🤖

SDE Resources

Context

Software Development is a huge area with subspecialties. This page aims to provide resources for anyone interested in a career transition into software development covering from educational resources to interview prep.

General

Here are the main components you will see on online resources:

Under software engineering here are the main concepts covered:

intro to programming	this is to introduce the basic foundations of software such as if else statements and for loops. This is to introduce a programming language to enable you to begin software
database	this covers how data is stored, when we shut off our machine, our apps restart, to retain previous information, this is stored into a database. This section usually teaches querying languages like SQL
data structures	this covers the basic types of structures available in coding for example, strings and integers are different in coding, and we have options to store them as a list etc..
algorithms	software is huge, so we learn about ways to optimize how we process large amount of information, the way of calculation/processing is called algorithms
dev ops	after you write your first code, they need to be managed, saved, deployed so its available to other ppl. These tools you will learn are called git, github, docker containers, kubernetes. Generally we don’t need an in-depth knowledge but enough to get things rolling.
testing	how do you make sure your code runs or doesn’t break anything else you or other people built? We write tests to make sure the code we write runs as expected. This is usually to learn about packages/methods for writing/running tests as we write code.

You might also stumble into another very popular area called data science. While they are both in the technology domain, the skillsets and focus is slightly different:

intro to python/R	most data science roles will require simple coding, when you have too much data, it will need to be done via coding as there is limit to what excel can do.
data analysis	you will usually learn how to analyze data, what this looks like is looking at a dataset, learning how to calculate median, averages, look for trends, graph out those trends and what conclusions you can drive from it. The common skills set under these are: Excel (to process data), SQL (to pull the data), tableau (to graph/visualize data)
machine learning concepts	a common theme in data science is to be able to drive business strategies with data, but with raw data there's not much business decisions can be made from it, so often, data will be fed through ML models so that it can predict what could happen next. Often you will see concepts covered like neural networks, decision trees or linear regression.

Backend vs Frontend vs Fullstack

You will hear these terms a lot. While it is all software engineering and all three still codes, there are some slight nuances.

Frontend

Typically refers to client side, this means they are usually working on the things you can see, such how the website layout looks, the interactions on the websites. This can include android/ios apps as well. Typically, keywords that will pop up are things like Javascript, React, Angular, HTML, CSS.

Backend

This refers to the server side. Usually, when client side makes a call, they call the server to get all the needed information to display on the front end. There's more emphasis on database and efficiency, if the server is slow, that's when you see your page just loads for a very long time, which is why server side have to ensure they return what client side needs in a timely manner. Keywords that indicate more backend could be database, APIs, Java, C++.

Fullstack

Fullstack refers to both frontend and backend. For example if you build your own website and has a server that connects to a database, this is typically fullstack. Fullstack just means the scope covers both client and server side.

Resources

Here are a couple resources, free or paid.

bootcamps	bootcamps are usually guided, there will be assigned instructors and a set timeline.	$$$
online courses	online courses are less guided, these are videos with hw that are not graded, they are self pace as well.	free-$
textbooks	these are just some of the books that have been recommended by colleagues	free-$
interviews	here are the general guidelines for interviews base on my own experience	free

Bootcamps

In general, bootcamps can be anywhere from 10 weeks to 6 months dependent on the program. Price-wise they usually fall around 10k range as well. Bootcamps are usually intensive, designed to teach you the minimum, enough to get you started and enough to get you to the interview stage. They usually cover all the main concepts listed above, but you will still need to put in work to grow your own portfolio and study algorithms to pass coding tests in interviews.

Online Courses

Online courses, imo, are a great way to get an introduction to what the industry is like. Most online courses will cover the introductory courses like programming or data structure and algorithms. They are often free to 50$ range, dependent on the certificate and who is providing the certificate.

📔 Online Courses

Textbooks

Textbooks, imo, are great to supplement if you have a firm grasp on introductory topics. This is because in the form of textbook, it doesn’t quite provide the guidance of setting up your machine and coding. But in exchange, in my experience, they cover more theoretical concepts such as explaining more in-depth into databases and memory, these are topics practical software engineering will not touch on as often.

📚 Textbooks

Interviews

This is just a general guideline for getting ready for interviews, a bit of tips and lookouts on resume, linkedin, getting interviews and going through interviews.

👔 Interview Prep Guidelines