

# 曹星明

男, 43岁, 籍贯: 湖南益阳, 居地: 河南信阳  
(+86) 135-2397-8101 [simeon.chaos@gmail.com](mailto:simeon.chaos@gmail.com)

## 学历

1989-1993: 国防科技大学计算机系本科

1994-1997: 空军雷达学院硕士 (人工智能)

## 工作

1993-1994, 1997-2009: 空军第一航空学院

2010-2013: 全职研究自然语言编程

## 我的项目

公开发布的项目

peasy  
taijilang: 太极语言

dao: 兼具lisp和prolog的特性, 综合逻辑范式和函数范式于一体的解释器和编译器, 基于continuation, 内置解析器。  
python实现 node.js实现 (使用coffee-script语言编写)

twoside  
killsola

splitter  
seer

pyshin  
hopen

学习和使用过的web开发技术:

未公开的项目

expressjs: hopen中用它作为服务器端框架

angularjs: hopen的客户端框架

jade: 最喜欢的模板语言。hopen就用它。可能有人注意到了: 我就是喜欢缩进风格的东西: py, coffee, jade, taijilang。感谢 Guido, Jeremy, TJ.  
web相关测试技术  
qunit: 在 hopen项目使用了  
seer项目 目使用了  
中使用了 karma, mocha, jasmine, qunit测试 chaijs, protractor 框架  
Meteor: 看了一些文档, 读了一些代码, 好复杂, 目前一点都没搞懂。  
Derbyjs: 这个要简单多了。但就是比Meteor发展慢, 流行度低。  
zope: web框架的鼻祖, python web框架的庞然大物, 所谓的 where zope lead, python follow. 很令人遗憾地看到它象夕阳般离去.....  
web2py: 很好学好用的框架, 遵循python带电池的哲学, 用它部分地开发了seer。但是不得不说: django还是python界的主流, 特别是在中国。

jquery: web开发者都知道

css预处理: sass

持续集成  
peasy项目使用  
travis.com实现了持续集成

工作流自动化  
gruntjs: gulpjs:  
hopen项目 因为gulp项目使用 的优点, grunt实现 在流程的自peasy, 自动化 twoside等项目中转换到了 gulpjs

长期使用的语言

C/C++: python: javascript:  
1993年- 2007年- 2011年至2007年主2011年主今主要使用的要使用的用的语言。 语言。 言。  
coffee-script: 我通过

熟悉的语言

lisp: scheme: haskell:  
2000年精研读过 学习过三次, 但是一年, 非SICP,从很丢脸的常喜欢它中对于编是, 至今的优雅和程得到了没有掌握强大, 然最多的启monad, 而对于以发 说明我确

了解过但是没有深入的语言  
两门当前最流行也是web开发最通用的两门语言, 也曾经偶尔试图进入, 但是纯粹因为个人喜好没有深入:

java: 一 php: 一方面 是觉直没有吸得它太罗引我。  
嗦 (语法

学习过的编程语言

coffee-script提高  
javascript  
开发效率。

下流行的  
看法：  
lisp的强  
大和优美  
是因为它赏它的优  
采用了 雅精简。  
sexpression受它影  
作为语 响，太极  
法，或者语言最开  
说直接使始使用[]  
用抽象语作为代码  
法作为具块定界  
体语 符，发布  
法，lisp 前才改成  
的宏及其使用更通  
将代码作用的{}。  
为数据的另外，太  
能力必须极语言有  
依赖于上比rebol的  
述特征，parse功能  
我一直有更方便的  
不同的观定制dsl的  
点。我认能力。

为，lisp  
的语法是  
因为受限  
于当时的  
条件（主  
要是语法  
解析技术  
方面，但  
不限于  
此）自然  
产生的结  
果，并在  
此种背景  
下产生了  
上述迷  
思，即使  
lisp最社  
区最核心  
的人物，  
包括  
McCarthy  
也有可能  
持有这种  
迷思。太  
极语言的  
实现打破  
了上述迷  
思，同时  
也实现和  
证明了我  
的观点。  
因为太极  
语言在外  
部语法上  
和其它高  
级语言同  
样灵活，  
但是在内  
核上对待  
程序的内部表示比  
lisp自身  
更为  
lisp：从  
源码到目  
标代码的  
转换过程  
中，程序  
始终保持  
嵌套列表

实是一个  
数学的  
loser。

繁琐，文  
件夹嵌套  
太多层  
次），另  
一方面可  
能是我开  
始了解的  
时候java  
生态太庞  
杂以至于  
我不知道  
从何入  
手。

其它了解过的语言：

smalltalk: J语言: 一  
安装过 门简洁的  
squeak, 数学家的  
感受过它语言。谁  
独特的风懂R语言  
格和OO 的，能不  
的风格。能解释一  
下为什么  
R语言现  
在能红起  
来，J语  
言依然很  
寂寞？

道德经：我必须把它放在第一位。说起来别人可能不相信，我从道德经得到了最多的软件设计和编程开发方面的心得感悟，请注意，都是直接关于软件和编程，而不是关于人生或者哲学。  
测试驱动开发（Kent Beck）：对于提高我的编程效率帮助最大的一本书。

拥抱极限编程(Kent Beck): 确实里面有很多东西都已经成为软件界以及我个人的日常实践。

人工智能手册

unixEric S.Raymond 很开眼界，很受启发

编译原理（龙书）第四版：我估计只能理解其中的百分之十，太极语言的实现大概只用到了其中的百分之一。

语言学教程：北京大学 胡壮麟等  
自然语言处理综论：Daniel

Jurafsky,James H.Martin等

语法哲学：丹麦

代码大全：长期的经典

on lisp: Paul Graham. The root of lisp  
的作者，在这本书里YCombinator  
的创立者对于宏有很深入的解  
说，

C++设计新思维-泛型编程与设计  
模式之应用 看的时候很震撼：原  
来C++还可以这样玩。后来用  
python的时候又很感叹：在一种环  
境下需要殚精竭虑的问题换个环  
境居然都不再成为问题。

《TCP/IP 详解卷一：协议》：我  
最不会的就是协议了。看了这本  
书的部分内容，对于各种协议我  
依然是一塌糊涂。

Lisp In Small Pieces: 我设计dao的时  
候，有什么lisp原语不会实现了，  
就到这本书里查代码。

重构--改善既有代码的设计 Martin  
Flower, 虽然我很少照搬这本书里  
面的重构处方，但是我越来越有

读过的一些  
书：

的形式，并且符合json的规范。因此，太极语言不但实现了完全可以和lisp相比的宏，而且在此基础上更进一步，每个程序都是自身的数据，每个程序都是自身的元程序，每个都通过运行自己的元程序产生自己的最终形式。

一个强烈体会：值得写很多遍的代码才是有价值的代码。我为了实现解析左递归，曾经重写过不下几十次，最终我实现了Peasy。然后在Peasy的基础上，我又实现了tajjilang，而在实现tajjilang时，解析器部分曾经经历了很多次涉及整体的重构，包括使用对象方法、函数方法解析缩进，使用基于token的方法和不用token的处理词法，最终才实现了一个健壮、模块化，可扩展的动态解析器。HTML & XHTML\_ The Definitive Guide: 要开发web应用，oreilly的这基本书应该是必读书目The Definitive Guide to javascriptCSS: The Definitive Guide

## 自我描述

本人性格目标明确，专注执着以至于偏执。对于已然认定之事能始终保持极大热情，甚至不惜牺牲大多数人看重的功名利禄。因为前一点曾被前辈评价为“有激情”，因为后一点而做出的某些决策也曾让人觉得极其不可思议。这些方面一直明显地体现在我的经历当中。

我与计算机领域结缘于1983年国防科大发明银河计算机的新闻，这新闻令我好奇计算机和人类思维的关系。后来我有幸因此而选择了学习和从事计算机专业。

1995年硕士选题时，我因为个人对编程的某些感悟而选择了自然语言编程作为研究对象，并将这一个兴趣保持至今。虽然当时纯粹是以探索性研究作为初衷，在坚持十多年后我多年的积累终于得到了很多意想不到的回报。互联网初兴之时，本人因偏于军队院校，业余时间又埋头研究自然语言编程相关的理论，一直对互联网保持着不敏感的状态。直到2007年，因为偶然的时机我才开始与互联网结缘，也打开了我的互联网思维，我领悟到无线和移动互联网将更加凶猛和广阔地渗透和冲击社会和生活。同时，我也觉察到，伴随着移动互联网，自然语言处理已经处于成熟的前夕，必将爆发巨大的实用价值，我也开始从互联网和社会化的角度重新思考自然语言编程。

发现和创造伴随着很多偶然的机遇。我从95年开始探索自然语言编程，开头十多年一直象牛一样不停寻找草料，咀嚼，消化，吸收，但是并未曾获得明确的心得感悟，更别提有什么重要发现和创造。直到2007年以后，长期的耕耘终于产出了一些果实。因为学习了大量自然语言处理方面知识和技术，同时广泛地了解和试验各种计算机语言及其设计思想、编译技术，我在这方面也有了自己的技术。2011年-2012年，我设计出了兼具lisp和prolog特性，综合逻辑范式和函数范式于一体的dao系统。我一直希望编程语言能够具有动态定制语法的能力，为实现此功能，我于2013年发现了一种新的解析器设计方案，发布了Peasy。最终，2012年，我在帮助十岁的儿子寻找更合适的编程软件时，面对孩子已经玩过很久的MIT的scratch，脑海中突然如同天启一般呈现出了我一直苦苦寻求的自然语言编程系统的整体设计思想。在基于这种设计思想实现该系统原型的过程中，命运之神又额外赐予我另一个成果：在今年3，4月份，我在使用各种web开发工具的时候，偶然地思考什么样的web框架是更好的框架。angularjs的双向绑定非常便利，我想到，如果双向绑定能够跨越前后端，应该会更加强大。我开始考虑能否自己设计一种这样的web框架。我首先的思路是要有一种合适的模板技术，我当时使用的是jade，它并没有提供我设想的这种功能。我开始比较我见到过的其它各种模板，也搜索互联网寻找其它相关的技术。然后我偶然地遇到了lispyscript这个项目，它的实现直接触发了我的思路。我在原来实现dao的基础上很快实现了太极语言。太极语言基本上实现了我个人关于高级语言的全部期望，同时，在设计过程中，我又非常偶然而幸运地得到了一个意向不到的重大发现：强大的元编程技术。综合以下三种创新：习惯语法结合lisp内核，动态语法，元编程技术，太极语言可以说是高级语言领域自1958年John McCarthy发明lisp以来的最重大进步。虽然太极语言还刚刚问世，但是我相信它会有广阔的空间和光明的前景。

在以互联网为基点探索和实现我所理解的自然语言编程系统的过程中，我接触和掌握了大量web相关的技术。最终，我也发现和自信已经对于新的技术趋势培养出了比较强的预见性、洞察力和敏感度。我于2007年开始接触python，为其简洁优雅所吸引，python后来一直为我的研究带来助益，同时我也得以了解LAMP开源范式，熟悉了zope，web2py等框架，发现了nlTK工具包。在python环境下，我实现了一个基于web的通用信息平台的原型(seer)。07年左右，当mootools和prototype还更流行的时候，我刚接触jquery即惊叹于它的优美设计，认定它将大兴于世。12年，在nodejs成熟和angular.js兴起的阶段，我适时切入这两个技术，掌握了MEAN(mongodb, expressjs, angularjs, nodejs)范式，并在此基础上完成了我的自然语言编程系统，该系统充分运用了MEAN全栈技术，使用了tab, tree, splitter, menu等诸多控件，并尝试多种开发方法，对比了使用和不使用angularjs，在angularjs框架下使用或不使用jquery，也试

验了部分bootstrap 控件，最终当发现某些现成控件带来诸多不必要的复杂性后，大部分控件都换成了自己的简单实现。<https://github.com/chaosim/splitter>控件是在这个过程我实现的一个简单的多窗格分隔条的简单开源项目。同时，在考察和试用了requirejs, seajs和browserify之后，我最终在所有的前后端项目采用的都是我自己实现的一个简单工具，该工具只有约100行，1kb，能够让模块同时运行在前端和后端。javascript 发布在<https://github.com/chaosim/twoside>，配合此工具，我也开发了一个gulpjs的插件，gulp-twoside, 发布在<https://www.npmjs.org/package/gulp-twoside>。与此同时，我也被漂亮的coffee-script所吸引，coffee大大提高了我的开发效率。