

The 10X Developer

101 TIPS

**for becoming a better
developer**

HARLEY FERGUSON

HARLEY FERGUSON

101 Tips For Becoming A Better Developer

101 tips that will turn you into a 10x developer.



First published by The 10X Developer 2023

Copyright © 2023 by Harley Ferguson

All rights reserved. No part of this publication may be reproduced, stored or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise without written permission from the publisher. It is illegal to copy this book, post it to a website, or distribute it by any other means without permission.

Harley Ferguson asserts the moral right to be identified as the author of this work.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book and on its cover are trade names, service marks, trademarks and registered trademarks of their respective owners. The publishers and the book are not associated with any product or vendor mentioned in this book. None of the companies referenced within the book have endorsed the book.

First edition

This book was professionally typeset on Reedsy.

Find out more at reedsy.com

For all the developers who want to be their best.

Contents

1	Essential	1
2	Coding	4
3	Practical	7
4	Environment	10
5	Team	13
6	Workspace	16
7	Growth	19
8	Mindset	22
9	Brand	25
10	Bucketlist	28
11	Finally...	31

1

Essential

“A problem well stated is a problem half solved.”

John Dewey

1. Learn clean code

Understanding the clean code philosophy will result in you writing better, more readable code.

2. Focus on problem solving

Writing code is just the tool. Problem solving is the actual skill. The better you get at problem solving, the better developer you'll become.

3. Done is better than perfect

The beauty of code is that it's never perfect. There is always a way to optimize it, make it more readable or more performant. Knowing when the problem is solved and not wasting more time on making changes that don't improve the product is important for any developer.

4. Find your niche

The software development industry is very broad. It bodes well to have a good understanding of all the areas to help round you out as a developer, but you'll get more opportunities, recognition and money if you become really good at one specific thing.

5. Learn how to learn

This industry is always evolving and so must you. Learning how to learn is crucial for any developer. Know what works best for you and use that whenever you're learning something new.

6. Take responsibility

For your code. For your work. For your career. It's up to you to become the best that you can.

7. Document your questions and answers

You're going to have thousands of questions in your career. It will save you time and make you more knowledgeable if you were to write down your questions, why you have them and ultimately, the answers. Problems have a way of repeating themselves, so why waste time finding the answer again when you could just reference your documentation?

8. Keep exploring

There are so many amazing tools, languages and stacks out there. Keep exploring and tinkering with things to help you find what area of the industry you really enjoy.

9. Stay informed

Stay in the loop with all of the latest trends, stacks and languages. Sometimes they stick so it's very useful to already have a good understanding of them. That'll put you ahead of others.

10. You vs. You

You're not in competition with anybody but yourself. Don't compare yourself with others. It's up to you to become the best version of yourself.

2

Coding

“Code is like humor. When you have to explain it, it’s bad.”

Cory House

11. Plan before you code

If you had a map for a maze, would you still run in blindly or would you use the map to navigate your way out? Writing code is one large maze. Plan your work which will create your map. Then start writing code.

12. Use a linter

Keeping your code formatting consistent saves you time and some eyesores.

13. Learn to debug

Knowing how to debug well will help you identify issues faster.

14. Write proper comments

In the codebase or in code review, comments go a long way to helping explain your thoughts. Be clear. Be concise.

15. Test your code

No item of work is truly complete until it's been thoroughly tested. Writing code to test your code is the sign of a truly great developer.

16. Write documentation

You know how your code works right now. That doesn't mean others will or that you'll remember how it works 6 months from now. Write documentation to save everyone time.

17. Understand deployments

It's crucial to understand how to get your code out into the world. Understanding deployments can help you to understand the full life cycle from first line of code written to first paying customer.

18. Understand best principles

Every aspect of code has best principles and they exist for a reason. Familiarize yourself and make use of them.

19. Write good commit messages

Explain what you did and why you did it. No more “I did things”. It’ll make everyone’s life easier.

20. Use ChatGPT

When in doubt, ask ChatGPT for the answer. Answers, code completion, technical documentation. The uses are endless and should be used by every developer to help increase their learning and output.

3

Practical

“First solve the problem. Then write the code.”

John Johnson

21. Use StackOverflow

For the longest time, StackOverflow has been the answer to 99.99% of developers' questions. Understanding how to navigate it, what to search for and how to find the best answer to question is a valuable skill.

22. Learn JavaScript fundamentals

JavaScript continues to be King Of The Internet. Knowing some (or a lot) of JavaScript will give you a lot of options in your career.

23. Learn how to deploy code

Understanding (and also creating) pipelines, setting up CI/CD is a great tool to have in your toolbox. Every bit of software needs to be released, so knowing how that works is crucial.

24. Understand different database types

Most platforms require a database. Knowing the difference between SQL, MongoDB, NoSQL, Collection/Document store etc. will help you make the best decision on what to use for your next project.

25. How to structure a database

After you've chosen a database, knowing how to best structure it is a skill on it's own. Spending time exploring how data should relate to one another and how to dictate those relationships is important.

26. Understand Git

Understanding the basics of Git can save you from a potentially catastrophic merge into the wrong branch.

27. Use coding tools

Tools like Postman, Fork and database visualizers are essential for making your life easier and increasing your productivity.

28. Understand how APIs work

Regardless if an API is RESTful or using GraphQL, knowing how APIs work and how to integrate with them is something every developer needs to know.

29. How to test your code

Functional, unit, end-to-end or automation tests. The more you know and can do, the better. Your team/company/clients/-partners will thank you for it.

30. How to write technical documentation

Writing code is sometimes easier than writing plain language to explain your code. However, it helps to provide context and explain the why and how of the code you wrote.

4

Environment

“Focus on being productive instead of busy.”

Tim Ferriss

31. Use productive extensions

Explore extensions for your IDE that will help make you more productive.

32. Limit notifications

Mute notifications during times where you really need to focus. Those emails and messages can wait.

33. Remove clutter

Have a clean desk. It's very easy for our desks to become cluttered, but keeping a clean desk can help our minds write clean code.

34. Remove distractions

Don't have distractions easily accessible. They'll take you out of your flow.

35. Maintain focus

Block out time to focus and get things done. Try to go for 30 minutes at a time. Have a goal, set the timer, get things done.

36. Invest in a good chair

Most developers are sitting for 8+ hours a day. Your body will thank you if you're able to acquire a chair that is comfortable and good for your posture.

37. Take a power nap

Coding is very intensive mentally and can cause fatigue. Take a 8-12 minute power nap in the middle of the day to recharge you for the next couple of hours.

38. Look after yourself

Eat right. Sleep enough. Exercise often. This will all make you mentally sharper, which results in you writing better code, making better decisions and having a better life.

39. Be comfortable

A good chair. A bottle of water. A pair of headphones. Some natural light. A good room temperature. All of these things help keep you comfortable, reduce you getting naturally distracted and induce focus. The more comfortable you are, the more productive you'll be.

40. Use a good machine

The better machine you have, the faster your code will compile and run. This might not be a possibility for everyone, but a better machine generally means you're saving more time.

5

Team

***“Talent wins games, but teamwork and intelligence
win championships.”***

Michael Jordan

41. Find a mentor

Having a mentor can help guide you in your career, learn from the mistakes of others and have someone to help chat through ideas.

42. Find a mentee

Having a mentee can help increase your leadership skills, pass on your knowledge to others and provide serious value to the mentee's career.

43. Communicate well

When you're in a team, communication is more important than anything else. Get really good at listening and speaking to your team.

44. Be open

We are all human. Being open when you're having a bad day or sharing some exciting news about your personal life helps create a bond with your team members. You are more than the code you write.

45. Raise your hand

When you don't know something, raise your hand. Be honest and don't pretend. Sometimes we just can't wrap our heads around something and we need that little bit of assistance to help us understand. Raising your hand earlier will help save time and get you back to writing great code faster.

46. Share your knowledge

Sharing what you know with those around you helps create an environment of growing, learning and improving. We can all help each other if we are more willing to share what we know.

47. Contribute during retrospectives

Retrospectives are designed to provide feedback, identify areas of improvement and share thoughts. If you don't participate, you can't complain if things don't get better.

48. Create a team agreement

Creating a team agreement helps set the standard and expectations for everyone in the team. Use a team agreement to ensure accountability within your team.

49. Win together, lose together

"There is no I in team". All success is shared and all failures are shared, regardless if one person was more involved.

50. Have team building sessions

Taking some time, as a team, to do something fun is crucial. Something that's not work related. Go out for a drink. Go bowling. Play some video games. It helps improve your bond and show that everyone is human with their own lives.

6

Workspace

***“Optimism is an occupational hazard of programming:
feedback is the treatment.”***

Kent Beck

51. Leave a toxic workplace

Plants don't grow in poisoned soil. Careers don't grow in toxic environments.

52. Participate in technical conversations

The water cooler talk for software developers generally gets very technical. Everyone wants to share the latest tool or language they've discovered. This is great as it helps inform and grow your knowledge. Get involved. Share your opinion. Listen to others.

53. Do more pair programming

Two brains are generally better than one when solving a complex problem. Beyond that, sitting on a call with another developer can give you more insights into their process and how they tackle their work.

54. Get involved in code reviews

The more code you review, the better developer you'll become. If you're a junior, you'll learn a lot from going through other people's code. If you're a senior, you'll be able to provide guidance to juniors and even learn a thing or two.

55. Give constructive feedback

Feedback is crucial for growth, but sharing feedback in a constructive way is even more important. You don't want to offend someone or come across as a tyrant. Share your feedback in a positive manner and they'll thank you for it.

56. Improve based on feedback

When you receive feedback, which is bound to happen at every point of your career, be objective and thankful. Too many developers can't handle criticism and take any feedback as an act of war. It's not. It's simply someone sharing information that they think will help you become better. Take the feedback, reflect on it, take action to improve.

57. Let your work speak

Politics and human nature often become overwhelming in the workplace. Instead of leaving things to chance or politics, let your work speak for itself. Be so good that nobody can dispute that you're the best.

58. Build relationships

The people you work with now may end up helping you in the future. They could help you find another position at a new company or even help you start your own. Build healthy and strong relationships with those you enjoy working with.

59. Make yourself known

In order to get more opportunities, a pay raise or a promotion, someone has to actually think of you. This only happens if they know who you are. Being confident and making it clear that you're a contender will get your foot in the door.

60. Create the culture

A company's culture is crucial for employee retention. Management likes to think that they create the culture, but it really comes down to the employees. If you want the culture to improve, it starts with you. Organize a team lunch or go chat to someone you've never spoken to before. Chances are people will love it and the idea will spread.

7

Growth

“Strive not to be a success, but rather to be of value.”

Albert Einstein

61. Learn from open-source

Open-source is one of the single, most beautiful things. Think about it. A endless supply of the highest quality code, written by some of the greatest minds in the industry. And you have free access to it. That's powerful.

62. Learn from your failures

You're going to fail. Hopefully a lot. The more you fail, the better you'll become. Make sure you learn from your failures, because if you don't, you really just wasted your time.

63. Celebrate your successes

It's easy to focus on a goal, achieve it and immediately move onto the next thing. That's a good mindset to have. You always want to keep growing, but take some time to enjoy your success. Celebrate it and take in the moment.

64. Know when to ask for help

Writing software is a collaborative endeavour, even if you're working alone in your room as you're still researching online. Get good at identifying when you're stuck and know who to ask for help.

65. Plan your next step

Your career is a series of steps. Know where you are, where you want to go and what the immediate next step is to get there. Then start working on taking that step.

66. Share what you know

Sharing what you know is so valuable. It helps solidify your knowledge, it helps others to extend their knowledge and it helps validate that you know what you're talking about.

67. Be ready to change your opinions

Having an opinion is great. However, the tech world moves so fast that your opinion may become invalid at some point. Stay up to date and regularly re-evaluate your opinions.

68. Understand people

The better you get at understanding people, the better team member and eventual leader you'll be. Understand what motivates people, how they think and what their goal is. This will give you perspective that will help you get the most out of the relationship.

69. Be accountable

You want to be known as the person that can be given any item of work and you'll get it done. Take responsibility and being accountable is the way to get this done.

70. Explore opportunities outside of coding

You can do a lot in this industry without writing a line of code. Teaching, giving a talk, consulting or creating content are all ways to leverage the skills you've learned as a developer and turn it into a new opportunity.

8

Mindset

“Experience is the name everyone gives to their mistakes.”

Oscar Wilde

71. Set your goals

We all need goals to help guide us and inform every decision we make. You need to have something to work towards and that comes by setting goals.

72. Know your next step

When you have a decision to make ask yourself, “Will this bring me closer to my goal?”. If the answer is yes, then do it. That’s your next step.

73. Inspire others

The more you inspire others, the more people will want to work with you. The more people want to work with you, the more opportunities you'll have.

74. Stay humble

You'll achieve a lot in your career. That's amazing, but don't let that form an ego. Nobody wants to work with the person with a huge ego. Stay humble and grounded.

75. Discover your purpose

We all need a reason why we do things. Knowing your why and your purpose will help you identify what type of work you want to do. Doing work that aligns with your purpose will help you to enjoy your work endlessly.

76. Work smart (and sometimes hard)

Working hard will only get you so far. Working smart is how you can make the most of your career. Determine what you should focus on, what makes you most productive and what you enjoy doing. Then spend as much time in that state of mind as possible.

77. Turn off

When you're not working, turn off properly. Don't look at your emails, messages or social media. Just focus on being in the moment and recharging so that you're ready to tackle work the next day.

78. Be reflective

Your career is a journey and you'll get better by being reflective and thinking about your decisions and how you can improve.

79. Reset

Take time to reset. Sometimes we just need an extra day to help us fully recharge and clear our mind. When you're feeling like that, don't ignore it. Avoid burnout by resetting at the right time.

80. Everything is an opportunity

Adjust your mindset to view everything as an opportunity. You may not always be able to work on the project you want or work with a cool stack. See each change as an opportunity to learn and grow and you'll become an unstoppable force.

Brand

“In order to be irreplaceable, one must always be different.”

Coco Chanel

81. Use LinkedIn

LinkedIn is great for expanding your network, sharing what you know, building an audience and demonstrating your experience.

82. Use Twitter

Twitter is useful to stay informed on what the latest trends are and to get advice from those you look up to.

83. Build a portfolio website

A website that demonstrates you, your skills and your experience is like having the best CV ever made, permanently available for anyone who wants to work with you.

84. Create content

Creating content, in any form, helps package and ship your knowledge. This is valuable and could turn into it's own form of income.

85. Start a blog or newsletter

A blog or newsletter is a great way to write down your thoughts, opinions and experiences. Some people like to do this just for themselves and others use it to build their audience. Either way, writing things down helps solidify your knowledge and process your thoughts.

86. Be a guest on a podcast

There are so many podcasts out there looking for guests. Reach out and have a conversation to help spread your reach and share what you know.

87. Host a podcast

If you have the skills and confidence, hosting a podcast is an amazing way to build your brand and create a network.

88. Attend talks or other networking events

There are so many events out there, both in-person and remote. Find one that you like. Attend it. Learn from it. Network with others there.

89. Speak at an event

If you're able to find an opportunity to speak at an event, take it. It will help you share your knowledge, build an audience and validate you and your brand.

90. Engage online with other developers

Reaching out to developers, both those you admire and fans of yours is how you can build relationships and create more opportunities.

Bucketlist

“It’s not a bug; it’s an undocumented feature.”

Anonymous

91. Stay up all night coding

Nothing will make you feel more like a developer than drinking far too many energy drinks and pulling an all-nighter.

92. Attend a hackathon

A bunch of developers. A couple of days. A lot of energy drinks. A lot of code. Now that’s a fun time.

93. Accidentally delete a database

You can't call yourself a developer until you've felt the sweat and anxiety that comes from accidentally deleting a database.

94. Having a commit message "Did stuff"

I don't condone this, but at least once you've got to have this commit message in the history log.

95. Use too many if statements instead of a switch statement

Probably the most common mistake that developers make.

96. Spend too much time customizing the theme of your IDE

There's nothing like wasting a whole day making your IDE look just the way you want it.

97. Spend hours on a bug that's a one character fix

Enough said.

98. Buy noise-canceling headphones

Can you really focus unless the outside world doesn't exist?

99. Create an infinite loop

When you write one so bad that you have to restart your machine, that's when you know you've made it.

100. Have something go wrong in a demo

"The demo gods were not with me today."

11

Finally...

“People rarely succeed unless they have fun in what they are doing.”

Dale Carnegie

101. Have fun

Fall in love with code. Fall in love with always learning. Fall in love with building something out of nothing. Enjoy yourself. That's when you'll succeed.

