Fizz buzz - LuaX demo

Christophe Delord - http://cdelord.fr/fizzbuzz

Thu Dec 14, 2023

Disclaimer

This document is not about Fizzbuzz.
This document is a suggestion to simplify the build process of software projects.
Fizzbuzz is just an application example.

Introduction

Lots of software projects involve various tools, free as well as commercial.

Lots of different data formats and scripting languages

- projects are less scalable and harder to maintain
- sharing data is painful and counter productive

e.g.: Bash, Python, Javascript, plain text, YAML, JSON, XML, CSV, INI, TOML, ...

Suggested solution

- a **common, simple and powerful data format**
- and a **reduced set of highly configurable tools**.

Lua - General programming language

Lua is a

- powerful,
- efficient,
- lightweight,
- embeddable scripting language.

It supports

- procedural programming,
- object-oriented programming,
- functional programming,
- data-driven programming,
- and data description.

Why choose Lua

- proven, robust language
- fast
- portable
- powerful but simple
- small
- free

Get Lua

https://www.lua.org

LuaX - Extended Lua interpreter / compiler

Lua eXtended

- Lua interpreter and REPL
- based on Lua 5.4
- more built-in packages
- multiplatform Lua *compiler*
- zero dependency

Improved Lua prompt

- history
- human-readable tables

General modules

- `F`: functional programming
- `fs`: file system
- `sh`: shell commands

Math modules

- `mathx`: `math` extension
- `imath`: arbitrary precision
- `qmath`: rational numbers
- `complex`: complex numbers

And more...

- `crypt`: cryptography
- `lz4`: compression
- `lpeg`: parsers
- `luasocket`: network
- ...

Get LuaX

https://github.com/CDSoft/luax

Scripting with LuaX

Fully compatible with Lua

- general programming language
- good documentation

Lua tables

- common data format
- human-readable and structured

project/software configuration

- Lua tables
- project configuration
- software configuration
- readable by any Lua interpreter

Lua table usages

- documentation generation
- code generation
- test results
- test reports
- requirement coverage

Get LuaX

https://github.com/CDSoft/luax

bang - Ninja file generator scriptable in LuaX

bang

- Ninja file generator scriptable in LuaX
- Lua/LuaX macros

LuaX

- compiled with LuaX
- all LuaX modules available in bang build scripts

How does bang work?

- bang takes a build description (a LuaX script)
- and generates a Ninja file

Features

- ninja primitives (variables, rules, build statements, . . .)
- rule/build statement pairs described in a single function call
- file listing and filenames list management using LuaX modules
- functional programming (LuaX `F` module)
- pipe simulation using rule composition
- “clean”, “install” and “help” targets

Get bang

https://github.com/CDSoft/bang

ypp - text preprocessing

ypp

- minimalist and generic text preprocessor
- Lua/LuaX macros

LuaX

- compiled with LuaX
- all LuaX modules available in ypp macros

How does ypp work?

- ypp searches for Lua expressions
- and replaces their sources by their results

Features

- Lua expression evaluation
- file inclusion
- conditional text
- functional programming (LuaX `F` module)
- file management (LuaX `fs` module)
- requirement management (still experimental)

Get ypp

https://github.com/CDSoft/ypp

ypp examples

File inclusion

```
@include "file.md"
```

Lua definitions

```
@@[[
    local foo = 42
    N = foo * 23 + 34
    local function sq(x)
        return x*x
    end
    function sumsq(n)
        return F.range(N)
            : map(sq)
            : sum()
    end
]]
```

Lua macros

```
$$
\sum_{i=1}^{@N} i^2 =
    @sumsq(N)
$$
```

Example

$$\sum_{i=1}^{1000} i^2 = 333833500$$

Pandoc - Documentation converter / generation

What is Pandoc?

Pandoc is a swiss-army knife to convert from and to a bunch of document formats.

Why Pandoc?

Pandoc uses Lua scripts:

- custom readers
- custom writers
- Lua filters on Pandoc AST

Excellent documentation

Manual:
https://pandoc.org/MANUAL.html
Lua filters:
https://pandoc.org/lua-filters.html

Get Pandoc

https://pandoc.org

Panda - Pandoc addons

Panda

- Pandoc Lua filters
- variable expansion
- conditional blocks
- file inclusion
- script execution
- diagrams

Based on Pandoc Lua

- based on the Pandoc Lua interpreter
- LuaX specific functions are available

Get Panda

https://github.com/CDSoft/panda

Panda examples

Documentation extraction from source code

```
/*@@@
**`answer`** takes any question
and returns the most relevant answer.

Example:
``` c
    const char *meaning
        = answer("What's the meaning of life?");
```
@@@*/

const char *answer(const char *question)
{
    return "42";
}
```

Panda examples

Documentation extraction from source code

The Panda `doc` macro extract documentation (in Markdown) from an external file:

```
:::{doc=deep_thought.c}
:::
```

This is rendered as

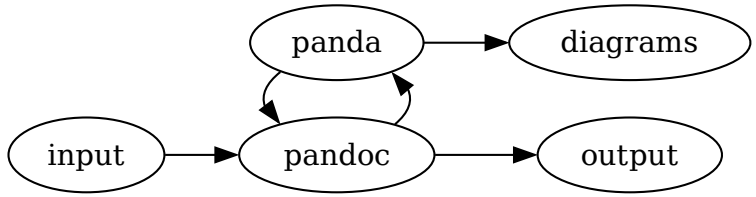
`answer` takes any question and returns the most relevant answer.
Example:

```
const char *meaning
    = answer("What's the meaning of life?");
```

Panda examples

Embedded diagrams

- code blocks
- replaced by an image by panda.



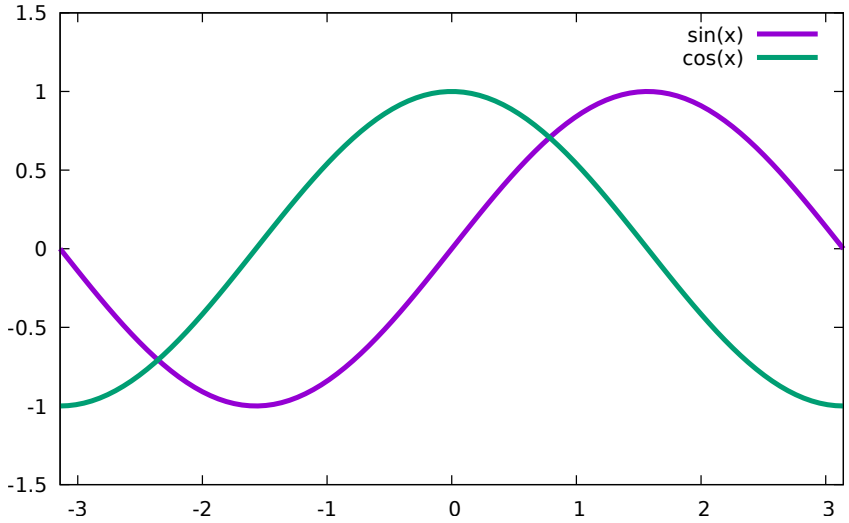
Example

````
```{.dot render="{{dot}}"}
digraph {
    rankdir=LR;
    input -> pandoc -> output
    pandoc -> panda -> {pandoc, diagrams}
    { rank=same; pandoc, panda }
    { rank=same; diagrams, output }
}
```
````

Panda examples

Embedded diagrams

- code blocks
- replaced by an image by panda.



Example

````
```{render="{{gnuplot}}"}
set xrange [-pi:pi]
set yrange [-1.5:1.5]
plot sin(x) lw 4, cos(x) lw 4
```
````

hey

hey

- single shell script
- install LuaX, ypp, pandoc, panda, ...

Simple installation

```
git clone https://github.com/CDSoft/hey
```

Simple usage

```
hey install all
```

Get hey

https://github.com/CDSoft/hey

Fizzbuzz

Concrete example of using

- LuaX
- YPP
- Pandoc
- Panda
- with a complete environment installed with `hey`

using Lua tables everywhere

- requirement database (still experimental)
- tests results
- test report

to specify

- examples using ypp and Panda macros
- requirement management

and test

- test execution
- test report

Complete Fizzbuzz example

- github.com/CDSoft/fizzbuzz
- cdelord.fr/fizzbuzz:
  - fizzbuzz.pdf
  - fizzbuzz.html

References

- **Fizzbuzz repository**: https://github.com/CDSoft/fizzbuzz
- **Lua**: https://www.lua.org
  - **Lua documentation**: https://www.lua.org/manual/5.4/
- **LuaX**: https://github.com/CDSoft/luax
- **bang**: https://github.com/CDSoft/bang
- **ypp**: https://github.com/CDSoft/ypp
- **Pandoc**: https://pandoc.org
  - **Pandoc manual**: https://pandoc.org/MANUAL.html
  - **Pandoc Lua filters**: https://pandoc.org/lua-filters.html
- **Panda**: https://github.com/CDSoft/panda
- **hey**: https://github.com/CDSoft/hey

Questions

Web / email

cdelord.fr

Github

github.com/CDSoft

Fizzbuzz

github.com/CDSoft/fizzbuzz
cdelord.fr/fizzbuzz
cdelord.fr/fizzbuzz/fizzbuzz.pdf

LinkdIn

linkedin.com/in/cdelord/