28. Imposter Syndrome

Different terms used to describe Imposter Syndrome

- Impostor phenomenon
- Impostor experience
- Impostor syndrome

What is Imposter Syndrome?

The idea that you've only succeeded due to luck, and not because of your talent or qualification. This includes doubting your abilities and feeling like a fraud.

Your journey

Whenever you are doubtful of your skill or abilities. Remember how far you have come across. Be proud of your progress, however little it may be.

5 hats of Imposter Syndrome

- The Perfectionist - set excessively high goals for themselves
- The Superwoman/man - trying to do extra work
- The Natural Genius - trying to learn/do something quickly
- The Soloist - trying to do/solve problems alone
- The Expert - trying to know or master everything

It's important to identify what shape your impostor syndrome is taking and taking corrective actions to overcome it's effects on you.

Overcoming Impostor Syndrome

- acknowledging & accepting the feelings or thoughts
- value constructive criticism
- ask for help
- continue to practice the skill
- know what you don't know

continue to practice code & upskill

You must continue to practice to code wether they are personal hobby projects, or learning tutorials. They key to reducing your impostor syndrome is making continuous effort at learning and practice to become more confident of your skill and craft.

Don't Forget Where you Started

One day, not so long ago, you were just starting. A beginner, that wouldn't be able to pick out an integer from a float. Now you've written full-stack applications, have worked in a team using Git, have not only talked the talk but are walking the walk.

When your fingers are at a keyboard, magic happens. An empty folder is quickly populated with a city of files filled with business of data, and citizens of functions. From nothing: something!

This should not be taken lightly. Don't just look, but feel where you are now compared to that humble beginning. Say it aloud:

> I can write programs. I know multiple programming languages. I can read documentation. I can build a web application, and already have!
> Like one of your favourite booleans, that statement is `true`.
> Spend real time looking back to admire your progress, reflect on the present and what needs to be done, and keep an eye to what is yet to come!

Where You're Going

Seniors don't know everything. They oft' just know how to get to an answer faster. They've learned how to learn, how to parse documentation and take the important bits quickly. If you do something everyday, sure you'll memorize the commands you use all the time, but more important than that in this forever shifting landscape is your ability to adapt.

Be willing to try, have something not work, and to try again. Experiment. `git commit`, break things, and roll back. Try things! Keep trying things until it works. Persistance and tenacity are the key.

Be Proud

- You've come a long way.

- You've grown, and learned, a lot!
- You **are** a developer! No question.
    - You HAVE written code that works.
    - You've built a project with a team.
    - You've built a web application.
    - You **ARE** a developer! Say it aloud, and let it sink in.

Network

- Volunteer for local, national, and online groups.
    - Volunteer as a mentor, or teacher.
    - Volunteer to do a talk on a tech or programming topic.
- Attend events (since COVID started, many are available online now too!)
    - Experience new and exciting topics curated by industry leaders.
    - Socialize after any talk or event to meet new developers and potential employers.
- Join web developer Slack groups (check your town, your province, and for national groups.)
    - Here you'll meet people!
    - Ask for help.
    - Offer help!
    - Advertise your talks!
    - Hear all sorts of perspectives from different developers.
- Consider Canada Learning Code, and have a look at MeetUp!

Resources

- Lighthouse Labs Breakout: Learning How to Learn and Problem Solving
- DevDocs.io
- Offline Documentation
    - MacOS: Dash
    - Windows: Velocity
    - Linux: Zeal
- Ladybug Podcast
- Rubber Duck Debugging
- Note-Taking

    - Obsidian
- Time Tracking
    - TickSpot
    - Harvest
    - There are Likely Free Alternatives...
    - Simpler the better, you just want to know how long certain types of tasks and projects are taking you!
- The Mythical Man Month
- Agile versus Waterfall

Be...

- Adaptable
- Willing-to-Learn
- a Team Player
- Tenacious
- ***You!***

After all, you've made it this far. If you keep going, where will you end up? Let's find out.

Keep on keepin' on.

Psst... looking to get a head-start on your dictionary?

Try springboarding from a list like this:

- Front-end Technologies
    - HTML5
    - CSS3
    - Fonts
    - Box Model
    - Flex
    - Grid
    - Media Queries
    - Common Frameworks (Bootstrap, Tailwind)
    - Common Language Extensions (Sass, Less)
    - JavaScript

    - ES Modules (import, export)
    - Ajax
    - Common Libraries (jQuery, React)
    - Common Language Extensions (TypeScript, CoffeeScript)
- Back-end Technologies
    - Node.js (CommonJS syntax)
    - Web Server
    - Database
    - SQL Language (CRUD, JOINs)
    - Relationships (One-to-One, One-to-Many, Many-to-Many)
    - ERD
    - Scripts (Interpreted Programming Language Files or Compiled Executables)
    - MVC
    - Routing
- Language Concepts and Constructs
    - Variables and Constants
    - Data-Types (Boolean, String, Integer, Float / Double)
    - Arithmetic Operators (`+` , `-` , `*` , `/` , `%`)
    - Comarison Operators (`==` , `<` , `>`)
    - Indexed Arrays / Lists
    - Associative Arrays / Maps / Hashes
    - Functions
    - Parameters
    - Return
    - Arguments
    - Envocation
    - Classes

View on Compass | Leave Feedback