How we learn a language

@ryanstenhouse

http://www.freeagent.com

freeagent

Hello!

- @ryanstenhouse
- http://ryanstenhouse.eu

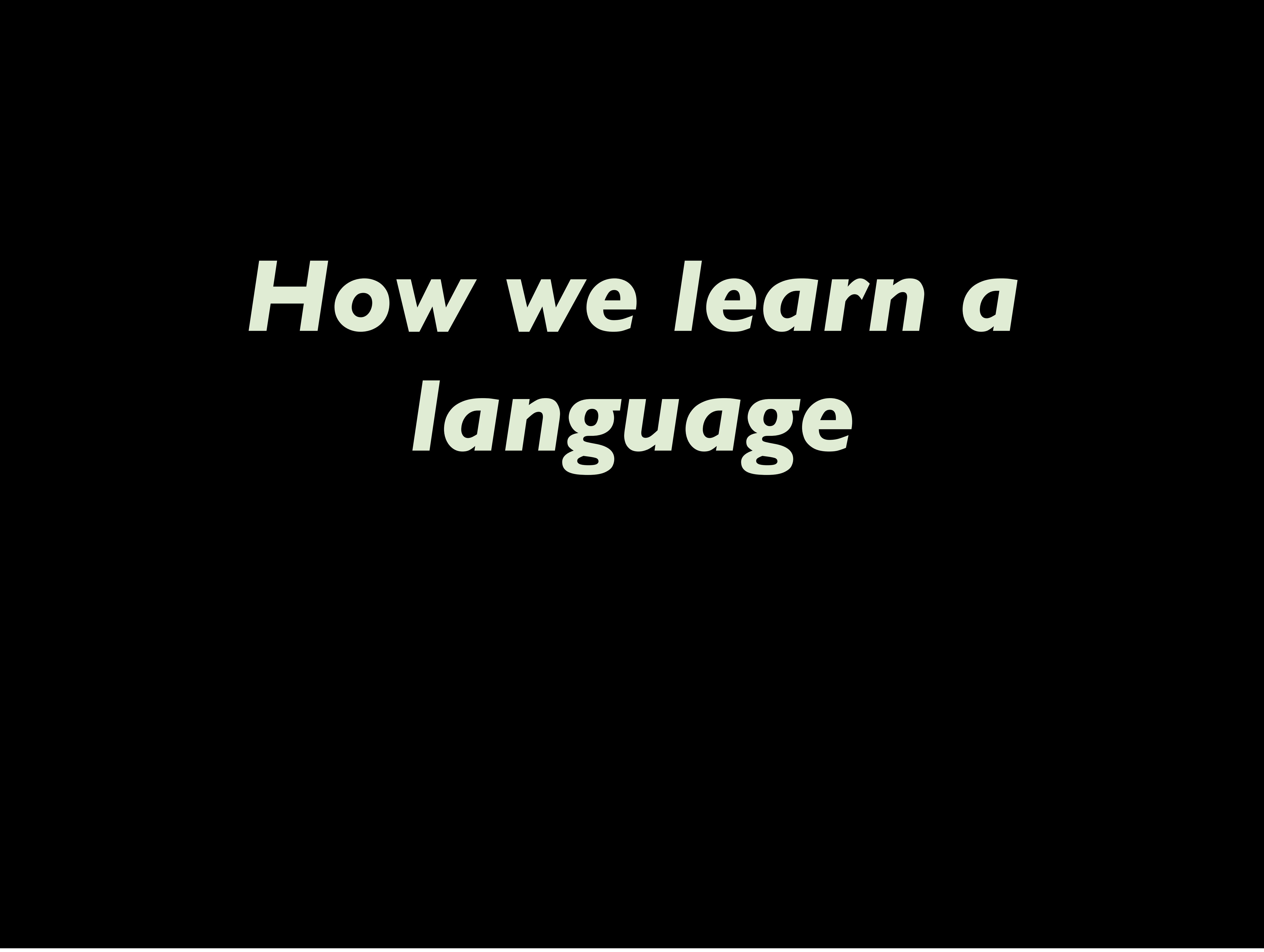
Saturday, 30 June 12

Hi! I'm Ryan Stenhouse, I write Ruby for FreeAgent here in Edinburgh, and I've got a lot to say about languages, how we learn them and how people communicate.

I'd like to open with a couple of questions, nothing too onerous – hands up if you came to Ruby from another programming language.

Great! Keep them up if Ruby was the 2nd programming language you learned.

And lastly, hands up if it was the 3rd or later?
Brilliant – Thanks! I'll come back to that later.



How we learn a language

Saturday, 30 June 12

What I'm wanting to talk with you about today is "how we learn a language".

I'm quite opinionated on this subject.

Like many (if not most) of you here, I speak a fair few languages:

* English
* Ruby
* Japanese
* Javascript
* French
* There are more...

Some of them I learned in a classroom, others in front of my computer, or by reading books or listening to mp3s / tapes. I guess most of you learned your languages using one or more of these methods.

I'll be going into that a bit later.

http://ryanstenhouse.eu/talks/how-we-learn-a-language for picture credits.

First though; I'm going to tell you a wee story:

> In 2005, I went to Japan for the first time. This was also the first time I learned how to eat food with chopsticks.

> I was at a small local restaurant in Shibuya, not all that far from the train station (can't remember its name) with a friend. We had a really friendly waiter who had a wicked sense of humour.

> I asked him for a fork, he looked at my friend, they shared a look, this moment of inexplicable unspoken communication. Both nodded, he pointed to the chopsticks, wished me 'good luck' and left.

> So, faced with the choice of:

>   1. Eat with Chopsticks

>   2. Starve

>   3. Eat with fingers

I chose '1'. If you've ever seen me eat with chopsticks, you'll see that I do have a rather unique technique.

Necessity drives a lot of what we do. How many of you (hands up again please, sorry), learned Ruby (or another programming language) because you _had_ to for one reason or another, probably due to work?

http://ryanstenhouse.eu/talks/how-we-learn-a-language for picture credits.

You’ve already learned another language!

Saturday, 30 June 12

If you've had your hand up at all this afternoon, _you've already learned another language_, sure, it was a comparatively simple computer language rather than a complex beast like a real-live natural language; but the skills you need to acquire to learn either are the same; and the application of those skills is also the same - and I'm here to talk a but more about them today.

Hopefully I'll be able to make it clear that the problem of learning a natural language isn't all that dissimilar to learning a programming language, and that the approach to the problems is the defining point, and that by thinking about them a little differently; you can become conversant with reasonable, but not already taxing levels of effort.

You've done it before!

Saturday, 30 June 12

I know this, because _you've done it before_.

I'm going to talk about two languages I've learned, or rather, I am still learning. They're quite different from one another, both are from the Far East and have mixed reputations with regard to how difficult they are to learn.

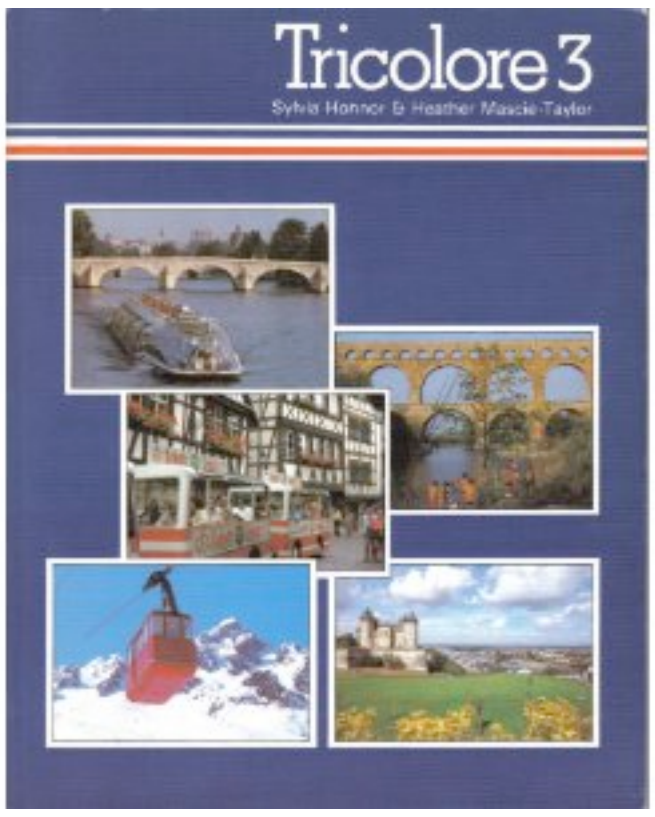
One I studied through immersion, fighting my way against a strange grammar and some very strange idioms. I was forced by necessity to engage with the culture and community supporting the learners of the language and found it invaluable.

The other also had a strange grammar and its own idiomatic strangeness that to an outsider like me were almost incomprehensible. I studied this same way I was taught French in High School - from textbooks and other examples, online media, video presentations and even podcasts.

écoutez et répétez

Turns out that kind of learning doesn't work out too well for me; especially when it makes me flash back to horrible French lessons form a disinterested teacher.

To this day, the phrase "écoutez et répétez" sends a small shudder right down my spine.

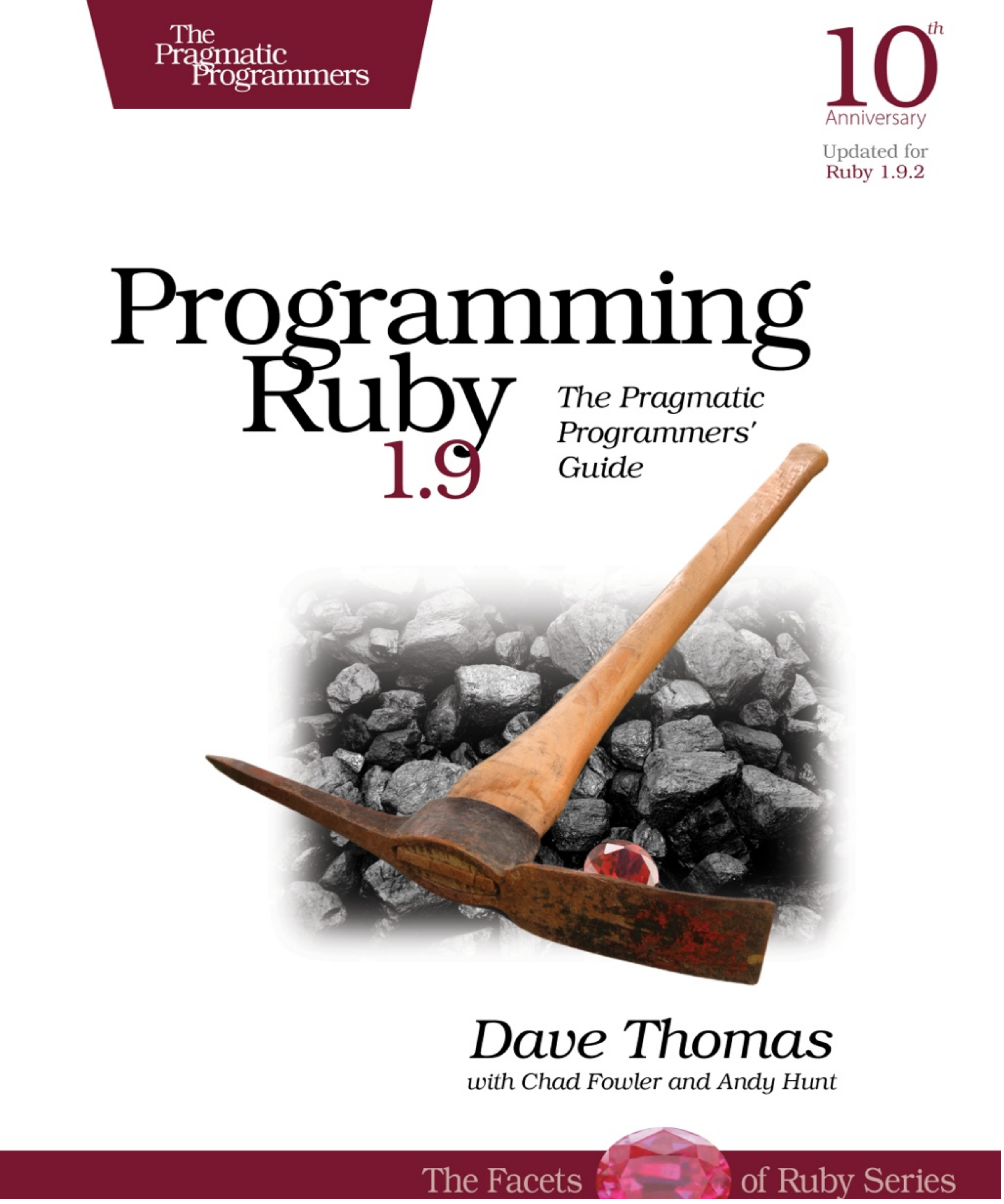


http://ryanstenhouse.eu/talks/how-we-learn-a-language for picture credits.

Saturday, 30 June 12

We were using first edition Tricolore textbooks from the 1980s and listening to tapes which were just as old. Never once since have I ever felt the need to ask: "ou est la discotheque" in anger.

I think that 'repeat from a tape' learning detaches you from the living heart of a language and the associated culture and brings in an artificial 'other' component - the 'textbook' and the idea that a textbook can tell you everything you need to know about a language just be reading aloud or writing phrases over and over.



The Pragmatic Programmers

10th Anniversary

Updated for Ruby 1.9.2

Programming Ruby 1.9

The Pragmatic Programmers' Guide

Dave Thomas

with Chad Fowler and Andy Hunt

The Facets of Ruby Series

Programming Ruby is available at
http://pragprog.com

You wouldn't sit with the Pickaxe open in front of you and read it from cover to cover without having a computer to hand to type the Ruby into, right? It's kinda the same thing. You need to interact with something that speaks the language you’re learning.

The languages I specifically want to talk with you about today are Ruby and Japanese, and my experiences with learning and continuing to learn both.

First, a wee bit of my history with languages:

Shibuya, 2005 (I’m an awful photographer)

I made my first attempt at learning Japanese from the summer of 2002 until May 2005, when I visited Japan for the first time. Since it was a 'real' language, I approached learning Japanese the same way I was taught French in school. You know, the illusion that constant droning repetition leads to understanding.

I thought it went well, until I found myself in Tokyo and unable to understand a word anyone was saying, including my friend and guide for some of the trip. I didn't know Japanese folks spoke so fast in real life, and I certainly hadn't considered dialects of the language as being significantly different. I was an idiot, but I was also learning in a bubble – these things didn't occur to me.

~250 miles apart

Sounds like a different language

Tokyo
Osaka

Saturday, 30 June 12

My friend was from Osaka and spoke kansai-ben, a rather different animal to the kyōtsūgo (Common Language based on Tokyo's dialect of Japanese) I had studied.

A while after I got back from Japan, I trained as a Teacher of English as a Foreign Language. I had a fantastic instructor, who managed to install a need for immersion, rather than rote being the best way to learn a language. She then processed for an entire day to instruct the class in Turkish; was very informative and quite good fun.

Map (c) 2012 Microsoft and Terra Italy, From http://binged.it/Oy4PMV

Saturday, 30 June 12

In 2007 I got my first _real_ Ruby job working for PCCL in Rosyth writing financial applications.

I needed to get up to speed quickly, so immersed myself in the code of the application, looking up bits I didn't understand or couldn't figure out in The Pickaxe; the reference section alone is worth buying the book for..My background in various disciplines of programming languages helped greatly, and I was able to draw on that experience to figure out what was going on and where.

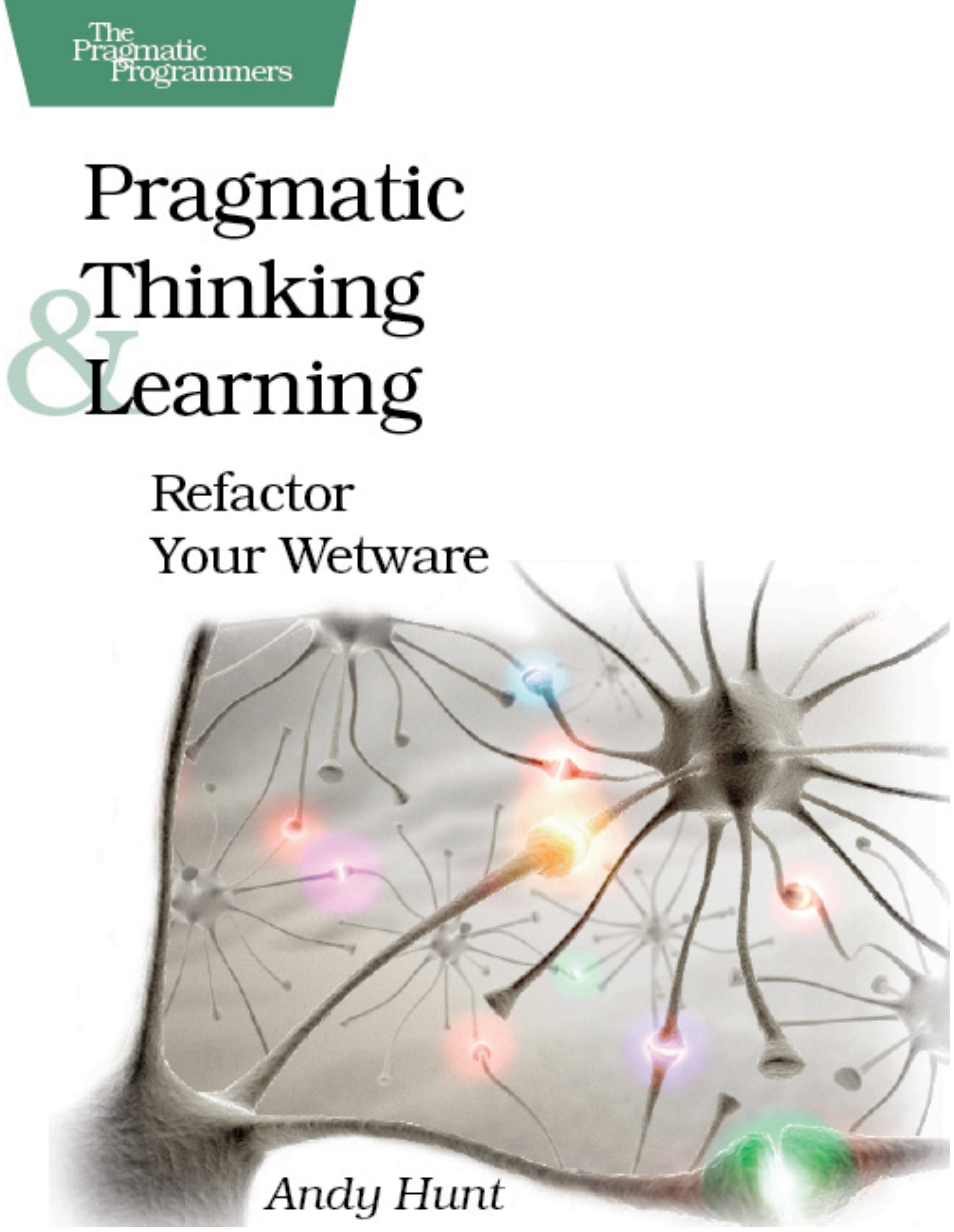
Everything I head already learned from playing around with Ruby before crystallised and I was productive in a number of weeks and fluent in a matter of months, with a few trips onto IRC and onto mailing lists to get questions answered, not to mention the wonderful support my colleagues offered at the time.

We’re problem solvers

It was about this time that I started picking up learning Japanese again, and radically changed my approach. That's what I'm here to talk about. Immersive learning was great for me, but I wondered if I could approach things a bit more like learning to write a programming language.

Here's what I mean:

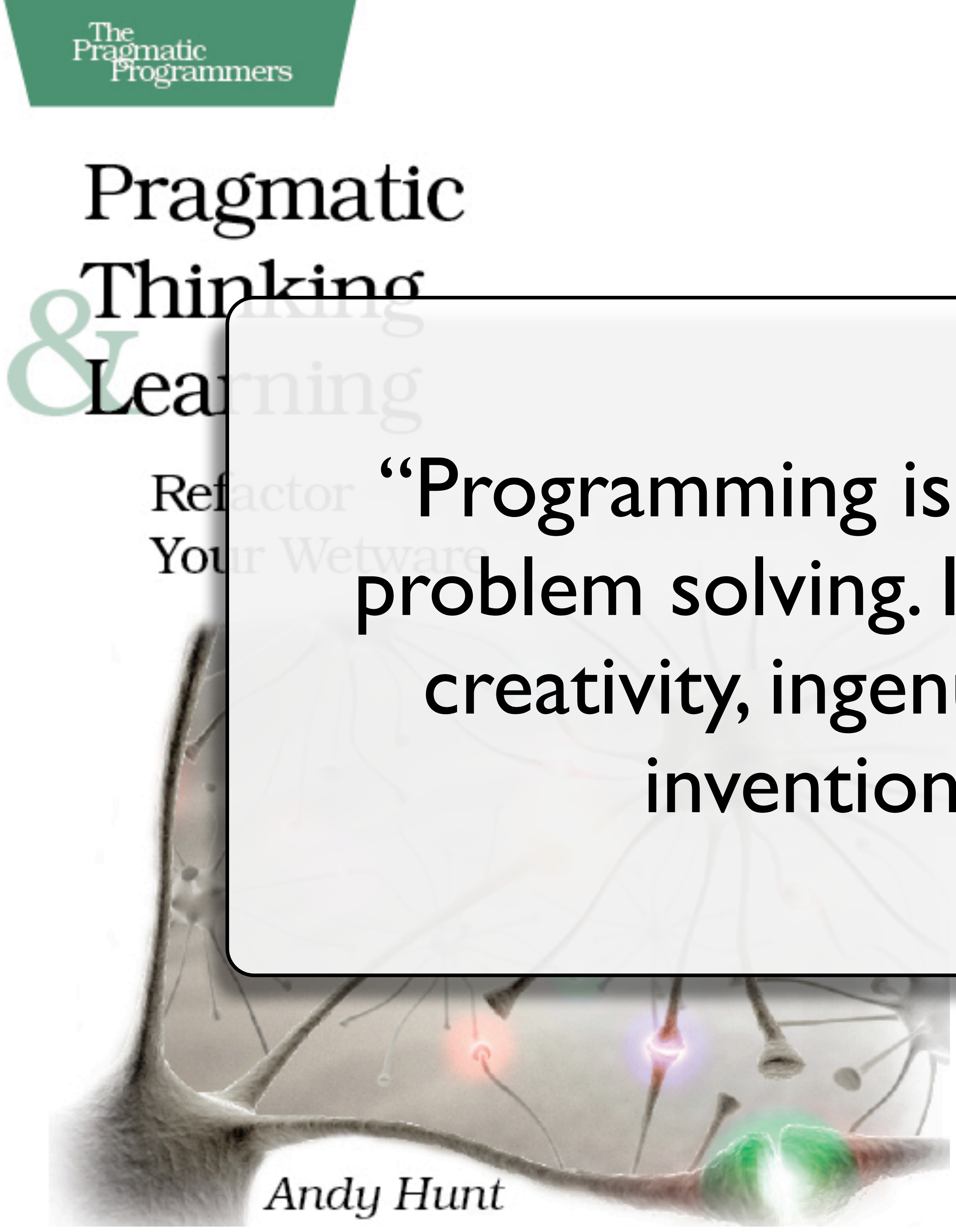
We're fantastic problem solvers. We have to be, it's part of our craft.



Pragmatic Thinking & Learning is available at
http://pragprog.com

Saturday, 30 June 12

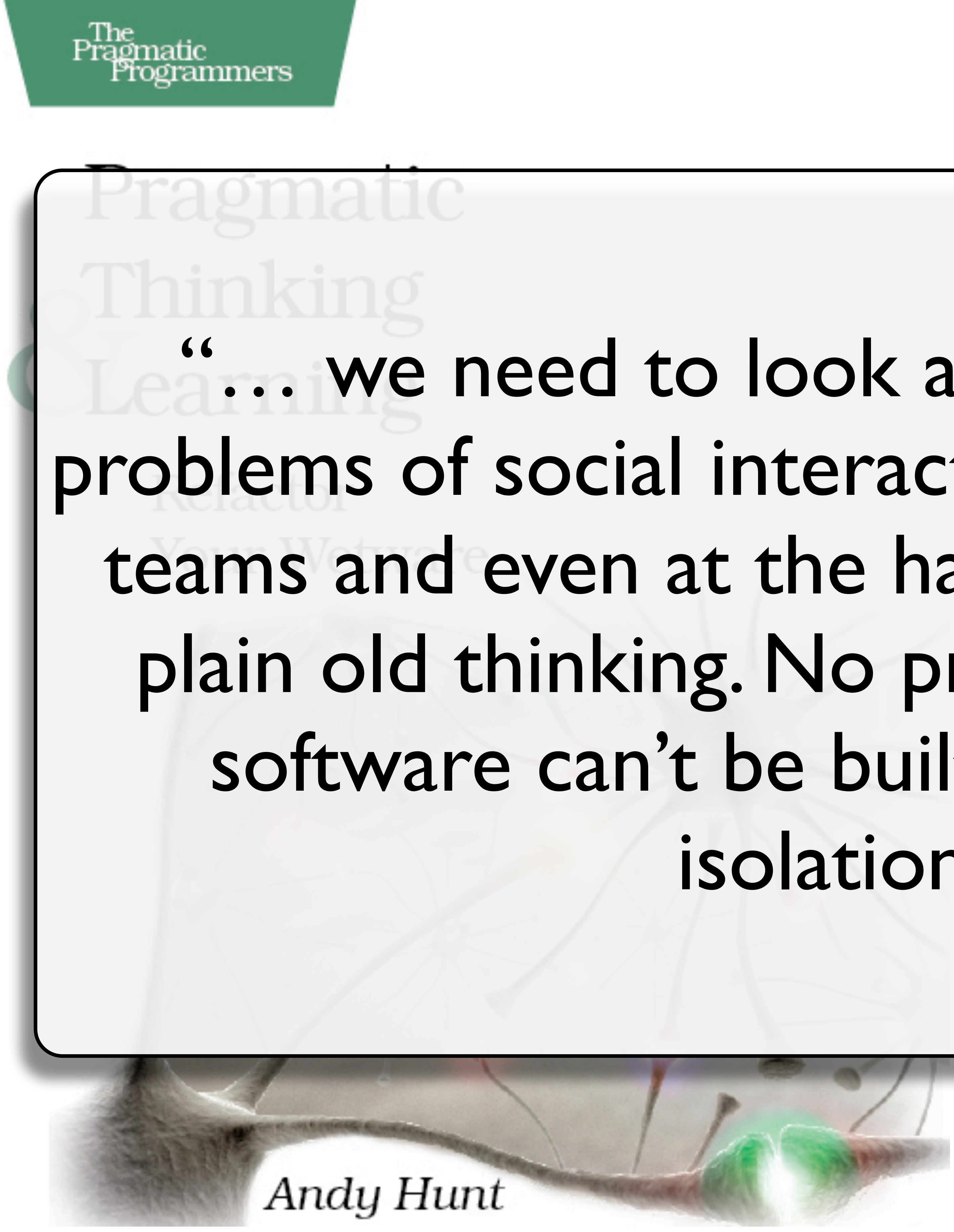
Andy Hunt of the Pragmatic Programmers says it better than I could in the introduction of his book "Pragmatic Thinking and Learning: Refactor your wetware":



“Programming is all about problem solving. It requires creativity, ingenuity, and invention.”

Pragmatic Thinking & Learning is available at
http://pragprog.com

> Programming is all about problem solving. It requires creativity, ingenuity, and invention.



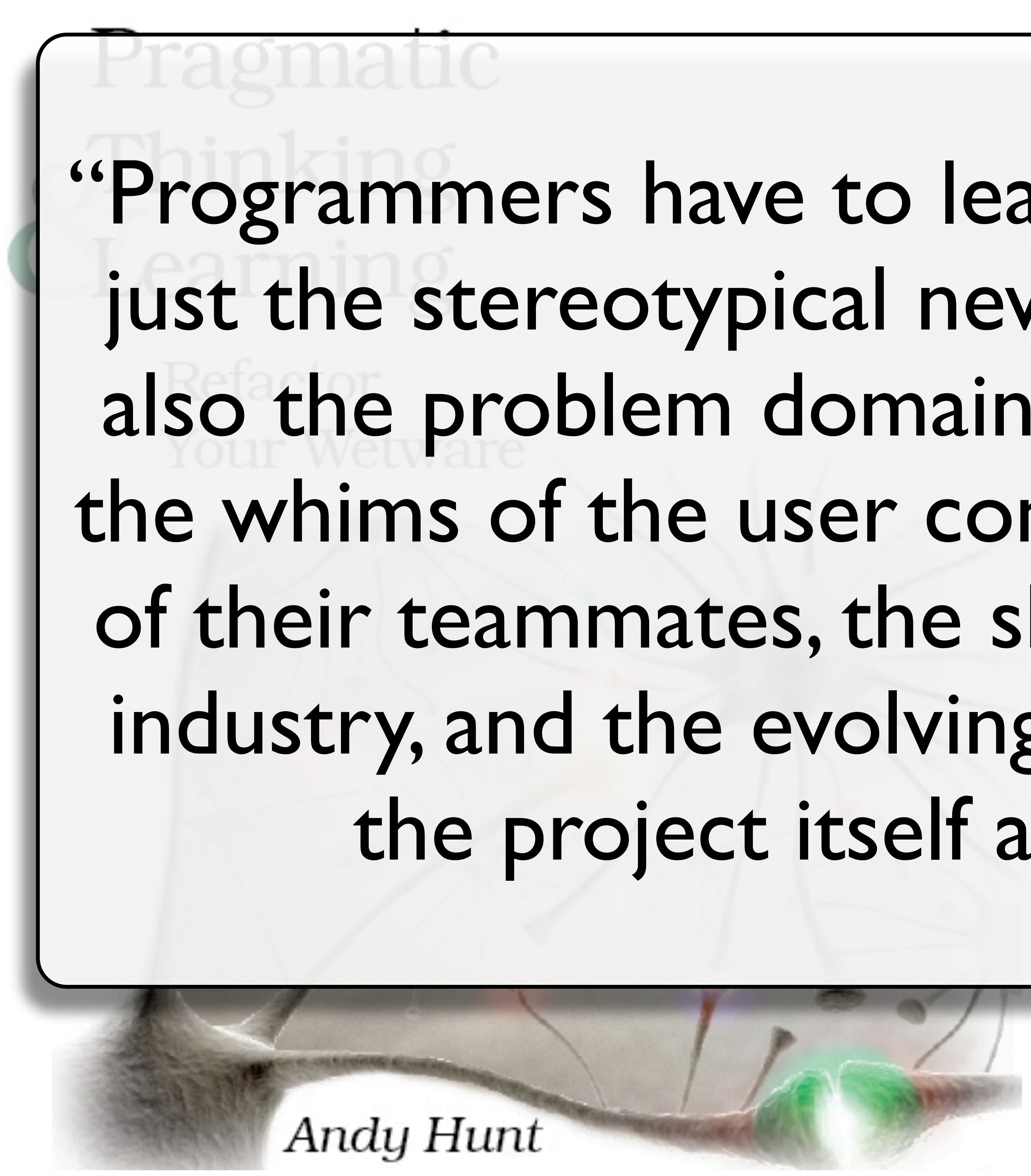
“… we need to look at the really hard problems of social interaction in and between teams and even at the harder issues of just plain old thinking. No project is an island; software can’t be built or perform in isolation.”

Pragmatic Thinking & Learning is available at http://pragprog.com

Saturday, 30 June 12

> … we need to look at the really hard problems of social interaction in and between teams and even at the harder issues of just plain old thinking. No project is an island; software can’t be built or perform in isolation.

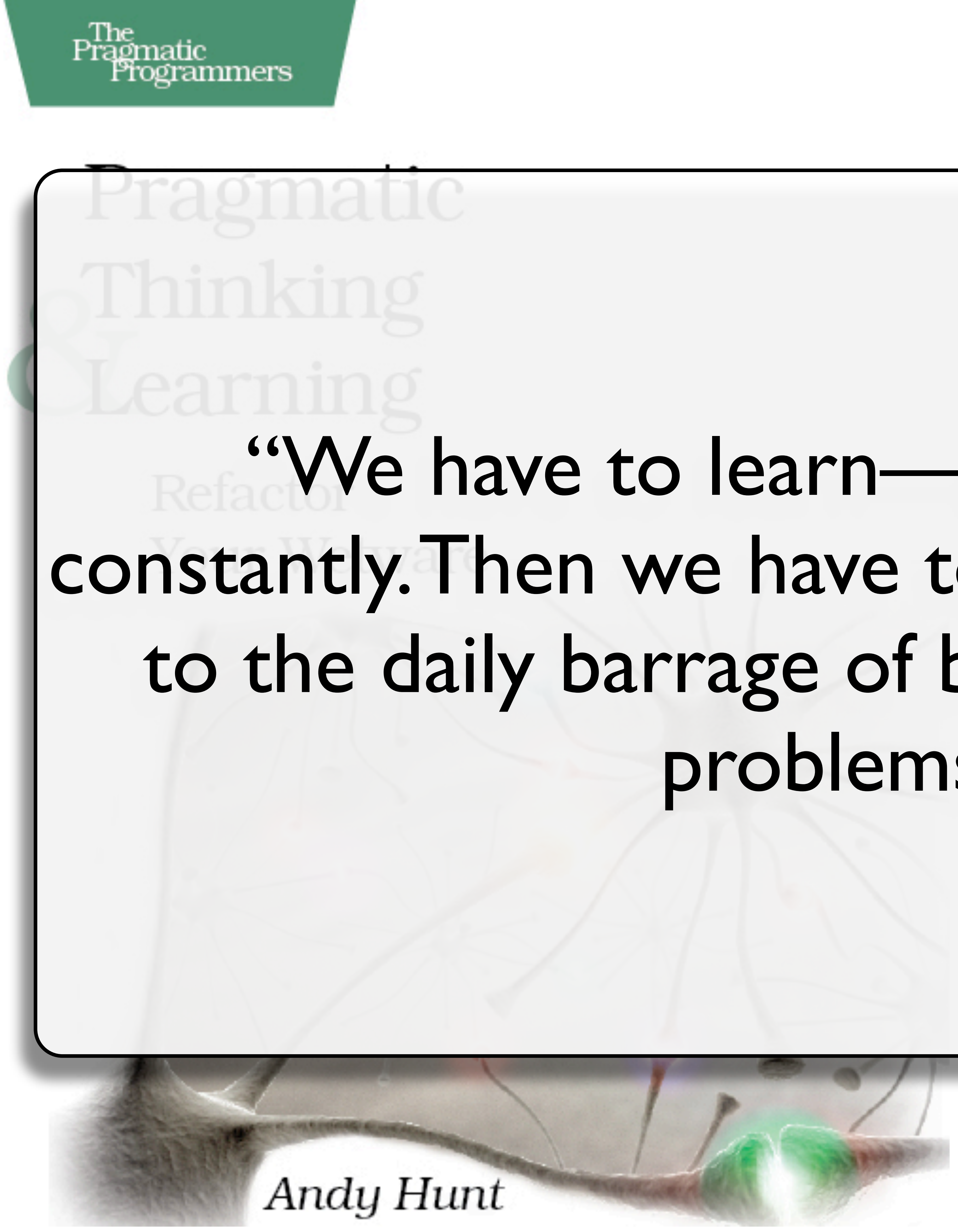
The Pragmatic Programmers



“Programmers have to learn constantly—not just the stereotypical new technologies but also the problem domain of the application, the whims of the user community, the quirks of their teammates, the shifting sands of the industry, and the evolving characteristics of the project itself as it is built.”

Pragmatic Thinking & Learning is available at
http://pragprog.com

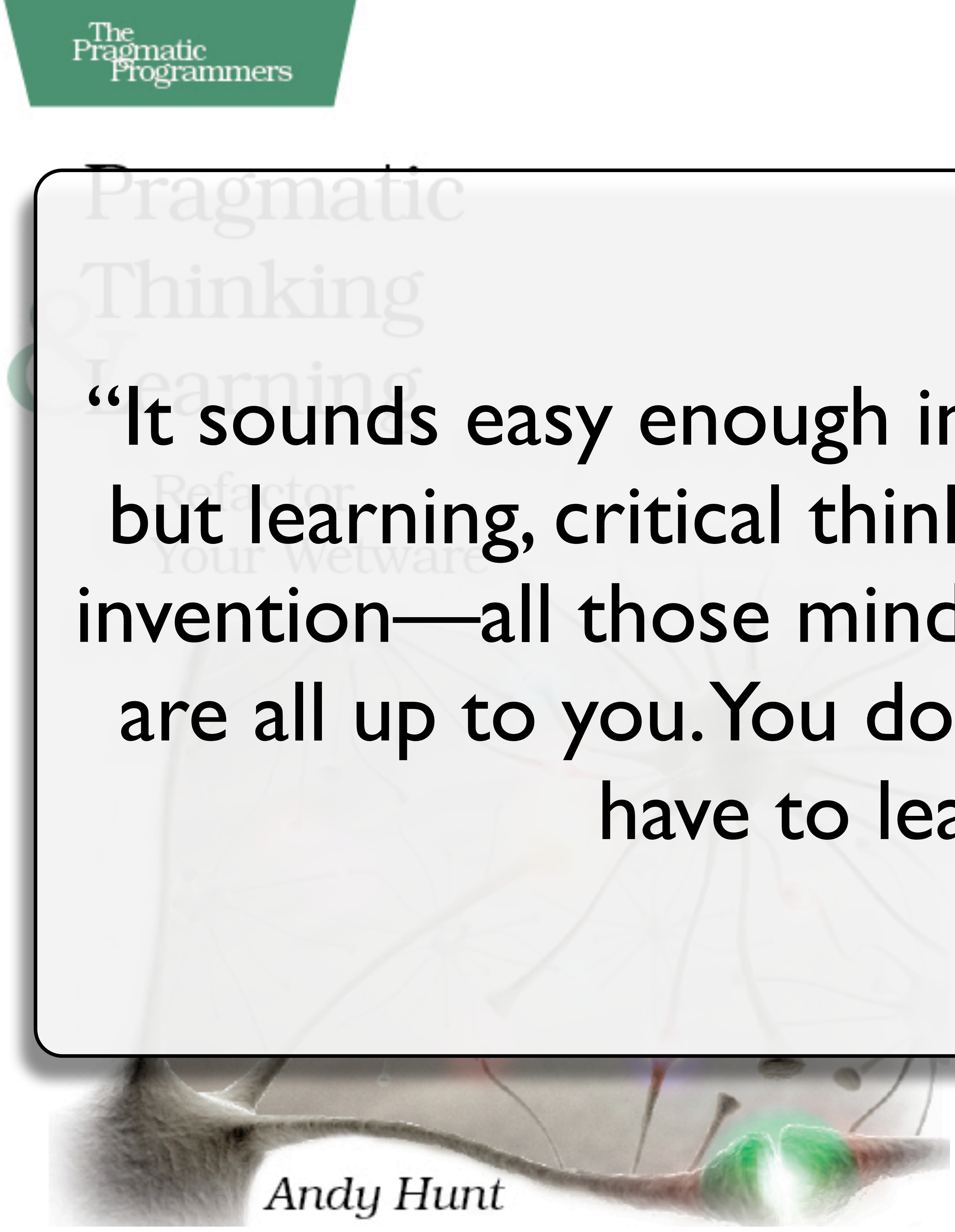
> Programmers have to learn constantly—not just the stereotypical new technologies but also the problem domain of the application, the whims of the user community, the quirks of their teammates, the shifting sands of the industry, and the evolving characteristics of the project itself as it is built.



“We have to learn—and relearn—constantly. Then we have to apply this learning to the daily barrage of both old and new problems.”

Pragmatic Thinking & Learning is available at
http://pragprog.com

> We have to learn—and relearn— constantly. Then we have to apply this learning to the daily barrage of both old and new problems.



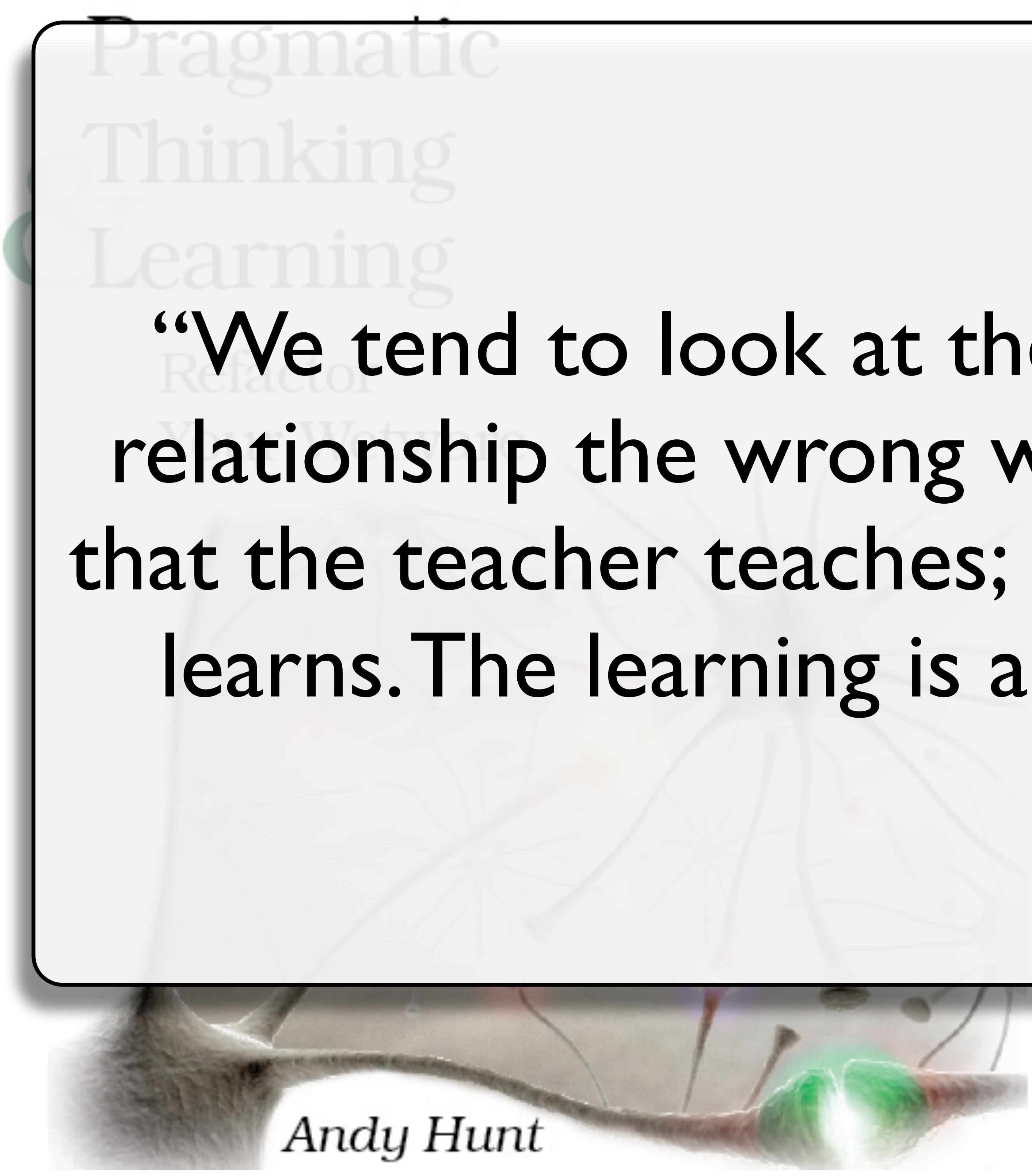
“It sounds easy enough in principle perhaps, but learning, critical thinking, creativity, and invention—all those mind-expanding skills—are all up to you. You don’t get taught; you have to learn.”

Pragmatic Thinking & Learning is available at http://pragprog.com

Saturday, 30 June 12

> It sounds easy enough in principle perhaps, but learning, critical thinking, creativity, and invention—all those mind-expanding skills—are all up to you. You don’t get taught; you have to learn.

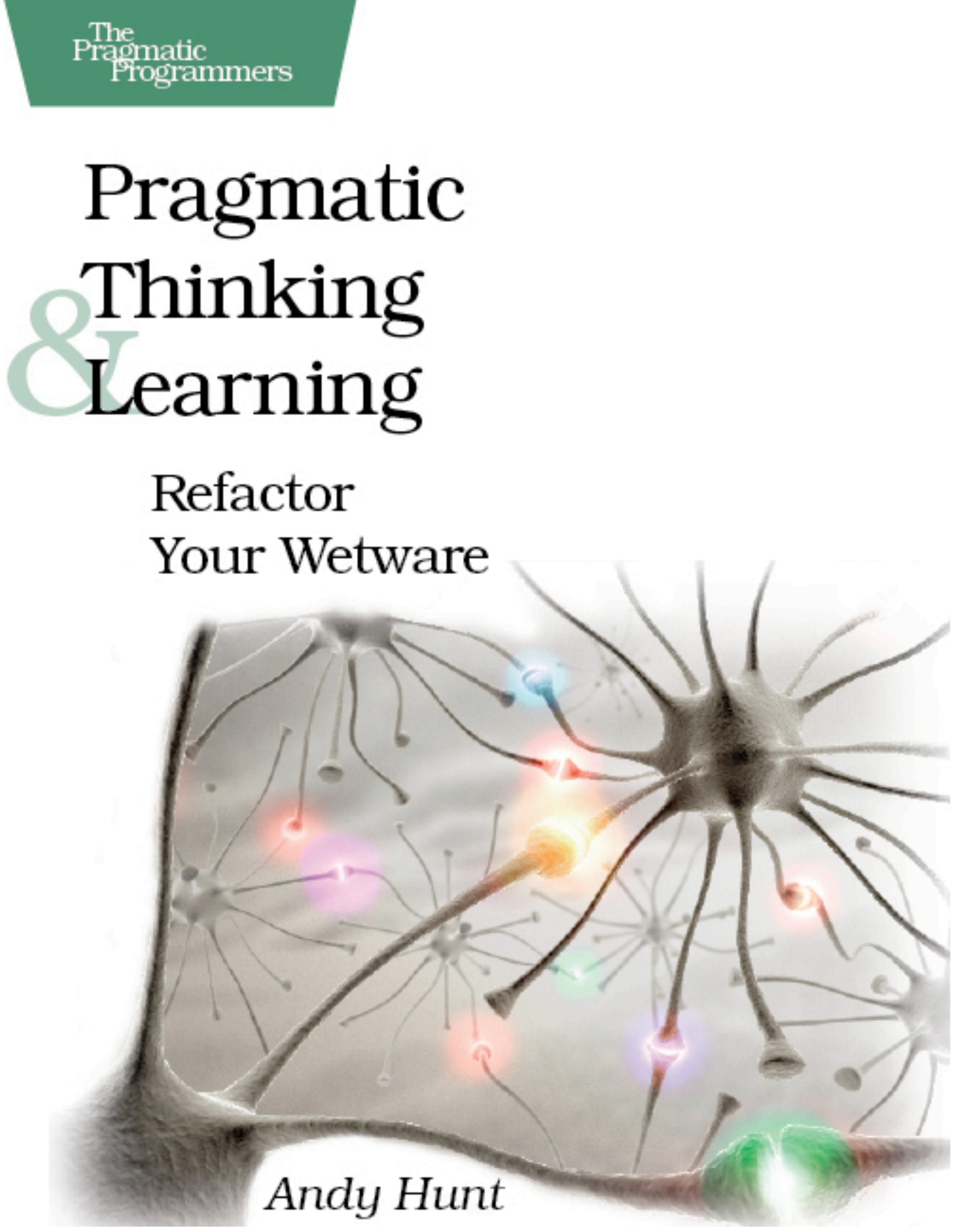
The Pragmatic Programmers



“We tend to look at the teacher/learner relationship the wrong way around: it’s not that the teacher teaches; it’s that the student learns. The learning is always up to you.”

Pragmatic Thinking & Learning is available at
http://pragprog.com

> We tend to look at the teacher/learner relationship the wrong way around: it’s not that the teacher teaches; it’s that the student learns. The learning is always up to you.

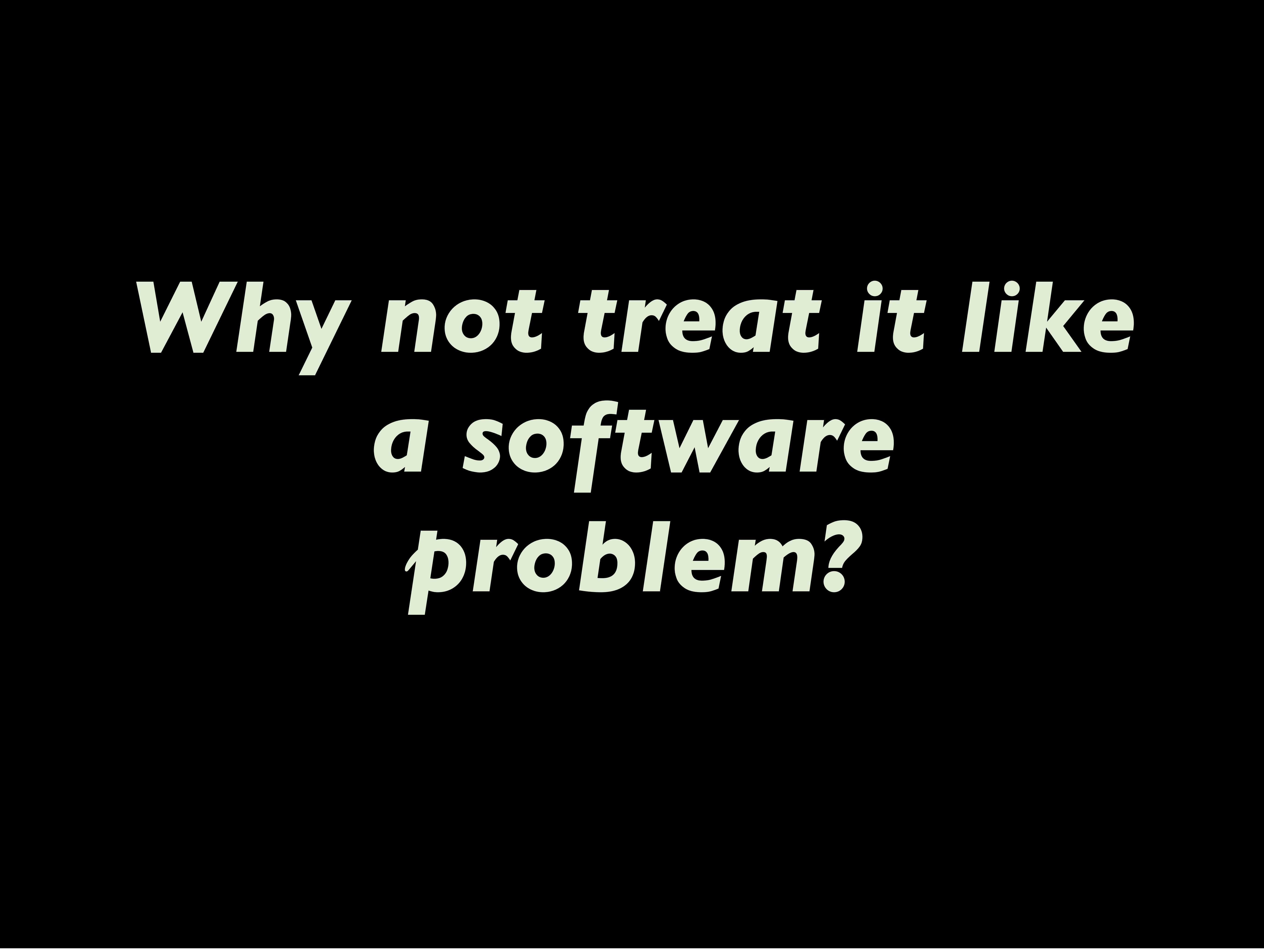


Pragmatic Thinking & Learning is available at
http://pragprog.com

If you haven't read this book yet, go and buy it now. Well, not _now_ now. After! I wish I had read it years ago.

Why not treat learning to speak another language like a software problem?

I say you've all learned a language before - I'm deliberately not making a distinction between a natural language and a computer one.

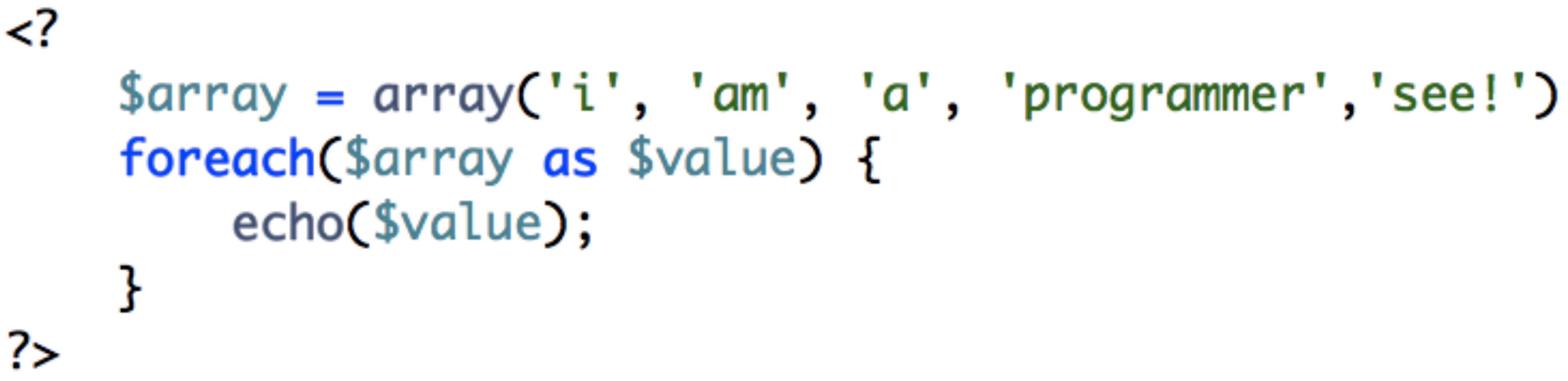


Why not treat it like a software problem?

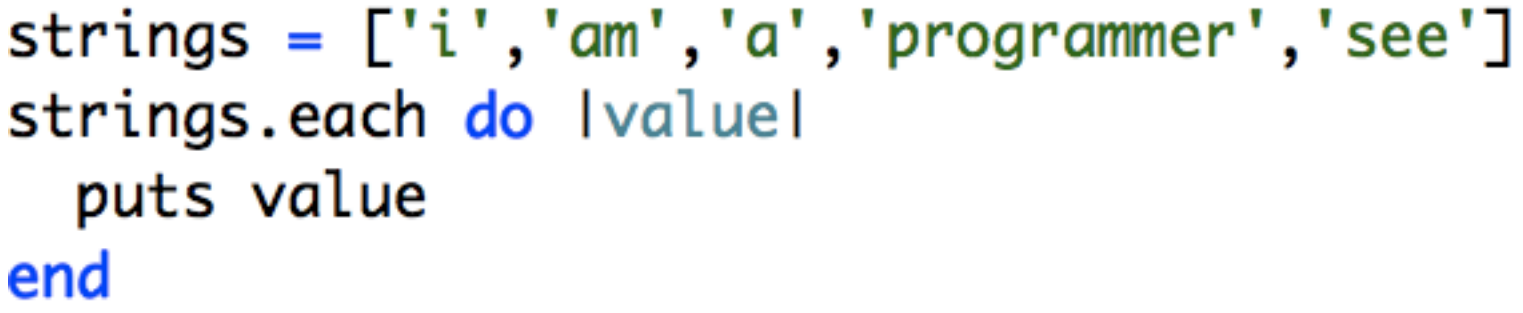
Saturday, 30 June 12

Why not treat learning to speak another language like a software problem?

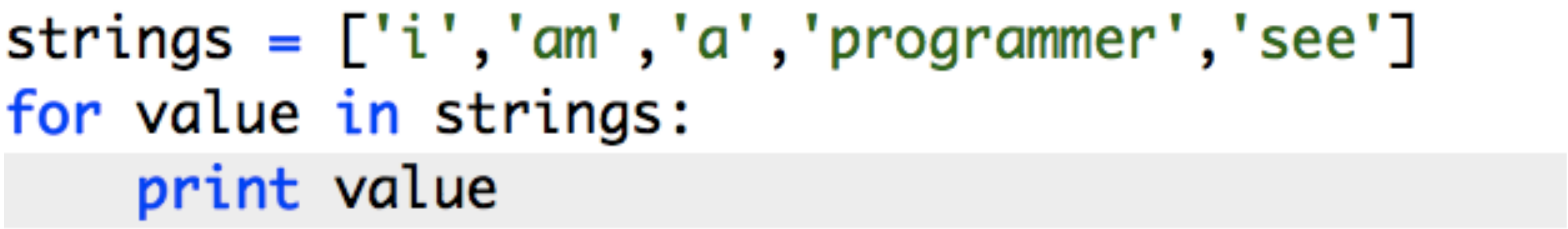
I say you've all learned a language before - I'm deliberately not making a distinction between a natural language and a computer one.



```
<?
    $array = array('i', 'am', 'a', 'programmer','see!')
    foreach($array as $value) {
        echo($value);
    }
?>
```



```
strings = ['i','am','a','programmer','see']
strings.each do |value|
  puts value
end
```



```
strings = ['i','am','a','programmer','see']
for value in strings:
    print value
```

Call out – What’s the top language?
And the bottom one?

You look at these and know they mean the same thing. Why?

Go on, answer, don’t be shy..

Porting == Translation.

BALLE

BALL

ボール

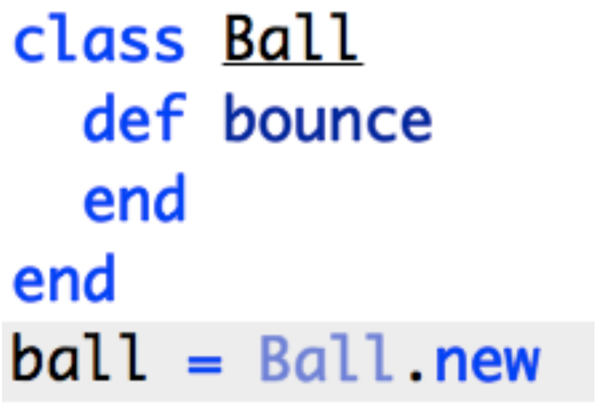
Saturday, 30 June 12

Call out - What’s the top language?
And the bottom one?

You look at these and know they mean same thing. Why?

Go on, answer, don’t be shy..

Porting == Translation.



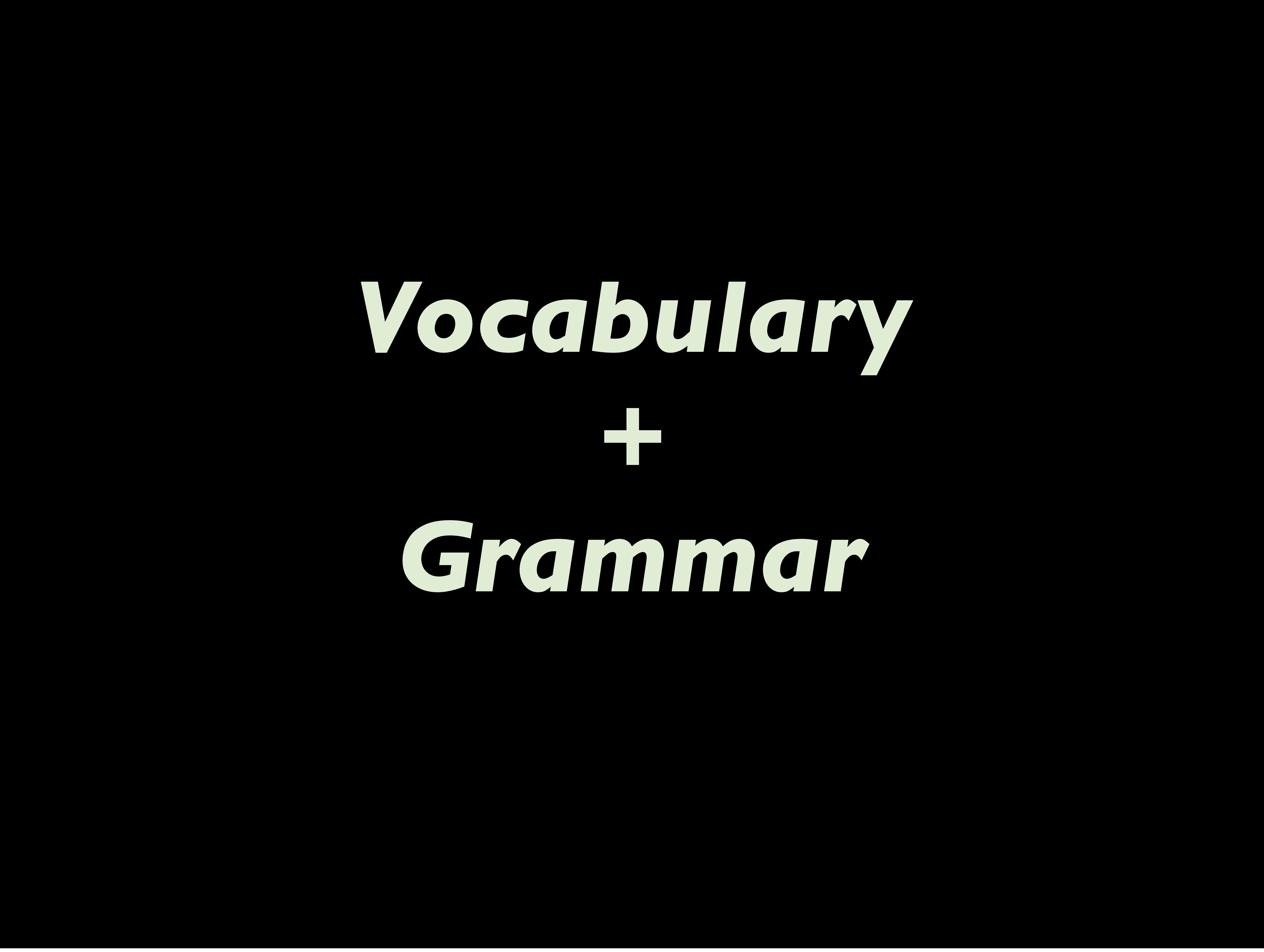
```
class Ball
  def bounce
  end
end
ball = Ball.new
```

BALLE

ボール

Saturday, 30 June 12

Why not this? They’re all balls, right? Everything else is an abstraction.



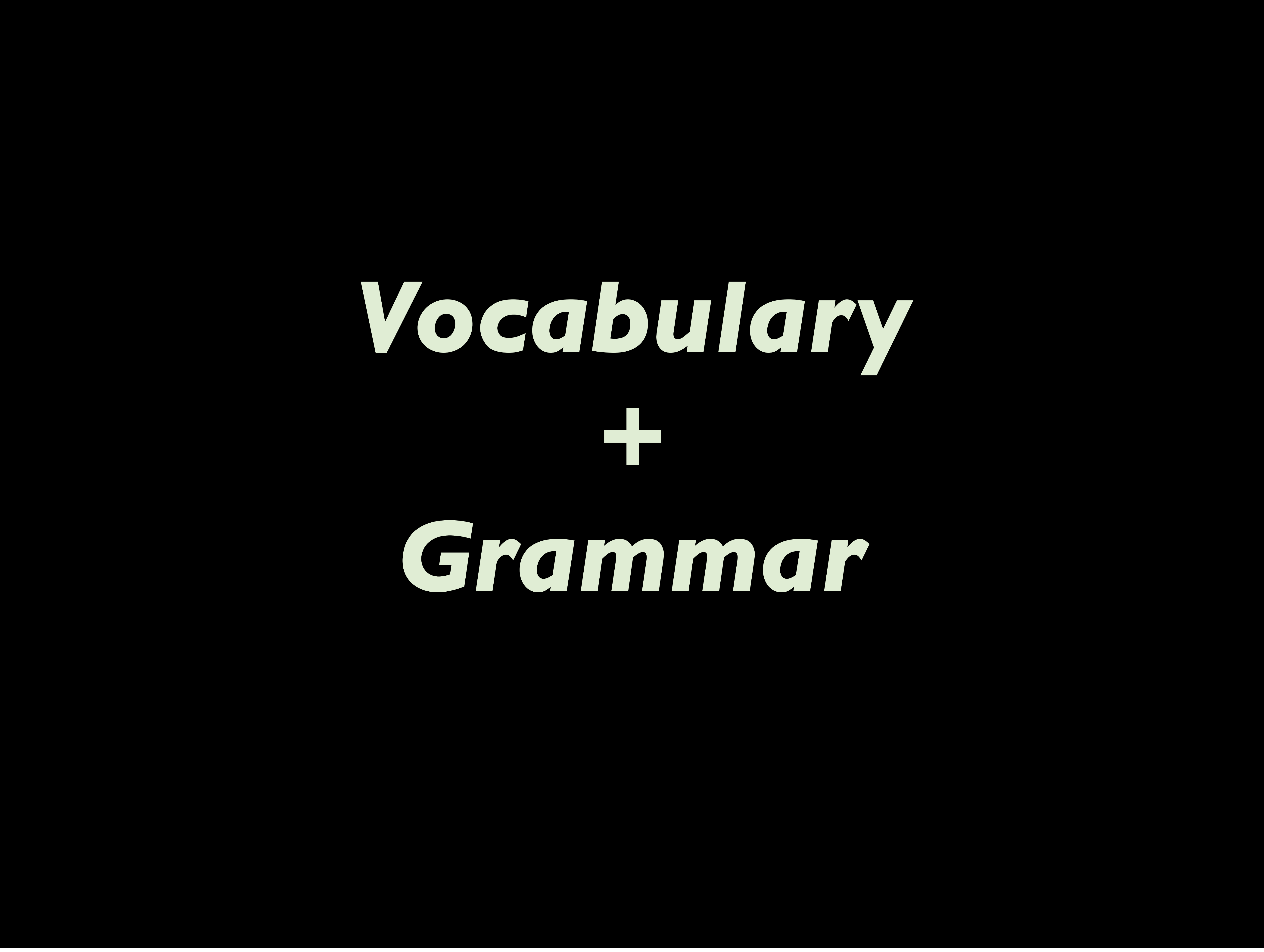
Vocabulary
+
Grammar

Saturday, 30 June 12

Your Vocabulary is the set of words within a language that are familiar to that person.

A grammar is the set of structural rules that govern the composition of a language.

Together, they more or less make up a language.

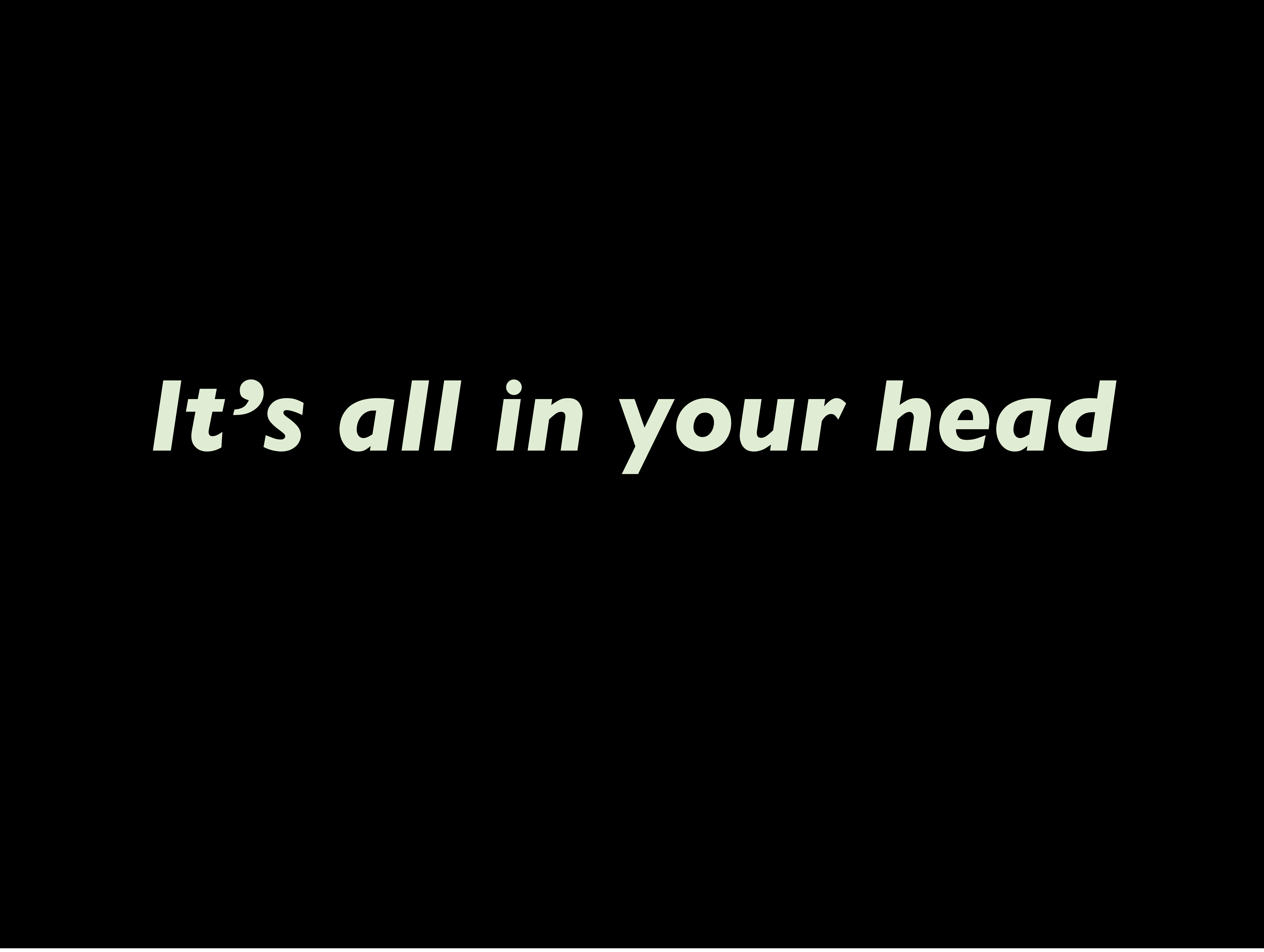


Vocabulary
+
Grammar

When you’re learning a new language, you pick up the vocabulary first. You look for things that are familiar to you based on your experience, you ask people. Even if you just hold up a ball and point to it you can be told that it’s “une balle”.

That’s how you’re able to pick up a new programming language quite quickly – the vocabulary is largely known. The main differences for programming languages are to do with the grammar.

You engage with the community and consult resources on your own terms to study the grammar.



It’s all in your head

Saturday, 30 June 12

The difference is a mental one - you think one is hard because you're more experienced with the other. Nonsense!

The same analytical approach you take to learning another programming language can be applied.

It’s all communication.

Saturday, 30 June 12

A programming language is how you _speak_ to a computer.

Learning how to speak to a _person_ is the same problem.

Break it down

Natural languages have at least three components, how they’re read, how they’re written and how they’re spoken.

Grammar, Vocabulary, Constructs, Abstractions.

Iterate

You can learn to speak a natural language without learning to read and write it.

It’s a great first step; since it builds vocabulary.

You can iterate later.

Engage

Saturday, 30 June 12

Seek out other learners of your language, and hang out on mailing lists, Stack Exchange and IRC channels. Harass folks on Skype to practice - all the tools you’d use to learn a programming language.

You’re good at this! You must be, you’ve already learned other languages.

Try it.

Saturday, 30 June 12

Prove me wrong. Here in Edinburgh, starting in August I'll be organising language meet-ups. Come along and study.

More Online + Credits

- http://ryanstenhouse.eu/talks/how-we-learn-a-language

We're hiring.

DUDE, FOR RAILS?

E PRAETERITO AD FUTURUM

freeagentcentral.com

http://www.freeagent.com/jobs

freeagent