



D03 - Ruby on Rails Training

Gems

Stéphane Ballet balletstephane.pro@gmail.com
42 Staff pedago@staff.42.fr

Summary: There is always a Gem for.

Contents

I	Preamble	2
II	instructions	3
III	Specific Rules of the day	5
IV 00	Exercise: I like	6
V	01 Activity: ft_wikipedia	8
VI	Exercise 02: TDD	10
VII	Exercise 03: Rails	11

Chapter I

Preamble

The RS7000 is a major groove generation workstation!

It's sort of like Akai's MPC-series (aim for bumps), sampling-combining and sequencing, with an added purpose internal synth engine. The RS7000 is PARTICULARLY suited for dance, techno, hip hop, R & B, and ambient genres.

The sampler section Consists of a 4MB (expandable to 64MB) sampler (5 kHz to 44.1kHz or 32kHz to 48kHz via digital option board). You can use it to sample external sounds, re-sample the RS7000's sounds itself, or load samples from a variety of common formats ! Auto-beat slicing lets you easily sample any loops or sounds and sync them to your sequence tempo ! All the professional sampling and editing features you'd expect are here, and more !

The tone generator offers 62-voice polyphonic AWM2 synthesis, with over 1,000 synth sounds and 63 drum kit sounds (all via ROM). Here you'll find the resonant filters (6 types), advanced LFO modulation, BPM-synchronized LFO waveforms, and more ! Edits made to the internal sounds, as well as to any samples are all stored within your sequence patterns.

The Sequencer is the real meat of the RS7000, where you make music out of the sounds it's got and that you've put into it ! It offers pattern-based recording with 16 tracks each, and a 200,000 note-per-song capacity. Linear sequencer sequencing, like you would do using a software sequencer like Cubase, is also supported by the RS7000. Pattern-based sequences can be converted to the linear format as well. Realtime, grid and step recording methods are also available. Linking patterns into songs can be done in real time and meticulously tweaked.

Total MIDI control, real-time hands on control, 18 assignable knobs and two pads, a Master effect section (with a multi-band compressor, slicer, isolater, other DJ-style master effects), and more make the RS7000 the most professional quality groove/loop/dance machine out there !

Chapitre II

Consignes

- Only this page serve as a reference: Do not fi rm the hallway noise.
- The subject can change up to an hour before rendering.
- If no contrary information is explicitly present, you must assume the following language versions:
 - Ruby 2.3.0
 - for d09 Rails> 5
 - but for all the other days rails 4.2.6
 - HTML5
 - CSS 3
- We prohibit EXPRESSLY using keywords while, for, redo, break, retry and until in Ruby source code that you will make. Any use of these keywords is considered cheating (and / or unfit), giving you the score of -42.
- The exercises are precisely ordered from simple to more complex. In any case, we do not bear attention or take into account a complex exercise if a simpler exercise is not very successful.
- Attention to the rights of your fi les and your directories.
- You must follow the rendering process for all your exercise: only the present work on your GIT repository will be evaluated in defense.
- Your exercises will be evaluated by your pool mates.
- You should leave in your repertoire no other fi le than those explicitly speci fi ed by the forward drills.
- You have a question ? Ask your neighbor to the right. Otherwise, try your left neighbor.
- Your reference manual called Google / man / Internet /
- Think discuss on the forum of your pool Intra!
- Read the examples carefully. They might require things that

not otherwise specified in the subject ...

- Please, by Thor and Odin! Re fl échissez dammit!


chapter III

Règles spécifiques de la journée

- Tout les fichiers rendus seront dotés d'un shebang approprié ET du flag de warning.
- Aucun code dans le scope global. Faites des fonctions ou des classes !
- Chaque rendu doit etre une Gem (sauf pour l'exercice 03 !)
- Chaque Gem doit utiliser minitest, la license MIT et ne doivent proposer aucun code de conduite.
- Ces Gems sont à vocation pédagogiques : ne vous occupez pas des repos d'upload !
Commentez les lignes dans le .gemspec.
- Chaque Gem doit inclure les tests demandés dans l'exercice et ceux-ci doivent être executés par un bundle exec rake dans le dossier racine de la gem
- Aucun import autorisé, à l'exception de ceux explicitement mentionnés dans la section "Fonctions Autorisées" du cartouche de chaque exercice.

Chapitre IV

Exercice 00 : J'aime

	Exercice : 00 Exercice
00 : J'aime	
Dossier de rendu : <i>ex 00</i> / Fichiers à rendre :	
deephthought	
Fonctions Autorisées : colorize	
Remarques : n/a	

Créez une gem, votre première avec nous .

Vous allez ainsi mettre au monde une portion de code réutilisable par la planète entière Procurant de manière portable le code suivant :

```
require 'colorize' class
Deepthought
  def initialize end

  def respond(question)
    if question == "The Ultimate Question of Life, the Universe and Everything"
      puts "42".green return "42"

    else puts "Mmmm i'm bored".red

      return "Mmmm i'm bored" end end end
```


Excitant , non ? !

Votre Gem doit répondre aux spécificités suivantes :

- Nom de Gem : `deepthought`
- Des tests ? oui bien sûr ! : utilisez `minitest`
- Un code de conduite ? Non
- Licence : MIT
- Version : `'0.0.1'`
- La commande `grep -Hrn 'TODO' --color=always`, exécutée à la racine de votre Gem, ne doit RIEN retourner.
- Vous devez écrire le test qui vérifie que `Deepthought.new` retourne bien l'objet attendu
- Vous devez écrire le test qui vérifie la valeur de retour des deux cas de la méthode `'respond'`

Chapitre V

Exercice 01 : ft_wikipedia

	Exercice : 01 Exercice 01 :
ft_wikipedia	
Dossier de rendu : ex 01/ Fichiers à rendre : ft_wikipedia	
Fonctions Autorisées : nnokogiri/open-uri	
Remarques : n/a	

Un fait intéressant de Wikipedia, est la "route vers la philosophie" Si on clique sur le premier lien en minuscule de la première description de n'importe quelle page, dans 94% des cas, on atteint la page philosophie.

De mon experience, cela n'a jamais dépassé 35 liens...

Pour en être sur, vous devez creer une gem "ft_wikipedia" qui s' utilise et s' affiche comme ceci :

```
Ft_wikipedia.search("Kiss")
First search @ :https://en.wikipedia.org/wiki/Kiss https://en.wikipedia.org/wiki/Love
https://en.wikipedia.org/wiki/Affection https://en.wikipedia.org/wiki/Disposition
https://en.wikipedia.org/wiki/Habit_(psychology) https://en.wikipedia.org/wiki/Behavior

https://en.wikipedia.org/wiki/American_and_British_English_spelling_differences https://en.wikipedia.org/wiki/English_orthography
https://en.wikipedia.org/wiki/Orthography https://en.wikipedia.org/wiki/Convention_(norm)
https://en.wikipedia.org/wiki/Norm_(philosophy) https://en.wikipedia.org/wiki/Sentence_(linguistics) https://en.wikipedia.org/wiki/Word
https://en.wikipedia.org/wiki/Linguistics https://en.wikipedia.org/wiki/Science https://en.wikipedia.org/wiki/Knowledge
https://en.wikipedia.org/wiki/Awareness https://en.wikipedia.org/wiki/Conscious https://en.wikipedia.org/wiki/Quality_(philosophy)
https://en.wikipedia.org/wiki/Philosophy => 19
```

Le programme liste toutes les urls visitées, et retourne le nombre de liens qu'il a du traverser avant la page "https ://en.wikipedia.org/wiki/Philosophy".

Sometimes some research as "matter" going in circles! ! you **must** handle this case with an exceptional lifting "StandardError" and a ffi expensive as follows:

```
Ft_wikipedia.search ( "matter")
First search @ https://en.wikipedia.org/wiki/matter https://en.wikipedia.org/wiki/Atom
https://en.wikipedia.org/wiki/Matter https://en.wikipedia.org/wiki/Atom

Loop detected there is no way to philosophy here => nil
```


Sometimes some research as "Effects_of_blue_lights_technology" are dead ends! ! you **must** manage it with exceptional lifting "StandardError" and AF expensive fi as follows:

```
Ft_wikipedia.search ( "Effects_of_blue_lights_technology")
First search @ https://en.wikipedia.org/wiki/Effects_of_blue_lights_technology Dead end => nil
```

You **must** also write all the tests that prove the good operation of your program with the following search "directory", "problem", "einstein", "kiss", "matter".

chapter VI

Exercise 02: TDD

	Exercise 02 Exercise
02: TDD	
Render Folder: <i>ex 02</i> / Files to make: Tailiste	
Permitted functions: n / A	
Remarks: n / A	

TDD, or Test Driven Development, is a practice highly prized venue beyond the world of web development

We provide you a Gem empty for you implementer. Tests were already written there. I will not dwell on the features of this Gem, since the purpose of the exercise is that you must guess.


To validate the exercise must ABSOLUTELY than :

- La commande "gem build" à la racine du dossier de votre Gem s'exécute sans erreur (outre les warnings d'aide, de manque de description et de homepage manquante)
- la dernière ligne de l'affichage de la commande "rake" soit :

```
"16 runs, 16 assertions, 0 failures, 0 errors, 0 skips"
```

Chapitre VII

Exercice 03 : Rails

	Exercice : 02
Exercice 03 : Rails	
Dossier de rendu : <i>ex 02/</i> Fichiers à rendre	
: HelloWorld	
Fonctions Autorisées : n/a	
Remarques : n/a	

Aaaah ! Hello World ! le célèbre. Vous voici donc maintenant au pied du mur ... Vous devez installer rails sur vos machines, faire une page contenant un titre "Hello World !", et lorsque vous lancez le serveur inclus dans rails, cette page doit être la première affichée.

Google est remplis de bons conseils, cependant j'en ai un autre pour vous : documentez vous avant de lancer des installations

Il est primordial que votre installation se déroule correctement. Etant donné que vous aurez à le refaire plusieurs fois, autant prendre un bon départ ! !

Pour valider l'exercice, il faudra :

- Installer la Gem rails, dépendances nécessaires incluses
- Faire en sorte que le serveur inclus dans rails démarre (la commande est : "rails serveur")
- Faire en sorte que la première page de votre site (accessible via "http ://localhost :3000/") soit une page HTML affichant "Hello World !" dans une balise de titre "<h1>"