

Init

Tech & Mindset

Keep technological watch

Keep yourself updated

- YC's hacker news / lobste.rs
- Reddit
- Product Hunt
- Medium
- Dribbble
- TechCrunch

- Angellist
- Twitter
 - Some example: Julia Evans, Paul Graham, DHH, etc
- Newsletter
 - ThisWeekIn(your techno)
 - Company blogs
 - Reco = Netflix/Spotify
 - Signal v noise

- Youtubes
 - The family
 - Conferences
- Meetup
- Read books
- Github Trends & follow users
- Become friend with other Tech people

Choose your tech stack

Choose a language

- Not an easy task
- Productive language
- Choice based on application scope

Languages History

C

- 1970
- Made for Unix

C++

- 1985
- Extend C with classes

PHP

- 1994
- Lib C for the web
- On all server providers

Java

- 1995
- Portable / Compiled for JVM

Python

- 1990
- Code readability - made for scientists
- Made for Math & IA

Ruby

- 1995
- Programmer happiness - productivity - made for devs
- Slow as fuck
- Exists for Ruby on Rails

JS

- 1995
- Made for front
- Become backend language with nodejs (v8 - 2008)

Go

- 2009
- 23 keywords
- Compiled with GC

Exotic 🌴

- Smalltalk: First object language
- Lisp: First fonctionnal language
- Scala: Work with all Apache tools, run on JVM
- Erlang: Made for messaging, network

Exotic 🌴

- Rust: Made for safety
- Swift: Made for Apple ecosystem
- Kotlin: New Android language
- TypeScript: Typed javascript

Choose a tool

- Go on GitHub
 - stars / commit / issues / contributors
- Find tutorials
- Read the fucking manual

Licence basics

- Open Source vs Free Software
- GPL / MIT / Apache / BSD

Business

Do you really need to code

No. At first

MVP or Tracer bullet

Use tracer bullets to find the target

- Users get to see something working early
- Developers build a structure to work in
- You always have something to demonstrate

Iterate

- Get feedback
- Watch your KPIs (but don't be crazy)

Pragmatic programmer

- DRY (Don't repeat yourself)
- KISS (Keep it simple, stupid!)

Orthogonality

The basic idea of orthogonality is that things that are not related conceptually should not be related in the system. Parts of the architecture that really have nothing to do with the other, such as the database and the UI, should not need to be changed together. A change to one should not cause a change to the other.

– David Thomas

source

