

# Useful Stuff

Farrel Lifson

[farrel.lifson@aimred.com](mailto:farrel.lifson@aimred.com)  
<http://www.aimred.com>



# Agenda

- Ruby Facets
- Sequel
- HAML/SASS

# Facets

- “Ruby Facets is the single largest collection of core extension methods and standard library additions available for the Ruby programming language”
- Facets/Core and Facets/More API
- Atomic requires on the method level
- <http://facets.rubyforge.org>

# Usage

- Array

```
1 [0, 1, 2].pad(6, 0)
2 # => [0, 1, 2, 0, 0, 0]
3 [1, 2, 3].rotate(2)
4 # => [2, 3, 1]
```

- Comparable

```
1 4.at_least(5) # => 5
2 4.at_most(5) # => 4
3 4.clip(2, 7) # => 4
4 9.clip(2, 7) # => 7
```

# Usage

- Enumerable

```
1 (1..5).partition_by { |n| n % 3 }
2 # => { 0 => [3], 1 => [1, 4], 2 => [2, 5] }
3 ["foo", "bar", "a"].commonality{|str|
4     str.length }
5 # => { 3 => ["foo", "bar"] }
6 [1, 1, 2, 3, 3, 4, 5, 5].occur(2)
7 #=> [1, 3, 5]
```

# Usage

- Float

```
1 4.555.round_at(0) #=> 5.0
```

```
2 4.555.round_to(0.1) #=> 4.6
```

- Integer

```
1 5.even? # => false
```

```
2 6.multiple?(2) # => true
```

```
3 3.of {|i| i+1} # => [1,2,3]
```

# Facets/More

Annotations AOP Advice AnsiCode Association AutoArray  
AutoReload BasicObject Behavior BinReadable Buildable Bytes  
CacheDelegator Capsule ClassExtension ClassMethods Cloneable  
CompareOn ConsoleUtils Coroutine Crypt Curry Daemonize Date  
Dependency Dictionary Downloader Elementor EnumerablePass  
EventHook Expirable File FileUtils Functor GetoptLong Heap Infinity  
Inheritor Instantiable Instantise Interface Interval IORedirect Lazy  
Let LinkedList Logger LRUcache MapSend Memoize Minitar  
Multipliers Multiton NackClass Namespace NullClass OpenObjects  
Overload Paramix Pathname Pool Prepend ProgressBar Prototye  
Random RBconfig RBsystem Recorder Recursive Reference  
Reflection Remain RWDelegator Semaphore Shellwords SMTP\_TLS  
Snapshot Stylize Sync Timer Times TracePoint Tuple Typecast  
Uninheritable UploadUtils Version YAML ZipUtils



# Sequel

- “Sequel is a lightweight ORM library for Ruby.”
- Active Development
  - 0.0.1 – March 2007
  - 0.3.4 – November 2007
- Lead Developer: Sharon Rosner
- <http://sequel.rubyforge.org>



# Database Support

- Database adapters for
  - ADO
  - DBI
  - MySQL
  - ODBC
  - Oracle
  - PostgreSQL
  - SQLite 3

# Usage

```
1 require 'sequel/sqlite'
2 DB = Sequel.sqlite('database.db')
3 DB.create_table :items do
4   column :name, :text
5   column :price, :float
6 end
7 items = DB[:items]
8 items << { :name => 'abc',
           :price => rand * 100 }
```

# Usage

```
1 dataset = DB[:managers].where(  
    :salary => 5000..10000).order(:name, :department)
```

```
1 dataset = DB.query do  
2   from :managers  
3   where :salary => 5_000..10_000  
4   order_by :name, :department  
5 end
```

# Usage

```
1 DB[:items].filter { :price < 100 }.sql
2 #=> "SELECT * FROM items WHERE (price < 100)"
3 DB[:items].filter { :name =~ 'AL%' }.sql
4 #=> "SELECT * FROM items WHERE
      (name LIKE 'AL%')"
5 DB[:items].filter { :state.nil? }.sql
6 #=> "SELECT * FROM items WHERE (state IS NULL)"
7 DB[:items].filter { (:x + :y) > :z }.sql
8 #=> "SELECT * FROM items WHERE ((x + y) > z)"
9 DB[:items].filter { :price < :AVG[:price] +
100 }.sql
10 #=> "SELECT * FROM items WHERE
      (price < (AVG(price) + 100))"
```

# Usage

```
1 t = Time.now - 86400 # 1 day ago
2 DB[:items].filter { :stamp <= t }.sql
3 #=> "SELECT * FROM items WHERE
      (stamp <= '2007-09-03T13:32:29+03:00')"
```

```
4 RUBY = 'ruby'
5 DB[:items].filter { :category == RUBY }.sql
6 #=> "SELECT * FROM items WHERE (category = 'ruby')"
```

# Usage

```
1 DB[:items].filter { :id == [1, 2, 3] }.sql
2 #=> "SELECT * FROM items WHERE
      (id IN (1, 2, 3))"
3 DB[:items].filter { :id == (4..6) }.sql
4 #=> "SELECT * FROM items WHERE
      (id >= 4 AND id <= 6)"
5 DB[:orders].filter {
      :price >= DB[:items].select(:price) }.sql
6 #=> "SELECT * FROM orders WHERE
      (price >= (SELECT price FROM items))"
```

# Usage

```
1 DB[:items].filter do
2   :x == 1
3   :y > 2
4   :z < 3
5 end.sql
6 #=> "SELECT * FROM items WHERE
      ((x = 1) AND (y > 2) AND (z < 3))"
```

# Models

```
1 class Post < Sequel::Model(:posts)
2   set_primary_key [:category, :title]
3   one_to_one :author, :from => Author
4
5   subset(:old_posts) { :stamp < 30.days.ago }
6   subset :invisible, :visible => false
7 end
8 post = Post['ruby', 'hello world']
9 post.author = Author['Farrel', 'Lifson']
10 Post.filter.invisible
```



# HAML/SASS

- “Haml is based on one primary principal. Markup should be beautiful.”
- Rails or standalone
- <http://haml.hamptoncatlin.com/>

# HTML

```
<div id='content'>  
  <div class='left column'>  
    <h2>Welcome to our site!</h2>  
    <p>  
      <%= print_information %>  
    </p>  
  </div>  
  <div class="right column">  
    <%= render :partial => "sidebar" %>  
  </div>  
</div>
```



# HAML

```
#content
```

```
  .left.column
```

```
    %h2 Welcome to our site!
```

```
    %p= print_information
```

```
  .right.column= render :partial => "sidebar"
```

# SASS

```
#main p
```

```
  :color #00ff00
```

```
  :width 97%
```

```
.redbox
```

```
  :background-color #ff0000
```

```
  :color #000000
```

# CSS

```
#main p {  
  color: #00ff00;  
  width: 97%; }  
  
#main p .redbox {  
  background-color: #ff0000;  
  color: #000000; }
```