



THE JR. DEVELOPER SURVIVAL GUIDE - OR - SHIT I WISH PEOPLE HAD TOLD ME

A LIST OF ADVICE, ANECDOTES, AND OTHER A-WORDS THAT WILL LEAD YOU
FROM BEING A USELESS LUMP TO ACTUALLY BEING PRODUCTIVE
(OR AT LEAST NOT BE IN THE WAY)

Introductions

- * I'm James

Introductions

- * I'm James
- * I make software

Introductions

- * I'm James
- * I make software
- * I got in to software later than most, in my early 30s

Introductions

- * I'm James
- * I make software
- * I got in to software later than most, in my early 30s
- * I know some stuff, and anyway, they asked me to talk, not you so shut up and listen to me for an hour.



Introductions 2: The Desolation of Smaug

- * I work for Lochbridge

Introductions 2: The Desolation of Smaug

- * I work for Lochbridge
- * I work at OnStar Corp. on the connected car platform

Introductions 2: The Desolation of Smaug

- * I work for Lochbridge
- * I work at OnStar Corp. on the connected car platform
- * Get in touch!
 - * @kroysemaj
 - * jamesoleanyork@gmail.com

Introductions 3: Revenge of the Sith

- * Software development is a 2nd career for me

Introductions 3: Revenge of the Sith

- * Software development is a 2nd career for me
- * I started out as a professional musician

Introductions 3: Revenge of the Sith

- * Software development is a 2nd career for me
- * I started out as a professional musician
- * I got into software when I decided I liked to do things like pay bills and eat food

Introductions 3: Revenge of the Sith

- * Software development is a 2nd career for me
- * I started out as a professional musician
- * I got into software when I decided I liked to do things like pay bills and eat food
- * Having software be my 2nd career gives me a different perspective from someone who's only ever done this job, hence the potential value of that perspective

Mission Statement

My goal with this talk is to:

Mission Statement

My goal with this talk is to:

- * Provide a mix of technical and interpersonal advice I wish I'd had when I started in the software field

Mission Statement

My goal with this talk is to:

- * Provide a mix of technical and interpersonal advice I wish I'd had when I started in the software field
- * Deliver some hard-hitting TruFax™ about the unjust universe we all live in

Mission Statement

My goal with this talk is to:

- * Provide a mix of technical and interpersonal advice I wish I'd had when I started in the software field
- * Deliver some hard-hitting TruFax™ about the unjust universe we all live in
- * Give a short list of tasks to undertake as you enter a job search to make you more attractive to a prospective employer or a more valuable member to a team you're already on

Trigger Warning

There are some feels ahead



TECHNOLOGY

SPECIFIC TOOLS, KNOWLEDGE, AND SKILLS
TO KNOW AND PRACTICE



Technical shizzle wizzle

- * Here follows a list of technical stuff you should know before trying to find a job

Technical shizzle wizzle

- * Here follows a list of technical stuff you should know before trying to find a job
- * If you already have a job and don't know this stuff, then you should learn them

Technical shizzle wizzle

- * Here follows a list of technical stuff you should know before trying to find a job
- * If you already have a job and don't know this stuff, then you should learn them
- * It is by no means a comprehensive list

VERSION CONTROL

Version Control

- * via Wikipedia

Version Control is the management of changes to documents, computer programs, large web sites, and other collections of information.

Version Control

- * via Wikipedia

Version Control is the management of changes to documents, computer programs, large web sites, and other collections of information. Changes are usually ... blah blah blah blah

Version Control

- * via Wikipedia

Version Control is the management of changes to documents, computer programs, large web sites, and other collections of information. Changes are usually ... blah blah blah blah

- * Here's the bottom line: When two developers are working on the same project, version control is what keeps them from killing each other

Version Control

- * via Wikipedia

Version Control is the management of changes to documents, computer programs, large web sites, and other collections of information. Changes are usually ... blah blah blah blah

- * Here's the bottom line: When two developers are working on the same project, version control is what keeps them from killing each other
- * Version Control Systems (VCS) are an incredibly important aspect of working on a team. Sadly, it rarely receives much — if any exposure — in many academic settings.

Version Control II: The Search for Curly's Gold

- * There are lots of VCS out there but they all fall into two main types: Centralized and Distributed

Version Control II: The Search for Curly's Gold

- * There are lots of VCS out there but they all fall into two main types: Centralized and Distributed

Centralized - All the revision control happens on a central server that is the single master copy of the software

Version Control II: The Search for Curly's Gold

- * There are lots of VCS out there but they all fall into two main types: Centralized and Distributed

Centralized - All the revision control happens on a central server that is the single master copy of the software

Distributed - Distributed revision control conducts synchronization by exchanging patches (change-sets) from peer to peer

Git 'er done

- * So which one should you learn?

Git 'er done

- * So which one should you learn?
- * Like a lot of aspects of computer science, conceptual learning is more important than specific tool mastery

Git 'er done

- * So which one should you learn?
- * Like a lot of aspects of computer science, conceptual learning is more important than specific tool mastery
- * That said, learn Git.

Y U NO LERN GIT?

- * Git has fast become a major standard in distributed VCS and is now an expected part of the modern software developer's toolkit

Y U NO LERN GIT?

- * Git has fast become a major standard in distributed VCS and is now an expected part of the modern software developer's toolkit
- * Git is crazy fast, all but fail-proof, and cheap as **FREE!**

Y U NO LERN GIT?

- * Git has fast become a major standard in distributed VCS and is now an expected part of the modern software developer's toolkit
- * Git is crazy fast, all but fail-proof, and cheap as **FREE!**
- * The one downside is there's a non-trivial* learning curve to Git

Y U NO LERN GIT?

- * Git has fast become a major standard in distributed VCS and is now an expected part of the modern software developer's toolkit
- * Git is crazy fast, all but fail-proof, and cheap as **FREE!**
- * The one downside is there's a non-trivial* learning curve to Git
- * I took about four good tries before I finally understood Git

Gitting Git

- * There are tons of tutorials online out there to get your started

Gitting Git

- * There are tons of tutorials online out there to get your started
- * My favorite is [gitimmersion.com](https://www.gitimmersion.com)

Gitting Git

- * There are tons of tutorials online out there to get your started
- * My favorite is gitimmersion.com
- * It's a series of 50+ short labs that have you working with repos and doing all the stuff you'd need to do on a team

Git(Hub) 'er done

- * In addition to learning to use Git, you should have an account on github.com

Git(Hub) 'er done

- * In addition to learning to use Git, you should have an account on github.com
- * In case you're not familiar with it, GitHub is a web-based hosting service for software projects that use Git for its version control

Git(Hub) 'er done

- * In addition to learning to use Git, you should have an account on github.com
- * In case you're not familiar with it, GitHub is a web-based hosting service for software projects that use Git for its version control
- * GitHub has become the gold standard for open-source software projects

Git(Hub) 'er done

- * In addition to learning to use Git, you should have an account on github.com
- * In case you're not familiar with it, GitHub is a web-based hosting service for software projects that use Git for its version control
- * GitHub has become the gold standard for open-source software projects
- * It is also a great way to share examples of your code with potential employers

JAVASCRIPT

Pretty Much just JavaScript

- * It has gotten to the point where JavaScript is unavoidable

Pretty Much just JavaScript

- * It has gotten to the point where JavaScript is unavoidable
- * Even if you don't want to do program in it full time you should be conversant with it and its quirks

Pretty Much just JavaScript

- * It has gotten to the point where JavaScript is unavoidable
- * Even if you don't want to do program in it full time you should be conversant with it and its quirks
- * If you ever plan to program for the web, this is not optional

DATA STORAGE

Data Storage

- * Databases are fairly common topic in many academic settings so this is likely something you have some familiarity with

Data Storage

- * Databases are fairly common topic in many academic settings so this is likely something you have some familiarity with
- * That said, there are a bunch of different kinds of DBs out there that you should be aware of

Different Types of DBs

Relational - MySQL, PostGres (most common)

Different Types of DBs

Relational - MySQL, PostGres (most common)

GraphDBs - Not as common but worth at least reading up on

Different Types of DBs

Relational - MySQL, PostGres (most common)

GraphDBs - Not as common but worth at least reading up on

NoSQL - These have gained a lot of popularity in recent years and are worth knowing. Notable examples are CouchDB and MongoDB

THE ENGLISH LANGUAGE

No, that is not a joke

- * Programming is less about computers and more about language

No, that is not a joke

- * Programming is less about computers and more about language
- * If you suck at communicating, you will write uncommunicative code and the people that work on your code after you will want to throat-punch you

No, that is not a joke

- * Programming is less about computers and more about language
- * If you suck at communicating, you will write uncommunicative code and the people that work on your code after you will want to throat-punch you
- * Don't get throat punched. Learn to write. Learn to speak. Learn to edit

No, that is not a joke

- * Programming is less about computers and more about language
- * If you suck at communicating, you will write uncommunicative code and the people that work on your code after you will want to throat-punch you
- * Don't get throat punched. Learn to write. Learn to speak. Learn to edit (!!!)

That is a joke

- * Programming is less about computers and more about language
- * If you suck at communicating, you will write uncommunicative code and the people that work on your code after you will want to throat-punch you
- * Don't get throat punched. Learn to write. Learn to speak. Learn to edit (!!!)*****

That is a joke

- * Programming is less about computers and more about language
- * If you suck at communicating, you will write uncommunicative code and the people that work on your code after you will want to throat-punch you
- * Don't get throat punched. Learn to write. Learn to speak. Learn to edit (!!!)*****\$\$\$\$\$

That was a joke

- * Programming is less about computers and more about language
- * If you suck at communicating, you will write uncommunicative code and the people that work on your code after you will want to throat-punch you
- * Don't get throat punched. Learn to write. Learn to speak. Learn to edit (!!!)*****\$\$\$\$\$WTFBBQ!!!1

Wherein Our Hero Nearly Says Something Stupid

So there's an addendum to this topic on communication that I've thought a lot about. It's kind of a touchy subject so I want to spend some extra time with it. I might say something that makes me sound like an asshole. Just hear me out before you start throwing things.



Communication Breakdown

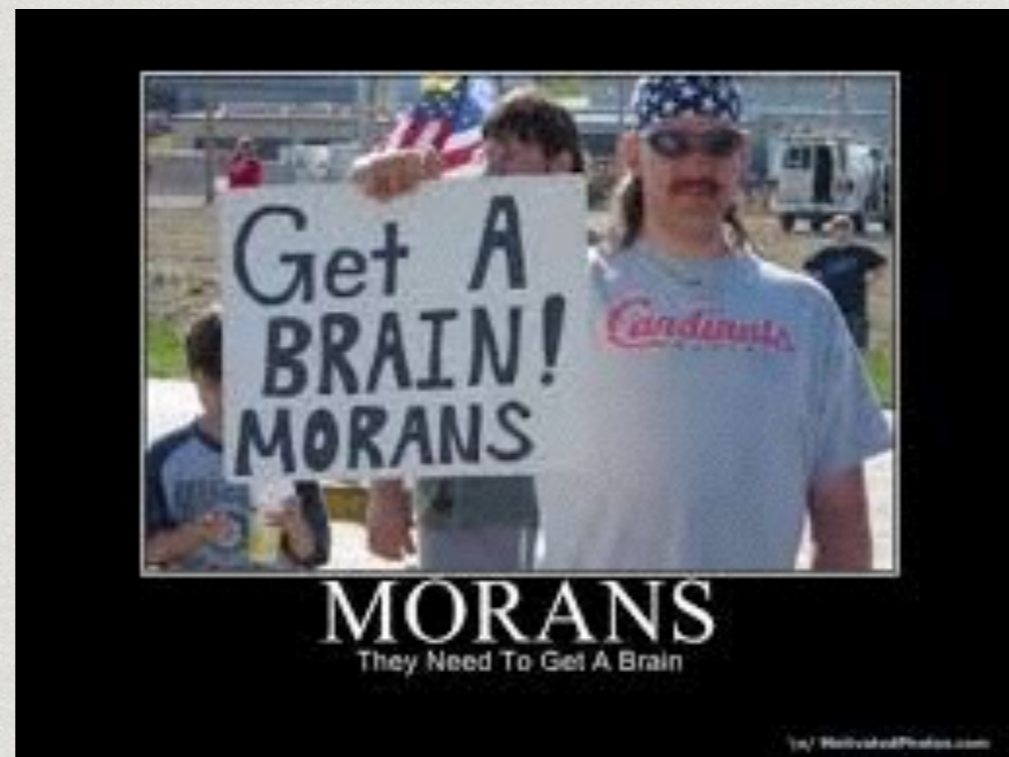
- * The burden of being understood lies with you

Communication Breakdown

- * The burden of being understood lies with you
- * To me, this means you may need consider the severity of your accent, if you have one.

Communication Breakdown

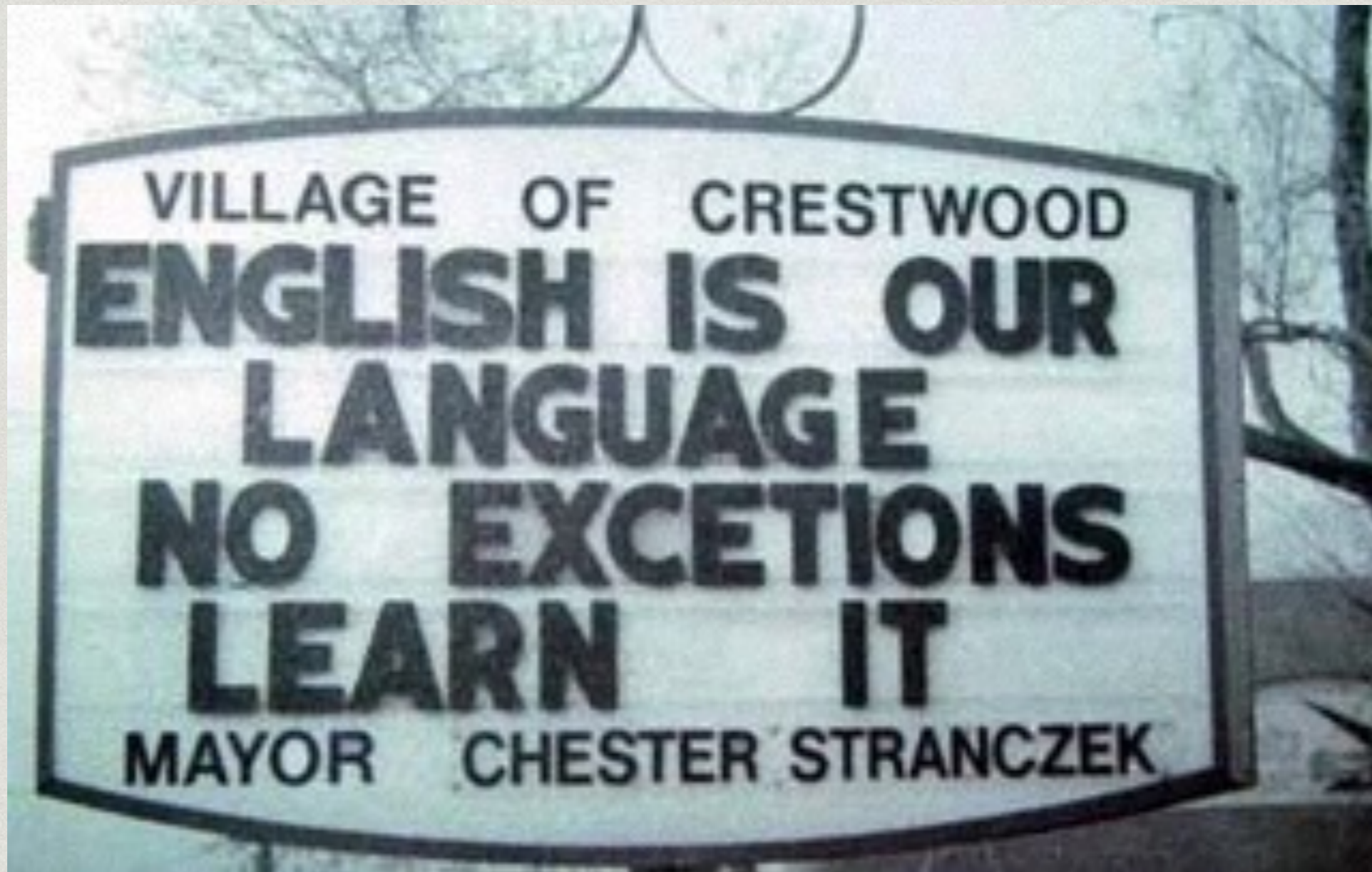
- * The burden of being understood lies with you
- * To me, this means you may need consider the severity of your accent, if you have one.
- * Now wait! This is not what I'm going for



Or this



There are a lot of these on the internet



Check me out while I clarify

- * The only point I'm making is that it is on you to be understandable to your teammates.

Check me out while I clarify

- * The only point I'm making is that it is on you to be understandable to your teammates.
- * Remember, good code is communicative. You also need to be able to form your ideas and thoughts succinctly to your team.

Check me out while I clarify

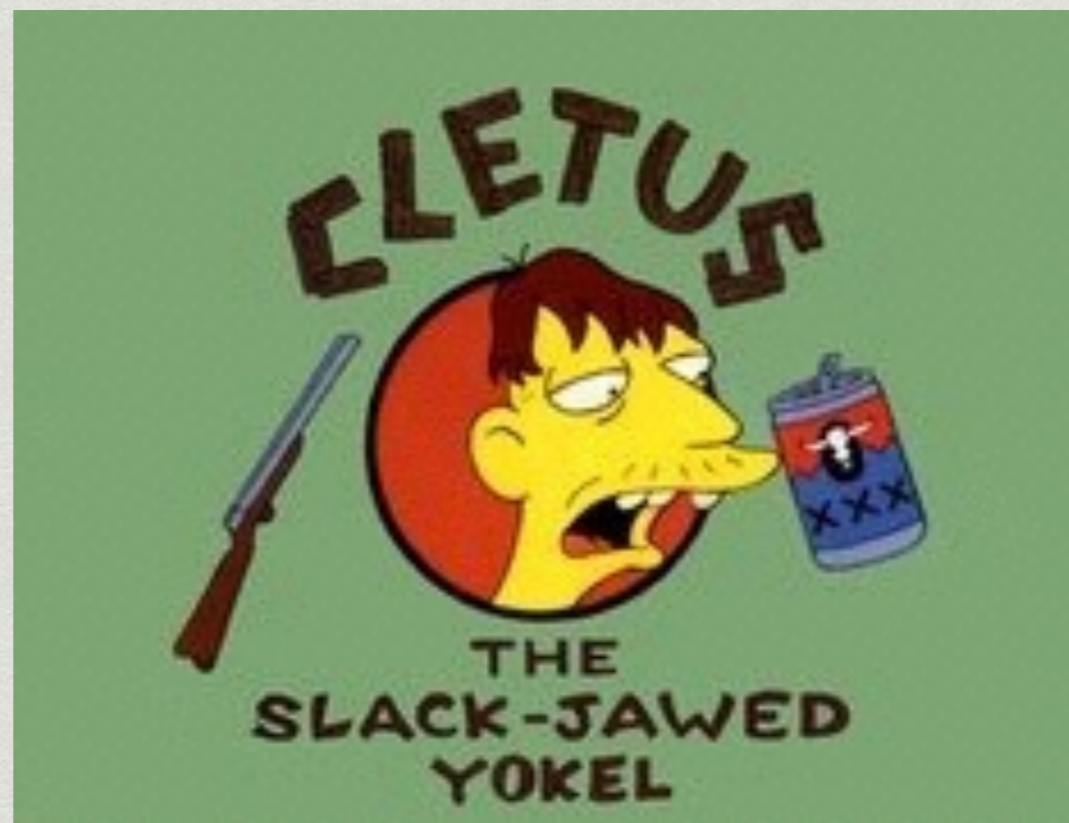
- * The only point I'm making is that it is on you to be understandable to your teammates.
- * Remember, good code is communicative. You also need to be able to form your ideas and thoughts succinctly to your team.
- * I've been on teams where the language barrier was almost insurmountable.

Check me out while I clarify

- * The only point I'm making is that it is on you to be understandable to your teammates.
- * Remember, good code is communicative. You also need to be able to form your ideas and thoughts succinctly to your team.
- * I've been on teams where the language barrier was almost insurmountable.
- * To a certain extent you learn to 'translate' thick accents, but you shouldn't have to.

What a twist!

Note, I include regional accents and dialects in this. I'm originally from the South and there are people I know who may as well be speaking another language to someone who speaks ESL.



When you point a finger...

The flip side of this is, as 'Muricans, we native speakers need to do our best to actively listen to our teammates, try to bridge any communication problems. The speaker may be self-conscious about being understood in an non-native language.



CS FUNDAMENTALS

More like FUN-damentals

- * For instance, you should know what arrays, stacks, queues, binary trees, linked lists, bitmaps, and Bloom filters all are

More like FUN-damentals

- * For instance, you should know what arrays, stacks, queues, binary trees, linked lists, bitmaps, and Bloom filters all are
- * Confession time: / don't even know what all that stuff is

More like FUN-damentals

- * For instance, you should know what arrays, stacks, queues, binary trees, linked lists, bitmaps, and Bloom filters all are
- * Confession time: / don't even know what all that stuff is
- * But I'm learning and so should you

Fundamentals 2: Pig in the City

- * This may seem like a weird one if you have a degree in Computer Science or Engineering

Fundamentals 2: Pig in the City

- * This may seem like a weird one if you have a degree in Computer Science or Engineering
- * But here is a secret

Fundamentals 2: Pig in the City

- * This may seem like a weird one if you have a degree in Computer Science or Engineering
- * But here is a secret
- * Not everyone* in this field has a degree in Computer Science or Engineering
 - * - Where 'Not Everyone' equals a lot of people

Fundamentals 3: The Last Stand

- * But ignorance is no excuse, so even if you don't have a CS degree, you still need to know this stuff. It's all still really important

Fundamentals 3: The Last Stand

- * But ignorance is no excuse, so even if you don't have a CS degree, you still need to know this stuff. It's all still really important
- * I've *personally* blown some otherwise awesome interviews because of these exact holes in my knowledge base

Fundamentals 3: The Last Stand

- * But ignorance is no excuse, so even if you don't have a CS degree, you still need to know this stuff. It's all still really important
- * I've *personally* blown some otherwise awesome interviews because of these exact holes in my knowledge base
- * Don't be like me; learn this stuff. Don't blow awesome interviews.

Fundamentals IV: The Quest for Peace

So where should you start?

Fundamentals IV: The Quest for Peace

So where should you start?

**Data Structures: Arrays, Stacks, Queues, Binary Trees,
Linked Lists, Bitmaps, and Bloom filters**

Fundamentals IV: The Quest for Peace

So where should you start?

Data Structures: Arrays, Stacks, Queues, Binary Trees, Linked Lists, Bitmaps, and Bloom filters

Algorithms and the application thereof: More importantly, knowing about the constant, linear, quadratic, logarithmic, 'Big O' implications of these operations. Ditto for space

Fundamentals IV: The Quest for Peace

So where should you start?

Data Structures: Arrays, Stacks, Queues, Binary Trees, Linked Lists, Bitmaps, and Bloom filters

Algorithms and the application thereof: More importantly, knowing about the constant, linear, quadratic, logarithmic, 'Big O' implications of these operations. Ditto for space

Robert LaFore's, Data Structures & Algorithms in Java is a very good place to start. There are also free courses online from places like Stanford and MIT that you can go through.

Technology Wrap Up

- * Learn Git



Technology Wrap Up

- * Learn Git
- * Get a GitHub profile



Technology Wrap Up

- * Learn Git
- * Get a GitHub profile
- * Learn JavaScript, even if you don't want to



Technology Wrap Up

- * Learn Git
- * Get a GitHub profile
- * Learn JavaScript, even if you don't want to
- * Hone your communication skills, all of them



Technology Wrap Up

- * Learn Git
- * Get a GitHub profile
- * Learn JavaScript, even if you don't want to
- * Hone your communication skills, all of them
- * Get your CS Fundamentals down solid



MEATSPACE

INTERPERSONAL SKILLS,
TRUFAX™, AND THINGS TO
DO THAT WILL GENERALLY
MAKE YOUR LIFE EASIER



GET INVOLVED

Here begins the TruFax™

- * Even if you're getting a 4-year degree from a respected university. They won't teach you everything you need to know.

Here begins the TruFax™

- * Even if you're getting a 4-year degree from a respected university. They won't teach you everything you need to know.
- * One great way to fill in the cracks in your knowledge is to get involved in the local developer community.

Why Get Involved

- * Brain Knowledge — Colleges can't teach you everything and their instruction will likely be dated, technology moves crazy fast

Why Get Involved

- * Brain Knowledge — Colleges can't teach you everything and their instruction will likely be dated. technology moves crazy fast
- * Contacts — You get to meet/learn-from some of the smartest folks in your area

Why Get Involved 2: Electric Boogaloo

- * Job Opportunities — Lots of developer events are sponsored by businesses who are hiring

Why Get Involved 2: Electric Boogaloo

- * Job Opportunities — Lots of developer events are sponsored by businesses who are hiring



Ways to Get Involved

- * Join a User Group — Groups of enthusiasts around a certain technology. Usually feature short talks on specific subjects. — You're doing an excellent job with this one

Ways to Get Involved

- * Join a User Group — Groups of enthusiasts around a certain technology. Usually feature short talks on specific subjects. — You're doing an excellent job with this one
- * Go to a Code Retreat — A developer practice event where developers get together to practice development. Code Craftsman Saturdays!

Ways to Get Involved

- * Join a User Group — Groups of enthusiasts around a certain technology. Usually feature short talks on specific subjects. — You're doing an excellent job with this one
- * Go to a Code Retreat — A developer practice event where developers get together to practice development. Code Craftsman Saturdays!
- * Attend a developer conference — You're doing an excellent job with this one

Ways to Get Involved

- * Join a User Group — Groups of enthusiasts around a certain technology. Usually feature short talks on specific subjects. — You're doing an excellent job with this one
- * Go to a Code Retreat — A developer practice event where developers get together to practice development. Code Craftsman Saturdays!
- * Attend a developer conference — You're doing an excellent job with this one
- * I have lots of others, just ask. Which leads me to...

LEARN TO ASK FOR HELP

More TruFax™

- * You don't know everything

More TruFax™

- * You don't know everything
- * I don't know everything

More TruFax™

- * You don't know everything
- * I don't know everything
- * Nobody knows everything

More TruFax™

- * You don't know everything
- * I don't know everything
- * Nobody knows everything
- * When you don't know something it's okay to ask for help

You know nuffin' Jon Snuh

- * It can be intimidating but it's totally cool

You know nuffin' Jon Snuh

- * It can be intimidating but it's totally cool
- * We've all been there

You know nuffin' Jon Snuh

- * It can be intimidating but it's totally cool
- * We've all been there
- * We are all still there

You know nuffin' Jon Snuh

- * It can be intimidating but it's totally cool
- * We've all been there
- * We are all still there
- * We will all be there forever

You know nuffin' Jon Snuh

- * It can be intimidating but it's totally cool
- * We've all been there
- * We are all still there
- * We will all be there forever
- * If you're the kind of person who simply has to have all the answers, all the time — you should find another job

If you need help... try this



It can be a little humbling...

But

It can be a little humbling...

But

A. That's good! Humility begets a good attitude for working on a team

It can be a little humbling...

But

- A. That's good! Humility begets a good attitude for working on a team
- B. The more you ask, the more you will know

It can be a little humbling...

But

- A. That's good! Humility begets a good attitude for working on a team
- B. The more you ask, the more you will know
- C. The more you know, the more you can...

LEARN TO HELP OTHERS

The Flip Side of Asking

- * When you *do* know something, teach it to others

The Flip Side of Asking

- * When you *do* know something, teach it to others
- * You will gain a far deeper understanding of that thing

The Flip Side of Asking

- * When you *do* know something, teach it to others
- * You will gain a far deeper understanding of that thing
- * This is the dirty secret of teaching, it's a tool for you (the teacher) to learn something over and learn it better

Science!

Dale's Cone of Experience

Some smart guys in Texas did a study on memory, their conclusions were that, in general people remember:

Science!

Dale's Cone of Experience

Some smart guys in Texas did a study on memory, their conclusions were that, in general people remember:

1. 10% of what they read

Science!

Dale's Cone of Experience

Some smart guys in Texas did a study on memory, their conclusions were that, in general people remember:

1. 10% of what they read
2. 20% of what they hear

Science!

Dale's Cone of Experience

Some smart guys in Texas did a study on memory, their conclusions were that, in general people remember:

1. 10% of what they read
2. 20% of what they hear
3. 30% of what they see

Science!

Dale's Cone of Experience

Some smart guys in Texas did a study on memory, their conclusions were that, in general people remember:

1. 10% of what they read
2. 20% of what they hear
3. 30% of what they see
4. 50% of what they see and hear (See what I did there?)

Science!

Dale's Cone of Experience

Some smart guys in Texas did a study on memory, their conclusions were that, in general people remember:

1. 10% of what they read
2. 20% of what they hear
3. 30% of what they see
4. 50% of what they see and hear (See what I did there?)
5. 70% of what they say

Science!

Dale's Cone of Experience

Some smart guys in Texas did a study on memory, their conclusions were that, in general people remember:

1. 10% of what they read
2. 20% of what they hear
3. 30% of what they see
4. 50% of what they see and hear (See what I did there?)
5. 70% of what they say
6. 90% of what they say and do (Teaching)

Where can you practice?

- * TA with group that teaches kids or adults interested in learning to code

Where can you practice?

- * TA with group that teaches kids or adults interested in learning to code
- * Teach a class at your local library. They eat that kind of thing up.

Where can you practice?

- * TA with group that teaches kids or adults interested in learning to code
- * Teach a class at your local library. They eat that kind of thing up.
- * Show a friend the basics of coding

MOST IMPORTANTLY

The Most Important Thing

The *Most* Important Thing

Don't be an Asshole

Seriously. It is that simple.

- * This probably seems like I'm just making another joke

Seriously. It is that simple.

- * This probably seems like I'm just making another joke
- * I am not

Seriously. It is that simple.

- * This probably seems like I'm just making another joke. I am not
- * You could be wonderfully talented and knowledgable, but if you're a pain to get along with ...



Pop Quiz Hot Shot

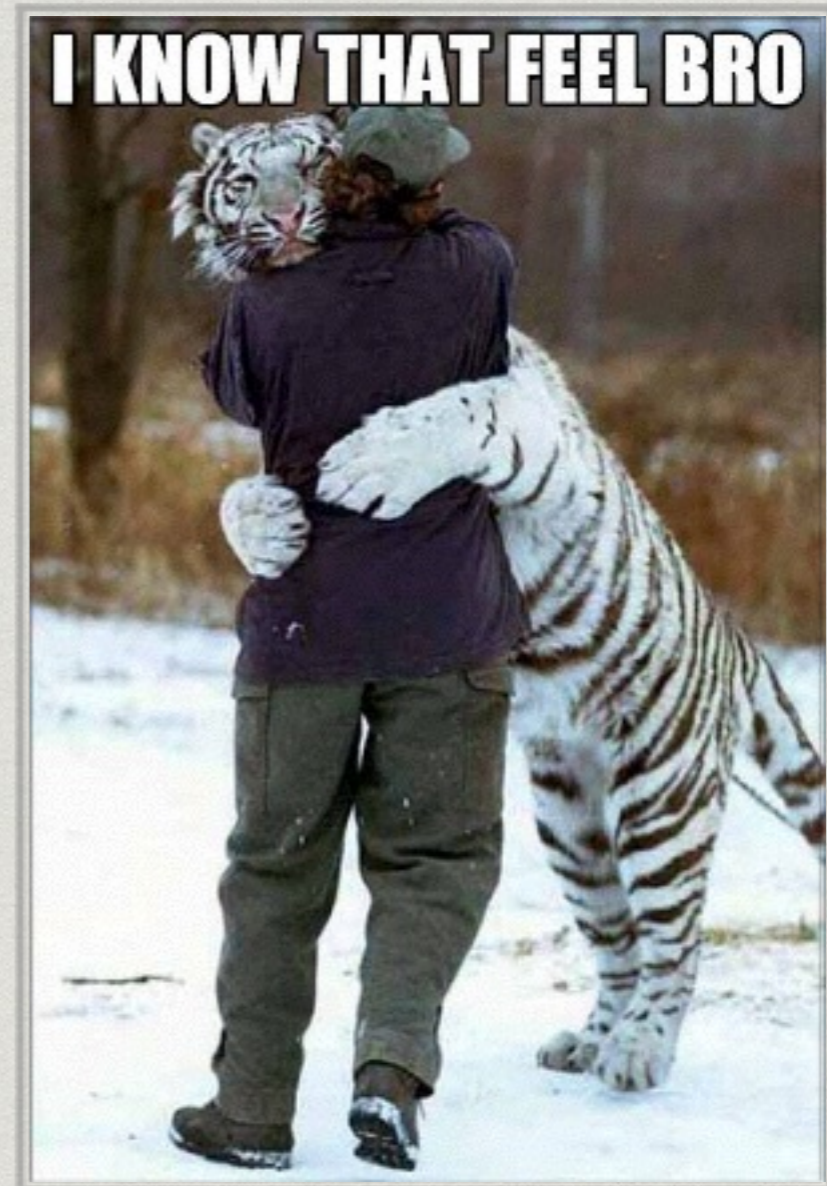
- * Do you every worry that you're being an Asshole?

Pop Quiz Hot Shot

- * Do you every worry that you're being an Asshole?
- * Congrats! You're probably not. Self-reflection is the path to non-assholery

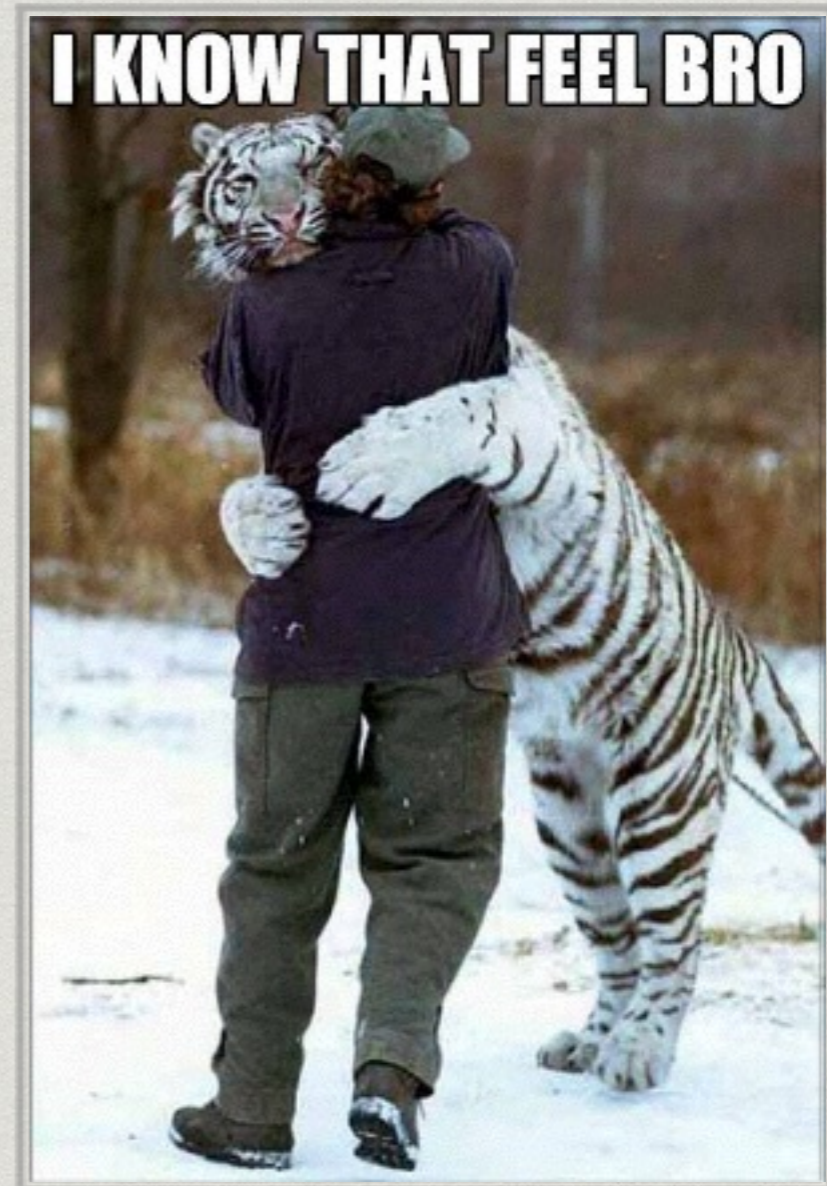
Meatspace Wrap

- * Get Involved
 - * User Groups, Code Retreats, & Conferences!



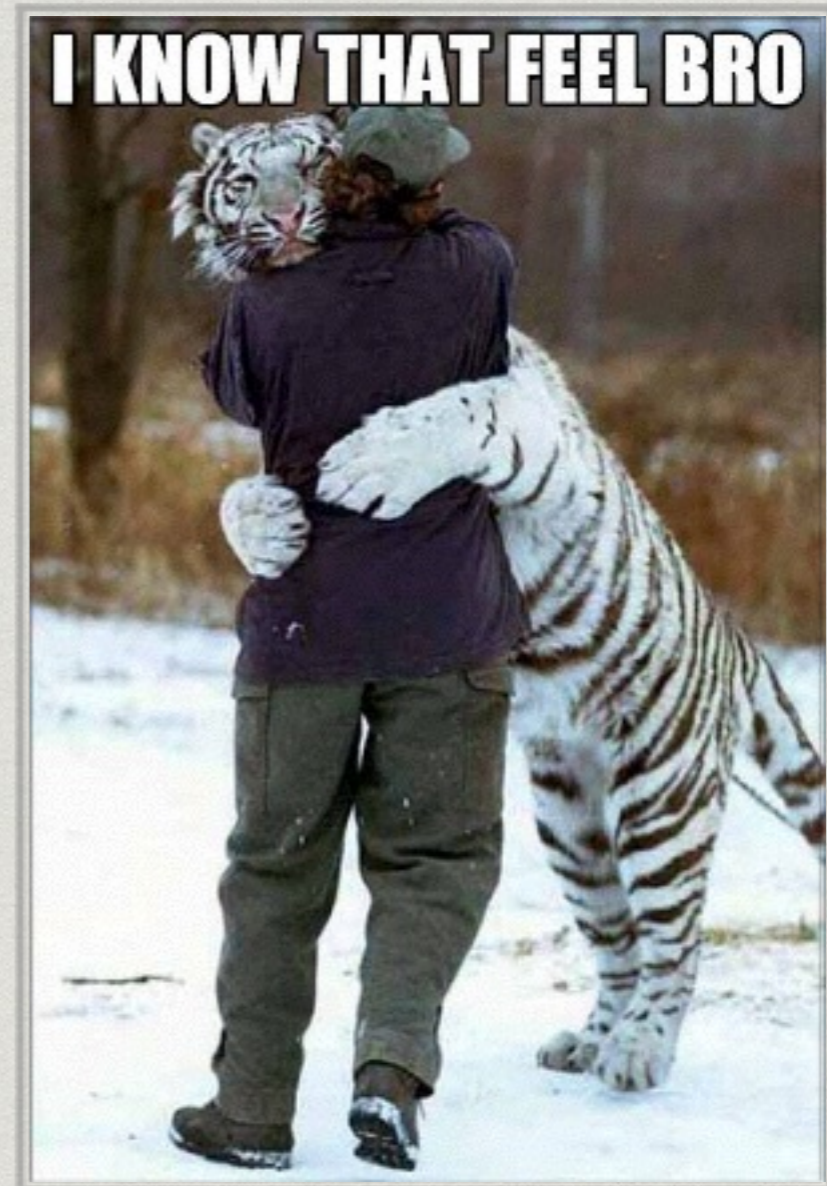
Meatspace Wrap (Mmm...)

- * Get Involved
 - * User Groups, Code Retreats, & Conferences!
- * Learn to Ask for Help
 - * Swallow your pride



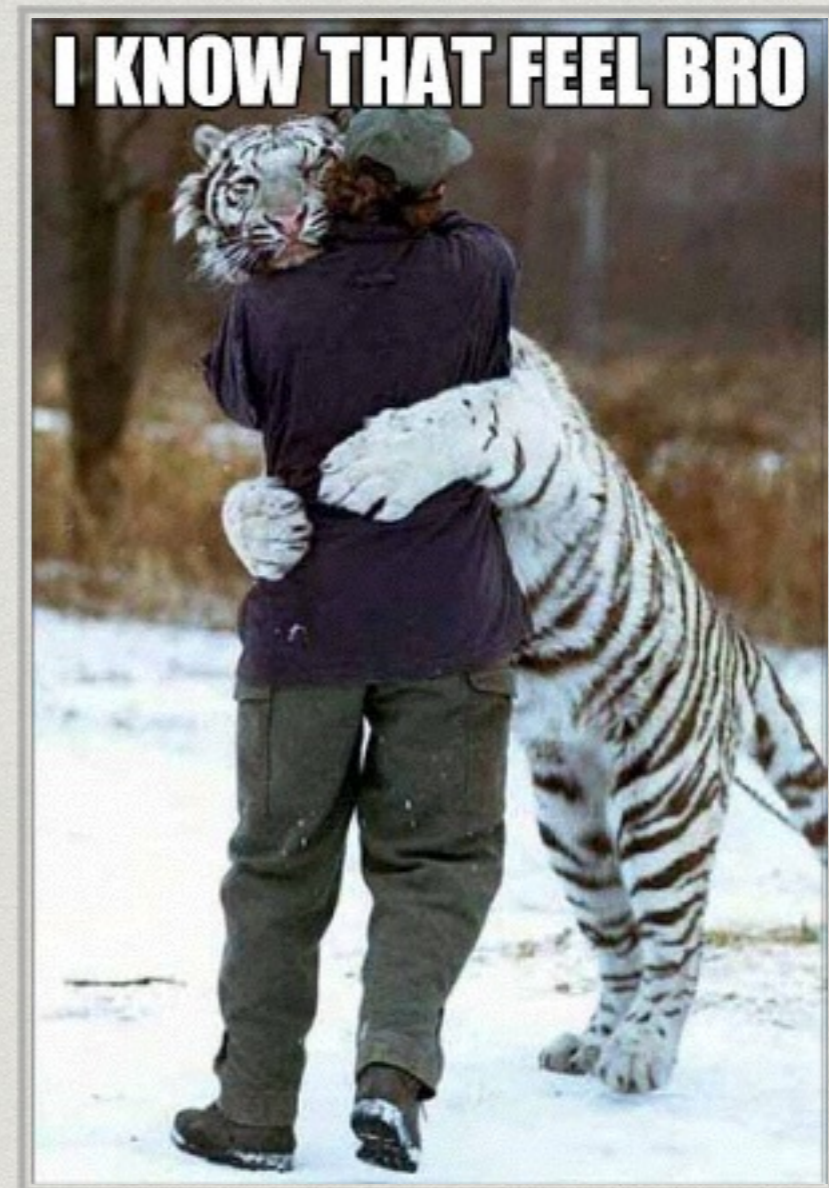
Meatspace Wrap (Mmm...)

- * Get Involved
 - * User Groups, Code Retreats, & Conferences!
- * Learn to Ask for Help
 - * Swallow your pride
- * Learn to Help Other People



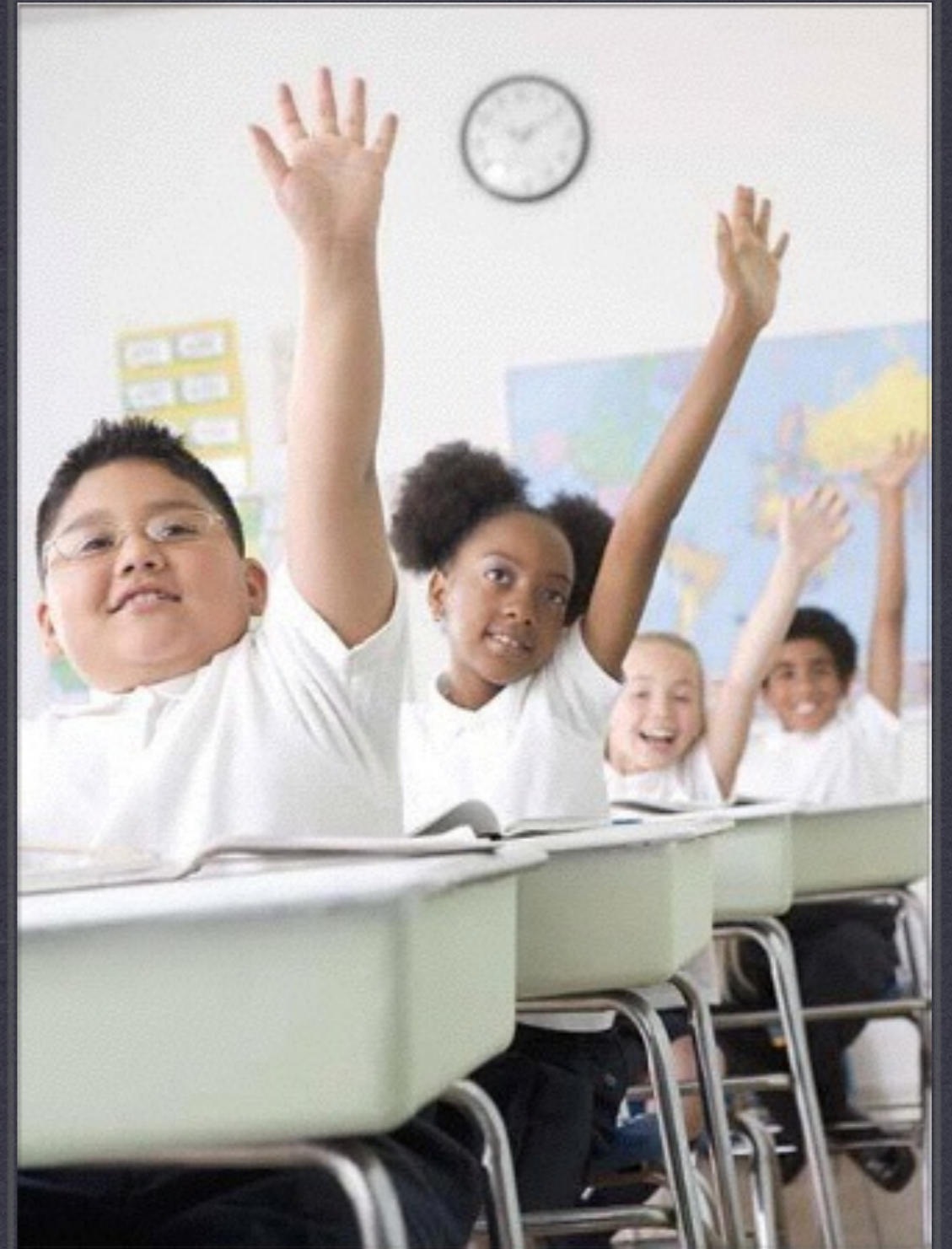
Meatspace Wrap (Mmm...)

- * Get Involved
 - * User Groups, Code Retreats, & Conferences!
- * Learn to Ask for Help
 - * Swallow your pride
- * Learn to Help Other People
- * *Most Important!*
 - * Remember Wheaton's Law



FIN

**QUESTION AND
ANSWER TIME!**



I was James York!

- * If you have questions or want advice or just want to tell me my hair is dumb. Get in touch!
- * @kroysemaj, @DetCraftsman
- * `“JamesYork”.reverse()` //not a real function
 - * JUST FYI. I measure my self-worth by my number of twitter followers.
- * Email - kroysemaj@gmail.com
- * github - kroysemaj
- * other... you get the idea.

THANK YOU!