 2.3.1p112 (2016-04-26) [x86_64-linux-gnu]
```

Maintenant, vous pouvez installer n'importe quel gem du système pour tous les utilisateurs via

```
sudo gem install gemname .
```

Remarque:

- Cette méthode installe et définit ruby et rubygems pour tous les utilisateurs
- Requiert `sudo` pour installer des gems (`sudo gem install rails`) mais n'exécutez pas `bundle` tant que `root`. S'il a besoin d'installer des gemmes, il demandera le mot de passe `sudo`.

Changer de source de pierres précieuses

```
# list gem sources:
gem sources -l
# remove default gem source:
gem sources -r https://rubygems.org/
# add other gem sources:
gem sources -a https://ruby.taobao.org/
```

Lire Démarrer avec Rubygems en ligne: <https://riptutorial.com/fr/rubygems/topic/1039/demarrer-avec-rubygems>

Chapitre 2: Anémone

Syntaxe

- Utilisez `Anemone::Core.new(url, options)` pour initialiser le robot
- Utilisez le bloc `on_every_page` pour exécuter du code sur chaque page visitée
- Utilisez la méthode `.run` pour démarrer l'analyse. Aucun code à l'avance ne déclenchera réellement des appels GET.

Paramètres

Paramètre	Détails
URL	URL (y compris le protocole à analyser)
options	hash optionnel, voir toutes les options ici

Remarques

- Le robot ne visitera que les liens situés sur le même domaine que l'URL de départ. Il est important de savoir que lorsque vous traitez des sous-domaines de contenu tels que `media.domain.com` ils seront ignorés lors de l' `domain.com`
- Le robot d'exploration est conscient de HTTP / HTTPS et restera par défaut sur le protocole initial et ne visitera pas d'autres liens sur le même domaine.
- L'objet `page` dans le bloc `on_every_page` ci-dessus a une méthode `.doc` qui renvoie le document Nokogiri pour le corps HTML de la page. Cela signifie que vous pouvez utiliser les sélecteurs Nokogiri à l'intérieur du bloc `on_every_page` tel que `page.doc.css('div#id')`
- D'autres informations pour commencer peuvent être trouvées [ici](#)

Exemples

Analyse de site de base

```
pages = []
crawler = Anemone::Core.new(url, options)
crawler.on_every_page do |page|
  results << page.url
end
crawler.run
```

Lire Anémone en ligne: <https://riptutorial.com/fr/rubygems/topic/6445/anemone>

Chapitre 3: Changer la source de gem dans Gemfile

Exemples

Utiliser le repo github pour une gemme spécifique dans Gemfile

Au lieu de

```
gem 'rails'
```

Vous pouvez spécifier une combinaison utilisateur / repo github avec

```
gem 'rails', github: 'rails/rails'
```

Utiliser ref, branch, git pour gem spécifique dans gemfile

```
gem 'any gem', git: 'any repo', branch: 'specific branch of that repo', ref: 'reference no.'
```

ref spécifie un commit individuel. **branch** spécifie la branche git à extraire.

Lire [Changer la source de gem dans Gemfile en ligne](https://riptutorial.com/fr/rubygems/topic/6356/changer-la-source-de-gem-dans-gemfile):

<https://riptutorial.com/fr/rubygems/topic/6356/changer-la-source-de-gem-dans-gemfile>

Chapitre 4: JSON

Introduction

Le `json` gem for Ruby permet l'analyse et la création de JSON.

Syntaxe

- `JSON.parse (json_document_string) =>` renvoie un Hash du document JSON
- `JSON.generate (ruby_hash) =>` renvoie un document JSON sous la forme d'une chaîne

Paramètres

Paramètre	Détails
<code>json_document_string</code>	Un document JSON sous la forme d'une chaîne
<code>ruby_hash</code>	Tout objet Hash

Exemples

Hash to JSON

```
require 'json'
data = {"test" => 123}
puts JSON.generate(data)
```

JSON à Hash

```
require 'json'
document = "{\"test\":123}"
puts JSON.parse(document)
```

JSON alternatif à Hash

```
require 'json'
data = JSON['{"test":23}'] # => {"test"=>23}
```

ou

```
require 'json'
data = JSON[["{"test":23}"]] # => {"test"=>23}
```

Hash alternatif à JSON

```
require 'json'  
document = JSON 'test' => 23 # => "{\"test\":23}"
```

ou

```
require 'json'  
document = JSON['test' => 23] # => "{\"test\":23}"
```

Lire JSON en ligne: <https://riptutorial.com/fr/rubygems/topic/8894/json>

Chapitre 5: Maruku

Introduction

Maruku est un interprète de Markdown pour Ruby. Il permet d'exporter Markdown au format HTML et PDF.

Syntaxe

- `Maruku.new (str) =>` analyse le Markdown dans un objet Maruku
- `Maruku # to_html_document =>` renvoie l'objet Maruku en tant que document HTML complet (sous forme de chaîne)

Paramètres

Paramètre	Usage
<code>str</code>	Chaîne de démarcation à analyser

Exemples

Lecture du Markdown dans Maruku

```
require 'maruku'

str = "
  # Title
  * List 1
  * List 2
  * List 3
"

markdown = Maruku.new(str)
puts markdown.inspect
```

Markdown to HTML

```
require 'maruku'

str = "
  # Title
  * List 1
  * List 2
  * List 3
"

markdown = Maruku.new(str)
puts markdown.to_html_document
```

Lire Maruku en ligne: <https://riptutorial.com/fr/rubygems/topic/8893/maruku>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec Rubygems	Community , halfelf , Killerpixler , Lahiru , mlabarca
2	Anémone	Killerpixler , Punknoodles
3	Changer la source de gem dans Gemfile	Divya Sharma , giniouxe , Killerpixler
4	JSON	Punknoodles
5	Maruku	Punknoodles