

Ruby Gems

Εισαγωγή

Με βάση τα όσα έχουμε ήδη πει πλέον είμαστε σε θέση να τρέχουμε κώδικα, καθώς επίσης να ορίζουμε μεθόδους και κλάσεις. Όλες οι γλώσσες προγραμματισμού έχουν τις δικές τους βιβλιοθήκες για τον ίδιο απλό λόγο, να μην ανακαλύπτουμε ξανά τον τροχό. Φυσικά και η Ruby έρχεται με τις δικές της βιβλιοθήκες και μάλιστα μπορείτε και εσείς να συνεισφέρεται προσφέροντας τις δικές σας βιβλιοθήκες.

Στη Ruby οι βιβλιοθήκες λέγονται gems και ο τρόπος διαχείρισής τους προσωπικά μου θυμίζει τον τρόπο εγκατάστασης προγραμμάτων των unix συστημάτων. Για τα παραδείγματα που ακολουθούν θα χρησιμοποιήσω την τελευταία έκδοση των Ubuntu, αν και όσα αναφέρονται ισχύουν και για Windows/Mac/Bsd. Να σημειώσουμε μόνο πως στην περίπτωση των Windows πρέπει είτε να απενεργοποιήσετε το UAC είτε να ανοίξετε την κονσόλα ως administrator, διαφορετικά το σύστημα θα σας γκρινιάζει ότι δεν έχετε δικαιώματα. Στα Linux θα ενημερώνεστε από την κονσόλα ώστε ότι χρειάζεται να το τρέξετε με sudo.

Η εφαρμογή rubygems

Όπως είπαμε τα gems είναι οι βιβλιοθήκες της Ruby. Εγκαθιστώντας τη Ruby στα Windows δε θα χρειαστείτε τίποτα άλλο. Ρίξτε όμως μία ματιά στον RubyGems Documentation Server που εγκαταστάθηκε μαζί με τη γλώσσα. Στην ουσία είναι documentation server όπου μπορείτε να δείτε το documentation του gem.

Στα linux δώστε την εντολή

```
gem -v
```

Εγώ π.χ. έχω την 1.3.7 και είναι το πρόγραμμα μέσω του οποίου διαχειριζόμαστε τα gems. Εάν δεν έχετε κάποια έκδοση απλά δώστε

```
apt-get install rubygems
```

ή π.χ. στα Fedora μέσω του yum και γενικότερα εγκαταστήστε το rubygems από την κονσόλα δίνοντας την κατάλληλη εντολή ανάλογα με τη διανομή σας.

Οι Mac users να ξέρετε πως εάν δεν αγοράσατε πριν πολλά χρόνια το μηχάνημά σας θα έχετε προεγκατεστημένη την ruby 1.8.7, οπότε ελέγξτε μήπως έχετε ήδη κάποια έκδοση του rubygem, δίνοντας gem -v, όπως είπαμε.

Η εφαρμογή rubygems αναλαμβάνει να επικοινωνεί με το <http://rubygems.org> που είναι το κυριότερο repo από gems καθώς και με όσα άλλα του ορίσετε εσείς.

Διαχείριση gems

Αφού έχετε τελειώσει με το rubygems μπορείτε να δείτε ποια gems έχετε ήδη εγκατεστημένα με την εντολή gem list.

Εγώ π.χ. λαμβάνω το εξής:

```
konstantinos@ubuntu:~$ gem list
```

```
*** LOCAL GEMS ***
```

```
actionmailer (2.3.5)
actionpack (2.3.5)
activerecord (2.3.5)
activeresource (2.3.5)
activesupport (2.3.5)
adzap-ar_mailer (2.1.10)
calendar_date_select (1.16.3)
rack (1.0.1)
rails (2.3.5)
rake (0.9.2.2)
```

Όπως βλέπετε, αρχικά ενημερωνόμαστε ότι πρόκειται για τη λίστα με τα local gems, ενώ σε παρένθεση εμφανίζεται η έκδοση του gem. Να αναφέρω μόνο ότι στα παραπάνω gems το gem rails είναι το γνωστό Ruby on Rails (RoR) που μας επιτρέπει να φτιάχνουμε ιστοσελίδες, και με κάποια database αν θέλουμε, ενώ το rake είναι η εξέλιξη του γνωστού make.

Επιστρέφω ξανά στον αριθμό που μας φανερώνει την έκδοση του gem. Να έχετε υπόψη ότι μπορείτε στο ίδιο μηχάνημα να έχετε πολλαπλές εκδόσεις του ίδιου gem αλλά και της ruby. Θα δούμε παρακάτω πως μπορείτε να γράψετε τον κώδικά σας ώστε να ελέγχετε ποια έκδοση χρησιμοποιείται

Αυτό που θέλω να τονίσω είναι ότι η δυνατότητα πολλαπλών εκδόσεων δίνει ευελιξία στον προγραμματιστή καθώς μπορεί να προσδιορίσει ποια έκδοση επιθυμεί από κάθε gem και πριν ξεκινήσει η εκτέλεση του κώδικα να βεβαιωθεί ότι αυτά υπάρχουν στο μηχάνημα. Ειδικά σε unix συστήματα αν η ruby διαπιστώσει ότι κάτι μας λείπει μας ρωτάει εάν θέλουμε να το εγκαταστήσουμε (ως root user).

Προσθήκη repo

Είναι πολύ πιθανόν να χρειαστείτε κάποιο νέο repo. Για το σκοπό αυτό απλά δώστε την εντολή:

```
gem sources -a <repo_link>
```

Εγκατάσταση/Απεγκατάσταση/Αναζήτηση gem

Λογικά, εάν τώρα προχωρήσατε στην εγκατάσταση της γλώσσας δε θα έχετε και πολλά gems στη διάθεσή σας. Έστω, λοιπόν ότι θέλετε να βάλετε τη rails ή π.χ. το gem bio, που σας επιτρέπει να διαχειριστείτε πολύ εύκολα ακολουθίες DNA.

Για να τα εγκαταστήσετε δώστε την εντολή:

```
gem install rails  
ή  
gem install bio
```

Αυτό θα εγκαταστήσει την τελευταία έκδοση του gem. Εάν θέλετε κάποια συγκεκριμένη έκδοση απλά δηλώστε το στο τέλος της εντολής, π.χ. `gem install rails -v 3.0`. Επίσης μπορείτε να δώσετε `gem install rails -v 3.0 --include dependencies` προκειμένου να εγκαταστήσετε τα οποιαδήποτε dependencies, κάτι που ειδικά για τη rails συστήνεται ανεπιφύλακτα. Ομοίως με `gem uninstall` μπορείτε να απεγκαταστήσετε το gem που δε χρειάζεστε πια ή κάποια συγκεκριμένη έκδοση του gem.

Μετά από κάποιο χρονικό διάστημα πιθανόν να έχετε πολλά gems και διάφορες εκδόσεις. Για αυτό καλό είναι να μάθετε την εντολή

```
gem search <gem_name>.
```

Η εντολή αυτή ψάχνει μόνο στο μηχανήμά σας. Στην περίπτωση που θέλετε να κοιτάξετε στα repos τι gem υπάρχουν χρησιμοποιήστε την εντολή με την παράμετρο `-r`:

```
gem search -r <gem_name>
```

Περισσότερα

Για να μην αραδιάζουμε απλώς τις εντολές καλό είναι να ρίξετε μία ματιά σου <http://docs.rubygems.org/read/chapter/10> όπου παρουσιάζονται όλες οι εντολές καθώς επίσης και να διαβάσετε από την κονσόλα το help δίνοντας:

```
gem help commands.
```

Πως χρησιμοποιώ τα gems στον κώδικά μου;

Ωραία όλα όσα είδαμε αλλά αυτό που πραγματικά μας νοιάζει είναι το πως θα αξιοποιούμε τα gems στον κώδικά μας.

Μια μικρή αναδρομή

Θα σας ταξιδέψω λοιπόν λίγο στον κόσμο της βιολογίας και για την ακρίβεια της γενετικής για να θυμηθούμε αυτά που κάναμε στο λύκειο.

Θυμάστε στο DNA τα 2 ζευγάρια a-t, c-g δηλαδή τις τέσσερις βάσεις αδενίνη, θυμίνη, κυτοσίνη και γουανίνη; Αυτά λοιπόν συνδέονται μόνο με τον παραπάνω τρόπο δηλαδή μόνο a-t και t-a και c-g ή g-c. Επειδή τα ζευγάρια είναι μοναδικά μπορούμε να υπολογίσουμε τη συμπληρωματική ακολουθία DNA, ξέροντας δηλαδή τον ένα κλάδο της έλικας να βρούμε το ζευγάρι του. Μεταξύ άλλων, είναι ένας σημαντικός λόγος για το γεγονός ότι το DNA μπορεί να αυτοδιπλασιαστεί.

Έτσι, πολύ συχνά, για λόγους που δε θεωρώ σκόπιμο να τους αναλύσω, θέλουμε να αντιστρέψουμε μία ακολουθία DNA και να πάρουμε τη συμπληρωματική της, δηλαδή το συμπληρωματικό κλάδο της έλικας. Πρέπει να έχουμε στο νου μας όμως ότι εκτός από το να πάρουμε το συμπληρωματικό των βάσεων πρέπει να αντιστρέψουμε και τη φορά που διαβάζουμε την ακολουθία.

Αντί, λοιπόν να ανακαλύπτουμε κάθε φορά τον τροχό, υπάρχει το αντίστοιχο gem με το όνομα `bio` που παρέχει έτοιμη αυτή τη μέθοδο μαζί φυσικά με άλλες.

Ο κώδικας

Ανοίξτε λοιπόν τον αγαπημένο σας text editor και σηκώστε τα μανίκια. Αρχικά θα πρέπει να δηλώσουμε στον κώδικα ότι θέλουμε να χρησιμοποιήσουμε κάποιο gem. Για αυτό ως πρώτη πρώτη εντολή δίνουμε την:

```
require 'rubygems'
```

Έπειτα θέλουμε να ενημερώσουμε τη ruby ότι θα χρησιμοποιήσουμε το gem `bio`. Ομοίως, λοιπόν, γράφουμε:

```
require 'bio'
```

Εδώ πρέπει να δώσουμε λίγη προσοχή και να τονίσουμε ότι με την παραπάνω δήλωση θα χρησιμοποιήσουμε την πιο πρόσφατη έκδοση του gem `bio`. Εάν το πρόγραμμά μας

απαιτεί κάποια συγκεκριμένη έκδοση τότε το δηλώνουμε αμέσως μετά. Και αυτό μπορούμε να το κάνουμε με διάφορους τρόπους. Π.χ.

```
require 'bio' = 1.3
```

Με την ίδια λογική μπορούμε να χρησιμοποιήσουμε τα γνωστά σύμβολα ανισότητας προκειμένου να ορίσουμε ότι θέλουμε οποιαδήποτε έκδοση νεότερη ή παλιότερη από μία συγκεκριμένη, εξασφαλίζοντας έτσι ότι δε θα υπάρχουν προβλήματα λόγω αλλαγών στη βιβλιοθήκη/gem. Στην περίπτωση που δεν έχουμε εγκατεστημένο το gem ο κώδικας δε θα εκτελεστεί. Παρατηρήστε ότι σύμφωνα με την παραπάνω λίστα των τοπικών gems δεν είναι εγκατεστημένη κάποια έκδοση του gem bio, προκειμένου να δούμε τι θα γίνει σε αυτή την περίπτωση.

Γράφουμε λοιπόν τον κώδικα του αρχείου gem.rb:

```
require 'rubygems'
require 'bio'
seq=Bio::Sequence::NA.new('atgcatgcaaaa')
puts seq
puts seq.complement
```

Τις δύο πρώτες εντολές τις εξηγήσαμε πιο πάνω. Η τρίτη εντολή είναι από τη βιβλιοθήκη Bio. Πηγαίνουμε στην υποκλάση της Bio με το όνομα Sequence και στην υποκλάση αυτής NA και καλούμε τη new για να φτιάξουμε ένα αντικείμενο τύπου NA, δηλαδή DNA sequence. Ως όρισμα δίνουμε την ακολουθία DNA ως string. Η επόμενη εντολή μας εκτυπώνει την ακολουθία που φτιάξαμε, ενώ η τελευταία εντολή καλεί τη μέθοδο complement που μας επιστρέφει τη συμπληρωματική (complement) ακολουθία DNA και την εκτυπώνουμε με την puts. Παρατηρήστε τον τρόπο κλήσης της μεθόδου, η οποία γίνεται κατευθείαν από το αντικείμενο seq.

Υπενθυμίζω ότι για να τρέξουμε το πρόγραμμα, πηγαίνουμε με την κονσόλα στο σχετικό φάκελο και δίνουμε την εντολή ruby gem.rb . Έτσι, αν τα κάναμε όλα καλά λαμβάνουμε το εξής στην κονσόλα:

```
konstantinos@ubuntu:~$ ruby gem.rb
/usr/local/lib/site_ruby/1.8/rubygems/custom_require.rb:31:in `gem_original_require':
no such file to load -- bio (LoadError)
    from /usr/local/lib/site_ruby/1.8/rubygems/custom_require.rb:31:in `require'
    from gem.rb:3
```

Αν παρατηρήσετε λοιπόν, λαμβάνουμε ένα μήνυμα λάθους στη γραμμή 3¹ εκεί δηλαδή

¹ Η πρώτη γραμμή περιέχει σχόλια όπως θα δείτε. Εάν το σβήσετε τότε στο μήνυμα θα σας λείπει στη γραμμή 2. Όπως και να έχει η ruby θα σας υποδείξει την πραγματική γραμμή με το λάθος.

που έχουμε την εντολή require 'bio'. Και μάλιστα το ίδιο το μήνυμα μας ενημερώνει πως ζητάμε ένα gem που δε βρίσκει: no such file to load --bio(LoadError). Για να μπορέσουμε λοιπόν να εκτελέσουμε το συγκεκριμένο κώδικα πρέπει να έχουμε το bio στο μηχάνημά μας. Το εγκαθιστούμε λοιπόν. Προσοχή! Η εγκατάσταση γίνεται με sudo. Διαφορετικά, θα λάβετε ένα μήνυμα σαν και αυτό:

```
konstantinos@ubuntu:~$ gem install bio
ERROR: While executing gem ... (Gem::FilePermissionError)
You don't have write permissions into the /usr/lib/ruby/gems/1.8 directory.
```

Ως sudo λοιπόν έχουμε:

```
konstantinos@ubuntu:~$ sudo gem install bio
Successfully installed bio-1.4.2
1 gem installed
Installing ri documentation for bio-1.4.2...
Installing RDoc documentation for bio-1.4.2...
```

Όπως καταλαβαίνετε το 1.4.2 είναι η έκδοση του gem και για τους πιο παρατηρητικούς είναι η τελευταία, καθώς δεν ορίσαμε κάποια συγκεκριμένη.

Μια ιδιαίτερα σημαντική παρατήρηση είναι η τελευταία γραμμή (Installing RDoc documentation for bio-1.4.2...) καθώς εκτός από το gem εγκαθίσταται και το RDoc του gem. RDoc είναι τα ruby documentations. Πολύ απλά είναι η επεξήγηση των κλάσεων και των μεθόδων προκειμένου να μπορείτε να το διαβάσετε και να ξέρετε τι μπορείτε να κάνετε με το gem που μόλις βάλατε αλλά και πως.

Πλέον μπορούμε να δοκιμάσουμε να ξανατρέξουμε τον κώδικα μας και έτσι έχουμε:

```
konstantinos@ubuntu:~$ ruby gem.rb
atgcatgcaaaa
ttttgcatgcat
```

Ο κώδικας πλέον μπορεί να εκτελεστεί κανονικά. Πράγματι, αρχικά βλέπουμε την ακολουθία DNA που ορίσαμε και στην επόμενη γραμμή τη συμπληρωματική της. Παρατηρείστε ότι η συμπληρωματική ακολουθία είναι και ανεστραμμένη Δείτε για παράδειγμα τις τέσσερις αδενίνες (a) στο τέλος της αρχικής μας ακολουθίας, που γίνονται τέσσερις θυμίνες (t) στην αρχή της συμπληρωματικής ακολουθίας.

Πως Δημιουργώ τα δικά μου gems;

Εάν έχετε φτιάξει κάποια δική σας βιβλιοθήκη στη ruby και θέλετε να τη μοιραστείτε με τον υπόλοιπο κόσμο, δεν έχετε παρά να φτιάξετε το δικό σας gem και να το ανεβάσετε σε κάποιο gem repo όπως π.χ. το <http://rubygems.org> που είδαμε στην αρχή. Η

διαδικασία είναι πολύ απλή:

1. Βεβαιωθείτε ότι ο κώδικάς σας είναι άρτιος, προσωπικά συστήνω να χρησιμοποιείτε την κλάση UnitTest για να φτιάξετε τα tests σας.
2. Βεβαιωθείτε ότι ο κώδικάς σας είναι καλά τεκμηριωμένος. Αυτό φυσικά ισχύει σε όλες τις γλώσσες προγραμματισμού. Στη ruby έχετε υπόψη ότι η πλειονότητα των προγραμματιστών δε θέλουν να του εξηγείτε τι κάνετε σε κάθε γραμμή. Αυτό θεωρούν ότι πρέπει να μπορούν να το καταλάβουν διαβάζοντας τον κώδικά σας. Αρκεί, λοιπόν, να περιγράψετε το τι κάνουν οι κλάσεις και οι μέθοδοί σας. Μπορείτε να βρείτε αρκετές σχετικές συμβουλές στο διαδίκτυο και πιστέψτε με είναι αρκετά απλό. Να σημειώσουμε ότι τα σχόλιά σας χρησιμοποιούνται για να φτιαχτεί με μία εντολή αυτόματα το RDoc του gem σας.
3. Φτιάξτε ένα gem specification file. Αυτά τα αρχεία έχουν την κατάληξη .gemspec και συντάσσονται με συγκεκριμένο τρόπο. Αν κάνετε μία αναζήτηση στο διαδίκτυο θα βρείτε πολλά παραδείγματα και συμβουλές. Στην παρουσίαση που σας προανέφερα θα βρείτε και τη δομή του gemspec για το gem bundler, το οποίο μπορείτε να συμβουλευτείτε. Γενικά έχετε υπόψη ότι θα δώσετε τα στοιχεία σας καθώς και τις απαιτήσεις του gems σας όπως για παράδειγμα dependencies από άλλα gems.
4. Δώστε την εντολή:

```
gem build <gemspec_file_name>
```

Με αυτόν τον τρόπο θα φτιάξετε το gem σας το οποίο είναι πλέον έτοιμο για να το διανέμετε σε κάθε ενδιαφερόμενο.

5. Ανεβάστε το gem σας σε κάποια repository.

Επίλογος

Αυτή τη φορά ασχοληθήκαμε με τα gems της Ruby που συνιστούν τις βιβλιοθήκες της. Μπορείτε και εσείς να φτιάξετε τις δικές σας βιβλιοθήκες και να μοιραστείτε τα “πολύτιμα πετράδια” σας με τον υπόλοιπο κόσμο. Κλείνοντας να αναφέρουμε ότι ο τρόπος διαχείρισης των gems στην πράξη σας διευκολύνει πολύ για τη διανομή του κώδικά σας, καθώς δε χρειάζεται να αγχώνεστε εάν τρέχει ή όχι ο κώδικάς σας σε κάποιο μηχάνημα, εάν έχουν οριστεί οι μεταβλητές περιβάλλοντος ή ποια έκδοση κάθε βιβλιοθήκης πρέπει να τρέχετε. Για όσους ενδιαφέρονται μπορείτε να δείτε στο <https://show.zoho.com/public/myrrdin/Ruby%20gems%20are%20pragmatic%20programmer's%20best%20friends> την παρουσίαση που έγινε (γραμμένη για Windows) για αυτό το θέμα στην ομάδα thessaliniki.rb, The Thessaloniki Ruby Users Group (<https://plus.google.com/u/0/117820512877082997368/posts>, στο google+) στην οποία θα χαρούμε πολύ να σας δούμε, καθώς οι συναντήσεις μας είναι ανοιχτές.

Τέλος, για τους οπαδούς του Unix, να αναφέρουμε ότι η Ruby σας δίνει τη δυνατότητα να ορίσετε συγκεκριμένα περιβάλλοντα και μπορείτε να τρέξετε τεστ προκειμένου να βεβαιωθείτε εάν τρέχει ή όχι και γιατί ο κώδικάς σας σε αυτά τα περιβάλλοντα,

γλυτώνοντας σας από πολύ κόπο και άγχος. Ως περιβάλλον εννοούμε την έκδοση της Ruby αλλά και τα gem που την συνοδεύουν. Ρίξτε μία ματιά στο <https://rvm.beginrescueend.com/>. Δυστυχώς σε windows συστήματα δεν υπάρχει αυτή η δυνατότητα. Ας ελπίσουμε ότι σύντομα αυτό θα επιλυθεί.