

起航



Part1. 欢迎加入训练营

一、前端如何学习成长

1. FE coder也是coder，coder的成长路线
2. FE coder的特殊学习路线
3. 知识体系建立的重要性
4. 每个知识点如何去找寻关联

二、前端成长路径

1. 不同阶段需要的关键能力
2. 不同阶段的基本必备以及向上突破点

预习资料

<https://github.com/JacksonTian/fks>

前端基础及入门知识体系较为完整的repo

引用和列举了许多前端相关的必备工具，可以作为入门或基础知识点辅助提升

Part2. 基础概念 - 指针 | 闭包 | 作用域

一、作用域 & 提升

什么是作用域链？

作用域与函数提升？

作用域与变量提升？

提升优先级与块级作用域？

考察重点：

1. 作用域判断 —— 常见于笔试/机试
2. 变量提升优先级 —— 综合题中的一部分判断
3. 块级作用域
循环中的块级作用域（常见基础）
利用块级处理单次应用变量
利用块级优化性能

二、上下文 & 指针

什么是上下文？ —— 用于完整理解承上启下逻辑

面试如何考察？

1. 函数中的上下文 —— 函数表达式、匿名函数、嵌套函数
2. 隐式绑定 —— 对象、数组等引用关系逻辑
3. 显式绑定 —— call | bind | apply区别与关联

4. 类相关的上下文 —— new 的加入
同步 & 异步的区别呢？

5. 如何突破作用域限制？

考察重点：

1. 各种类型上下文指向 —— 笔试
2. 显示绑定与隐式绑定的对比
3. 手写显示绑定方法
4. 各种this的合理利用与继承 => 与原型链相关整合

三、 闭包

1. 什么是闭包 —— 函数和他周围状态的引用捆绑在一起的组合

2. 闭包的形态

函数作为返回值

函数作为参数

函数嵌套

3. 立即执行函数

闭包为何经久不衰？

函数封闭作用域将代码天然地隔绝开来，从而实现局部的封闭与隔离

闭包则是将函数内部和函数外部连接起来的桥梁。
在模块的隔绝基础上实现了参数变量的打通与传递

考察重点：

1. 几种不同闭包形式的写法
2. 函数嵌套与HOC
3. 立即执行函数与相关问题 => 与模块化知识点整合
4. 实现步进执行与方法封装

预习资料

<https://www.liaoxuefeng.com/wiki/1022910821149312/1023021053637728>

Javascript高级编程、犀牛书以及各种站点的知识点汇总都可以作为基础系统性学习的参考

有时间的情况下建议完整阅读，时间紧张的情况下，建议将基础部分务必详解后

可以将书本作为工具书，遇到问题后查询回顾