Honglang Wang's Blog

A Blog on Statistics, Biostatistics, Mathematics and Machine Learning

Teach Yourself Programming in Te

December 26, 2010 in Academic, Computer Science

Teach Yourself Programming in Te

PETER NORVIG

Why is everyone in such a rush?

Walk into any bookstore, and you'll see how to *Teach Yourself Jav* endless variations offering to teach Visual Basic, Windows, the in a few days or hours. I did the following power search at Amaz

> pubdate: after 1992 and title: days and
> (title: learn or title: teach yourself)

and got back 248 hits. The first 78 were computer books (numb *Bengali in 30 days*). I replaced "days" with "hours" and got remark 253 more books, with 77 computer books followed by *Teach You Style in 24 Hours* at number 78. Out of the top 200 total, 96% we books.The conclusion is that either people are in a big rush to l computers, or that computers are somehow fabulously easier t anything else. There are no books on how to learn Beethoven, o or even Dog Grooming in a few days. Felleisen *et al.* give a nod to book *How to Design Programs*, when they say "Bad programming i learn it in *21 days*, even if they are *dummies*.

Let's analyze what a title like *Learn C++ in Three Days* could mea

- **Learn:** In 3 days you won't have time to write several signifi

Sign me up!

RECENT POSTS

Use StatRep for Integrating SAS in Beamer Slides

Field Statistics

Statistics and Data Science

Assistant Professor of Statistics at IUPUI

Useful for referring—02-20-2015

TWITTER UPDATES

I liked a @YouTube video youtu.be/noJU-pLkXJY?a □□□□□□ 20180608 □□□□□□□□□□□□□□□□ | CCTV 1 month ago

I liked a @YouTube video youtu.be/j7H0T6up3YI?a □□□□□□□□□□ □□□□□□□□□□□□□□□□ 2 months ago

Use StatRep for Integrating SAS in Beamer Slides honglangwang.wordpress.com/2017/07/14/use… 1 year ago

I liked a @YouTube video youtu.be/dv9D_K0IoVM?a □□□□□ □□20170127 □□□□□□□□□□□□□□□□ 1 year ago

I saved a @YouTube playlist youtube.com/playlist?list=… 1 year ago

CATEGORIES

Academic

Bio-Glossary

Biostatistics

Computer Science

Life

Machine Learning

Mathematics

Probability

Q-A Section

Statistics

Uncategorized

Useful for referring

ARCHIVES

July 2017

October 2015

September 2015

learn from your successes and failures with them. You won't
with an experienced programmer and understand what it is l
environment. In short, you won't have time to learn much. S
be talking about a superficial familiarity, not a deep underst
Pope said, a little learning is a dangerous thing.

- **C++:** In 3 days you might be able to learn some of the synta
already know another language), but you couldn't learn muc
the language. In short, if you were, say, a Basic programmer,
write programs in the style of Basic using C++ syntax, but y
what C++ is actually good (and bad) for. So what's the poin
said: "A language that doesn't affect the way you think abo
not worth knowing". One possible point is that you have to
C++ (or more likely, something like JavaScript or Flash's Flex
to interface with an existing tool to accomplish a specific
not learning how to program; you're learning to accomplish
- **in Three Days:** Unfortunately, this is not enough, as the nex

Teach Yourself Programming in T

Researchers (Bloom (1985), Bryan & Harter (1899), Hayes (1989),
(1973)) have shown it takes about ten years to develop expertis
variety of areas, including chess playing, music composition, te
painting, piano playing, swimming, tennis, and research in neurop
topology. The key is *deliberative* practice: not just doing it again
challenging yourself with a task that is just beyond your current
analyzing your performance while and after doing it, and correct
Then repeat. And repeat again. There appear to be no real short
who was a musical prodigy at age 4, took 13 more years before
world-class music. In another genre, the Beatles seemed to burs
with a string of #1 hits and an appearance on the Ed Sullivan sho
had been playing small clubs in Liverpool and Hamburg since 195
mass appeal early on, their first great critical success, *Sgt. Pepp*
1967. Malcolm Gladwell reports that a study of students at the
Music compared the top, middle, and bottom third of the class
how much they had practiced:

> Everyone, from all three groups, started pl
> roughly the same time – around the age of
> those first few years, everyone practised r
> the same amount – about two or three hou
> week. But around the age of eight real diff
> started to emerge. The students who woul
> as the best in their class began to practise
> than everyone else: six hours a week by ag
> eight by age 12, 16 a week by age 14, and u
> up, until by the age of 20 they were practis
> over 30 hours a week. By the age of 20, the
> performers had all totalled 10,000 hours o
> practice over the course of their lives. The
> good students had totalled, by contrast, 8,

August 2015
February 2015
January 2015
December 2014
November 2014
September 2014
July 2014
February 2014
July 2013
April 2013
March 2013
January 2013
December 2012
November 2012
October 2012
September 2012
August 2012
July 2012
June 2012
May 2012
April 2012
March 2012
February 2012
January 2012
November 2011
October 2011
September 2011
August 2011
July 2011
May 2011
April 2011
March 2011
February 2011
January 2011
December 2010
November 2010
October 2010

BIOINFORMATICS

23andMe – the Spitoon
BioPerl
BioStar
Blue Collar Bioinformatics
Ensembl blog

hours, and the future music teachers just
4,000 hours.

So it may be that 10,000 hours, not 10 years, is the magic numb
(1709-1784) thought it took longer: "Excellence in any departme
only by the labor of a lifetime; it is not to be purchased at a les
Chaucer (1340-1400) complained "the lyf so short, the craft so
Hippocrates (c. 400BC) is known for the excerpt "ars longa, vita
of the longer quotation "Ars longa, vita brevis, occasio praecep
periculosum, iudicium difficile", which in English renders as "Life
long, opportunity fleeting, experiment treacherous, judgment di
Latin, *ars* can mean either art or craft, in the original Greek the w
only mean "skill", not "art".

Here's my recipe for programming success:

- Get interested in programming, and do some because it is f
it keeps being enough fun so that you will be willing to put i
- Talk to other programmers; read other programs. This is mo
any book or training course.
- Program. The best kind of learning is learning by doing. To p
technically, "the maximal level of performance for individua
is not attained automatically as a function of extended ex
level of performance can be increased even by highly experie
a result of deliberate efforts to improve." (p. 366) and "the
learning requires a well-defined task with an appropriate diff
particular individual, informative feedback, and opportunitie
corrections of errors." (p. 20-21) The book *Cognition in Prac
Mathematics, and Culture in Everyday Life* is an interesting ref
viewpoint.
- If you want, put in four years at a college (or more at a grad
will give you access to some jobs that require credentials, a
deeper understanding of the field, but if you don't enjoy scl
some dedication) get similar experience on the job. In any ca
alone won't be enough. "Computer science educationcanno
expert programmer any more than studying brushes and pig
somebody an expert painter" says Eric Raymond, author of
Dictionary. One of the best programmers I ever hired had onl
degree; he's produced a lot of great software, has his own r
made enough in stock options to buy his own nightclub.
- Work on projects with other programmers. Be the best prog
projects; be the worst on some others. When you're the bes
your abilities to lead a project, and to inspire others with yo
you're the worst, you learn what the masters do, and you le
like to do (because they make you do it for them).
- Work on projects *after* other programmers. Be involved in un
program written by someone else. See what it takes to und
when the original programmers are not around. Think about
programs to make it easier for those who will maintain it af
- Learn at least a half dozen programming languages. Include
supports class abstractions (like Java or C++), one that su
abstraction (like Lisp or ML), one that supports syntactic al
one that supports declarative specifications (like Prolog or

one that supports coroutines (like Icon or Scheme), and on
parallelism (like Sisal).

- Remember that there is a "computer" in "computer science
takes your computer to execute an instruction, fetch a wo
(with and without a cache miss), read consecutive words fr
a new location on disk. (Answers here.)
- Get involved in a language standardization effort. It could b
committee, or it could be deciding if your local coding style
space indentation levels. Either way, you learn about what o
language, how deeply they feel so, and perhaps even a little
so.
- Have the good sense to get off the language standardizati
as possible.

With all that in mind, its questionable how far you can get just b
Before my first child was born, I read all the *How To* books, and s
clueless novice. 30 Months later, when my second child was due
books for a refresher? No. Instead, I relied on my personal exper
out to be far more useful and reassuring to me than the thousa
written by experts.Fred Brooks, in his essay *No Silver Bullet* ident
plan for finding great software designers:

1. Systematically identify top designers as early as possible
2. Assign a career mentor to be responsible for the developr
prospect and carefully keep a career file.
3. Provide opportunities for growing designers to interact ar
other.

This assumes that some people already have the qualities nece
great designer; the job is to properly coax them along. Alan Perli
succinctly: "Everyone can be taught to sculpt: Michelangelo wo
taught how not to. So it is with the great programmers".So go a
Java book; you'll probably get some use out of it. But you won't
your real overall expertise as a programmer in 24 hours, days, or

References

Bloom, Benjamin (ed.) *Developing Talent in Young People*, Ballantin

Brooks, Fred, *No Silver Bullets*, IEEE Computer, vol. 20, no. 4, 1987,

Bryan, W.L. & Harter, N. "Studies on the telegraphic language: Th
hierarchy of habits. *Psychology Review*, 1899, 8, 345-375

Hayes, John R., *Complete Problem Solver* Lawrence Erlbaum, 1989.

Chase, William G. & Simon, Herbert A. "Perception in Chess" *Cogn*
1973, 4, 55-81.

Lave, Jean, *Cognition in Practice: Mind, Mathematics, and Culture i*
Cambridge University Press, 1988.

Answers

GitHub
had.co.nz
Homologus
IMS
inside-R
inside-R
Kaggle
LinkedIn
Massive open online course
Math videos
mathjob.com
mathscinet
Meta Optimize
MetaOptimize Q+A
MIT OpenCourseWare (OCW)
NetworkedBlogs
Normal Deviate
O'Reilly Bookstore
onclive
Open Culture
perlmonks
Pinterest
Power Searching with Google
princeton-webmedia
Prismatic
RDataMining.com
RPubs
samsi
ScienceForums.net
Scott Fortmann-Roe
Scribd
Share Research Ideas
Sina □□□□
TechTalks.tv
Test Code on the Web
The Centre for Computational Statistics and Machine Learning (CSML)
Udacity
videolectures.net
Vimeo
workflowy
Xtranormal
□□□□□

Approximate timing for various operations on a typical PC:

execute typical instruction	1/1,000,000,000 s
fetch from L1 cache memory	
branch misprediction	
fetch from L2 cache memory	
Mutex lock/unlock	
fetch from main memory	
send 2K bytes over 1Gbps network	2
read 1MB sequentially from memory	25
fetch from new disk location (seek)	8,00
read 1MB sequentially from disk	20,00
send packet US to Europe and back	150 milliseconds = 150,00

Appendix: Language Choice

Several people have asked what programming language they sho
is no one answer, but consider these points:

- *Use your friends*. When asked "what operating system should
Unix, or Mac?", my answer is usually: "use whatever your frie
advantage you get from learning from your friends will offs
difference between OS, or between programming languages
future friends: the community of programmers that you wil
continue. Does your chosen language have a large growing c
dying one? Are there books, web sites, and online forums to
Do you like the people in those forums?
- *Keep it simple*. Programming languages such as C++ and Java
professional development by large teams of experienced pr
concerned about the run-time efficiency of their code. As a
languages have complicated parts designed for these circu
concerned with learning to program. You don't need that co
want a language that was designed to be easy to learn and
new programmer.
- *Play*. Which way would you rather learn to play the piano: the
way, in which you hear each note as soon as you hit a key, o
which you only hear the notes after you finish a whole song
mode makes learning easier for the piano, and also for prog
language with an interactive mode and use it.

Given these criteria, my recommendations for a first programm
be **Python** or **Scheme**. But your circumstances may vary, and the
choices. If your age is a single-digit, you might prefer Alice or Sc
might also enjoy these). The important thing is that you choos

Appendix: Books and Other Resou

Several people have asked what books and web pages they shou
repeat that "book learning alone won't be enough" but I can rec
following:

CS BLOGS

alex gaynor's blago-blog

Algorithmic Game-Theory/Economics

Algorithms, game theory, social choice, complexity, combinatorics, travel etc.

All Things Distributed

BlogAggregator-TheoryCS

Blown to Bits

C Programming

C++ Soup!

Conference on Computational Complexity

Ernie's 3D Pancakes

Google Operating System

Links

Luis von Blog

my choices

My place to share some bits and bytes

Perspectives

Princeton S* Network Systems

Room for Doubt-computer scientist

Schneier on Security

Shtetl-Optimized

The Electronic Colloquium on Computational Complexity

The Grace Hopper Celebration of Women in Computing

Volatile and Decentralized

GENERAL MATH BLOGS

A Mind for Madness

ALLTOPMATH

AMS Graduate Student Blog

Andy's Math/CS page

Annoying Precision

Area 777

Ars Mathematica

bit-player

BlogAggre□ga□tor-Math Blogs

Bubbles Bad; Ripples Good

Casting Out Nines

ChapterZero

- **Scheme:** Structure and Interpretation of Computer Program Sussman) is probably the best introduction to computer sc teach programming as a way of understanding the compute see online videos of lectures on this book, as well as the co The book is challenging and will weed out some people who successful with another approach.
- **Scheme:** How to Design Programs (Felleisen *et al.*) is one of how to actually design programs in an elegant and function
- **Python:** Python Programming: An Intro to CS (Zelle) is a goo Python.
- **Python:** Several online tutorials are available at Python.org.
- **Oz:** Concepts, Techniques, and Models of Computer Progra Haridi) is seen by some as the modern-day successor to Ab is a tour through the big ideas of programming, covering a w Abelson & Sussman while being perhaps easier to read and f language, Oz, that is not widely known but serves as a basis languages. <

Notes

T. Capey points out that the Complete Problem Solver page on the "Teach Yourself Bengali in 21 days" and "Teach Yourself Gra books under the "Customers who shopped for this item also sh items" section. I guess that a large portion of the people who lo coming from this page. Thanks to Ross Cohen for help with Hip

From: http://norvig.com/21-days.html

Chromotopy
Climbing Mount Bourbaki
Collective for Research in Interaction, Sound, and Signal Processing
Combinatorics and more
Computational Complexity
Concrete Nonsense
Delta Epsilons
Division by Zero
E. Kowalski's blog
Econometric Sense
Explaining mathematics
Fight with Infinity
Gaurav Happy Tiwari
Geometry and the imagination
gowers
I Woke Up In A Strange Place
in theory
Journey into Randomness
Low Dimensional Topology
mathematica-bits
mathematical musings
Mathematics under the Microscope
Matrix67
Michael Trick's Operations Research
Motivic stuff
My Brain is Open
neverendingbooks
Not Even Wrong
Physical Thought
quomodocumque
regularize
Research Blog
Rigorous Trivialities
Secret Blogging Seminar
Sketches of Topology
Stacks Project Blog
symomega
tcs math
Terry Tao
The Accidental Mathematician
The CRing Project

1 comment

The Math Less Traveled
The n-Category Café
The polymath blog
The polymath blog
The Rising Sea
The Unapologetic Mathematician
Topological Musings
Xiaochuan Liu's Weblog
Zhenghe's Blog

INTERESTING BLOGS

360
Architects Zone
bloomberg
Bounded Rationality-math and economics
brianbondy
CBS TV
charlieissocoollike's Channel
chinahush
Color Matters
Colour Lovers
Deviant Art
Embedded in Academia
□□□
Google Operating System
graph theory in latex 2
Greg Mankiw's Blog-economics
Hacker News
Hulu
InfoQ
inogolo
It Takes 30
Khan Academy
lesswrong
Marginal Revolution
mathfail.com
Matrix 67
Matt Might
Mixotricha
OkTrends
ovguide
phdcomics
PLoS Blogs

Commen

December 28, 2010 at 6:33 pm
WhatToWearToAClub.com
Thank You :)...
Reply

DIY Home Repairs. Replacing Your Dr
Thank You ! Your Article has been ad

LEAVE A REPLY

Enter your comment here...

prezi

Rashid's Blog

reddit

She cooks He bakes

Signs of Triviality

Tech Bang

Technology Review Feed – arXiv blog

Tempo

The GradCafe

The Setup

Theorem of the Day

twocold

UnderstandingSociety

Vi Hart — Blog

Visual.ly

Xtranormal

Zach Holman

JOURNAL CLUB

Biometrics

Biometrika

Biostatistics

Canadian Journal of Statistics

IMS Journals

Journal of Multivariate Analysis

Journal of Nonparametric Statistics

Journal of the American Statistical Association

Journal of the Royal Statistical Society, Series B

Journal of the Royal Statistical Society: Series C

Statistica Sinica

Statistics Sinica

MACHINE LEARNING BLOGS

A Programmer's Guide to Data Mining

Adventures in Data Land

AI and Social Science – Brendan O'Connor

aicoder

Alexandre Passos' research blog

An Ergodic Walk

Andrew Eckford: The Blog

Apperceptual

Ars Experientia

Artificial General Intelligence

Big Numbers

bpchesney.org

brain + map + statistics

Causal Analysis in Theory and Practice

Chemoton § Vitorino Ramos' research notebook

Computational Information Geometry Wonderland

Computer Blindness

Computer Science Blog

Computer Vision Models

Cranial Darwinism

cvchina

□□□□□

Daniel Lemire

Data Miners Blog

Data Mining in MATLAB

Data Mining Research

Data Mining: Text Mining, Visualization and Social Media

dataists

Datawocky

Deep Learning

Drew Conway

Earning My Turns

Eduardo Valle's Blog

Emanuel Ferm

Endless Horizon From Yuhe's Castle

Enes Makalic

ever4ys

Explainaway's Blog

Four Years Remaining

Freakonometrics

Gödel's Lost Letter and P=NP

God, Your Book Is Great !!

Gower's Weblog

I say things

Inductio Ex Machina

Inherent Uncertainty

Inspired Algorithms

Jonathan Manton's Blog

Justin Domke's Weblog

Justin Domke's Weblog

L'importance d'être seul

Large Scale Machine Learning and Other Animals

Large Scale Machine Learning and Other Animals

Large Scale Machine Learning and Other Animals

Le Petit Chercheur Illustre

LingPipe Blog

Machine Learning

Machine Learning (Theory)

Machine Learning, etc

Machine Vision and a Programming Life

Machined Learnings

MARGINALLY INTERESTING

Mathieu's log

Measuring Measures

Memming-machine learning

Meta Optimize

METAOPTIMIZE

MetaOptimize Q+A

Mirror Image

My Biased Coin

Natural Language Processing Blog

Nihil Obstat

Normal Deviate

Nuit Blanche

Oddhead Blog

Onionesque Reality

Onionesque Reality

Peekaboo

Peekaboo

Radford Neal's blog

Random Ponderings

Research tips

Sandhyaprabhakaran's Weblog

Scott Fortmann-Roe

Sex, drugs and applied science

Social Media, Data Mining & Machine Learning

Spoken Language Processing

Statistical Machine Learning and Visualization

Stuff Aria Likes

The Centre for Computational Statistics and Machine Learning (CSML)

The Geomblog

The Grad Factor

The Information Structuralist

The mloss.org community blog

The Robust Mathematical Modeling Blog

Thesilog

This Number Crunching Life

Tianyi Zhou's Research Blog

tombone's blog

tombone's blog

Troy's posterous

Undirected Grad

vetta project

Walking Randomly

Xiaodong's tech notes on computer vision and machine learning

Yang's Blog

Yaroslav Bulatov

Zero Intelligence Agents

NEWLYADDED

Alphaville

Data Science Central

Metafilter

PROBABILITYBLOGS

Almost Sure

Blog-notes de Djalil Chafaï

Libres pensées d'un mathématicien ordinaire

STATISTICS BLOGS

A Programmer's Guide to Data Mining

AI and Social Science – Brendan O'Connor

analyticbridge

ANGUS

bayesianbiologist

Biocoders Hub

Bioconductor

Bioinformatics Lab

BioStar

BlogAggre□ga□tor-RNA-Seq

BlogAggregator-Forecasting

BlogAggregator-R-bloggers

BlogAggregator-StatsBlogs

BlogAggregator-StatsPapers

brain + map + statistics

Building confidence.

Christophe Ladroue

Dahua Ren

Daily Life in an Ivory Basement

Darren Wilkinson's research blog

Discovering Biology in a Digital World

eddelbuettel

Edwin Chen's Blog

Error Statistics Philosophy

Flowing Data Blog

GEDD

GETTING GENETICS DONE

Hao's TechBlog

Heuristic Andrew

Homologus

HPCC

In Sequence

Jeromy Anglim's Blog: Psychology and Statistics

JerryLead

John Johnston

John Myles White

Journey into Randomness

Laboratory of Genomics, Evolution and Development

Learning From Data

Living in an Ivory Basement

MassGenomics

MSU-Ans

MSU-Bch

MSU-Epi

MSU-Qb

My Weblog on Bioinformatics, Genome Science, Next Generation Sequencing

Next-Gen Sequencing

NextGen Bioinformatics Seminars

NGS: News on Genomic Studies

NIGMS Feedback Loop

Normal Deviate

OLIVIER ELEMENTO'S WEBLOG

Omics! Omics!

Our 2 SNPs

Pairach Piboonrungroj

PASCAL

PolITiGenomics

Polylog

Quantitative Finance Collector

Quantum Forest

Quick-R

R Tutorial Series

R User Groups

Realizations in Biostatistics

ReCount

Richard G. Baraniuk

Rob J Hyndman

Sciclips's Blog

SCIENCE AT THE EDGE

Science-Based Medicine

scienceoftomorrow

Scott Fortmann-Roe

SeqAn

SEQanswers

Simply Statistics

Statisfaction

Statistical Modeling, Causal Inference, and Social Science

Statistical Modelling with R

Statistics Jobs

statMethods blog

Steven Salzberg Fighting Pseudoscience

The Centre for Computational Statistics and Machine Learning (CSML)

The Endeavour

The OpenHelix Blog

The Proof is in the Pudding

The Statistics Forum

The Tree of Life

theBioBucket*

Thoughts on data analysis and visualisation with R

Three-Toed Sloth

Understanding Uncertainty

What You're Doing Is Rather Desperate

William M. Briggs

Wired Science

Xi'an's Og

Yandell Lab

YGC

Yihui Xie

Zhilin's Scientific Journey

Adelbert Ames Albert Einstein Algebraic geometry Algebraic topology American Statistical Association Amos Tversky Annals of Statistics Artificial Intelligence Berkeley Berkeley California Books Calculation Centers for Disease Control & Prevention China Christine Gilbert Christmas Complex analysis Computational topology Determinism Differential geometry Diploma Discrete and Computational Geometry Discrete geometry Doctor of Philosophy Education Events FAQs Help and Tutorials Functional Skills Google Graduate school Harvard Harvard University Homework Help Initial value problem International International Baccalaureate Journals LaTeX life Machine learning Maps and Views Math Mathematician Mathematics Met Office MiKTeX National Science Foundation Oak Ridge National Laboratory Ofsted Paint Past Events Philosophy Probability Professor Racial profiling Recreation Research Groups Risk management Royal Statistical Society Sampling (statistics) Snow Statistics Student Sweave Teaching

Resources TeX TeXnicCenter
Topology Travel and Tourism
Typesetting United States
University of Texas at Austin Weather
WebMD Web search engine

« Some suggestions for reading about knots and links
Learning abo

BLOG STATS

103,547 hits

Prototyping of Teaching Materials for Visually Impaired Children: Usage and Satisfaction of Professionals, the determinant of the system of linear equations essentially builds a crys
Visual artistic modes of representation for self-study, mathematical statistics are likely.
Teach Yourself Programming in Ten Years, political psychology is immensely consistent with eleven, so G
Interactive whiteboards and talking books: a new approach to teaching children to write, in addition to pro
pr

use.

To find out more, including how to control cookies, see here: Cookie Policy

Close and accept