



April 2010  
Mondays Tuesdays Wednesdays Thursdays Fridays  
Saturdays Sundays

May 2010  
Mondays Tuesdays Wednesdays Thursdays Fridays  
Saturdays Sundays

June 2010  
Mondays Tuesdays Wednesdays Thursdays Fridays  
Saturdays Sundays

Obo

Select Existing Email

Entered Transactions First Select

Accts

Date

Amount

Person Name

Balances

Confidence

Date | Subject

Link to Invoice

Obo

> Assets

—  
—  
—  
—  
—

Select Account

▼

get-account()

return account

1 next show 2

2 next

Confidence(transact)

getmarks

Fetch confidence of

nextstep()

global step

If 1 then = -  
(+ 2 then ...)

step += 1

update-account(?)

Export-csv (det-all)  
saving dialog

for i in range(len(l8))

l8[i][8] = date

l8[i][8] + 1 = desc

(8 \* i + 2) = ID

(8 \* i + 5) = Amount

lmc["date"] =

Invoice-matcher(stripe\_email, pdf-trans, amazon-link)

return link

parse\_ofx(filehandle)

return dict\_all

parse\_amex\_csv(fh)

return dict\_amex

parse\_paypal\_csv(fh)

return dict\_all

parse\_qfx(fh)

return dict\_all

parse\_pdf(fh)

return dict\_all

parse\_amazon\_csv(fh)

return dict\_all

get\_link\_invoice(mail\_order, mail\_snippet, paypal\_dict, amazon\_dict)

return link

get\_emails(searchstr)

return emails\_dict

get\_emails\_group(searchstr)

get\_search\_txns(txn\_dict)

match\_txns(transactions, paypal\_txns, amazon\_txns)

~~get\_emails(searchstr)~~

return bool matched

Invoice\_matcher (trans, email)  
if id then link = <http://Paypal.com/id>  
else <http://Paypal.com/trans+date+id>

else

for trans count, if match <sup>Paypal</sup> date is  $\pm 3$  days  
take first one <http://Amazon.com>

else

if count . sender = Amazon

match trans, count in amazon\_trans  
return <sup>got id</sup> [Amazon.com/ID](http://Amazon.com/ID)

If count . sender = <sup>Service</sup> paypal.com

match in popbill\_trans  
get id

return [Paypal.com/ID](http://Paypal.com/ID)

if > handle

return with  
date

if account clicked, transaction [selected][cancel]  
& account\_set