

# Ruby

Farrel Lifson  
<http://www.aimred.com>

Ruby Is Not PHP

# History

- Initial Release - 1995
- Rails - 2004
- Ruby 1.9 - Dec 2007



# Who's Using It?

- NASA
- Amazon
- Toyota
- Sun Microsystems (JRuby)
- Microsoft (IronRuby)

# Design Principles

*“Often people, especially computer engineers, focus on the machines. They think, “By doing this, the machine will run faster. By doing this, the machine will run more effectively. By doing this, the machine will do something something something.” They are focusing on machines. But in fact we need to focus on humans, on how humans care about doing programming or operating the application of the machines. We are the masters. They are the slaves.”*

Yukuhiro “Matz” Matsumoto

# Design Principles

Principle Of Least Surprise

# Design Principles

Do Not Repeat Yourself

# Objects Everywhere

```
1 5.times do |number| # Integers
2   puts(number)
3 end
4 5.2.floor           # Floats
5 "yrub".reverse.upcase # Strings
6 [1,3,5,9,7].sort   # Arrays
7 1..10.to_a         # Ranges
8 /?ub?/.match("quby") # Regexp
9 :method_name.to_s  # Symbols
```

# Objects Everywhere

```
1 class Person
2   attr_accessor :name, :date_of_birth
3   def initialize(name, date_of_birth)
4     @name, @date_of_birth = name, date_of_birth
5   end
6 end
7 p = Person.new("Farrel Lifson",
8               Date.civil(1979,1,5))
9 p.name # => "Farrel Lifson"
10 p.date_of_birth.to_s # => "1979-01-05"
```

# Objects Everywhere

Objects are never closed

```
1 class <<p
2   def describe
3     "My name is #{@name} and
4     I am #{(Date.today -
5             @date_of_birth).div(365)}
6     years old"
7   end
8 end
9 p.describe # => "My name is Farrel Lifson
10          #    and I am 28 years old"
```

# Objects Everywhere

## Classes are objects

```
1 integer_class = 5.class
2 integer_class.to_s # => Fixnum
3 integer_class == Fixnum # => true
4 Fixnum.public_instance_methods.sort[0..4]
5 # => ["%", "&", "*", "**", "+"]
6 Fixnum.respond_to?(:zero?) # => true
```

# Objects Everywhere

Objects never closed

+

Classes are objects

=

Classes are never closed

# Objects Everywhere

```
1 class Integer
2   def days
3     self * 86400
4   end
5   def from_now
6     Time.now + self
7   end
8 end
9 5.days.from_now
10 # => Sat Sep 22 01:19:10 +0200 2007
```

# Objects Everywhere

Code blocks are objects

```
1 5.times do |number|
2   puts(number)
3 end

4 times_3 = lambda do |x|
5   x * 3
6 end

7 times_3.call(5) # => 5
```

# Objects Everywhere

```
1 def render_text(text, &renderer)
2   renderer.call(text)
3 end
4 stars = lambda { |text| "***#{text}***" }
5 render_text("stars", &stars)
6 # => "***stars***"
7 render_text("bars") do |text|
8   "---#{text}---"
9 end
10 # => "---bars---
```

# Duck Typing

```
1 def times_3(object)
2   object * 3
3 end
4 times_3(5)
5 # => 15
6 times_3("5")
7 # => "555"
```

“If it walks like a duck...”

# Modules

Acts as a namespace

```
1 result = Math::sin(Math::PI)
2 module Pet
3   class Dog
4   end
5 end
6 fido = Pet::Dog.new
```

# Modules

## Mixins

```
1 class Dog
2   attr_accessor :name
3   include Comparable
4   def <=>(other)
5     self.name <=> other.name
6   end
7 end
8 fido > rover # => false
9 fido == fido # => true
```

# Modules

- Enumerable mixin module
- Define 'each' method and get  
all?, any?, collect, detect, each\_cons, each\_slice,  
each\_with\_index, entries, enum\_cons, enum\_slice,  
enum\_with\_index, find, find\_all, grep, include?, inject, inject,  
map, max, member?, min, partition, reject, select, sort,  
sort\_by, to\_a, to\_set, zip  
for free

```
1 [1,5,6,2].sort # => [1,2,5,6]
```

```
2 [1,2,3,4].map{|e| e**2} # => [1,4,9,16]
```

```
3 [1,2,"3",4].all?{|e| e.is_a?(Numeric)}  
# => false
```

# Metaprogramming

```
1 class Dog
2   attr_accessor :name
3 end
4 class Dog
5   def name
6     @name
7   end
8   def name=(value)
9     @name = value
10  end
11 end
```

# Metaprogramming

```
1 class Dog
2   def method_missing(name, *args)
3     STDOUT.puts("Called #{name} with
4                 #{args.size} parameters")
5   end
6 end
7 fido.meow
8 # => Called meow with 0 parameters
9 Customers.find_by_email("info@aimred.com")
```

# Metaprogramming

```
1 class Question < ActiveRecord::Base
2   has_many :answers
3 end
4 Question.find(:all).each do |question|
5   puts "#{question.text}"
6   puts "#{question.answers.first.text}"
7 end
```

# Other Chainsaws

## Threads

```
1 a = Thread.new do
2   50.times{ print("a") }
3 end
4 b = Thread.new do
5   50.times{ print("b") }
6 end
7 # => aaaaaabbbbbbaaaaaabbbb.....
```

# Other Chainsaws

## DRb – Distributed Ruby

```
1 url = "druby://localhost:3000"  
2 # server  
3 DRb::start_service(url, Person.new)  
4 # client  
5 DRb::start_service  
6 remote = DRb::DRbObject.new(nil, url)  
7 remote.name
```

# Other Chainsaws

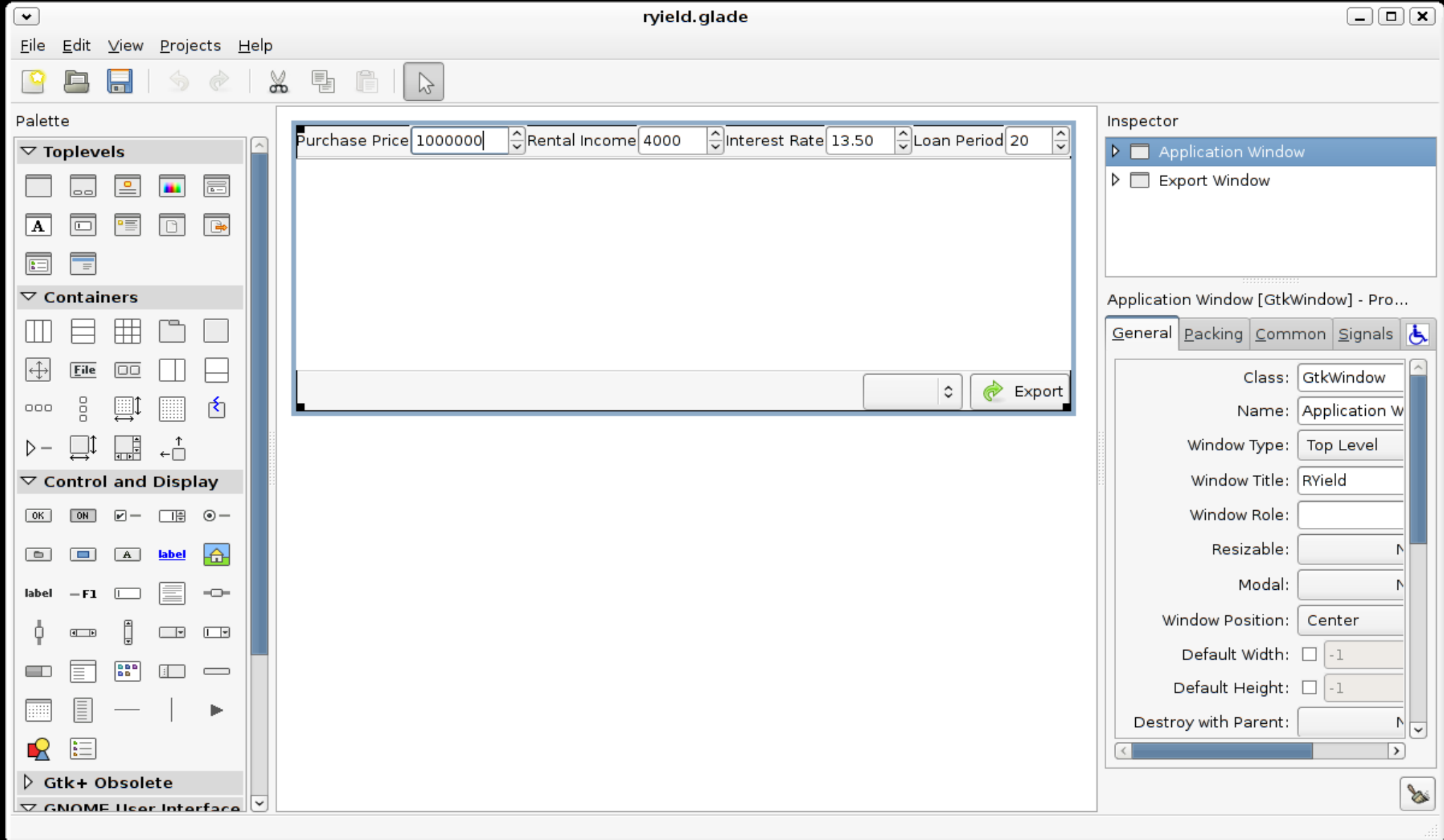
## ParseTree

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

# Not Just For Console

- GTK/GNOME
- QT/KDE
- Carbon
- FOX
- TK

# Not Just For Consoles





# Not Just For Consoles

**RYield**

Purchase Price: 1000000 | Rental Income: 4000 | Interest Rate: 13.50 | Loan Period: 20

Down Payment	Monthly Payment	Monthly Cash Flow	Annual Cash Flow	Return On Investment
R0	R12073.75	R-8073.75	R-96884.96	0.00%
R100000	R10866.37	R-6866.37	R-82396.47	-82.40%
R200000	R9659.00	R-5659.00	R-67907.97	-33.95%
R300000	R8451.62	R-4451.62	R-53419.47	-17.81%
R400000	R7244.25	R-3244.25	R-38930.98	-9.73%
R500000	R6036.87	R-2036.87	R-24442.48	-4.89%
R600000	R4829.50	R-829.50	R-9953.98	-1.66%
R700000	R3622.12	R377.88	R4534.51	0.65%
R800000	R2414.75	R1585.25	R19023.01	2.38%
R900000	R1207.37	R2792.63	R33511.50	3.72%
R1000000	R0.00	R4000.00	R48000.00	4.80%

Text | Export

# Links

<http://www.ruby-lang.org>

<http://www.aimred.com>

<http://www.buzzfuse.com>