

Computational Algebraic topology: Lecture 2

Alberto Paoluzzi

March 10, 2017

Literate Python Programming

- 1 Introduction to Literate Programming
- 2 LarLib Literate Programming Environment
- 3 Literate programming for Julia
- 4 References

Introduction to Literate Programming

Donald Knuth. “Literate Programming (1984)” in Literate Programming. CSLI, 1992, pg. 99

I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature. Hence, my title: “Literate Programming.”

Donald Knuth. “Literate Programming (1984)” in Literate Programming. CSLI, 1992, pg. 99

I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature. Hence, my title: “Literate Programming.”

Let us change our traditional attitude to the construction of programs:

- Instead of imagining that our main task is to instruct a computer what to do,
- let us concentrate rather on explaining to human beings what we want a computer to do.

The CWEB System of Structure Documentation

Donald Knuth. Addison-Wesley. 1994. pag. 1

The philosophy behind CWEB is that an experienced system programmer, who wants to provide the best possible documentation of his or her software products, needs two things simultaneously:

- a language like TeX for formatting,
- and a language like C for programming.

The CWEB System of Structure Documentation

Donald Knuth. Addison-Wesley. 1994. pag. 1

The philosophy behind CWEB is that an experienced system programmer, who wants to provide the best possible documentation of his or her software products, needs two things simultaneously:

- a language like TeX for formatting,
- and a language like C for programming.

Neither type of language can provide the best documentation by itself; but when both are appropriately combined, we obtain a system that is much more useful than either language separately.

The CWEB System of Structure Documentation

Donald Knuth. Addison-Wesley. 1994. pag. 1

- 1 The structure of a software program may be thought of as a “WEB” that is made up of many interconnected pieces
- 2 To document such a program we want to explain each individual part of the web and how it relates to its neighbors.

The CWEB System of Structure Documentation

Donald Knuth. Addison-Wesley. 1994. pag. 1

- 1 The structure of a software program may be thought of as a “WEB” that is made up of **many interconnected pieces**
- 2 To document such a program **we want to explain each individual part** of the web **and how it relates to its neighbors**.

The typographic tools provided by TeX give us an opportunity **to explain the local structure** of each part by making that structure visible, and the programming tools provided by languages like C make it possible for us **to specify the algorithms** formally and unambiguously.

The CWEB System of Structure Documentation

Donald Knuth. Addison-Wesley. 1994. pag. 1

Besides providing a documentation tool, CWEB enhances the C language by providing the **ability to permute pieces of the program text**, so that a large system can be understood entirely

The CWEB System of Structure Documentation

Donald Knuth. Addison-Wesley. 1994. pag. 1

Besides providing a documentation tool, CWEB enhances the C language by providing the **ability to permute pieces of the program text**, so that a large system can be understood entirely

- in terms of **small sections**
- and their **local interrelationships**

Daniel Mall. “Recommendation for Literate Programming”

The **key features** of literate programming are the organization of **source code into small sections** and the production of a **book quality program listing**.

- Literate programming is an excellent method **for documenting the internals of software products** especially applications with complex features.
- Literate programming is **useful for programs of all sizes**.
- Literate programming encourages **meaningful documentation** and the inclusion of details that are usually omitted in source code such as the
 - description of algorithms,
 - design decisions,
 - and implementation strategy.

LarLib Literate Programming Environment

Remote repository

LINK: <https://github.com/cvdlab/lar-cc>

cvdlab / lar-cc

Unwatch

10

Star

1

Fork

2

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Pulse

Graphs

Settings

Linear Algebraic Representation to Compute with Cellular (Co)Chains

Edit

Add topics

305 commits

3 branches

0 releases

3 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

plasm-language	update faces2cycles	Latest commit fbb1bdf on Jun 24, 2016
doc	extraction of cycles from surfaces	9 months ago
larlib	extraction of cycles from surfaces	9 months ago
src	update faces2cycles	9 months ago
test	bug removed from spacePartiton and bruteForceIntersect inserted	10 months ago
.gitignore	First commit	3 years ago
Makefile	bug removed from spacePartiton and bruteForceIntersect inserted	10 months ago
README.md	proved correctness of non-signed boundary operator	a year ago
TODO	added Lar structures	3 years ago

Literate programming environment

Grab the environment description: [frame.pdf](#)

The screenshot shows a GitHub repository interface for 'cvdlab / lar-cc'. The repository has 10 stars, 1 fork, and 2 forks. The 'Code' tab is selected, showing the file 'frame.pdf' in the 'doc' directory. The file is 292 KB and was last modified on Jan 20, 2015. The file content is a LaTeX document titled 'Literate programming IDE for LAR-CC *' by Alberto Paoluzzi, dated January 20, 2015.

cvdlab / lar-cc

Unwatch 10 Star 1 Fork 2

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Branch: master lar-cc / doc / pdf / frame.pdf Find file Copy path

plasm-language crumbs b415e23 on Jan 20, 2015

2 contributors

292 KB Download History

Literate programming IDE for LAR-CC *

Alberto Paoluzzi

January 20, 2015

Figure 2: The literate programming environment for `_larlib_`

Template to add YOUR module

- 1 Clone the repository on the local machine

```
$ git clone https://github.com/cvdlab/lar-cc.git
```

- 1 Grab the file: `template.tex` and complete by updating some fields:
 - Title <- moduleName
 - TheAuthor <- yourName
 - Date <- theDate
 - template (bib.bib file)
- 2 of course, save as `moduleName.tex` within the `src/tex/` folder of local repository

Literate programming for Julia

Slide_1

Slide_2

Slide_3

References