```
#!/usr/local/bin/perl

# This is the minimalist script to demonstrate the use of
# the cgi-lib.pl library
# Copyright (C) 1994 Steven E. Brenner
# $Header: /cys/people/seb1005/http/docs/web/RCS/minimal.cgi,v 1.1 1994/06/28 23:06:02 seb
1005 Exp $



if (&MethGet) {
  print &PrintHeader;
  print &PrintMessage;
} else {
    &ReadParse(*input);
    &CheckAddressee($input{'from'},$input{'subject'});
# addressee has "syr" in the from field so we move on
    &MailText(%input);
    print &PrintHeader, &PrintVariables(%input);
}

sub  CheckAddressee {
    local(@add,@machine);
    $_ = @_[0];

    if(@_[1] eq ""){
        print &PrintHeader;
        print <<SORRY;
<TITLE>No Can Do</TITLE>
<H1>No Can Do</H1>
<HR>
<H3>Please enter the subject of your mail in the subject field</H3>
<A HREF="http://altair.syr.edu:2024/cgi-bin/mail.cgi">Back to Mail Form</A>
SORRY
    exit(0);
    }

    tr/A-Z/a-z/;
    @add = split('@',$_);
    @machine = split('\.',$add[1]);
    if( $machine[@machine-2] ne "syr" ) {
        print &PrintHeader;
        print <<SORRY;
<TITLE>No Can Do</TITLE>
<H1>No Can Do</H1>
<HR>
<H3>I do not recognize your E-Mail. Please enter a SU address
in the "From" field of the Mail Form</H3>
<A HREF="http://altair.syr.edu:2024/cgi-bin/mail.cgi">Back to Mail Form</A>
SORRY
    exit(0);
    }
}

sub MethGet {
  return ($ENV{'REQUEST_METHOD'} eq "GET");
}

sub MailText {
    local (%in) = @_ if @_;
    open(TMP,">mail_csh$$");
print TMP <<EOM;

This message was mailed by
```

```
EOM
    print TMP $in{'from'};
print TMP <<EOM2;

The content of the message follows

EOM2
    print TMP $in{'content'};
    close(TMP);
    $return = `mail -s $in{'subject'} $in{'to'}< mail_csh$$`;
    if($return) {
        print &PrintHeader;
        print $return;
        exit(2);
    } else {
        print &PrintHeader;
        print <<OK;
<TITLE>Message Sent</TITLE>
<H1>Message Sent</H1>
<HR>
<H3>Your Message was sent.</H3>
<A HREF="http://altair.syr.edu:2024/cgi-bin/mail.cgi">Back to Mail Form</A>
OK
    exit(0);
    }                           #
}

# ReadParse
# Reads in GET or POST data, converts it to unescaped text, and puts
# one key=value in each member of the list "@in"
# Also creates key/value pairs in %in, using '\0' to separate multiple
# selections

# If a variable-glob parameter (e.g., *cgi_input) is passed to ReadParse,
# information is stored there, rather than in $in, @in, and %in.

sub ReadParse {
    local (*in) = @_ if @_;


  local ($i, $loc, $key, $val);

  # Read in text
  if ($ENV{'REQUEST_METHOD'} eq "GET") {
    $in = $ENV{'QUERY_STRING'};
  } elsif ($ENV{'REQUEST_METHOD'} eq "POST") {
    read(STDIN,$in,$ENV{'CONTENT_LENGTH'});
  }

  @in = split(/&/,$in);

  foreach $i (0 .. $#in) {
    # Convert plus's to spaces
    $in[$i] =~ s/\+/ /g;

    # Split into key and value.
    ($key, $val) = split(/=/,$in[$i],2); # splits on the first =.

    # Convert %XX from hex numbers to alphanumeric
    $key =~ s/%(..)/pack("c",hex($1))/ge;
    $val =~ s/%(..)/pack("c",hex($1))/ge;

    # Associate key and value
    $in{$key} .= "\0" if (defined($in{$key})); # \0 is the multiple separator
```

```
    $in{$key} .= $val;

  }

  return 1; # just for fun
}

# PrintHeader
# Returns the magic line which tells WWW that we're an HTML document

sub PrintHeader {
  return "Content-type: text/html\n\n";
}

# PrintVariables
# Nicely formats variables in an associative array passed as a parameter
# And returns the HTML string.

sub PrintVariables {
  local (%in) = @_;
  local ($old, $out, $output);
  $old = $*;  $* =1;
  $output .=  "<DL COMPACT>";
  foreach $key (sort keys(%in)) {
    foreach (split("\0", $in{$key})) {
      ($out = $_) =~ s/\n/<BR>/g;
      $output .=  "<DT><B>$key</B><DD><I>$out</I><BR>";
    }
  }
  $output .=  "</DL>";
  $* = $old;

  return $output;
}

# PrintVariablesShort
# Nicely formats variables in an associative array passed as a parameter
# Using one line per pair (unless value is multiline)
# And returns the HTML string.


sub PrintVariablesShort {
  local (%in) = @_;
  local ($old, $out, $output);
  $old = $*;  $* =1;
  foreach $key (sort keys(%in)) {
    foreach (split("\0", $in{$key})) {
      ($out = $_) =~ s/\n/<BR>/g;
      $output .= "<B>$key</B> is <I>$out</I><BR>";
    }
  }
  $* = $old;

  return $output;
}




sub PrintMessage {
    print <<EOM;
<HTML>
<HEAD>
<TITLE>Mail a message to PHY106 Instructors</TITLE>
```

```
</HEAD>
<BODY>
<H1>Mail a message to PHY106 Instructors</H1>
<HR>
Please complete all the fields below and then hit the button
<B>Send Mail</B>
<P>
Please enter your name in the next box. This email cannot be delivered without
your name.
<form method=POST>
<B>From:</B> <input name="from" SIZE=40>
<P>
<B>To:<B>      <select NAME="to">
        <OPTION> mcnamara@suhep.phy.syr.edu
        <OPTION> salgado@npac.syr.edu
        <OPTION> diglio@npac.syr.edu
        <OPTION> edlipson@mailbox.syr.edu
        <OPTION> marvin@suhep.phy.syr.edu
        <OPTION> smc@npac.syr.edu
        <OPTION> falcioni@npac.syr.edu
        <OPTION> jtgruzle@mailbox.syr.edu
        <OPTION> petrego@suhep.phy.syr.edu
        </SELECT>
<P>
Subject: <input name="subject" size=60>
<P>
<textarea name="content" rows="15" cols="72"></textarea>

<input type="submit" value="Send Message"> </form>
</BODY>
</HTML>
EOM
}
```