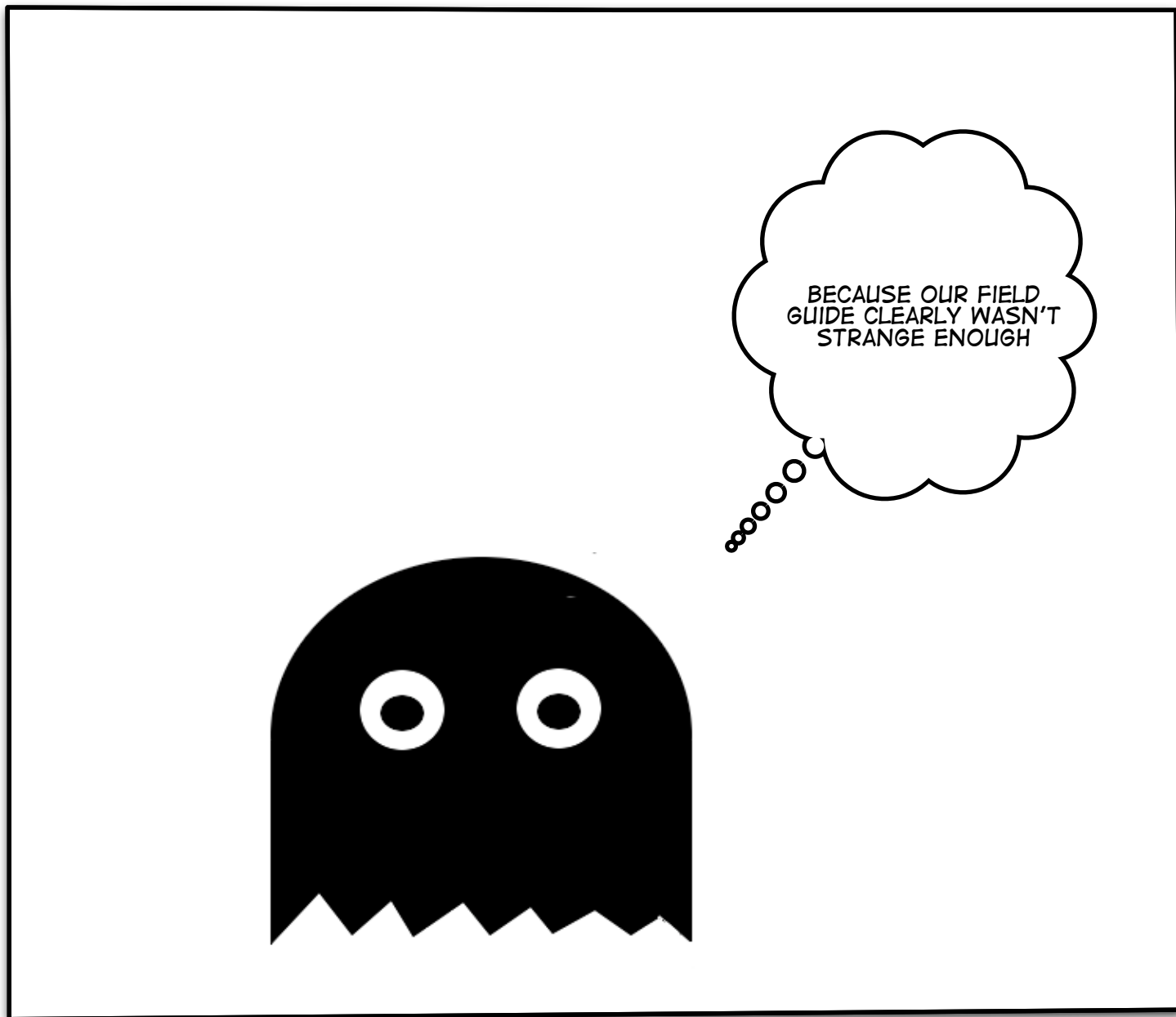
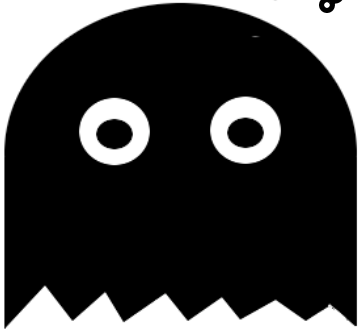


# Ruport in 10 minutes



RUBY REPORTS IS A SIMPLE, EXTENSIBLE REPORTING SYSTEM BUILT FOR RUBYISTS.

BLAH,  
BLAH  
I'M ALREADY  
BORED



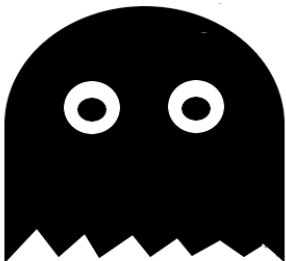
WE PROVIDE A SOLID DATA MODEL CAN BE EASILY HOOKED UP TO A NUMBER OF SOURCES, INCLUDING CSV FILES, ACTIVERECORD MODELS, AND RAW SQL CONNECTIONS THROUGH DBI.

OH, CSV?  
YEAH... I NEED TO  
DEAL WITH THAT.  
LET'S SEE WHAT  
YOU GOT, RUPORT.



```
INPUT = <<-EOS
KIND,NAME,GENDER,RATE
H,JOHN ADAMS,M,21.20
V,ABAGAIL ADAMS,F,0
H,GEORGE W BUSH,M,80.00
H,BORIS YELTSIN,M,50.00
H,JOHANNA KENNEDY,F,45.00
S,ALBERTA EINSTEIN,F,65200
V,MUHAMMAD ALI,M,0
S,LEONARDO DA VINCI,M,3711
S,BENJAMIN FRANKLIN,M,52113
H,TAMMY JEFFERSON,F,65.20
V,SHERLOCK HOLMES,M,0
V,HERCULE POIROT,M,0
EOS
```

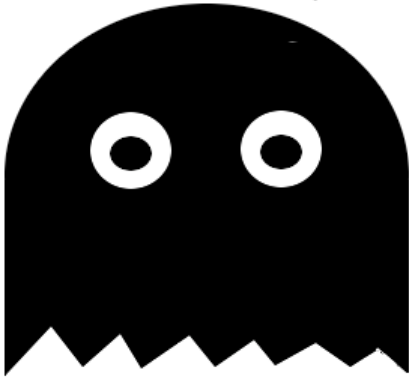
SO THE  
REPORT I WANT  
IS DEAD  
SIMPLE



I JUST WANT  
TO KNOW THE  
WEEKLY RATE  
OF PAY FOR THIS LIST  
OF EMPLOYEES

THE CALCULATIONS ARE  
TRIVIAL, HAVE A LOOK.

```
def adjusted_rate
  case(kind)
  when "H"
    rate.to_f * 7
  when "V"
    0
  when "S"
    rate.to_f / 52
  end
end
```



HMM...  
SO IT LOOKS LIKE RUPT'S  
DATA::TABLE AND DATA::RECORD  
CLASSES MIGHT DO THE TRICK.

IT LOOKS LIKE I CAN LOAD A  
CSV USING A CUSTOM RECORD,  
AND JUST CALCULATE  
THE WEEKLY RATE ON THE FLY

```
require "rubygems"
require "rreport"

class Employee < Rreport::Data::Record
  def adjusted_rate
    case(kind)
    when "H"
      rate.to_f * 7
    when "V"
      0
    when "S"
      rate.to_f / 52
    end
  end
end

table = Table(:string => INPUT, :records => true,
              :record_class => Employee) do |s,r|
  r.rate = r.adjusted_rate
  s << r
end
```

WE ALSO PROVIDE AN INSANELY EASY TO EXTEND FORMATTING SYSTEM WHICH PROVIDES A CONSISTENT INTERFACE AS WELL AS A SET OF HELPER FUNCTIONS FOR A NUMBER OF OUTPUT TYPES (HTML,PDF,CSV, AND PLAIN TEXT).

OKAY, THESE DEVELOPERS SERIOUSLY UNDERESTIMATE MY LAZINESS...

I'M JUST GOING TO TRY TO PRINT THE TABLE. IF THAT DOESN'T WORK I'M *SO* OUT OF HERE...



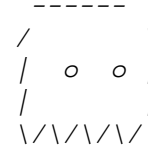
# PROGRESS!

```

++++ +++++ +++++ +++++ +++++
+ + + + + + +
++++ +++++ + + + +
+ + + + + + + +
+ + +++++ +++++ +++++ +++++

= ===== = = =====
= = = = = = =
= = = = = = =
= = = = = = =
===== ===== = =====

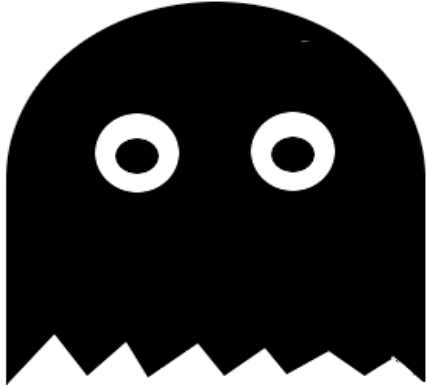
```



THAT'S ACTUALLY PRETTY COOL

puts table

<i>kind</i>	<i>name</i>	<i>gender</i>	<i>rate</i>
H	John Adams	M	148.4
V	Abigail Adams	F	0
H	George W Bush	M	560.0
H	Boris Yeltsin	M	350.0
H	Johanna Kennedy	F	315.0
S	Alberta Einstein	F	1253.84615384615
V	Muhammad Ali	M	0
S	Leonardo Da Vinci	M	713.673076923077
S	Benjamin Franklin	M	1002.17307692308
H	Tammy Jefferson	F	456.4
V	Sherlock Holmes	M	0
V	Hercule Poirot	M	0



THAT'S ALREADY PRETTY CLOSE TO WHAT I NEED.

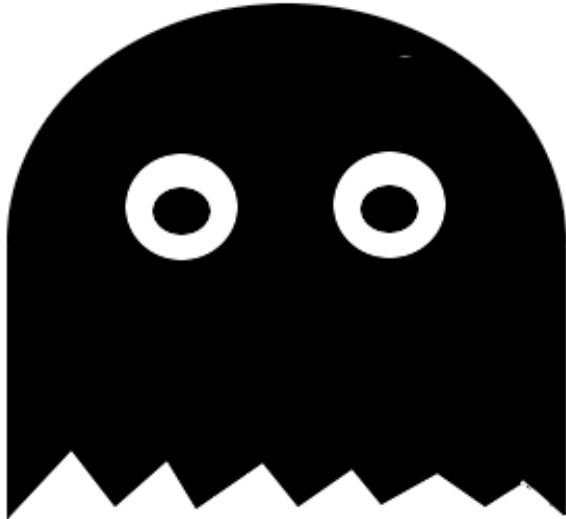
NOW I JUST WANT SOME BASIC SORTING.

I ALSO WANT TO CLEAN UP THE REPORT DATA A BIT

```
table.remove_column("kind")
table.sort_rows_by!(["gender", "rate"])
table.replace_column("rate") { |r| "%0.2f" % r.rate }
```

```
+-----+
|          name          | gender | rate |
+-----+-----+
| Abigail Adams         | F      | 0.00 |
| Johanna Kennedy       | F      | 315.00 |
| Tammy Jefferson       | F      | 456.40 |
| Alberta Einstein      | F      | 1253.85 |
| Muhammad Ali          | M      | 0.00 |
| Sherlock Holmes       | M      | 0.00 |
| Hercule Poirot        | M      | 0.00 |
| John Adams            | M      | 148.40 |
| Boris Yeltsin         | M      | 350.00 |
| George W Bush         | M      | 560.00 |
| Leonardo Da Vinci     | M      | 713.67 |
| Benjamin Franklin    | M      | 1002.17 |
+-----+-----+
```

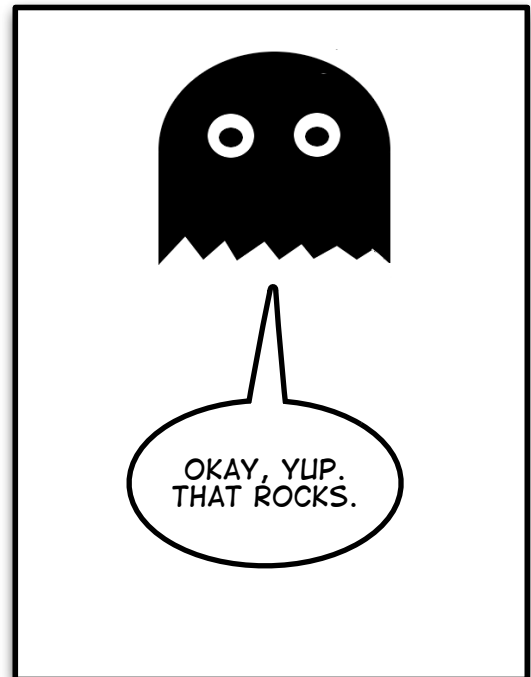
# SCHWEEET



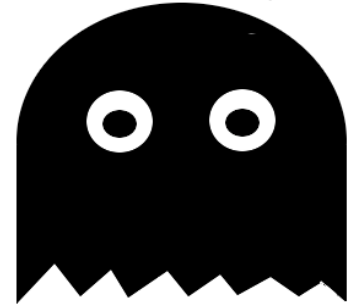
I HAVE TO ADMIT,  
THIS IS PRETTY EASY  
SO FAR.  
  
LET'S TRY THE  
PDF OUTPUT

```
puts table.to_pdf
```

name	gender	rate
Abigail Adams	F	0.00
Johanna Kennedy	F	315.00
Tammy Jefferson	F	456.40
Alberta Einstein	F	1253.85
Muhammad Ali	M	0.00
Sherlock Holmes	M	0.00
Hercule Poirot	M	0.00
John Adams	M	148.40
Boris Yeltsin	M	350.00
George W Bush	M	560.00
Leonardo Da Vinci	M	713.67
Benjamin Franklin	M	1002.17



```
ruby myscript.rb > foo.pdf
```



SOMETHING ABOUT  
THIS FEELS ITCHY THOUGH...

DO I REALLY NEED THAT  
SPECIAL RECORD CLASS?

ALSO...

RUPOORT IS SUPPOSED  
TO WORK WELL WITH FASTERCSV

MAYBE I CAN USE THOSE  
COOL CONVERTOR THINGIES...

```
require "rubygems"  
require "ruport"
```

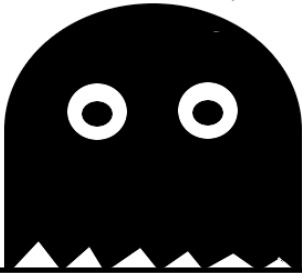
```
RATES = { "H" => lambda { |rate| rate * 7 },  
          "V" => lambda { 0 },  
          "S" => lambda { |rate| rate / 52 } }
```

```
table = Table(:string => INPUT, :csv_options => { :converters => :float } )  
table.replace_column("rate") { |r| ("%0.2f" % RATES[r.kind][r.rate]).to_f }  
table.sort_rows_by!(["gender", "rate"])  
table.remove_column("kind")
```

WHOA! IT'S HARD TO BELIEVE THAT THIS  
ACTUALLY DOES THE SAME THING AS MY FIRST PROGRAM!.

LOADING IN A CSV, GENERATING A CALCULATED FIELD,  
DROPPING A COLUMN, AND DOING A MULTILEVEL SORT IN  
7 LINES OF CODE. CONSIDERING I GET TEXT, CSV,  
HTML, AND PDF OUTPUT FOR FREE, I'LL TAKE THAT!

THE LAST THING TO DO IS ADD A HEADER  
TO THE REPORT WITH A TOTAL.



OKAY, THIS IS  
REALLY LAME.

RUPT'S TABLE FORMATTERS  
DON'T SEEM TO HAVE A CLUE  
WHAT A REPORT HEADER IS.

I REALLY HOPE  
THEIR FORMATTING SYSTEM IS AS  
'INSANELY EASY TO EXTEND'  
AS THEY CLAIM.

I REALLY ONLY NEED  
A PDF, SO LET ME  
TRY FOOLING WITH  
THAT.

```
class EmployeeRenderer < Ruport::Renderer
  stage :employee_report

  module Helpers
    def week_total
      "Employee rate per week. "+
      "Total paid: #{data.sum(:rate)}"
    end
  end
end

class EmployeeAsPDF < Ruport::Formatter::PDF

  renders :pdf, :for => EmployeeRenderer

  def build_employee_report
    pad(20) do
      add_text week_total, :font_size => 16,
              :justification => :center
    end
    draw_table(data)
    render_pdf
  end
end

puts EmployeeRenderer.render_pdf(:data => table)
```

Employee rate per week. Total paid: \$4799.49

name	gender	rate
Abigail Adams	F	0.00
Johanna Kennedy	F	315.00
Tammy Jefferson	F	456.40
Alberta Einstein	F	1253.85
Muhammad Ali	M	0.00
Sherlock Holmes	M	0.00
Hercule Poirot	M	0.00
John Adams	M	148.40
Boris Yeltsin	M	350.00
George W Bush	M	560.00
Leonardo Da Vinci	M	713.67
Benjamin Franklin	M	1002.17



THAT WAS ACTUALLY A  
LITTLE MORE WORK THAN  
WHAT I EXPECTED.

I THINK I SEE WHAT IT  
GAINED ME THOUGH..  
LET'S TRY IMPLEMENTING  
THE TEXT AND HTML  
OUTPUTS

```
class EmployeeAsTextual < Ruport::Formatter::HTML
  renders [:text,:html], :for => EmployeeRenderer

  def build_employee_report
    html { output << textile("h1. #{week_total}") }
    text { output << week_total << "\n\n" }
    render_table(data)
  end
end

puts EmployeeRenderer.render_html(:data => table)
puts EmployeeRenderer.render_text(:data => text)
```

YUP, AS I SUSPECTED.

ONCE YOU HAVE RENDERERS  
HOOKED UP, IT'S REALLY EASY  
TO HOOK IN NEW FORMATS.

```
<h2>Employee rate per week. Total paid: $4799.49</h2>
```

```
<table>
  <tr>
    <th>name</th>
    <th>gender</th>
    <th>rate</th>
  </tr>
  <tr>
    <td>Abigail Adams</td>
    <td>F</td>
    <td>0.00</td>
  </tr>
  <tr>
    <td>Johanna Kennedy</td>
    <td>F</td>
    <td>315.00</td>
  </tr>
```

Employee rate per week. Total paid: \$4799.49

name	gender	rate
Abigail Adams	F	0.00
Johanna Kennedy	F	315.00
Tammy Jefferson	F	456.40
Alberta Einstein	F	1253.85
Muhammad Ali	M	0.00
Sherlock Holmes	M	0.00
Hercule Poirot	M	0.00
John Adams	M	148.40
Boris Yeltsin	M	350.00
George W Bush	M	560.00
Leonardo Da Vinci	M	713.67
Benjamin Franklin	M	1002.17

OKAY, SO SOME OF THAT STUFF WAS AWESOME.

SOME PARTS MADE ME FEEL LIKE I WAS ATTACKING AN ANT-HILL WITH A BAZOOKA.

I LEARNED AFTERWARDS THAT EVEN THOUGH TABLES CAN'T TITLES ATTACHED TO THEM, GROUPS CAN. SO I REALLY JUST NEEDED SOMETHING LIKE:

```
table.to_group("My Table Header").to_pdf
```

STILL, THAT DOES LOOK A LITTLE HACKY, AND I SEE THAT A LOT OF TIMES, I'M GOING TO WANT TO ROLL MY OWN FORMATTERS WHEN USING RUPORT, SO I'M GLAD I GOT A CHANCE TO PLAY WITH THEM.

FOR NOW, I THINK I'LL GO BUG THE DEVELOPERS AT [LIST.RUBYREPORTS.ORG](http://LIST.RUBYREPORTS.ORG) OR STOP BY `#ruport` ON FREENODE AND SEE IF I CAN GET SOME MORE DETAILS ABOUT SOME OF THE OTHER FEATURES OF RUPORT.

IT LOOKS LIKE THERE ARE A WHOLE BUNCH OF HELPERS IN THESE FORMATTER BASE CLASSES, AND THE ACTIVERECORD SUPPORT LOOKS PRETTY INTERESTING TOO.

MAYBE SOME DAY, THEIR DOCUMENTATION WON'T INVOLVE STRANGE THEMES SUCH AS PACMAN GHOSTS AND PIRATES, BUT FOR NOW, I THINK I'VE SEEN ENOUGH TO GET STARTED.

IT LOOKS LIKE *RUPORT-LITIL* HAS SOME INTERESTING STUFF THAT I'LL HAVE TO TAKE A LOOK AT WHEN I KNOW A BIT MORE ABOUT WHAT MAKES RUPORT TICK. SINCE I NOW KNOW HOW MUCH TO PAY THESE FOLKS, MAYBE THE INVOICE SUPPORT IN THERE WOULD COME IN HANDY.

HAS IT BEEN 10 MINUTES YET? REPORTING SUCKS REAL BAD, WHETHER OR NOT I GET TO WORK IN RUBY, SO I THINK I'LL CALL IT A DAY.

