An anatomical drawing of a human torso, showing the muscles, internal organs, and skeletal structure. The drawing is detailed and realistic, with a focus on the musculature and the internal organs. The text is overlaid on the drawing in a large, bold, orange font.

Confessions
of a
Ruby Sadist

Philosophy

People will
press charges
if you hurt
them

Code

won't

So *Top* your
code

Make the
code your
bitch

Beat it into
submission

Ruby Sadism:

noun

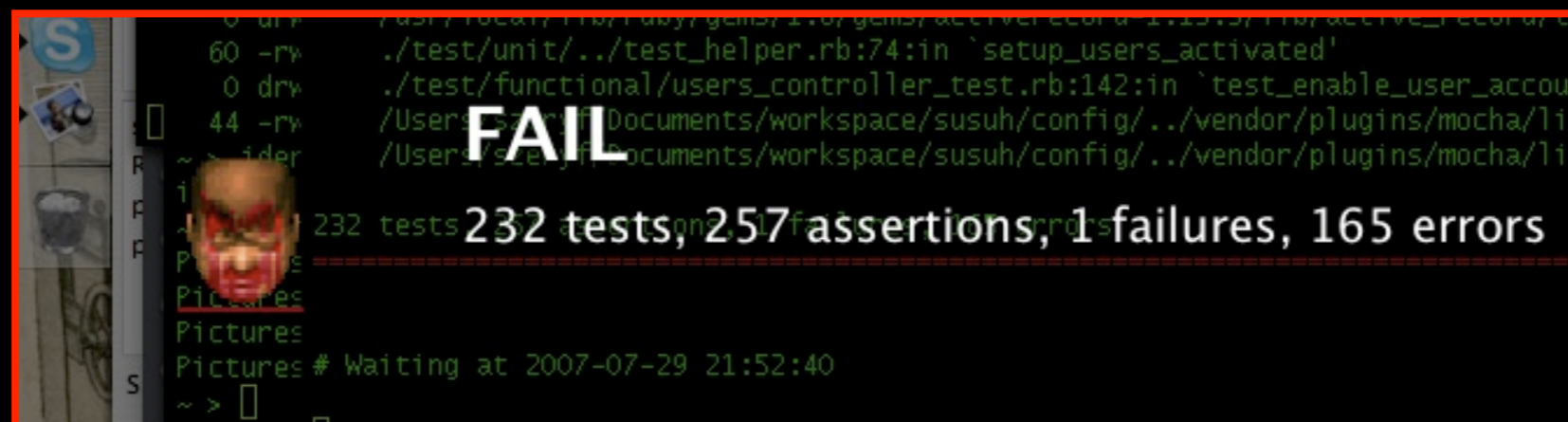
*Deriving pleasure from
inflicting pain on software.*

Esp. bad software.

Autotest

the Howitzer of Testing

- You probably didn't realize that using autotest is a form of ruby sadism and asceticism, did you?
- Poor doom guy...



```
60 -rw ./test/unit/./test_helper.rb:74:in `setup_users_activated'  
0 drw ./test/functional/users_controller_test.rb:142:in `test_enable_user_accou  
44 -rw /Users/szeryf/Documents/workspace/susuh/config/./vendor/plugins/mocha/li  
~/ider /Users/szeryf/Documents/workspace/susuh/config/./vendor/plugins/mocha/li  
i  
F  
F  
F  
P  
P  
P  
Pictures  
Pictures  
Pictures # Waiting at 2007-07-29 21:52:40  
~ > []
```

FAIL

232 tests, 232 assertions, 1 failures, 165 errors

```
Seattle.rb Rulez — emacs — 1

# ensure we add test_blah.rb when test_blah.rb itself updates
util_find_files_to_test(@test, @test => [])
end

def test_flunk
  # flunk "this is a demo. yay for failures!"
end

def test_handle_results
  @a.files to test.clear
-uu-:---F1 test autotest.rb 52% (145,4) (Ruby Outl Icy Abbrev)-----
512 % ruby -Ilib bin/autotest

-uu1:**-F1 autotest All (1,30) (Shell:run Icy Shell-Compile)-----
```



Heckle

Make Your Tests Cry

```
def is_awesome?  
  x if @awesome  
end
```

Started

.....
Finished in 1.638607 seconds.

10 tests, 42 assertions, 0 failures, 0 errors

```
def is_awesome?  
  x unless @awesome  
end
```

Started

.....
Finished in 1.314607 seconds.

10 tests, 42 assertions, 0 failures, 0 errors

← morphed

BAD!!!

Flog

Because Beatings Build Character

```
class Test
  def blah          # 11.2 =
    a = eval "1+1" # 1.2 + 6.0 +
    if a == 2 then # 1.2 + 1.2 + 0.4 +
      puts "yay"   # 1.2
    end
  end
end
end
```

```
Test#blah: (11.2)
  6.0: eval
  1.2: branch
  1.2: ==
  1.2: puts
  1.2: assignment
  0.4: lit_fixnum
```

Flay
The New Kid


```
% flay unit/itemconfig.rb  
Processing unit/itemconfig.rb...
```

```
Matches found in :when (mass = 572)  
  unit/itemconfig.rb:343  
  unit/itemconfig.rb:379  
  unit/itemconfig.rb:706  
  unit/itemconfig.rb:742
```

```
Matches found in :when (mass = 500)
```

```
... and so on
```

```

when /^(#{__item_numval_optkeys(tagid(tag0rId)).join('|')})/
  # conf =
  tk_split_simplelist(_fromUTF8(tk_call_without_enc(*(__item_confinfo_cmd(tagid(t
ag0rId)) << "-#{slot}"))))
  conf =
  tk_split_simplelist(tk_call_without_enc(*(__item_confinfo_cmd(tagid(tag0rId))
<< "-#{slot}")), false, true)

  if ( __item_confinfo_struct(tagid(tag0rId))[:default_value] \
    && conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]] )
    begin
      conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]] =
        number(conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]])
    rescue
      conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]] = nil
    end
  end
  if ( conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]] )
    begin
      conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]] =
        number(conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]])
    rescue
      conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]] = nil
    end
  end
end

```

```

when /^(#{__item_boolval_optkeys(tagid(tagOrId)).join('|')})/
  # conf =
  tk_split_simplelist(_fromUTF8(tk_call_without_enc(*(__item_confinfo_cmd(tagid(t
agOrId)) << "-#{slot}"))))
  conf =
  tk_split_simplelist(tk_call_without_enc(*(__item_confinfo_cmd(tagid(tagOrId))
<< "-#{slot}")), false, true)

  if ( __item_confinfo_struct(tagid(tagOrId))[:default_value] \
    && conf[__item_confinfo_struct(tagid(tagOrId))[:default_value]] )
    begin
      conf[__item_confinfo_struct(tagid(tagOrId))[:default_value]] =
        bool(conf[__item_confinfo_struct(tagid(tagOrId))[:default_value]])
    rescue
      conf[__item_confinfo_struct(tagid(tagOrId))[:default_value]] = nil
    end
  end
  if ( conf[__item_confinfo_struct(tagid(tagOrId))[:current_value]] )
    begin
      conf[__item_confinfo_struct(tagid(tagOrId))[:current_value]] =
        bool(conf[__item_confinfo_struct(tagid(tagOrId))[:current_value]])
    rescue
      conf[__item_confinfo_struct(tagid(tagOrId))[:current_value]] = nil
    end
  end
end

```

```

when /^(#{__item_numval_optkeys(tagid(tag0rId)).join('|')})/
  # conf =
  tk_split_simplelist(_fromUTF8(tk_call_without_enc(*(__item_confinfo_cmd(tagid(t
ag0rId)) << "-#{slot}"))))
  conf =
  tk_split_simplelist(tk_call_without_enc(*(__item_confinfo_cmd(tagid(tag0rId))
<< "-#{slot}")), false, true)

  if ( __item_confinfo_struct(tagid(tag0rId))[:default_value] \
    && conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]] )
    begin
      conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]] =
        number(conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]])
    rescue
      conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]] = nil
    end
  end
  if ( conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]] )
    begin
      conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]] =
        number(conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]])
    rescue
      conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]] = nil
    end
  end
end

```

```

when /^({__item_boolval_optkeys(tagid(tag0rId)).join('|')})$/
  # conf =
  tk_split_simplelist(_fromUTF8(tk_call_without_enc(*(__item_confinfo_cmd(tagid(t
ag0rId)) << "-#{slot}"))))
  conf =
  tk_split_simplelist(tk_call_without_enc(*(__item_confinfo_cmd(tagid(tag0rId))
<< "-#{slot}")), false, true)

  if ( __item_confinfo_struct(tagid(tag0rId))[:default_value] \
    && conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]] )
    begin
      conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]] =
        bool(conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]])
    rescue
      conf[__item_confinfo_struct(tagid(tag0rId))[:default_value]] = nil
    end
  end
  if ( conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]] )
    begin
      conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]] =
        bool(conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]])
    rescue
      conf[__item_confinfo_struct(tagid(tag0rId))[:current_value]] = nil
    end
  end
end

```

Shiny Shit
MmmmmmmCandy

So you've
seen
ParseTree

and probably
ruby2ruby

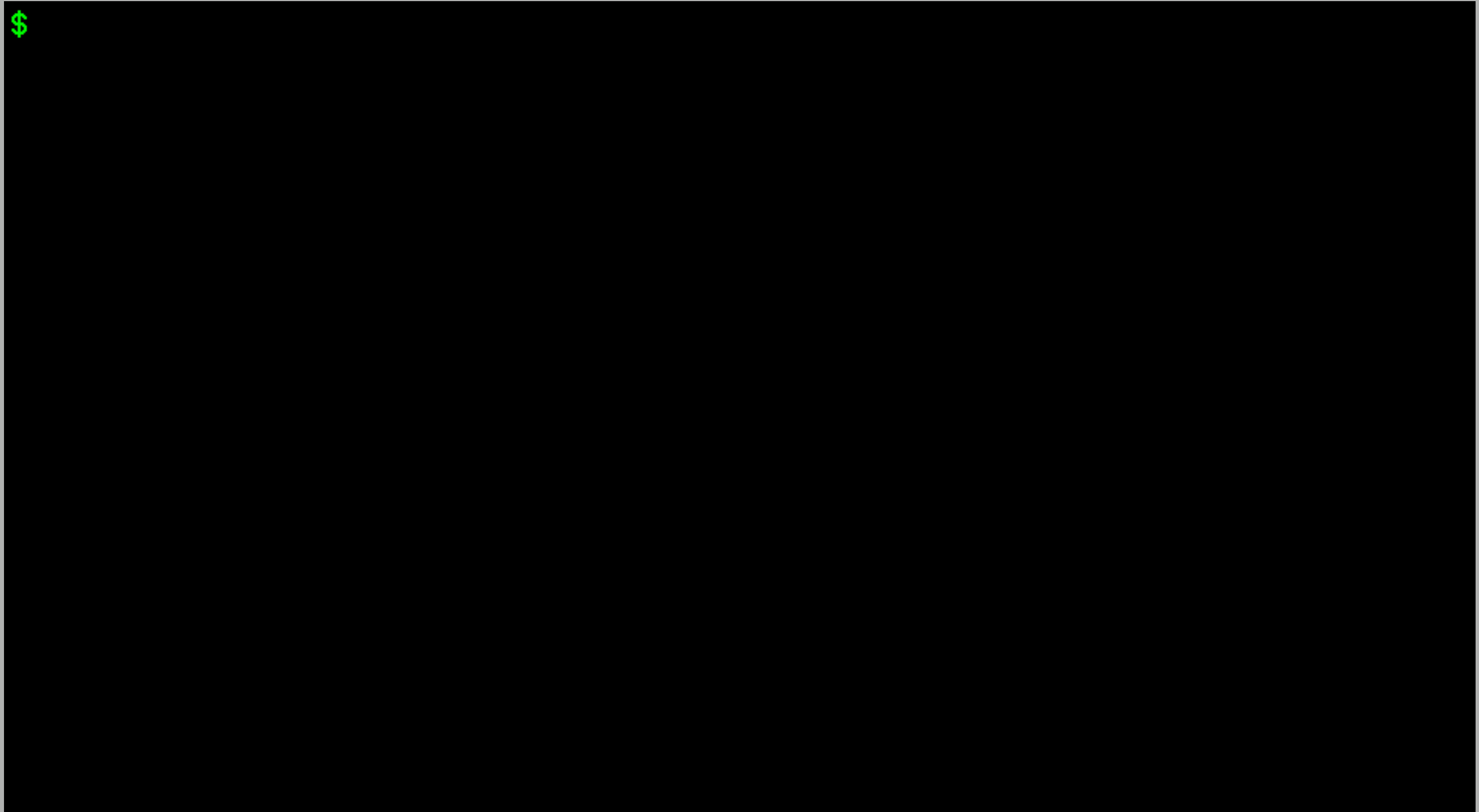
and you *may*
have played
with
RubyInline

but we also
have ruby2c

and a code
profiler

profiler +
ParseTree +
ruby2c +
rubyinline = ?

\$



rad

Thank
You