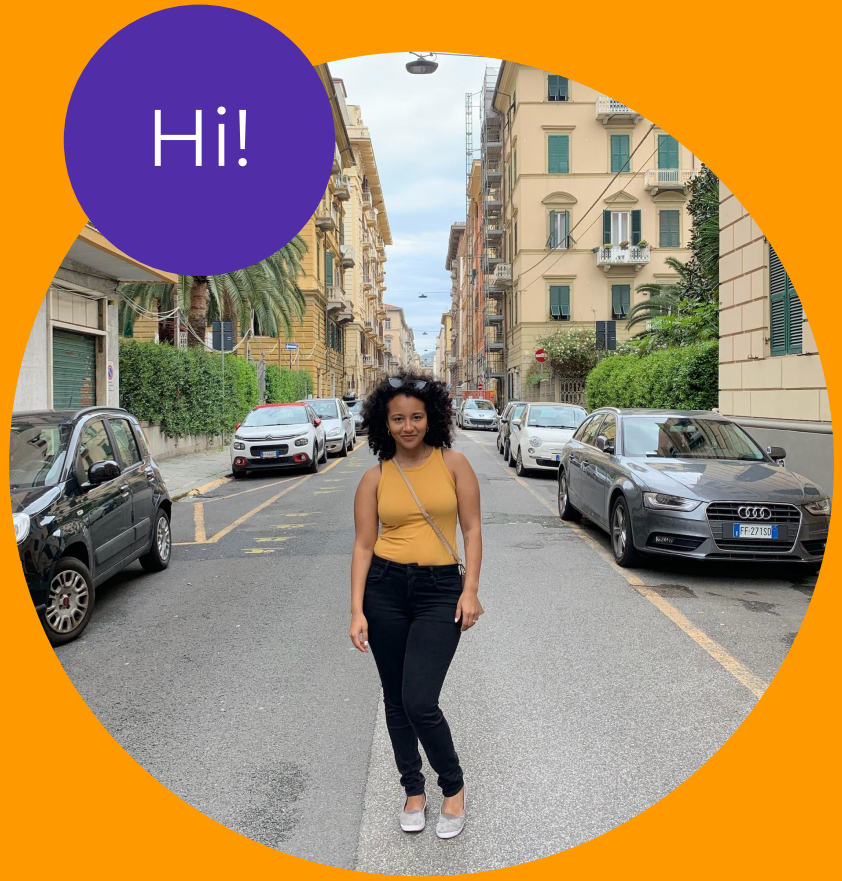
Accelerating Growth as a Developer

Taylor **Dennis**
twittter: **@tay_lurre**

About Me

- Michigan State Alum
- Non traditional background
  - 2 Job Life
- How I got into development
- Software Engineer at ShopRunner
  - Apprentice => Engineer
- JavaScript



Agenda

01

Working with Legacy Code

Why documentation is important, building empathy, using design patterns

02

Sharing what you learned (are learning)

The different ways to share what you learned, asking questions, and will you ever know it all?

03

Know your impact

Digging into how you provide value as a developer

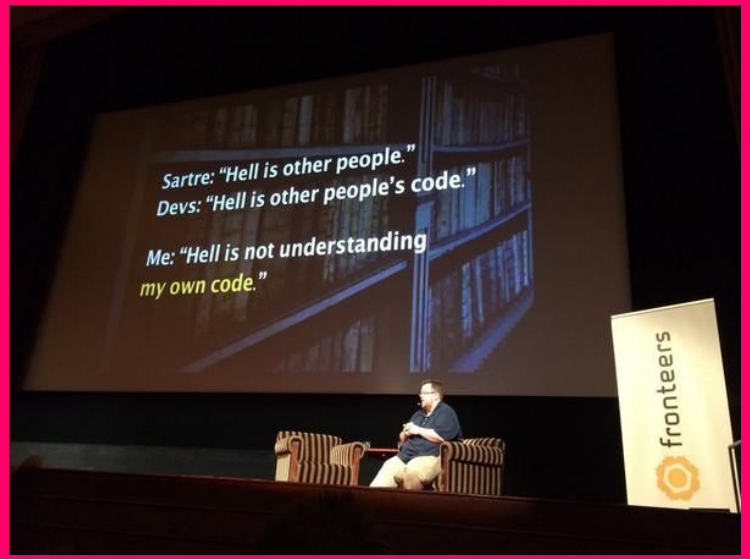
04

Find your tribe

Quick Dive into the types of people you should have in your tribe

01

Working with Legacy Code



Empathy through (Git) Blame

- Why did they use this pattern?
- This code is not clean
- How does this even work?
- Who introduced this bug?
- This naming convention makes no sense
- You could have written this code. Be the change you want to see

Documentation

is Important

Documenting in the code
Can be used to for API Docs, or provide reasoning for hack

Documenting for buy in
Providing a line of communication before implementing a major change

Documenting for self
Keeping track of what you worked on and are working on

Pull Request as documentation
Documenting what you did, making small readable pull requests, and adding quality steps to reproduce

Documenting for buy in

Action: Highlight action(s) you or the team are taking
Problem: Highlight the problem being solved
Solution: What steps are you taking to mitigate the problem
Benefits: What benefits does using the solution above provide
Considerations: (Optional only would recommend if you had an evaluation with a team first) Anticipating questions or counter solutions and why that approach wasn't taken

Documenting Pull Requests

- **Issue addressed**
- **Solution/ Technical**
- **Acceptance Criteria Met**
- **Steps to test the solution (if applicable)**

Documenting For Self

- **What did you work on?**
- **Any Blockers?**
- **Any new learnings?:** Articles you read
- **Any new computer science concepts learned?**

Creational

Structural

Behavioral

- Know what pattern you want to use and why
- Diagram or reason out your design before implementing
- Understand anti-patterns and why they may have been used in legacy code

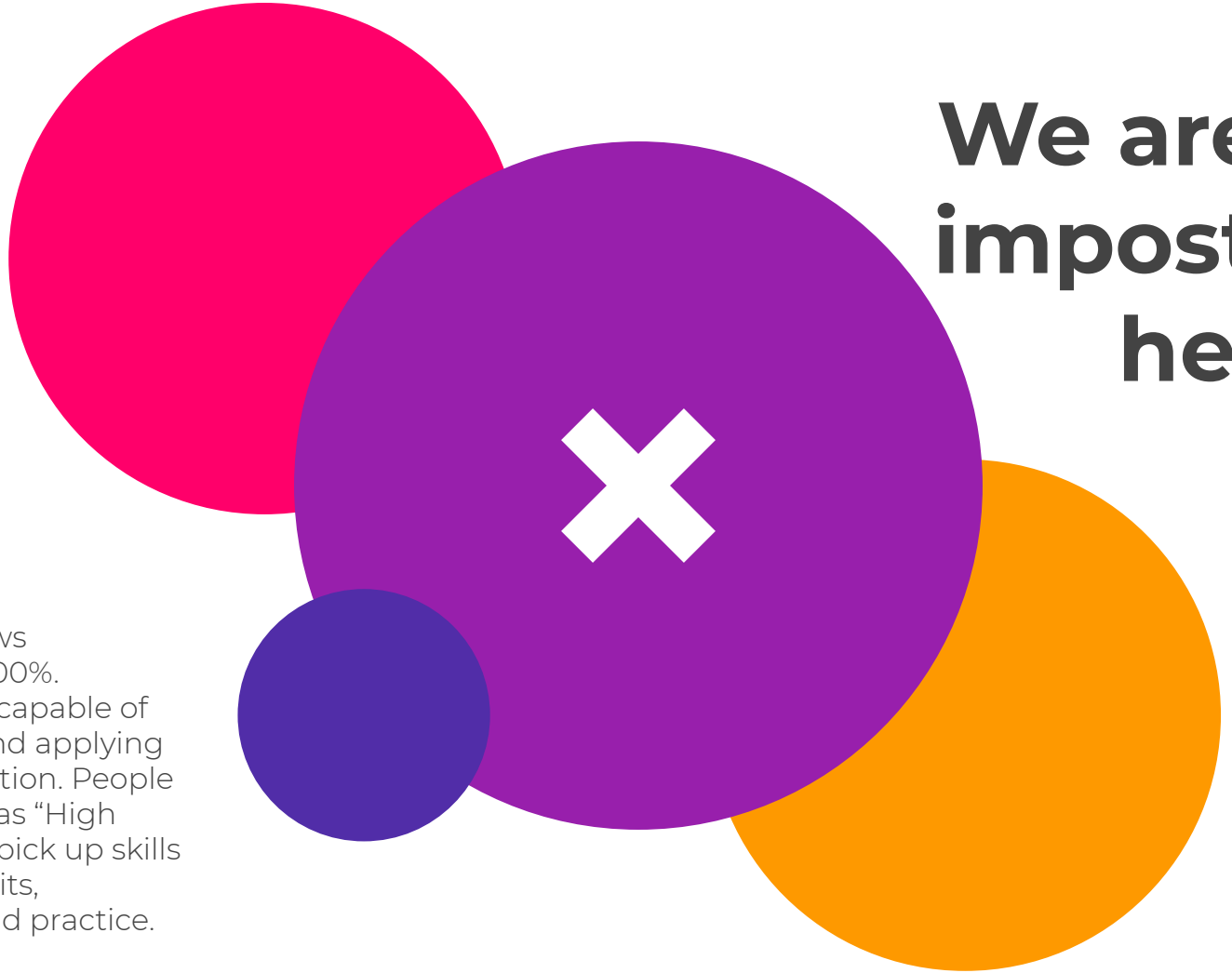
Resources

- https://addyosmani.com/resources/essentialjsdesignpatterns/book/
- https://www.pluralsight.com/courses/javascript-practical-design-patterns

02

Sharing what you learned (are learning)

We are all imposters here...



No one knows everything 100%. **Everyone** is capable of absorbing and applying new information. People we perceive as “High performers” pick up skills through habits, repetition and practice.

Sharing what you learned

(You don’t have to do all these things)

Write a blog

Commit it to your memory and share your voice

Give a talk

Internal talk at work, among peers

Tweet

There is a very engaging community of tech people on twitter

Code it

Taking what you learned and apply it and leave comments in your code or PR for yourself

Go to a meetup

Collaborate with others, pair program organically

Be Patient with yourself

No topic is too small to talk about

03

Know your impact

What is your ~~code~~ **impact**

- Peers
- The Organization
- Other Code
- Not everything you contribute will be in code

04

Find your tribe

It takes a village, here are **three** types of people you should have in your tribe

Peers

- Healthy peer relationships with people that challenge how you may approach a problem in a constructive way.
- Don't be afraid to ask why or have them provide feedback.

Mentors

- Servant Leadership
- Provide habits of someone that is efficient
- Ask questions and observe to pick up some of those habits

Advocates

- People you feel comfortable sharing your experiences in front that will acknowledge your experience
- People that can advocate on your behalf when you aren't in the room

Your **confidence** as a Developer should not come from your ability to have the answers right way**.** It should come from your ability to get to the correct **solution**.”

—Someone famous