

# 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.br
    content << cgi.b{"Subject: "} +
cgi.text_field("subject", "Hello!", 40)
    content << cgi.br
    content << cgi.b{"Your message:"}
    content << cgi.br
    content << cgi.textarea("msg", 50, 6)
    content << cgi.br
    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\n"
  body <<
"-----"
  cmdnd = "echo \" #{body} \""
  cmdnd << "| mail -s " + cgi["subject"].first
  cmdnd << " mailme@here.com -- -f " + cgi['from'].first
  system("cmdnd")
  mail_sent(cgi)
end

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

```

```

#!/usr/bin/ruby
# mailme.cgi

require "cgi"
require 'net/smtp'

cgi = CGI.new("html4")

from = cgi.params["from"]
subject = cgi.params["subject"]
msg = cgi.params["msg"]

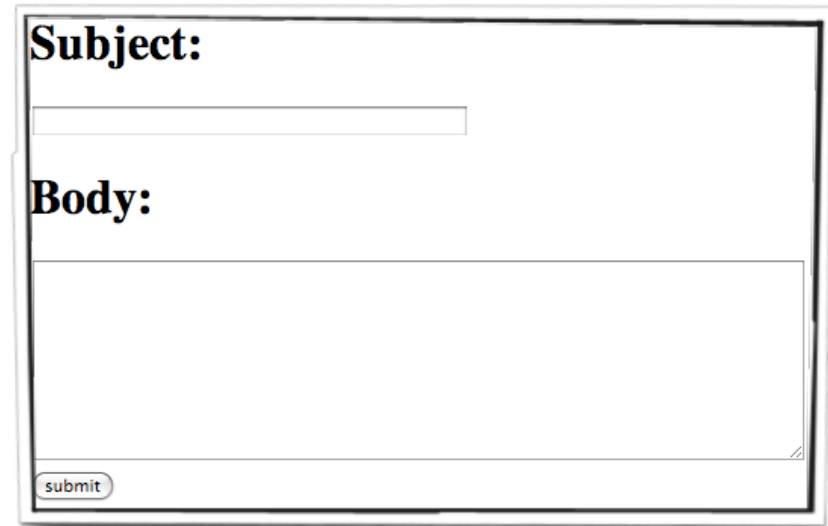
case cgi.params["submit"].first
when "send"
  cgi.out do
    send_mail(cgi)
  end
when "cancel"
  cgi.out do
    show_form(cgi)
  end
else
  cgi.out do
    show_form(cgi)
  end
end
end

```

# simple lessons learned

- CGI online / offline mode  
( $\exists$ 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

# Stephan

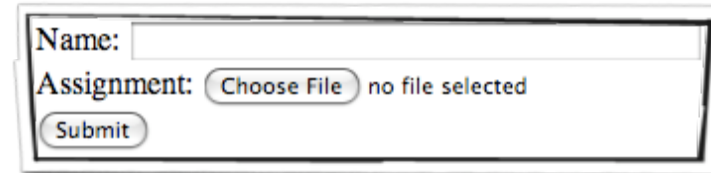


The image shows a simple, rectangular form with a thin border. At the top left, the word "Subject:" is written in a bold, sans-serif font. Below it is a single-line text input field. Further down, the word "Body:" is written in the same bold font. Below "Body:" is a large, empty rectangular area for text entry. In the bottom-left corner of the form, there is a small, rounded button labeled "submit". In the bottom-right corner, there is a small, faint icon of a right-angled triangle.

Simplicity defined 😊

Following the previous design,  
stripped off to the bare  
essentials

# Bryan



A screenshot of a web form with a white background and a thin black border. The form contains three elements: a text input field labeled "Name:" which is empty; a text input field labeled "Assignment:" containing the text "Choose File" and "no file selected"; and a rounded rectangular button labeled "Submit".

Simplicity redefined ☺

MIME-attachments allowed  
(metasend)

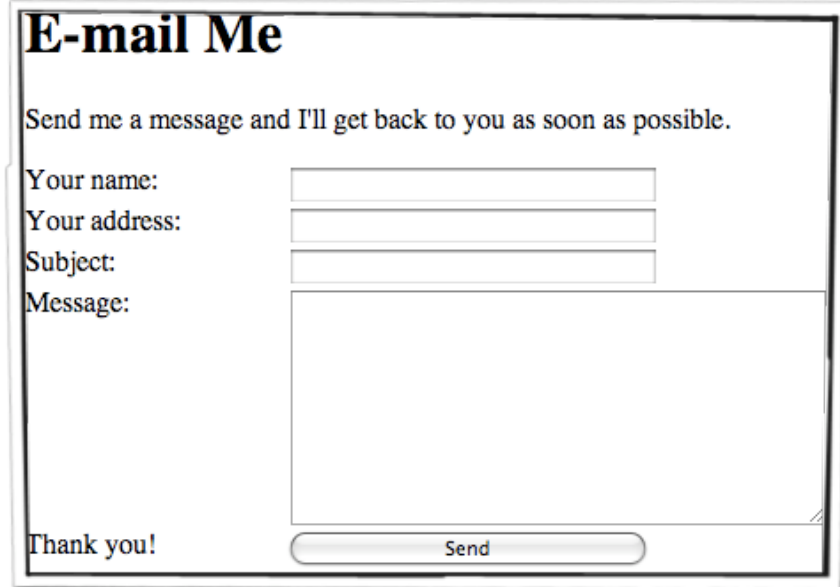
pre-cautious: stripping off special  
filename characters

# Brandon

Using CSS for alignment

address-pattern check (regexp)

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



**E-mail Me**

Send me a message and I'll get back to you as soon as possible.

Your name:

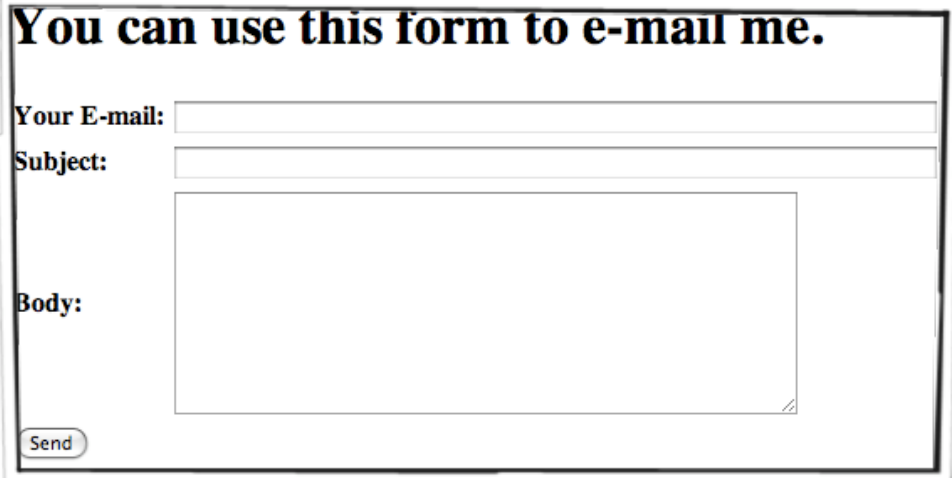
Your address:

Subject:

Message:

Thank you!

# Jason



**You can use this form to e-mail me.**

Your E-mail:

Subject:

Body:

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 ☠ !

(arghh, looks cool, but images are static ...)

## Drop box for Ian C

Please enter you name

Please enter the subject of your message

And now your subject (You must atleast fill this out)

Finally, a captcha test - read the middle one

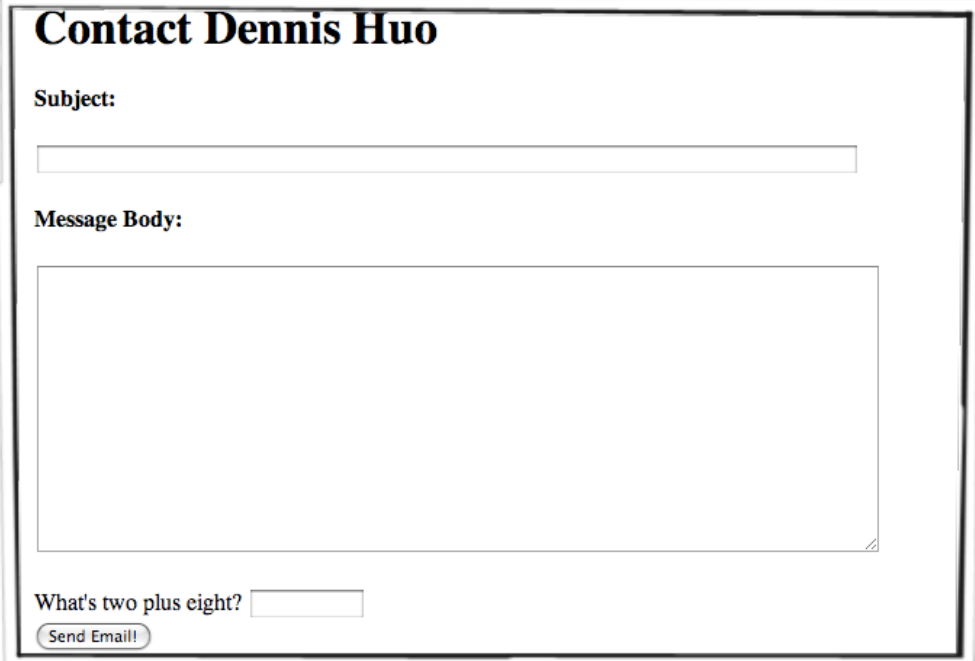
e5hb

xmQKi

6he3

# 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?



**Contact Dennis Huo**

Subject:

Message Body:

What's two plus eight?