



**Technology \* Economics \* Learning**

# **Becoming a Programmer?**



Finding a Economic Learning Path to Investment Return

# Programming = Mucho Dinero?


- My observations as an armchair economist:
- There are throngs of people studying programming and data science but there is yet an unsatisfied demand for qualified hires, because the complexity of technology outpaces the self-capitalization rate of many learners.
- It is hard to get the experience you need without already working on an enterprise technology project
- The number of **dependencies** in modern software development put a very large startup cost in front of the learner.


# It seems so easy





 **DHH**  @dhh · Jul 25, 2014

The original "How to build a blog in 15 minutes with Rails" from 2005 aka the wups video: [youtu.be/Gzj723LkRJY](http://youtu.be/Gzj723LkRJY) #RailsTurns10

 Ruby on Rails demo Watch later Share

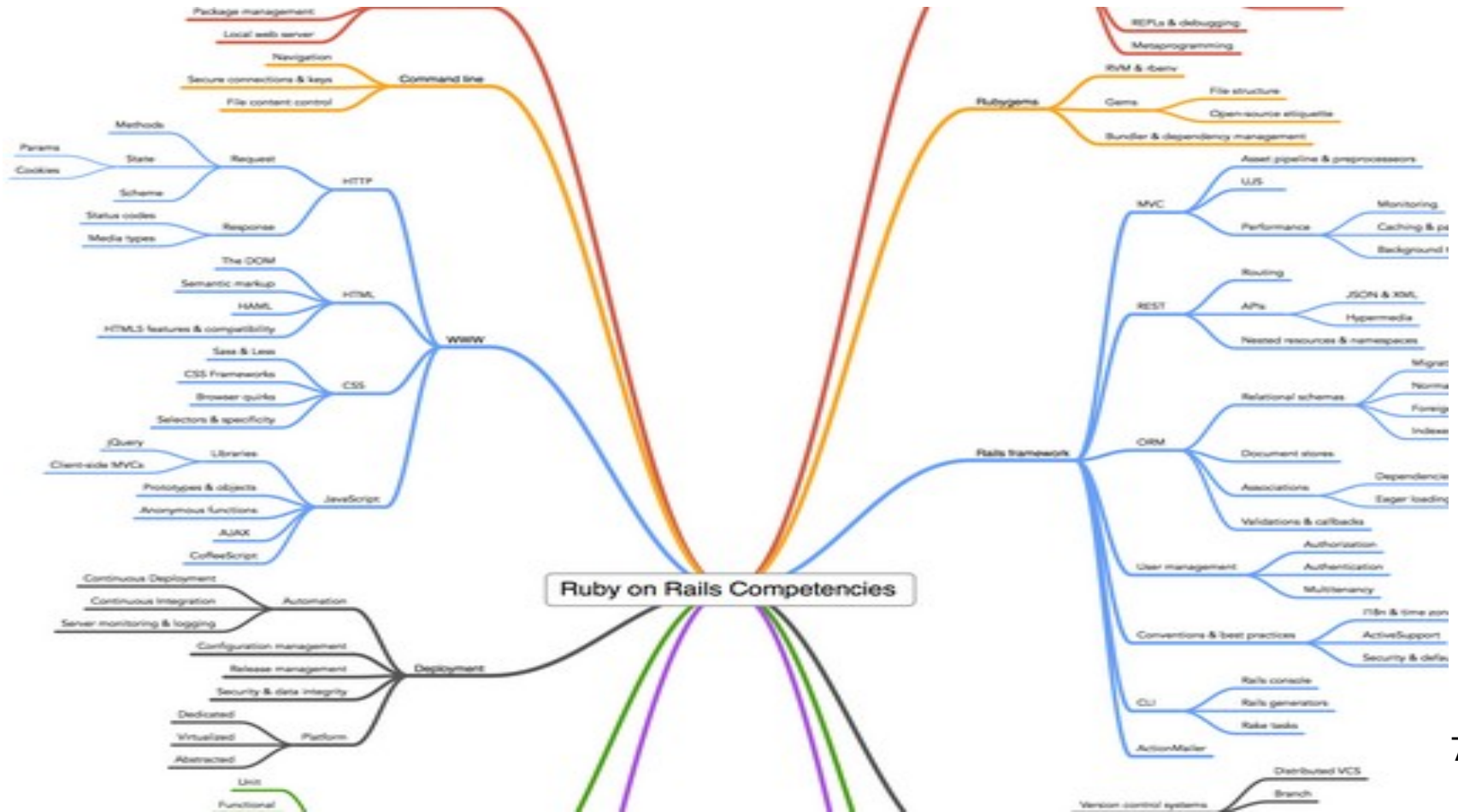
 **How to build a blog engine in 15 minutes with Ruby on Rails**  
<http://www.rubyonrails.org>  
By David Heinemeier Hansson, originally prepared for the Rails 1.0 conference in Brazil 2005

MORE VIDEOS

  0:01 / 16:12 CC Settings YouTube Fullscreen

Ruby on Rails demo  
David Heinemeier Hansson, the creator of the Ruby on Rails web development framework demonstrates how to create a blog engine in 1...  
[youtube.com](http://youtube.com)

# Code Fellows “This is why learning Rails is hard”





# The world programmers live in...

- Computer Science: Discrete Math, Algorithms
- Software Development: Agile project management, testing and verification
- High Level Programming: Ruby, Python, Java: Object-oriented and functional paradigms
- Low level Programming: C and C++: optimizing software for limited hardware resources
- Cloud management: Using Amazon Web Services / Microsoft Azure / Google Cloud



# ...is deep and complex.

- Networking: How packets move around a network, the levels of packet headers and the protocols that use them, security, firewalls, routing...
- Operating System (user) -- Dealing with files and processes, setting permissions, How to use and configure system utilities and daemons...
- Operating System (system) -- Concepts of what a file, or socket look like, how the OS determines what to do next...
- Hardware -- How a CPU processes instructions, speed costs of storage, storage failure, backup systems...



# What is a “dependency”?

- A "dependency" is a requirement of knowledge or implementation to accomplishing the desired result.
- E.g. learning machine learning depends on your ability to understand linear algebra and calculus.
- E.g. getting Ruby on Rails setup means first installing Ruby (and its gem manager) which requires installing a Ruby installer like rbenv.

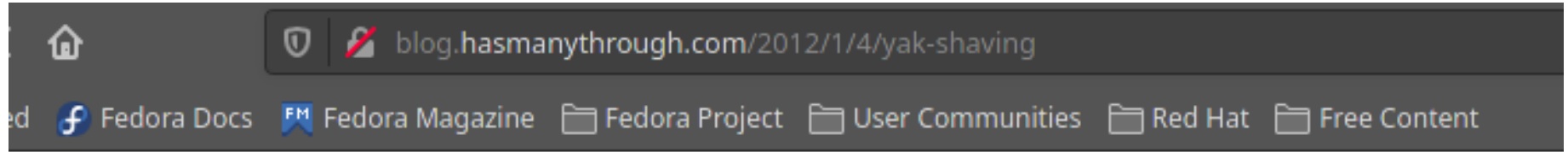


# Dependencies Lead to Yakshaving

- Recursion of dependencies: Having too many dependencies may cause what seems to be a simple task to take an infinite time to complete.
- The more yakshaving, the further the distance to seeing the results of your learning experiments.



# Yakshaving in the Real World



## has\_many :through geek by association

### Yak Shaving

— January 4, 2012 at 08:23 PST

As coders, most of us are not only familiar with the term [yak shaving](#), but spend many of our days doing nothing but. I often struggle to explain to non-technical folks what I actually spend my time doing when I'm working and what it feels like. This is the most accessible explanation I can come up with.

Say you want to go see a movie with your friend Joe. You can't get away with leaving the house when the kitchen is such a mess, so you have to load and run the dishwasher before you go. Unfortunately you're out of detergent, which means you have to run to the store to pick some up. You want to ride your bike or it will take too long to get to the store. But your bike's front tire is kind of low, so you have to pump it up first. However, your roommate borrowed your tire pump and you don't know where it is, so you have to go find your roommate and ask him about it. He's over at a neighbor's place having band practice, but it's just a short walk. When you get there they are in the middle of practice so you have to wait for a few minutes. The

# More Yakshaving

See this on Youtube:  
“Malcolm in the Middle  
light bulb”





# Thinking about economic return

- Value is created when a working product is delivered to a customer, or at least when someone other than the developer can see value in some output of labor.
- Learn something that has a minimum of "dependencies" that gives a new useful skill that will enhance productivity or value to people who have work that needs to be done.
- How can we make it easier to get closer to productivity?



# Reframe the Problem → Reduce Dependencies

- “I want to work at a big corporation on super-scale projects”  
→ “I want to be able to build database-driven websites”
- Get to productivity with less investment:
  - Reduce the areas of knowledge that are required
  - Reduce the number of tools involved

# LAMP: A Simpler Stack



- LAMP stack: Linux – Apache – MySQL – PHP
- Lamp stacks are easy to setup on your development machine (Windows, Mac, or Linux)
- Lamp stacks are inexpensive and dead-simple to get from shared-hosting companies. You also get bundled pre-configured services like email and backups.
- So, all you have to focus on is **PHP and MySQL**



# PHP and MySQL are useful

- Easy to start learning.
- Understand and extend off-the-shelf tools like Wordpress and Processwire.
- Learn good coding practices: object-oriented design, testing, and version control.
- SQL databases are fairly universal.



# A 3-stage Learning Path

1. PHP, MySQL, HTML and CSS

→ Add custom functionality to Wordpress or Processwire.

2. Javascript

→ Visualize data, create interactive pages, create browser-based games.

3. React frontend with Codeigniter API

→ Build a modern API-driven website.

# Learning Strategies

- **Practice:** Practice for an hour everyday, even if the hour is only 20 minutes.
- **Memorize:** Systematically memorize the keywords for the language and know the methods for types such as String and Array.
- **Experiment:** Push all the buttons.
- **Scrapbook:** Get an overview of the kinds of things that can be done with the technology. Keep a digital scrap book of links and screenshots of designs and functionality that you like. Try to implement them with what you currently know.





# Feedback is helpful

- Please Like and Subscribe. :-)
- I will try to answer questions or comments.

This video was made using Free and Open Source Software (FOSS) including Fedora Linux, LibreOffice Impress, and OBS Studio.