2020-08-30 12:16基本过完

行业特点与语言

Java特点：

经典工业型语言，面相对象，虽然笨重但是严峻，工作机会多，学了java后再学别的语言很快。Go除外

已入行服务端：

如果用php则推荐再学java

如果用java则推荐再学python

如果再多的话就没有太大的意义了！！！

已入行前端：

已经入行前端可以学node.js或者python

实用性考虑：

开发一个万把人访问的站点没必要用java或者golang，用动态语言成本会降低很多

国内国外差距：

国外：实用为主

国内：国内技术方向受大厂影响很严重

Php语言：

不是初学者的首选语言，适合找工作但是java范围更广

Python语言：

Python适合做第二语言，适合做数据分析和科学计算，深度学习但是门槛极高需要很高的学历和数学基础。不适合做找工作的第一目标.

python适合独立开发者，自由开发者，有情怀的开发者，python覆盖了除了底层外的所有应用场景，可以做web，做脚本做运维，统计分析，爬虫

C#语言：

语法非常好，但是基本上都是国企在用，用来做内部管理项目

Web工作理解

Web方向：前端/java/php

找工作：

前端岗位是服务端的好几倍，需求量大。因为服务端如果不发布版本基本就不用改动，而前端一旦改动就非常麻烦

纯技术发展路线（不包含管理路线）：

普通程序员--》高级程序员--》架构师--》技术总监--》CTO

前端：

需要一定审美能力，因为ui在很多小细节上其实是没办法面面俱到的，这个时候小细节就需要前端来把控了

必须有极强的耐心

适合女孩子

前端进入工程化的时间太短，是最近五年来才快速发展起来的，市面上的前端没有工作超过十年的。所以发展路线不明确。

很多公司的CTO和总监很少有纯前端出身的。

所以掌握服务端，更容易统领全局

前端主要负责人机交互，容易受到UI端影响

服务端：

必须要有一定的大局观，和架构能力

服务端未来转型机器学习，自然语言处理更快速

前端对比服务端：

前端可能变化快，但服务端稳定

服务端的学习相比前端相对轻松

很多前端的新模式，都来源于服务端的思想

服务端编程思想更通用，更接近本质

前端会将大量的时间浪费在ui上

全栈：

不用前后端都很精通，但是要具备前端的思维。很多大厂api都是由前端来写，后端都是集中在写微服务了。他们只提供数据，对于业务模型的整合发到前端中去，自己写的api再写页面会舒服一些。服务端现在主要是提供数据了

现在前后端分离，削弱了程序员的成长力。

纵向前后比横向拓展语言更划算。

前端很容易原地打转，因为学了vue再去学react和其他的框架其实就是原地打转，把时间花在纵向里面，把vue学精通，如果没有很强的工作需求就不去学太多的框架了。

***编程领域中谁离数据更近，谁就更有话语权***

***人要现实点，认清自己。吃饱饭是第一要务，进不了BAT也没关系，互联网大厂坑位就只有那一些***

***踏踏实实，做好手头工作最重要***

学生在学校应该怎么学习

***985***：

目标只有一个，进一线互联网公司

在校时学习优先数学、英语、算法、数据结构

具体则为：***高等数学，统计与概率论，线性代数***，复变函数，计算机原理，

时间有限则重理论，轻业务，语言选java

***211***：

能进一线进一线，不能进则选小而美的公司

学习路线最好保持跟985一致，需要偏向一些业务，动手实践，做项目

***普通高校***：

进一线难度较大。第一份工作薪资不是首要的，必须选择“有人用”的产品，优先选择互联网公司，不选择外包、传统软件行业。***因为技术的提升，必须有复杂业务的驱动，没人用的产品或者要求不高的产品，就无法提升技术***

学习路线：最好保持跟985学习路线一致，额外必须有大量实践项目。时间有限的情况下，轻理论，重点是做出成品高质量的项目

BAT工作机会的珍贵在于他们有高并发和分布式的业务机会和环境

跳槽

***什么时候跳槽***：

习惯是最可怕的杀猪刀

环境可以磨灭一个优秀的人

安逸可以磨灭一个人的上进心

技术陈旧、没有用户、软件代码质量没有高要求的公司，多跳跳不是坏事，***但要更上一层楼***

只要是公司越来越好那么换工作频繁些，就没什么问题，履历上一开始是普通公司，然后进入初创型的互联网公司，再到二线互联网，然后一线互联网，那么就没什么问题。

***是否需要考研***：

如果能考上985（最差211）的学校就考。不然对计算机行业来说意义不大

***做该做的事***：

人生很多的困境，都是因为适当的年龄，没有干合适的事情

***程序员的价值是什么？***

做出有人用的产品，成就感最直观的来源就是有人用你的产品。编程的路很长，必须要找到一定的成就感，才能维持你在这个路上越走越远

程序员高效学习方法

***编程高效学习方法：***

需要太强主观性的方法都不是什么好方法。

看慕课网视频最好的方法是先不看，自己写一遍，然后再看视频，才能把视频物尽其用！

不要去看什么太多的方法论，先写。

***被动驱动方式是一个很好的方式，找一个例子然后去实现一次，做一款高质量，业务复杂的项目，注重每个细节，死磕项目***

程序员的综合素养：

综合素质会影响你短期内的选择。当技术达到一定程度的时候，综合素质是你的瓶颈

30岁的程序员性价比其实是很低的，有家庭有孩子，薪水又不能降低因为要养家，不能跟20岁的程序员比，所以综合素质就很重要了。能够技术上一条道路走到黑的人很少很少很少。30岁之后拼的就是软技能了

***综合素质包括：***

交流能力，管理能力，认知能力，学习能力，任务规划能力等

以上能力，能难看一本书或者一个视频就能解决的。个人综合素质，很大程度上面是看生活环境所决定的。

***综合素质提升方式：***

1. 少盯着技术不放，少学习各种功能雷同的框架。多看书，文学，小说，历史，诗词皆可。人的自我意识就会帮助你进化

带有太强目的性学习，反而不利于成长

2、培养一个兴趣爱好，正向爱好即可

3、程序员的抽象能力和逻辑思维能力不差，但是审美和观察能力偏差

1. 忽略细节，成长会很慢

5、编程不是为了考试，不要用上学考试的思维来学习编程

编程要以解决问题为目标，遇到问题解决问题，技术才能成长

6、少学习功能雷同的框架，要懂得舍弃，卖东西的都说自己的东西好，但这个好，对你来说可能没有意义

推荐数单：

《代码大全2》（每一个做编程的都推荐看）

《Javascript高级编程》

《java核心技术（分上下两卷，偏基础，做java推荐看下）》/《java编程思想（偏思想）》

《算法图解》（入门级别算法书，图形化的方式去讲解）

语言学习推荐看书，框架推荐看书

《全球通史》

《少年得到》（古文、文学、历史）

程序员境界：

古人早就把职业分为了七层：奴、徒、工、匠、家、师、圣

奴：非自愿、靠人监督的人

徒：能力不足，肯自愿学习的人

工：老老实实，按规矩做事的人

匠：精通一门技艺或手艺的人

师：掌握了规律，又能将其传授给他人的人

家：有固定的信念，让别人生活更好的人

圣：精通事理，通达万物的人

职场：

坚韧一些，大部分的批评都是善意的

提升编程能力最快的方式/如何提问/编程思维怎么培养：

问：如何提问？

答：能自己解决，能搜索的绝对不麻烦别人，编程中80%能够通过调试解决，19%可以通过搜索解决，只有1%可能是真正需要提问解决的。

即使是要提问也要有：前因后果，有理有据，逻辑清晰，描述正确

问：怎么样培养自己的编程思维？

答：自我解决问题，别人讲出来的思维你会觉得就是这个样子的，并没有什么，但是自己想的时候就不是这么回事了。这也是一种被动式的培养方式

问：编程能力提升最快的方式是什么？

答：自我解决问题，寻找答案的思考过程。

云开发：

云开发是无服务器编程的一种，是一种特定场景下的serverless

Serverless是更进一步的云，适合微服务，目前不适合整站serverless

小程序云开发优势：

1. 统一集成开发环境
2. 省去运维、部署等麻烦事儿
3. 不需要域名（避免https）
4. 自带云存储（图片上传很方便，不用存在服务器中或者存在阿里云oss中）
5. 前端直接操作数据库

小程序云开发劣势：

适合个人/独立开发者，公司项目优先传统开发。

缺少完整的商业解决方案，如cms方案

LaaS：

PaaS：平台级服务

SaaS：软件级服务（石墨、在线文档）

FaaS：功能级服务（亚马逊AWS）

现代web架构：