Twitter Thread by Wogan



Wogan
@WoganMay



"I want to learn how to code, but have no idea where to begin."

I've been getting this question pretty frequently lately, and thought it was a good time to put together a thread to cover all the basics.

And here it is!

Instead of just throwing a few links at you, I first want to cover something much more fundamental: The *reason* you want to learn to code.

I've had to train a good number of developers in my time. The ones that succeeded had rock-solid, persistent motivations. Is that you?

Some good motivations for learning how to code:

* Enabling a career change
* Insatiable curiosity, eagerness to learn
* Frustration with inefficiency + desire to improve things
* Larger goals in mind, coding being a stepping stone

Some bad motivations for learning how to code:

* It seems popular
* Your friends/family have ideas for apps
* You want to make lots of money by sitting in front of a computer
* You bought an expensive MacBook and are struggling to justify the expense

For those, trade forex.

Motivation matters, because of the one thing that's universally true about coding: It's hard.

Depending on what you do, you may have to learn several languages at once. Tools constantly update. New approaches land in the market on an annual basis. You're constantly upskilling.

As with any great endeavour, it matters to have a "why" before you worry about the "what".

The first "real" thing I made was in 2006 - my own CMS, having become frustrated with the limitations of an existing one.

15 years later, that same motivation carries me every single day.

Once you know "why", it becomes easier to pick the things to learn. No matter the language, you're eventually going to cover:

* Variables
* Control structures
* Error handling
* Functions, maybe Objects

No one language is "better". Disregard claims to the contrary.

In fact, the opposite is very much true: Learning how to write statements in any one language is pretty easy.

Learning how to structure programs as they grow more and more complex is more challenging, but is much more transferable to other languages.

If you're setting out to be a "x" developer (Javascript, Python, Rust, etc) because you believe this one language is the best / will solve all your problems / is a universal hammer for every nail, you're limiting yourself.

I call this out because I've seen it so many times.

With all of that said, let's get to the heart of it: *actually* learning how to code. It's really just doing one thing over and over again:

Pick a language that's suited to the sort of problem you want to solve first:

* Working with data, automating a tedious process: Python
* The above but in Excel: VBA
* Building webpages: HTML and CSS
* Webpages that can store data: PHP
* Webpages that are interactive: Javascript

A good place to get a grip on the basics in a super-accessible way: https://t.co/4UWPe2PIMl

They have a solid platform for picking up languages, grouped by career paths, and you can get through most of it fairly quickly. But this is only the start of the journey.

When you have at least 1 language in your toolbox, set out to solve a problem that's personally frustrating for you.

Maybe it's a manual report you're doing in Excel, or creating a single page that can submit something to an API, or building a simple dashboard.

While solving it, take advantage of the one ubiquitous thing in recent history: Community.

StackOverflow, https://t.co/n0p1Uciwxg, https://t.co/MMW3zGZTwj or any other forum, find a place to network with others who are also starting out.

Learning from each other is gold.

By the time you've figured out a solution to your first problem, chances are you've already identified a bunch of others:

* A better way to re-do it
* A tool or library you didn't know about before
* An unexpected issue that hindered you

That's the next thing. Keep going.

If you solve enough problems (and pay attention to how you do it), you'll start seeing patterns: Re-usable tricks you can combine.

That's all programming really is. It looks big and complex on the outside, but it's actually just thousands of tiny solutions stuck together.

Possibly the most important: Don't get discouraged. I've seen too many people with potential start to doubt themselves when they compare what they're doing to what others are doing.

I can guarantee you that every single one of those people started in the same place you are.

To get you started, here's a handful of things you should know about upfront:

* Use a decent text editor. Microsoft Visual Studio Code is light, fast, free, and pretty good!

* HTML and CSS are good markup languages to learn, even if you don't end up using them all the time.

* Get a free https://t.co/KMuOQgZwaK account and spend some time learning how to use git. Source control is a basic competency you will need.

* https://t.co/F3HeR7lqe6 can publish basic websites from https://t.co/KMuOQgZwaK free of charge, in a very easy-to-use interface.

* You'll end up working with a database at some point. https://t.co/VG8EADun4f is a good place to start: it's free, popular, very simple to work with, most languages can connect to it, and it will teach you the basic SQL syntax you'll need for larger databases.

* You can learn a lot by pulling apart other people's source code - especially for building webpages. https://t.co/Da3PZZQv9O is a library with wide support and good examples to get you started.

* I like https://t.co/zzDE3BzOUM for taking it further, with lots of free components

* When you move beyond static sites, you'll need to learn how to take server-side code into production (so other people on the internet can use it). https://t.co/UMkXmF3eju does an excellent job on both their platform, and their community education guides.

* Since it's 2021, learn this right now: Everything should have HTTPS on it. https://t.co/vwqB7RiOoe is 100% free. Don't deploy a website without SSL.

* You may need your own domain name. https://t.co/WwJE93om85 is easily my favourite for price, ease of use, and reliability.

* If you need icons, images or artwork:

- https://t.co/NCrgrasz9n
- https://t.co/Dj3UWsEuwh
- https://t.co/ZrXU1Jm0dl
- https://t.co/rBp7waDpaN

* Free web fonts: https://t.co/HNr9WgT1dK

If you want to build mobile apps:

* Android apps are built with https://t.co/USk2B6OAzh
* iOS apps are built with https://t.co/yCZitv6Hgl
* https://t.co/fw2XnJkHbT lets you build for both
* Simplest start you can make: https://t.co/mPLnmBfFeu

If you want to build desktop apps:

* Build cross-platform using HTML, CSS and Javascript with https://t.co/gubXTnaHWx (there are many alternatives)
* For Windows desktop apps, definitely get Visual Studio Community (latest edition)
* iOS desktop apps, that's also Xcode.

If you want to start making money from writing code, that's a whole other thread - but the most direct trade is time-for-money via https://t.co/EaUibfQi36

If you can build, deploy, and run a service you want people to pay you for (in SA) I recommend https://t.co/PhZdGw6WyC

And I think that's everything for now! There's an awful lot to learn on this journey, but it's also an infinitely rewarding one. Learning how to code in any year is a smart move, but this is especially important in 2021.

If you do undertake this journey, let me know!