

工大ACM15天训练计划（工程实践方向）

15天计划池

- 概论[Done]
- Ubuntu安装，基础命令熟悉[Done]
- Github账号创建[Done]
- 后端语言基础入门[Done]
- 前端入门[Done]
- SQL使用入门[Done]
- 找到自己发展方向和兴趣点[Done]
- 做自己喜欢的项目（没有就增删改查和带登录的博客系统）
- 面向企业技术栈搭建[Done]
- 数据结构不可少

2018.08.15

1.查看ToACM导论，认清现状和未来公认发展方向

坑都被学长们踩过了，就像期末已经画好重点，多少你们的前辈因为没有意识到这些关键点虚度大学三年，让自己去了自己不满意的地方或是直接考研碰大运。

里面写的不保证适用性，请培养自己的独立思考能力。毕业后也不要被其他人、郭嘉、舆论牵着走。

对于互联网企业，只跟着老师上课的乖宝宝，简历注定被扔进垃圾桶。

2.先选择一门喜欢的后端语言

从现在起禁止使用VC++，工作时候任何情况下都用不到的技术，在学校也不要花时间和技能点放到这上面，请把时间花在刀刃上。

推荐选择Java技术栈，Java这几年招聘市场仍然非常广，各种中间件和解决方案非常多。

PHP技术栈入门快，适合快速成型的项目，但招聘市场趋于萎缩。

C、C++技术栈不推荐，招聘要求过高，非985、211、大硕或水平高请勿硬钢。但C、C++算法训练很有用。

Python、GO技术栈虽然不推荐，但你可以业余研究一下，写工具、爬虫有用。GO比较适合分布式多线程。

如果走前端路线，JS则是必修课。后端也可以使用Node实现。

注意，任何语言和技术都有它适合使用的场景。只ALL IN除了Java、PHP外一门语言比较危险，因为各位无论如何都去找工作吧。所以无论现在和未来，都不要把自己局限为某语言开发工程师。

这回让各位自行选择喜欢的后端语言，我和板砖个人推荐Java。因为我是PHP、Go、Python、JS技术栈，所以Java如果有疑问请联系板砖

禁止把时间浪费在比较语言和技术优劣，只有是否适合某个场景。把时间花在上手实践上

3.我今天做什么呢

1.找到喜欢后端语言后，那就准备快速上手吧

如何劝退新手？看巨厚的书、老师上课枯燥教学。所以能快速上手然后实践试错是王道。

2.那我去哪里学？

[菜鸟教程](#)

推荐入门的第一步

[菜鸟教程-java](#)

[菜鸟教程-php](#)

[菜鸟教程-python](#)

[菜鸟教程-go](#)

w3c

w3c比较适合PHP和前端技术栈

w3c-php

想要通过视频手把手教

[java1234学习路线图](#)

书籍推荐

SAMS出的xx入门经典是我大部分技术入门第一本书

包括前后端语言、正则表达式。但SQL之类的可以直接w3c解决。

[java入门经典](#)

[php入门经典](#)

当然也可以找电子书，原则上支持正版

3.晚上回复进度

不要求一天全学会，至少基础部分能过一遍。

语言学习要持续整个技术生涯，今天只是个开头。

4.明后天计划准备

1. 下载VirtualBox
2. 下载Ubuntu镜像
3. 下载Git工具

2018.08.16

1.Github账号创建

Github是未来你的第0号简历，越早创建、把练习和个人开发项目放到上面越好。

一般面试官如果看到你的简历上附有Github链接，则会点进去看看你的绿点（推代码情况）和项目情况。记住，潜然泪下的万字求职信不如一行高

质量代码。而最直观展示你学习、潜力、探索、实践成果，就是Github。

1.需要我做什么？

1. 访问[github](#)，创建一个账号。
2. 下载github客户端，搜索安装方法
3. 按照文档，创建rsa_key并添加到github上[创建rsakey](#),[添加key](#)
4. github上[新建一个库](#)，名字随意。推荐叫practice

下面针对使用命令行界面同学，如果使用图形化界面，请自行搜索图形化github使用。

1. 使用 `git clone`，将库克隆拉到本地
2. 在本地新的库随意创建一个文件
3. 使用 `git add -A`，`git commit -m '本次推送内容'`，`git push` 三个命令，将文件推到github上面。

无论图形化还是命令行方法，都需要掌握以下命令，是面试重点。

学习 `git add`，`git commit`，`git push`，`git pull`，`git branch`，`git checkout`，`git log`，`git reset` 含义及使用

2.群作业回复已创建的github账号。

3.持续跟着08.15继续入门语言

并且把跟着教程学习代码放到github上，以后所有练习和个人项目代码都需要放上面。

2.继续学习 8.15 选择的语言

创建Github相关比较简单，如果已完成 1 的内容，剩下的时间请将今天的练习代码push到github上。

3.附加任务 安装Ubuntu18.04 desktop

如果你使用的是MACOS，那可以忽略掉这一部分。

为什么要用Linux/Unix而不用Windows呢？因为所有的大型互联网公司服务器都运行在非windows操作系统上，日常维护、查线上问题都需要熟练掌握linux，没有掌握Linux命令直接过不了第一道简历筛选关。

可以自行搜索virtualbox安装ubuntu方法，虽然可能虚拟机运行比较慢，但一上来就让你装双系统或直接放弃windows全部使用Linux作为日常操作系统不是很友好。所以今天先安装上Ubutnu，可以自己随便操作操作，适应一下新操作系统。最好能运行terminal终端试一试能否用命令行控制你的Linux。

2018.08.17

1.Linux简单熟悉

需要了解 / 、 /home , /bin 等各个目录的含义。

2.Linux命令熟悉

Linux命令是面试重点，而且工作后几乎每天都会使用。

假如线上出现了线上事故，老大们都在看着，如何快速响应、定位、修复，也是晋升的必备能力。

1.基础命令

需要了解以下命令的作用和使用方法

ls , pwd , cd , mkdir , touch , mv , rm , cp , sudo , ps , top , history

2.工具使用

1. vim

因为线上服务器不会提供图形化编辑工具，所以一些工作会在vim中进行。对于一些高手来说，vim使用熟练了能够脱离鼠标，编码速度加倍。当然vim入门有一定曲线，需要背一些基础键位。

需要了解vim的安装

提示 `sudo apt-get install vim`

需要了解vim的基础使用

[打怪升级入门指南](#)

[cheatsheet键位图方便理解](#)

2. tar

压缩传文件必备，可以尝试自己打包和解包。

压缩样例: `tar zcf test.tar.gz waimai_*`

解压缩样例: `tar zxvf test.tar.gz`

3. ssh

连接服务器必备，未来如果有了自己的服务器，也可以使用ssh进行登录。目前可以先尝试自己连接自己。

```
ssh 127.0.0.1
```

有非常大概率会报错~那么如何解决呢？请自行搜索，并学习如何启动服务。

3.需要我今天做什么？

1. 继续熟悉昨天和前天学习语言基础
2. 入门Linux基础命令
3. 尝试使用Linux工具，提交感受和收获到作业中
4. 最后~执行命令 `cowsay "我碉堡了Σ(° °)"`
5. 尝试安装mysql相关工具。（提前熟悉）

2018.08.18

1.安装MySQL

推荐在Ubuntu上使用 `sudo apt install mysql-server` 来进行命令行安装。

非Ubuntu系统也可以尝试安装。

2.熟悉SQL语句

从这里开始，禁止使用图形化操作界面。一律使用命令行操作。

今天需要熟悉以下命令

创建数据库

```
CREATE DATABASE
```

使用数据库

```
USE
```

创建数据表

```
CREATE TABLE
```

增

```
INSERT
```

删

```
DELETE
```

改

```
UPDATE
```

查

```
SELECT
```

删除表\库

```
DROP
```

修改表结构

```
ALTER
```

这里有样例，可以直接执行体验一下，然后思考为什么这么写：

创建数据库

```
CREATE DATABASE test DEFAULT CHARSET utf8 COLLATE utf8_general_ci;
```

使用数据库

```
use test
```

创建数据表

```
CREATE TABLE testa (
```

```
id bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
order_id bigint(20) unsigned NOT NULL DEFAULT 0 COMMENT '订单  
id',  
pass_uid bigint(20) unsigned NOT NULL DEFAULT 0 COMMENT '用户  
编码',  
user_name varchar(64) NOT NULL DEFAULT "" COMMENT '用户名',  
total_amount bigint(20) unsigned NOT NULL DEFAULT 0 COMMENT  
'单笔订单总金额',  
status tinyint(4) DEFAULT 1 COMMENT '状态',  
order_time bigint(20) unsigned NOT NULL DEFAULT 0 COMMENT '下  
单时间',  
PRIMARY KEY ( id ),  
UNIQUE KEY order_id ( order_id ),  
KEY p_s_o ( pass_uid , status , order_time )  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='订单表';
```

增加数据

```
INSERT testa (order_id, pass_uid, user_name, total_amount)  
VALUE(123456, 654321, "测试用户", 200)
```

查询数据

```
SELECT * FROM testa WHERE pass_uid=654321
```

修改数据

```
UPDATE testa SET user_name="我修改了我自己" WHERE user_name="测试用  
户"
```

删除数据

```
DELETE FROM testa WHERE order_id=123456
```

删除表

```
DROP TABLE testa
```

删除库


```
DROP DATABASE test
```

3.思考为什么要设置成UTF8

数据库中文乱码是新手必经之路，不要慌。

4.进阶SQL语句

GROUP BY HAVING掌握较为困难，但却能解决大多数聚类的问题，我面试第一题就是让写SQL，写不出来的我就直接再见送走了。

```
GROUP BY .... HAVING
```

聚类函数

```
COUNT
```

```
SUM
```

```
AVG
```

综合练习：

这里是我面试真题：

数据库中按如下语句建表

```
CREATE TABLE waimai (  
  id bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
  order_id bigint(20) unsigned NOT NULL COMMENT '订单id',  
  pass_uid bigint(20) unsigned NOT NULL COMMENT '用户编码',  
  total_amount bigint(20) unsigned NOT NULL COMMENT '单笔订单总金额',  
  status tinyint(4) DEFAULT 1 COMMENT '状态',  
  order_time bigint(20) unsigned NOT NULL COMMENT '下单时间',  
  PRIMARY KEY ( id ),  
  UNIQUE KEY order_id ( order_id ),  
  KEY p_s_o ( pass_uid , status , order_time )  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='订单表';  
请使用SQL语句找出status为1、order_time大于1500000000、至少下过两单的pass_uid的历史下单总额
```

今天我要做什么？

- 1.安装MySQL
- 2.学习SQL基础语句
- 3.深入学习SQL语句
- 4.提交感想和心得作业

附：SQL语句快速入门

链接

看所有基础教程

高级教程中 TOP LIKE 通配符 IN BETWEEN JOIN INNER JOIN LEFT

JOIN RIGHT JOIN UNIQUE DROP ALTER

函数中

AVG COUNT SUM GROUP BY HAVING

从学习SQL第一天起，就必须养成这样的习惯：禁止一切使用SQL计算行为，所有计算请放到业务代码中进行。

2018.08.19

1.使用程序进行数据库增删改查

今天是个综合练习，请结合从第一天起的所有经验进行开发。

可能难度比较大一些，但能够用程序对数据库进行增删改查是关键一步。无论前后端，以后写接口还是写整个工程都会用到的必备入行技能，而且会伴随大半个职业生涯。

暂且可以不用管什么面向对象，直接一个流程下去对数据库进行增删改查即可。

不同的语言可能需要不同的拓展，如果是PHP需要apt安装Mysqli拓展，如果是JAVA则需要在lib中引入jdbc系列库。这地方如果不太清楚请请教群里学长。

参考链接：

[PHP数据库交互](#)

[JAVA数据库交互](#)

JAVA数据库交互最好还是看看网上的样例吧

2.将 1 中的代码推送到github上

如果还没有研究好github的同学请尽快搞定~

```
git add -A
```

```
git commit -m '和数据库交互'
```

```
git push
```

3.需要我做什么

1. 搜索你选择的语言如何进行增删改查。当然可以搜索样例开发过程和代码，模仿代码也是正确的最快入门方法。
2. 将今天的代码推到github上，并在任务中回复github链接
3. 虽然今天任务描述比较少，但却是关键一步。今天的任务会根据完成情况决定明天是否有新的任务。

2018.08.20

1.使用JSON格式输出数据库查询结果

昨天内容可能有点多，今天的任务非常简单，之前没有完成的任务也请赶上。

目前互联网公司很少用后端渲染前端，而是为了提高复用性和提供微服务，使用json格式和前端或其他模块进行数据交换。所以掌握如何使用json格式输出就相当于学到了接口的精髓。

各个语言都会有json库，例如PHP，直接调用json_encode、GO语言用simplejson库即可将字符串、数字、数组转换为json格式。

当然json也可以手动编写，语法也非常简单。有时候得到一长串json，可以在线、浏览器console或使用插件进行简单格式化。

json是个啥

json格式化

chrome FE万能工具插件

2.需要我做什么

1. 使用程序进行数据库查询，并将查询结果（要求多行结果）使用库方法转换为json格式，并输出。
2. 截图、提交作业
3. 之前的任务如果没有完成请尽快完成并提交~今天任务节奏舒缓。

2018.08.21

1.前端入门

前端能够为后端数据赋予灵魂，提供极高交互性，并理论上允许世界每一个人更方便使用你写的程序。

和后端一样，前端也是一个开发发展方向。如果对界面、交互感兴趣的同同学可以走这一块，但前端入行难度和后端一样。

掌握了前端和后端，自己SOLO项目打比赛也方便，毕竟请做好没人带着你做项目的心理准备。

虽然也有Bootstrap等前端框架能大幅度加速开发，但基本功必须扎实。

今天需要掌握html基础标签、css基础语法。

觉得w3school相关太多了，没有个重点请看群里《HTML和CSS入门经典第八版》，入门经典系列都是我特别喜欢的入门书。

实验楼HTML&CSS

w3school

2.需要我做什么

1. 排版并实现[train_html](#)静态页面。要求html和css文件分离保存。

2. 将代码推到github上，截图、提交作业
3. 之前的任务如果没有完成请尽快完成并提交~

2018.08.22

1.前端入门#2

今天和第一天后端学习节奏一致，请继续学习HTML、CSS。另一半时间，可以接触[Bootstrap前端框架](#)，能够快速加速开发。当然还有一些vue等MVVM框架，但学框架之前务必扎实基础。常用的标签和属性需要上手就能写。

请继续看群里《HTML和CSS入门经典第八版》，入门经典系列都是我特别喜欢的入门书。

[实验楼HTML&CSS](#)

[w3school html](#)

[w3school css](#)

2.需要我做什么

1. 继续想办法排版并实现[train_html](#)静态页面。要求html和css文件分离保存。
2. 继续将代码推到github上，截图、提交作业。
3. 查看并熟悉Bootstrap前端框架。
4. 之前的任务如果没有完成请尽快完成并提交~

2018.08.23

1.请设计一张用户数据表

包括自增id、用户id、用户名、密码、创建时间、用户状态(status)字段。

并插入到数据库中。

2.设计前端简易提交表单联动后端

模拟用户注册

`<html>` 套 `<form>`、`<input>`，允许用户输入注册信息:用户名、密码。后端进行校验后存入数据库中。

模拟用户登录 同上，允许用户输入登录信息:用户名、密码。通过后端验证用户是否合法。

难点:

1. 后端需要启动接口接收前端请求
2. 安全考虑，后端不能相信用户输入的所有数据。所以必须进行字段合法性校验
3. 前端如何提交数据到后端，后端如何返回数据给前端？
4. 拓展:了解什么是SQL注入

3.后端开接口、渲染前端、验证数据

首先，后端如果想要被前端访问到，需要开放某个接口供用户通过浏览器或其他终端根据IP、端口访问。其次，因为需要提交后返回是否成功，所以需要后端对前端进行渲染(吐 前端)。暂且先不管接口化，能吐前端就已经胜利一大半了 最后，因为安全原则，永远不要相信用户输入的任何数据。所以需要后端 对数据进行验证，一般验证规则是:类型、长度、正负(余额反向充值了解一下)、预防XSS、预防SQL注入(一般底层库也会做)、产品需求规则(例如不能输入2整个数字)、跨权限访问(例如产品允许通过userid查询好友用户信息，但有的恶意用户可能会随意输入用户id导致用户信息泄露)。

P.S.:我曾经在学弟开发的网站上注入无限循环《我的滑板鞋》，如果我想要玩高级点就直接DROP整个库。

4.需要我做什么

1. 设计用户表结构
2. 学习form语法
3. 后端开接口
4. 后端渲染前端
5. 后端数据校验、入库
6. 代码推github, 提交作业

2018.08.24

1.学习javascript基础语法

javascript赋予前端灵魂，能够对用户各种操作进行多种多样响应。今天需要了解基础语法和使用，详见群里《javascript入门经典》pdf，各位第一天已经选了喜欢的后端语言，基础语法和javascript类似，所以可以速推。

[菜鸟教程](#)

2.实际前端使用javascript

参考 08.25，对用户输入的用户名、密码进行验证。要求：

1. 用户名不能包含特殊字符(~!@#\$%^&*()_+{|:"<>?)
2. 密码必须包含大写、小写、数字，不能包括空格，且大于9位
3. 如不符合以上要求，请弹出警示框，并将光标自动focus到不合要求的输入框中。
4. 复习盒装模型(CSS) 盒装模型几乎是前端面试必考题，也是各种定位基础方法。
排版必备技能。

3.需要我做什么？

1. 学习 1 基础语法

2. 学习DOM盒装模型

3. 结合 08.23 实际前端输入框校验

注意，后端校验不可省略，前端任何校验都是可以绕过去的。

2018.08.25

1.学习jquery库

jquery对一些dom操作进行了简单封装。极大降低前端编码量,用起来非常简单。

但如果走前端职业方向的话，原生JS一定一定要掌握。不要怕麻烦。

jquery主要对dom进行操作，需要了解父子节点的概念和操作。

今天是缓冲时间，没有太多的额外任务。请将昨天的作业 2：

对用户输入的用户名、密码进行验证。

要求：

1. 用户名不能包含特殊字符(~!@#\$%^&*()_+{|:"<>?)
2. 密码必须包含大写、小写、数字，不能包括空格，且大于9位
3. 如不符合以上要求，请弹出警示框，并将光标自动focus到不合要求的输入框中。

使用jquery实现。

光标锁定使用focus api，可以自行搜索百度。

[菜鸟教程](#)

当然，如果有能力可以直接上vue等MVVM前端库。

2.需要我做什么？

1. 学习jquery基础使用
2. 将昨天的作业使用jquery实现
3. 代码推送github，提交github链接作业

2018.08.26

因为前端相关的知识太杂，所以今天强行完结前端.....

1.学习使用集成Bootstrap

Bootstrap是前端样式库，已经写好了大部分前端常用的组件，使用和集成非常简单。

曾经我挑战过24小时内开发一整套项目，能够完成挑战助力之一就是bootstrap和一套后端mvc框架。

BootStrap

前端在这里强行完结。如果要拓展可以研究npm、vue、react、native react、less

2.mvc入门

一般新手或是紧急项目写出来的代码杂糅前端、数据库交互、业务逻辑代码。跑倒是能跑起来，但之后自己维护或是交给其他同事维护就日了狗了。（写出这种代码也容易被骂）

所以MVC设计模式就上场了，C - Controller - 控制器，专门负责处理业务逻辑。M - Model - 模型，专门负责处理数据库交互。V - View - 视图，专门展示前端。其中C能够和M和V进行交互，V和M不能反向和C交互，V和M不能互相交互。

一般例子或是教程已经把MVC表现得很明显了，MVC如果用好了，感觉像是指挥交响乐一样，让数据在之中流转。

一般的MVC框架推荐

PHP

CI框架

JAVA

Spring MVC

一般文档写的比较全面，直接看很容易直接弃坑。所以正常的新技术学习方式：

1. 先快速搭起来框架
2. 看并模仿现有代码实现简单代码。可以通过视频或是博客快速入门。也可以通过github搜索现成的代码，当然项目star收藏数量越高越好。
3. 需要的时候再看具体的手册

最后记住，技术是让人类用的，为了吸引使用，入门文档不会

3.如何找到自己喜欢做的项目

1. 项目逼着你前进（e.g.还有半个月就要比赛/**就要用了，赶紧爆肝去做）
2. 你对现状的不满和通过技术手段将其改善（e.g.**系统/中间件过于辣鸡，我要重写）
3. 其他人或你的技术需要（e.g.辣鸡U盘总挂，写一个自动备份工具）
4. 练手用（e.g.了解kafka等业界常用中间件）

如果找不到请想想是不是太长时间没有关注业界了。

2018.08.27

1.快速学习正则表达式

正则表达式在爬虫和搜索上十分常用，面试上有时候也会问到。
参考群里的《正则表达式必知必会》，封面上写的10分钟就能学会。
不要滥用正则表达式，能用函数解决的问题尽可能用函数解决，糟糕的正则表达式回溯时间复杂度有可能降低性能

2.MVC设计模式结合ajax

之前的写法每点击一个按钮或链接，整个页面都会刷新来提交，这样用户

体验极差，因为需要加载整个页面，所以效率也差。因此ajax就是为了解决全页刷新的问题，也解决了控制器必须吐前端的问题，让控制器集中注意力于业务逻辑开发。

1. 搜索关键字 `jquery ajax`
2. 搜索关键字 `(你喜欢的语言) MVC接口开发`，提示：直接`echo json`字符串。
3. 结合一起开发
学会上面的技能，就可以对之前注册和登录功能进行改造，将同步请求改为异步请求。

3.要我做什么

1. 快速学习正则表达式（大约15分钟）
2. 学习`jquery ajax`
3. 学习`mvc`接口开发
4. 将改造后的代码推`github`，提交`github`作业链接

FINAL

常用架构

后端核心

PHP/JAVA/GO/Python/node...

MVC框架

包依赖工具（`mvn`、`composer`、`npm`、`godep`等）

前端核心

HTML (5) 、 CSS (3) 、 JavaScript

Vue、React、angular (MVVM)

Bootstrap

jquery

数据库

关系型 - MySQL、tidb

非关系型 - mongodb、ES

数据仓库 - HIVE、HDFS

时序数据库 - opentsdb

操作系统

CentOS

文档化

PHPDoc、JavaDoc

软件工程

代码整洁之道

设计模式

敏捷开发

搜索

awk

grep

find

正则表达式

单元测试

JUnit、PHPUnit...

版本管理

Git

缓存

Redis(常用) / tair
memcache（不常用）

检索

ElasticSearch

消息队列

kafka
rabbitmq

服务注册、发现

zookeeper

大数据计算

spark
storm
hadoop

图形化

Zeppelin
Grafana

内功

数据结构

算法

大数据量，高QPS并发经验