e-mail drop box

a piece of ruby-cake (~100 lines code)
key goals:

• Familiarize with Ruby for web-development
  • Use Ruby CGI class
• Use Ruby documentation
• Understand HTML forms and how they bind with Ruby CGI
• Interface with “system”
def show_form(cgi)
  cgi.form do
    content = ""
    content << cgi.h1 { "Send me an e-mail!\n" }
    content << cgi.hr
    content << cgi.b("From: ") + cgi.text_field("from", "Your e-mail address", 40)
    content << cgi.hr
    content << cgi.b("Subject: ") +
    cgi.text_field("subject", "Hello!", 40)
    content << cgi.hr
    content << cgi.b("Your message:")
    content << cgi.br
    content << cgi.textarea("msg", 50, 6)
    content << cgi.hr
    content << cgi.submit("cancel", "submit")
    content << cgi.submit("send", "submit")
    content
  end
end

def send_mail(cgi)
  body = "This e-mail was sent from:" +
  cgi["from"].first + "\n"
  body <<
  "------------------------------------------------------"
  cmnd = "echo " #{body} " | mail -s "
  cmnd << " mailme@here.com -f "
  cmnd << " + "
  system("cmnd")
  mail_sent(cgi)
end

def mail_sent(cgi)
  content = ""
  content << cgi.h1("Mail sent!")
  content << cgi.hr
  content << cgi.b("From: ") + cgi["from"].first + "\n"
  content << cgi.hr
  content << cgi.b("Subject: ") + cgi["subject"].first + "\n"
  content << cgi.hr
  content << cgi.b("Message: ") + cgi["msg"].first + "\n"
  content
end

#!/usr/bin/ruby
# mailme.cgi
require "cgi"
require 'net/smtp'

cgi = CGI.new("html4")
form = cgi.params["from"]
subject = cgi.params["subject"]
msg = cgi.params["msg"]

begin
  case cgi.params["submit"].first
  when "send"
    cgi.out do
      send_mail(cgi)
    end
  when "cancel"
    cgi.out do
      show_form(cgi)
    end
  else
    show_form(cgi)
  end
end

# send mail

def send_email(from, subject, msg)
  body = "This e-mail was sent from:" +
  from + "\n"
  body <<
  "------------------------------------------------------"
  cmnd = "echo " #{body} " | mail -s "
  cmnd << " mailme@here.com -f "
  cmnd << " + "
  system("cmnd")
end

from = form
subject = subject
msg = msg

send_email(from, subject, msg)
simple lessons learned

• CGI online / offline mode
  \( (\exists \text{REQUEST\_METHOD}) \)
determines CGI object capabilities

• Ruby + CGI = ♥

• Ruby : *do the same thing in multiple ways!*
  • practice, experience will allow you to appreciate this flexibility
Simplicity defined 😊

Following the previous design, stripped off to the bare essentials
Bryan

Simplicity redefined 😊

MIME-attachments allowed (metasend)

pre-cautious: stripping off special filename characters
Brandon

Using CSS for alignment

address-pattern check (regexp)

Using SMTP instead of system mail (works@betaweb?)
Jason

write mail on a file, then send ☁️
good for logging, special chars, but someone has to flush ...

Did it the hard way ( puts “HTML syntax”)
Ian

Including a captcha-test for spammers 🎃!

(аргхх, looks cool, but images are static ... )
Dennis

Including a random-question generator for spammer-filtering!

Lesson learned: there is always a smart spammers scarecrow...

Q: Can you think of analogous, AI-proof challenges?