

PLAIN TEXT ACCOUNTING

ACCOUNTING FOR POWER USERS

Paulo Flabiano Smorigo

p fsmorigo@{twitter, gmail, github, linkedin, ...}



v1.0 - Nov 2017

DISCLAIMER

We are talking about using terminal and edit text files manually (with some helpers). If you are not comfortable, or doesn't like it, maybe this is not suitable for you. At first sight, it might be stupid and not practical but you will be surprised on how the system works well.

WHAT WE'RE GOING TO TALK ABOUT?

- PART 1: Double-entry & account hierarchy
- PART 2: Plain Text Accounting
- PART 3: Hands-on with ledger-cli

PART 1

Double-entry & account hierarchy

The boring part

DOUBLE-ENTRY BOOKKEEPING SYSTEM

- Values are transferred from one account to another
- All accounts must balance to zero (error check)

SPREADSHEET

| Date | Operation | Value | Balance |
|-------|-----------|--------|---------|
| Sep 2 | Paycheck | \$1000 | \$1000 |
| Sep 6 | Fuel | \$150 | \$850 |
| Sep 6 | Groceries | \$300 | \$550 |
| Sep 9 | Rent | \$500 | \$50 |

DOUBLE-ENTRY

| Date | Operation | From | To | Value |
|-------|-----------|---------|--------|--------|
| Sep 2 | Paycheck | Company | Bank | \$1000 |
| Sep 6 | Fuel | Bank | Auto | \$150 |
| Sep 6 | Groceries | Bank | Market | \$300 |
| Sep 9 | Rent | Bank | House | \$500 |

DOUBLE-ENTRY

```
$ ledger -f examples/ex1.ledger register
```

| | | | | |
|------------|-----------|---------|---------|--------|
| 2017-09-02 | Paycheck | Bank | \$1000 | \$1000 |
| | | Company | \$-1000 | 0 |
| 2017-09-06 | Fuel | Fuel | \$150 | \$150 |
| | | Bank | \$-150 | 0 |
| 2017-09-06 | Groceries | Market | \$150 | \$150 |
| | | Bank | \$-150 | 0 |
| 2017-09-06 | Rent | House | \$500 | \$500 |
| | | Bank | \$-500 | 0 |

MUST BALANCE TO ZERO

```
$ ledger -f examples/ex1.ledger balance
```

| | |
|---------|---------|
| \$200 | Bank |
| \$-1000 | Company |
| \$150 | Fuel |
| \$500 | House |
| \$150 | Market |
| ----- | |
| 0 | |

AMERICAN APPROACH

- Assets (bank accounts, meal ticket, ...)
- Equity (opening balances, car, house, ...)
- Liabilities (credit card, ...)
- Income (salary, ...)
- Expenses (expenses :P)

ACCOUNTS HIERARCHY

```
Assets
  └─ Checking
  └─ Savings
    └─ Meal Ticket
Equity
  └─ Opening Balances
Liabilities
  └─ Credit Card
Income
  └─ Salary
    └─ Others
Expenses
  └─ Auto
  └─ Market
    └─ House
```

MUST BALANCE TO ZERO (V2)

```
$ ledger -f examples/ex2.ledger balance
```

| | |
|---------|-----------------|
| \$200 | Assets:Checking |
| \$800 | Expenses |
| \$150 | Auto |
| \$500 | House |
| \$150 | Market |
| \$-1000 | Income:Salary |
| ----- | |
| | 0 |

PART 2

Plain Text Accounting

Accounting ...with text

HISTORY

- In 2003, John Wiegley created Ledger
- The idea was welcomed by some technical folks
- 5+ actively-developed Ledger-likes
- In 2016, plaintextaccounting.org

LEDGER-LIKES

| Project | Start | Code | Committers |
|------------|-------|---------|------------|
| ledger-cli | 2003 | C++ | 112 |
| hledger | 2007 | Haskell | 67 |
| Beancount | 2008 | Python | 17 |

BUT WHY TEXT?

- It's valuable, be accessible for ever
- ... even without a specific software
- Easy to edit, search and organize
- ... as long as you are good with text editing, and ...
- ... you can version-control it! Hello git!
- Use your own security method (gpg, ecryptfs)

EDITOR SUPPORT

- **vim-ledger** for Vim (recommended :P)
- **ledger-mode** for Emacs
- **language-ledger** for Atom
- Also for Sublime, TextMate, VisualStudio...

BASIC

| <Date> <Description> | <Value> |
|-----------------------|---------|
| <Account:Sub-account> | <Value> |
| <Account:Sub-account> | <Value> |

| | |
|---------------------|---------|
| 2017-09-02 Paycheck | |
| Assets:Checking | \$1000 |
| Income:Salary | \$-1000 |

| | |
|---------------------|--------|
| 2017-09-02 Paycheck | |
| Assets:Checking | \$1000 |
| Income:Salary | |

Remember: balance to zero!

MULTIPLE ACCOUNTS

2017-03-31 Paycheck

| | |
|--------------------------|--------|
| Assets:Bank:Checking | \$1000 |
| Expenses:Taxes:Federal | \$200 |
| Expenses:Taxes:Syndicate | \$50 |
| Expenses:Medical | \$150 |
| Income:Salary | |

CLEAR / PENDING

2017-09-02 * Paycheck
Assets:Checking \$1000
Income:Salary

2017-09-02 ! Paycheck
Assets:Checking \$1000
Income:Salary

EFFECTIVE DATE

2017-09-06=2017-09-09 Rent

Expenses:House

\$500

Assets:Checking

2017-09-06 Groceries

Expenses:Market

\$50 ; [=2017-10-10]

Expenses:Market

\$50 ; [=2017-11-10]

Expenses:Market

\$50 ; [=2017-12-10]

Assets:Checking

COMMODITIES AND CURRENCIES

Cost per item

2010-05-31 Market
Assets:Fridge 35 apples @ \$0.42
Assets:Cash

2010-05-31 Market
Assets:Fridge 35 apples {=\$0.42}
Assets:Cash

Total cost

2010-05-31 Market
Assets:Fridge 35 apples @@ \$14.70
Assets:Cash

METADATA

```
2012-03-10 * KFC
    Expenses:Food           $20.00
    Assets:Cash
    ; :TAG:
```

```
2012-03-10 * KFC
    Expenses:Food           $20.00
    Assets:Cash
    ; :TAG1:TAG2:TAG3:
```

```
2012-03-10 * KFC
    Expenses:Food           $20.00
    Assets:Cash
    ; MyTag: This is just a bogus value for MyTag
```

BUDGETING AND FORECASTING

~ Monthly

| | |
|--------------------|-------------------------------|
| Expenses:Rent | \$500.00 |
| Expenses:Food | \$450.00 |
| Expenses:Auto:Gas | \$120.00 |
| Expenses:Insurance | \$150.00 |
| Expenses:Phone | \$125.00 |
| Expenses:Utilities | \$100.00 |
| Expenses:Movies | \$50.00 |
| Expenses | \$200.00 ; all other expenses |
| Assets | |

~ Yearly

| | |
|----------------------|----------|
| Expenses:Auto:Repair | \$500.00 |
| Assets | |

PART 3

Hands-on with ledger-cli

Show me the code

LEDGER-CLI SYNTAX

```
$ ledger [command] [options] [arguments]
register --file FILE (-f)
balance --begin DATE (-b)
print --end DATE (-e)
...
stats --effective
xml --market (-V)
convert --price-db FILE
...
--sort EXPR (-S)
...
```

Ledger will never alter your input file!

COMMON COMMANDS

```
$ man ledger
```

```
...
```

balance [report-query]

Print a balance report showing totals for postings that match report-query, and aggregate totals for parents of those accounts.

```
...
```

register [report-query]

List all postings matching the report-query. This is one of the most common commands, and can be used to provide a variety of useful reports.

```
...
```

EXAMPLE

```
$ cat examples/ex2.ledger
```

| | | |
|------------|-----------------|--------|
| 2017-09-02 | Paycheck | |
| | Assets:Checking | \$1000 |
| | Income:Salary | |
| 2017-09-06 | Fuel | |
| | Expenses:Auto | \$150 |
| | Assets:Checking | |
| 2017-09-06 | Groceries | |
| | Expenses:Market | \$150 |
| | Assets:Checking | |
| 2017-09-06 | Rent | |
| | Expenses:House | \$500 |
| | Assets:Checking | |

EXAMPLE: REGISTER

```
$ ledger -f examples/ex2.ledger reg checking
```

| | | | | |
|------------|-----------|-----------------|--------|--------|
| 2017-09-02 | Paycheck | Assets:Checking | \$1000 | \$1000 |
| 2017-09-06 | Fuel | Assets:Checking | \$-150 | \$850 |
| 2017-09-06 | Groceries | Assets:Checking | \$-150 | \$700 |
| 2017-09-06 | Rent | Assets:Checking | \$-500 | \$200 |

EXAMPLE: BALANCE

```
$ ledger -f examples/ex2.ledger bal  
  
$200  Assets:Checking  
$800  Expenses  
$150   Auto  
$500   House  
$150   Market  
$-1000 Income:Salary  
-----  
0
```

ANOTHER EXAMPLE

```
$ cat examples/ex3.ledger
```

| | | | |
|-----------------------|------|-----------------|----------------------|
| 2017-09-02 | * | Paycheck | |
| | | Assets:Checking | \$1000 |
| | | Income:Salary | |
| 2017-09-06 | * | Fuel | |
| | | Expenses:Auto | 40 GAS @ \$3.75 |
| | | Assets:Checking | |
| 2017-09-06 | ! | Groceries | |
| | | Expenses:Market | \$50 ; [=2017-10-10] |
| | | Expenses:Market | \$50 ; [=2017-11-10] |
| | | Expenses:Market | \$50 ; [=2017-12-10] |
| | | Assets:Checking | |
| 2017-09-06=2017-09-09 | Rent | | |
| | | Expenses:House | \$500 |
| | | Assets:Checking | |

ANOTHER EXAMPLE

```
$ ledger --file ex3.ledger --market --effective --sort date reg
```

| | | | | |
|------------|-----------|-----------------|---------|--------|
| 2017-09-02 | Paycheck | Assets:Checking | \$1000 | \$1000 |
| | | Income:Salary | \$-1000 | 0 |
| 2017-09-06 | Fuel | Expenses:Auto | \$150 | \$150 |
| | | Assets:Checking | \$-150 | 0 |
| 2017-09-06 | Groceries | Assets:Checking | \$-150 | \$-150 |
| 2017-09-09 | Rent | Expenses:House | \$500 | \$350 |
| | | Assets:Checking | \$-500 | \$-150 |
| 2017-10-10 | Groceries | Expenses:Market | \$50 | \$-100 |
| 2017-11-10 | Groceries | Expenses:Market | \$50 | \$-50 |
| 2017-12-10 | Groceries | Expenses:Market | \$50 | 0 |

SCRIPTING

- Itau case. Let's go...

LINKS & REFERENCES

- This presentation
<https://github.com/pfsmorigo/reveal.js/tree/pta>
- Ledger: Command-Line Accounting
<http://ledger-cli.org/3.0/doc/ledger3.html>
- Plain Text Accounting
<http://plaintextaccounting.org/>
- Ledger CLI cheatsheet
<https://devhints.io/ledger>

THE END

Thanks! Questions?

NONE - FADE - SLIDE - CONVEX - CONCAVE - ZOOM
BLACK - WHITE - LEAGUE - SKY - BEIGE - SIMPLE - SERIF - BLOOD - NIGHT - MOON - SOLARIZED