

拉勾教育

— 互联网人实战大学 —



大前端 就业急训营

「课程大纲」v1.1

大前端就业急训营

0 基础到高薪就业，多个热门项目实战
带你成为企业真正需要的前端人才

唯一不可或缺

门槛更低

更适合初学者学习

140万+ 岗位空缺

更易上手

每月大厂内推

金牌老师一对一指导

高薪就业



8 大阶段
体系化进阶

280+
课时

114+
核心知识点

24 个
企业实战案例

1v1
实时在线辅导

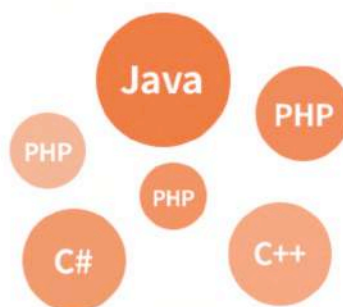
终身
技术生态圈

FRONT END
前端
占据互联网行业的半壁江山



JavaScript

VS



竞争激烈的后端语言

Atwood 定律：“任何可以用 JavaScript 来写的應用，最终都将采用 JavaScript 来写”

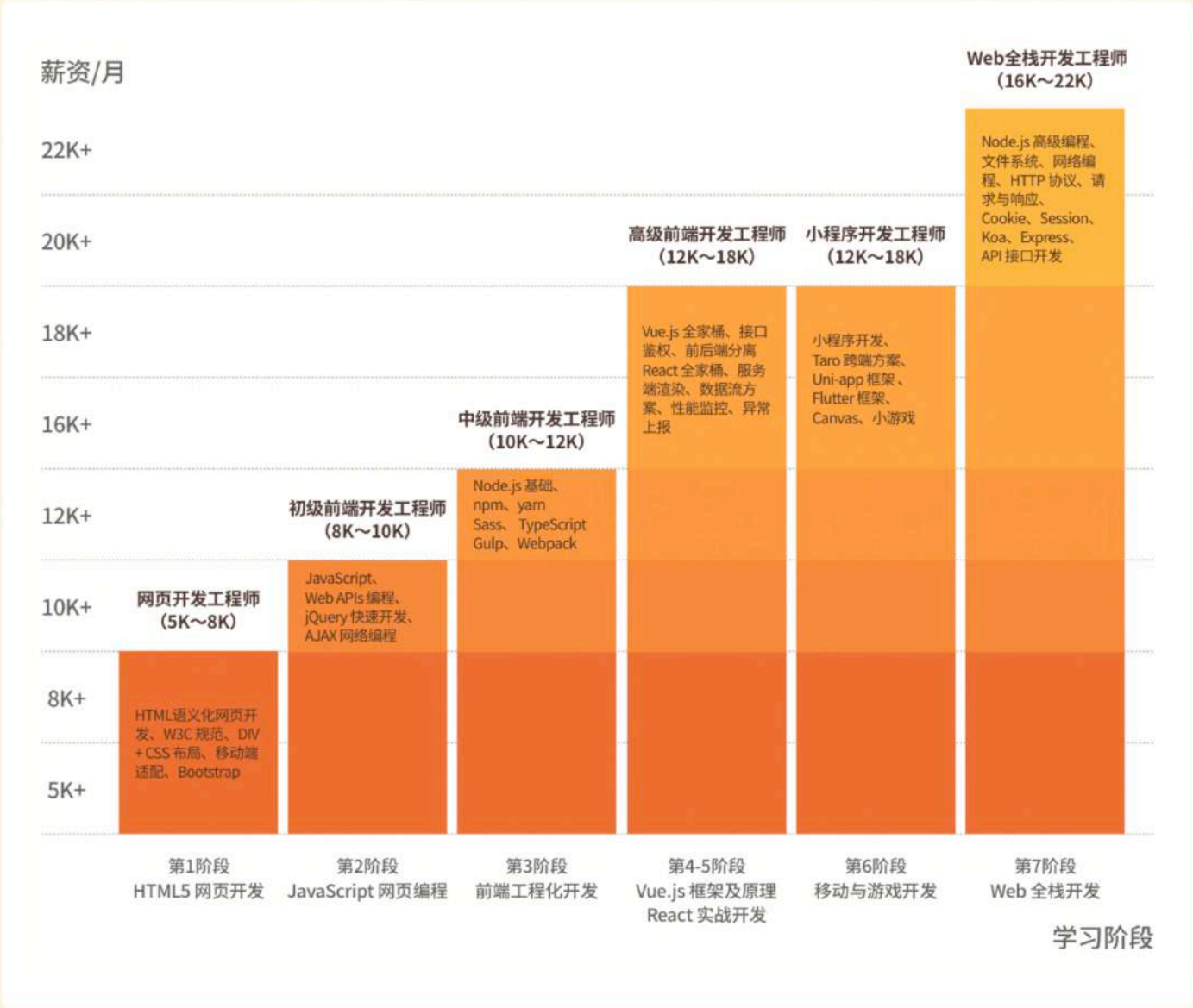
FRONT END
高薪就业才是你的核心目标

从 0 基础到「高薪」就业
180 天即可获得 2-4 年的工作经验 快速成为高级前端工程师

01 前端工程师养成路径



02 阶梯式内容设置，科学成长计划



0 基础不用怕 跟着学
你也能成为别人眼里的前端高手

01 结合拉勾 90万+ 企业真实用人需求，覆盖更全面技术点，实时更新，满足企业刚需

拉勾求职绿色通道 直通一线互联网公司面试官



大厂定向直推

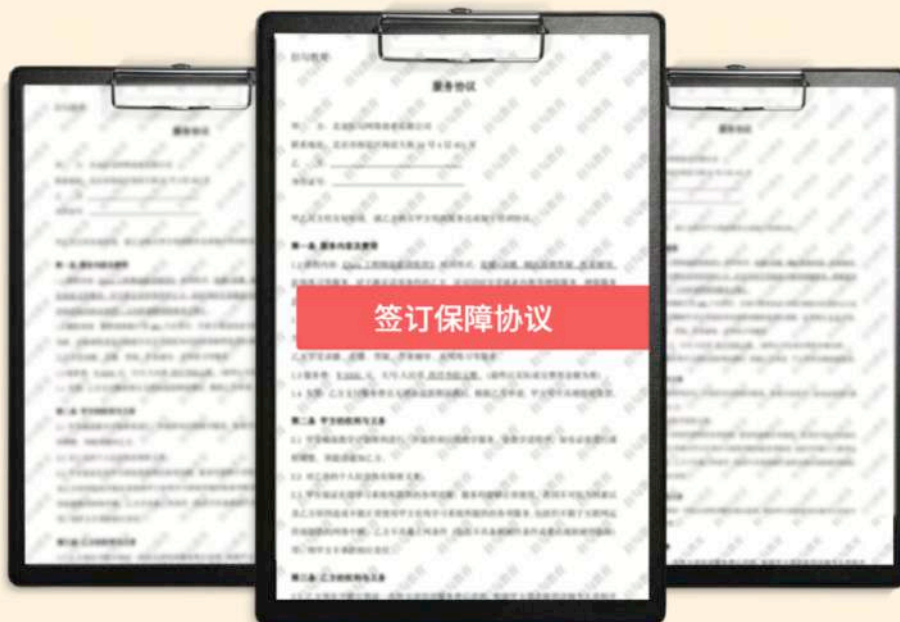


1v1 面试指导、面试复盘



告别筛选
简历直达技术leader

提供保障协议



支持信用卡分期 0手续费 0利息
轻松入学 帮你减轻负担

| | |
|-------|---|
| 匹配岗位 | 网页开发工程师 (5K~8K) |
| 学习目标 | <ul style="list-style-type: none"> - 正确认识前端以及前端开发在互联网中究竟属于一个怎样的工种 - 了解快速成为前端大神所必要的学习路径及阶段侧重点 - 掌握当下前沿的 Web 标准组成，了解下一步的发展趋势 - 掌握 HTML + CSS 配合设计软件切图，开发静态网页 - 通过不同的屏幕适配方案开发能够在移动端正常使用的网页 - 了解网站运行的原理及服务端客户端交互的过程 |
| 核心知识点 | HTML + CSS、W3C 规范、语义化网页开发、基础的 SEO、盒子模型、媒体查询、Flex 布局、移动端适配、CSS 动画、3D 效果、Bootstrap、rem/vw/vh 等新单位、HTTP 协议 |
| 项目案例 | <ul style="list-style-type: none"> - 『拉勾网』PC 端网页项目 - 『京东商城』移动端网页项目 - 『携程旅行』移动端网页项目 - 『苏宁易购』移动端网页项目 |





| | |
|-----------------|--|
| 1) HTML 网页开发 | <ul style="list-style-type: none">- 互联网开发储备知识- HTML 的基本概念- HTML 语法快速掌握- HTML 基本组成结构分析- HTML 常用标签（标题、段落、图像、超链、表格、列表、表单、布局等） |
| 2) CSS 层叠样式表 | <ul style="list-style-type: none">- CSS 概念和作用- CSS 语法及常用属性- CSS 选择器及应用- 最为核心的层叠式特性- CSS 常用字体、文本样式- CSS 盒模型、布局基本功- CSS 背景相关属性和特性- CSS 浮动和定位 |
| 3) 拉勾静态网页项目 | <ul style="list-style-type: none">- 页头页尾实现- 版心设计、页面布局实现- 网站开发流程- WEB 开发常见规范- 网站开发制作技巧 |
| 4) HTML5 + CSS3 | <ul style="list-style-type: none">- HTML5 概述- HTML5 新增语义化布局标签- HTML5 新增多媒体标签- HTML5 新增表单标签和属性- CSS3 新增属性- CSS3 新增 3D、动画制作 |
| 5) 移动端网页项目 | <ul style="list-style-type: none">- 移动端基础- 视口概念及价值- 移动开发选择和技术解决方案- 流式布局、项目- Flex 布局常见属性、原理、项目- Rem 适配布局- 媒体查询、移动端适配- Less 语法及编译工具- rem 布局方案及相关项目- Bootstrap CSS 框架实战- 响应式布局及相关项目 |



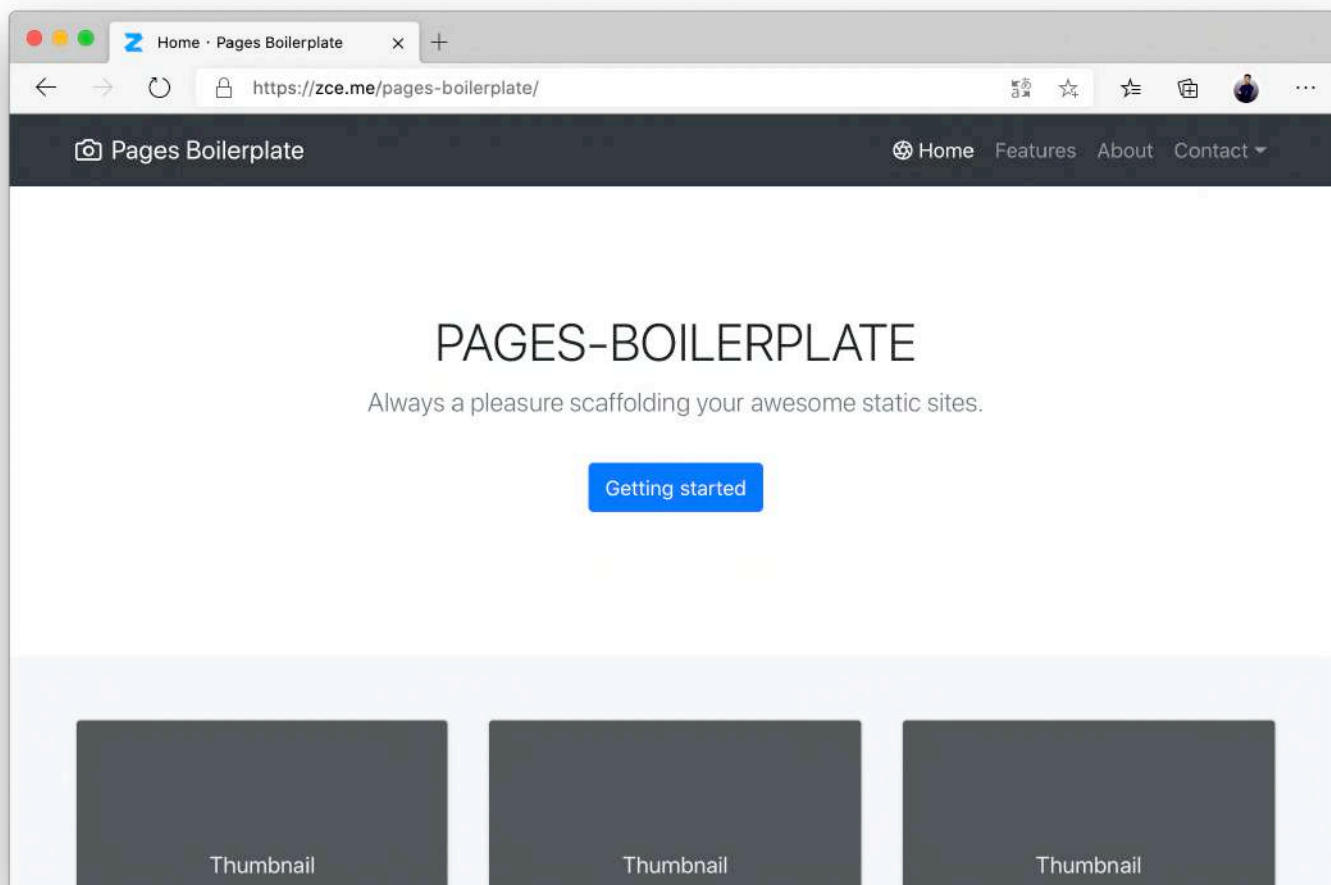
| | |
|-------|--|
| 匹配岗位 | 初级前端开发工程师 (8K~10K) |
| 学习目标 | <ul style="list-style-type: none">- 快速掌握你的第一门编程语言：JavaScript- 深入理解 JavaScript 的高级特性及原理- 通过大量网页应用的开发锻炼提高编程能力- 掌握网页开发中局部业务功能的设计及编码- 能够使用 jQuery 提高网页应用开发效率- 能够使用 AJAX 完成应用与服务端的数据通信 |
| 核心知识点 | 基本的数据结构、JavaScript 语法、Web APIs 编程、闭包、递归、原型、原型链、面向对象、正则表达式、ES6、ES7、jQuery 快速开发、jQuery 常用插件、jQuery 原理、AJAX 网络编程、Axios |
| 项目案例 | <ul style="list-style-type: none">- 网页轮播图特效案例- 元素入场动画特效案例- 『贪食蛇』小游戏项目- 『网络留言板』网页应用项目- 『豆瓣电影榜单』网页应用项目- 『网页记事本』网页应用项目- 『Todos 待办清单』网页应用项目- 『购物车』局部功能案例 |





| | |
|----------------|---|
| 1) JS 基础语法 | <ul style="list-style-type: none">- JavaScript 概述及基本语法- 字面量概念和变量机制- 原始数据类型、数组类型- 操作符、运算符、表达式和语句- 条件分支语句、循环语句- 函数基础、函数递归、作用域- 预解析、执行原理- 对象的创建和使用- 内置对象 (Math、Date、Array、String 等) |
| 2) Web APIs 编程 | <ul style="list-style-type: none">- Web APIs 介绍、组成部分- DOM 概念、操作 HTML 属性与 CSS 样式- DOM 事件操作、节点操作- BOM 的概念、顶级对象 window- 定时器和延时器、简单运动动画实现- location、history、navigator- 偏移量、滚动偏移、页面特效制作 |
| 3) JS 高级特性 | <ul style="list-style-type: none">- JavaScript 面向对象编程- 面向对象概述、创建对象、构造函数- 原型、原型链- 随机方块游戏、贪吃蛇游戏制作- 函数进阶、高阶函数、函数闭包、正则表达式 |
| 4) jQuery 快速开发 | <ul style="list-style-type: none">- jQuery 介绍、对象和 DOM 对象- jQuery 的选择器、操作样式、注册事件- jQuery 操作标签属性、类名操作- jQuery 操作元素内容、动画、节点操作- jQuery 操作元素的尺寸和位置- 网页内多库共存问题- jQuery 插件的使用 |
| 5) AJAX 网络编程 | <ul style="list-style-type: none">- AJAX 概述、快速上手- HTTP 协议 (IP、端口、域名、DNS、URL)- XMLHttpRequest 对象解析- HTTP 协议、GET / POST 请求- 同步与异步模式下的差异- 留言板案例、JSON、电影榜单案例- 封装 AJAX 调用、回调 / 委托- 跨域问题、JSONP 和 CORS 解决方案- jQuery 中的 AJAX、Axios 介绍和使用- XHR 2.0 新特性、会员管理综合案例 |
| 6) ES6+ 新特性 | <ul style="list-style-type: none">- 块级作用域、模板字符串、对象与数组的解构、rest 操作符- 函数进阶 (箭头函数、默认参数)、对象和数组的扩展用法- Proxy、Reflect、Map、Set、Symbol、for...of、迭代器模式、生成器函数- 新特性编译工具 (Babel) 的使用、Polyfill: CoreJS 标准库 |

| | |
|-------|---|
| 匹配岗位 | 中级前端开发工程师（10K~12K） |
| 学习目标 | <ul style="list-style-type: none">- 能够配置前端工程化所必要的环境和工具，并理解其中关系- 掌握如何借助 NPM 工具自动化的管理、维护项目依赖- 掌握通过 Less/Sass、Babel 等工具提升编码效率，减少项目维护成本- 能够使用 ESLint 等工具规范项目编码，确保项目的代码质量- 能够使用 Gulp 设计一个前端自动化开发工作流，提升开发效率- 了解如何针对不同类型的项目设计合理的工程化项目 |
| 核心知识点 | Git、Node.js 基础、npm / yarn、Less / Sass、PostCSS、CSSNext、Babel、ESLint、StyleLint、npm scripts、Gulp、Rollup、Webpack |
| 项目案例 | <ul style="list-style-type: none">- 通用前端工程化项目模板- 搭建一个可以复用的项目模板，提升后续开发效率- 『静态站点项目』工程化项目 |





1) 工程化基础储备

- 工程化组成和项目中的表现
- 工程化与工具之间的关系
- Windows 常用命令 (cd、dir、cls、exit 等)
- Linux 常见命令 (ls、pwd、cp、ln、cat、vi、chmod 等)
- Git 原理及操作命令、分支概念及操作命令
- GitHub 仓库使用
- SVN 工具介绍
- Node.js 介绍与环境配置
- 使用 node 执行 js 文件
- 浏览器环境下的 js 与 Node.js 环境下的区别

2) 脚手架工具

- 脚手架设计思想与本质作用
- 常用的脚手架工具一览
- Yeoman 的基本使用以及自定义 Generator
- Yeoman Sub Generator 特性
- 基于 Yeoman 创建适合自己的脚手架工具
- Plop 生成器的基本使用
- 使用 Plop 提高项目创建同类文件的效率
- 脚手架工作原理剖析、手写一个自己的脚手架工具

3) 自动化构建

- 如何使用自动化构建提高开发效率
- 常用的自动化构建任务执行器
- npm scripts & script hooks
- Grunt 工具的使用及扩展开发
- Gulp 的使用以及任务结构
- 基于 Gulp 创建自动化构建工作流
- 封装独立的自动化构建工具

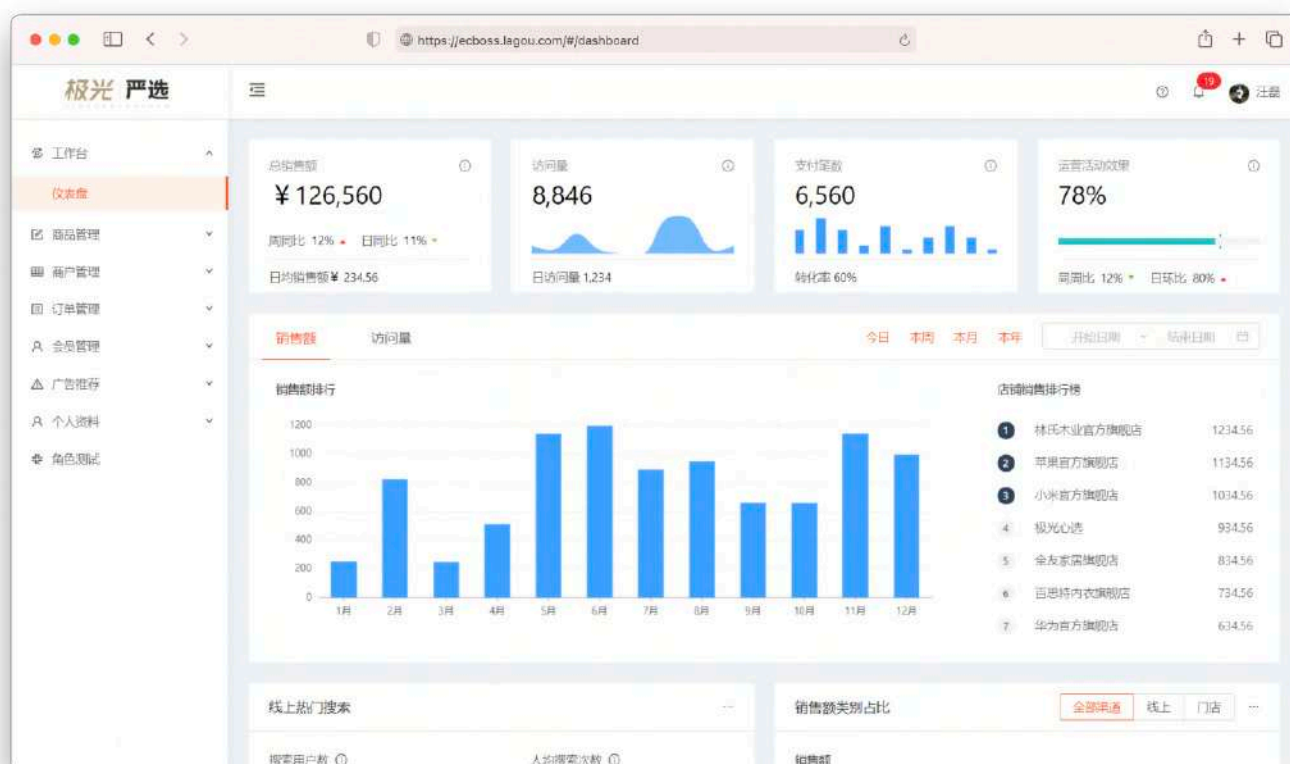
4) Webpack 模块化

- 模块化标准与规范
- ES Modules 标准的支持情况
- Webpack 打包工具的基本使用、配置详解
- Webpack 打包过程和打包结果分析
- Webpack 中资源模块的加载 (Loader)
- 如何开发一个 Webpack 的 Loader
- Webpack 的插件机制、开发一个 Webpack 插件
- Webpack 周边生态 (Dev Server、HMR、Proxy)
- Webpack 高级特性 (Tree-shaking、sideEffects)
- Webpack 打包过程及打包结果的优化
- 其他常见的打包工具 (Rollup、Parcel)

5) 规范化标准

- 常见的代码 Lint 工具 (ESLint、Stylelint)
- 创建项目或者团队专属的 Lint 规则 / 风格
- 通用型代码格式化工具 Prettier
- 结合自动化工具或者 Webpack 的使用
- 配合 Git Hook 确保源代码仓库中代码的质量
- 结合脚手架、自动化、模块化、规范化搭现代化前端工程

| | |
|-------|--|
| 匹配岗位 | 高级前端开发工程师（12K~18K） |
| 学习目标 | <ul style="list-style-type: none"> - 理解数据驱动形开发与传统界面驱动形开发的差异及各自优缺点 - 能够掌握 Vue.js 主流框架的使用以及全家桶配置搭建 - 了解如何借助 MVVM 思想组织大型应用项目的结构 - 能够使用数据流管理方案解决超大型应用中数据扭转的问题 - 了解如何使用服务端渲染方案解决 SPA 应用的性能及 SEO 问题 - 理解 MVVM 框架原理，开发一个类似 Vue.js 的框架 |
| 核心知识点 | Vue.js、插值表达式、常用指令、计算属性、侦听器、事件处理、组件、自定义事件、生命周期、MVVM 思想、defineProperty / Proxy、Vue Router、Vue CLI、Vue Devtools、Vue.js 组件化开发、Element、Vuex、前后端数据交互、JWT 接口鉴权、Vue.js 3.0、Composition APIs |
| 项目案例 | <ul style="list-style-type: none"> - 项目一：『拉勾心选』电商后台管理系统 - 项目二：『拉勾心选』电商门户系统 - 项目三：『拉勾心选』数据实时监测平台 |





1) Vue.js 框架基础

- Vue.js 基本使用、模板插值语法
- 指令的概念和基本用法
- 双向数据绑定、MVVM 设计思想、Vue Devtools
- 事件绑定、属性绑定、分支循环结构
- Class 和 Style、样式处理
- 过渡动画、表单操作、自定义指令
- 计算属性、过滤器、侦听器、Vue 的生命周期
- 基于 Vue CLI 的图书管理 (CRUD) 案例

2) 客户端路由

- 单页应用的概念
- 路由及 Vue Router 概述
- 路由原理分析
- 基本使用步骤、子路由用法
- 动态路由的用法和参数传递
- 编程式路由规则和用法
- 基于路由的案例

3) Vue.js 实战开发

- 基于 Webpack 搭建 Vue 开发环境
- Vue 项目工程化及 Vue CLI 脚手架使用
- Vue 单文件组件使用及优势
- 组件化思想、注册组件
- 组件间传值、组件插槽
- 基于组件的案例
- 第三方 UI 组件库 Element-UI
- Vue.js、Vue CLI、Vue Router、Axios 结合开发
- Vue 调试工具使用
- 注册登陆、身份认证、权限鉴定
- 常规后台业务 (CRUD、表单处理、文件上传等...)
- 发布与部署 (CI/CD)

4) Vue 核心实现原理

- MVVM 框架介绍与双向数据绑定
- Vue 框架基本结构、compile 模块
- fragment 及子节点的解析步骤
- 解析指令的基本步骤
- 解析 v-on 指令、通过 compileUtil 简化代码
- 解析插值表达式、显示复杂数据
- 数据响应式实现相关源码分析
- Vuex 数据流方案
- Vuex 基本用法与概念分析
- 基于 Vuex 的案例

5) Vue.js 高级进阶

- Vue Router 原理
- Vuex 原理
- Vue SSR 扩展
- 深入组件封装
- 关于 Vue.js 的大厂面试题

| | |
|-------|---|
| 匹配岗位 | 高级前端开发工程师（12K~18K） |
| 学习目标 | <ul style="list-style-type: none">- 理解数据驱动形开发与传统界面驱动形开发的差异及各自优缺点- 能够掌握 React 框架的使用以及全家桶配置搭建- 了解如何借助 MVVM 思想组织大型应用项目的结构- 能够使用数据流管理方案解决超大型应用中数据扭转的问题- 了解如何使用服务端渲染方案解决 SPA 应用的性能及 SEO 问题- 了解 React 在编程能力方面的优势 |
| 核心知识点 | React、JSX 语法、元素渲染、组件 & Props、State & 生命周期、组件、事件处理、条件渲染、列表渲染、React Router、Create React App、Redux、Mobx、React Hooks、服务端渲染 |
| 项目案例 | <ul style="list-style-type: none">- 项目一：『拉勾课堂』后端管理平台- 项目二：『拉勾课堂』门户系统项目 |





1) React 框架基础

- React 基本用法、JSX 基础语法
- 组件化开发思想分析
- 用函数方式创建组件、用类方式创建组件
- 组件的内部状态处理、组件的生命周期
- 模板语法、JSX 进阶特性
- 高阶组件概念和应用
- 非父子组件之间传值

2) 前后端交互

- 接口调用、fetch api 基本用法
- fetch api 不同请求方式的参数传递
- fetch api 响应数据处理
- axios 用法强化
- 基于接口的图书系统案例

3) 客户端路由

- 前端路由的本质分析
- 前端路由的实现原理-基于 hash
- 前端路由的实现原理-基于 history
- React Router 模块基本用法
- 嵌套路由用法、路由重定向用法
- 路由参数的传递方式、程式化路由的用法
- 自定义路由链接样式

4) React 综合项目

- 项目开发流程分析
- 项目业务分析、技术选型分析
- 初始化项目结构
- 基于 git 管理项目
- 注册、登录、退出
- cookie、session、token 演进分析
- token 登录流程分析
- 项目主页模块、咨询模块、聊天模块、个人中心
- WebSocket 基本使用
- 性能优化
- 提取第三方 cdn 链接
- 配置 gzip 打包效果
- 项目打包生成 dist 目标内容
- 配置服务器环境、部署上线
- 项目代码获取与打包

5) Redux 状态管理

- 状态管理模块用法与核心概念分析
- 状态管理必要性分析
- action、reducer、store
- redux 数据交互流程分析
- 基于状态管理的案例
- 案例功能分析
- 初始化项目、实现业务功能
- 将数据基于 redux 进行重构

| | |
|-------|--|
| 匹配岗位 | 小程序开发工程师 (12k-18k) |
| 学习目标 | <ul style="list-style-type: none"> - 了解微信小程序的设计理念以及与 App 有何不同 - 掌握如何使用前端技术开发一款自己的微信小程序 - 能够通过接入第三方云服务，不用服务端支持也能开发小程序 - 了解 Web、App、小程序、H5 等多终端统一开发解决方案：uni-app - 了解如何能够使用 Flutter 技术开发跨平台的移动 App - 了解前端技术在微信小游戏 / H5 游戏开发中的应用 |
| 核心知识点 | 开发工具使用、小程序配置详解、小程序 UI 开发、WXSS 样式、小程序 API 编程、事件处理、生命周期、wxs 抽象、网络编程、Taro 方案、uni-app 框架、Flutter 框架、Canvas、小游戏 |
| 项目案例 | <ul style="list-style-type: none"> - 项目一：『AI 测颜值』小程序项目 - 项目二：『本地生活』- 类似大众点评的小程序项目 - 项目三：『疯狂猜成语』微信小游戏项目 |





1) 小程序开发基础

- 微信开发的概念
- 微信开放平台、公众平台
- 小程序和手机 App 之间的本质区别
- 小程序中的组件和 API
- 小程序开发账号的注册和设置
- 小程序开发者工具的使用
- 小程序项目的结构、页面的结构
- 小程序常用的 UI 组件、样式
- wxss 与选择器、尺寸单位 rpx
- @import 样式导入、全局样式与局部样式
- 全局配置、页面配置
- 小程序的生命周期

2) 小程序的视图与逻辑

- 数据绑定与事件绑定
- wxs 脚本概述、基础语法
- 内嵌 wxs 脚本和外联 wxs 脚本
- 页面渲染、页面事件、页面导航
- 自定义组件、组件传值
- 网络数据请求

3) uni-app 框架

- 环境搭建、页面外观设置
- 数据绑定、生命周期
- 组件使用、条件编译、计算属性
- uni 中的事件
- 导航跳转、组件通信
- 项目实战、跨平台支持

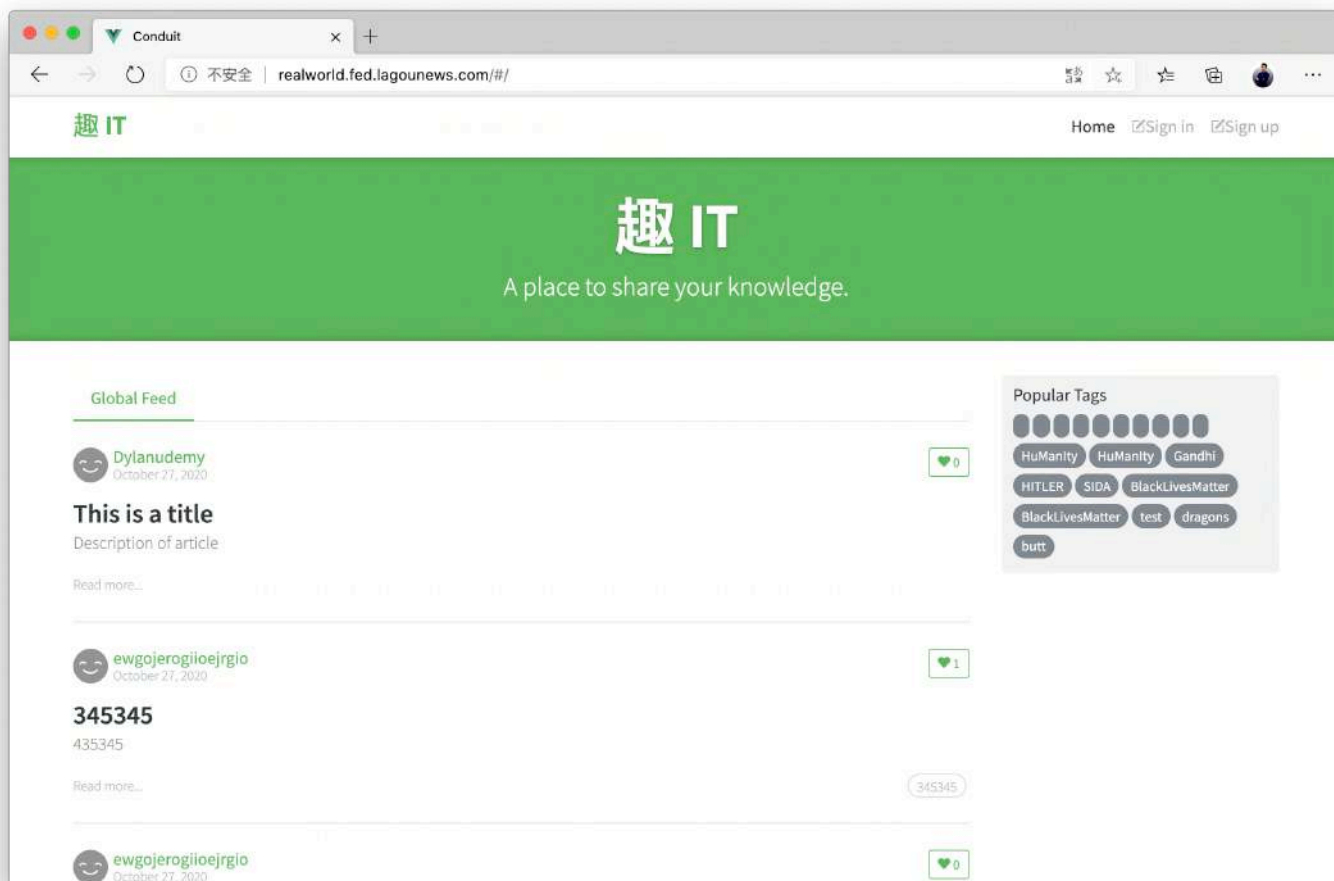
4) 小程序项目实战

- 初始化项目
- 首页功能的开发
- 分类页面的开发
- 商家列表页面的开发
- 商家详情页面的开发
- 个人中心页面的开发

5) Flutter 原生开发

- Flutter 概述以及 Windows / macOS 环境搭建
- Dart 语言快速上手、包管理工具
- Flutter 快速上手、开发体验、路由和导航
- UI 开发：内置 Material Design 和 Cupertino (iOS风格) Widget
- 常用 Widget、表单组件、布局方式
- 数据响应：界面状态管理
- 网络编程以及相关第三方包
- Native 功能和 SDK 的调用

| | |
|-------|---|
| 匹配岗位 | Web 全栈开发工程师（16K~22K） |
| 学习目标 | <ul style="list-style-type: none">- 了解 Node.js 的定位及对前端开发行业的意义- 掌握基于 Node.js 平台的 CLI 工具开发- 能够通过 Node.js 完成基本的服务端应用开发- 能够掌握基本的 MySQL 数据库操作- 能够使用 Express / Koa 完成服务端接口的开发- 能够掌握 Cookie、Session 等状态保持方案 |
| 核心知识点 | Node.js、CommonJS、文件系统、异步编程、动态网站、HTTP 协议、请求与响应、Express、Cookie、Session、Koa、接口开发 |
| 项目案例 | - 『趣 IT』 - 类似知乎的在线社交媒体平台 |





1) Node.js Web 基础

- 核心模块、fs 模块、path 模块
- HTTP 模块搭建 Web 服务器
- 浏览器和服务器的通信过程
- 创建 Web 服务器步骤
- 接收浏览器的请求、给浏览器做出响应
- 解决中文乱码
- 根据请求 URL 返回静态资源
- 静态 Web 服务器开发综合项目案例
- 动态网站的定义、动态渲染列表
- 表单处理
- art-template 模板引擎的使用
- 第三方模块、npm 的使用
- 全局安装和本地安装
- package.json
- require 的模块加载机制
- 中间件介绍及主要分类
- 内置中间件与第三方中间件
- 应用程序级别中间件、路由级别中间件
- 错误处理中间件
- Cookie & Session

2) Express 与 MySQL

- Express 介绍及使用
- Express 快速搭建 Web 服务器
- 基本路由、处理静态资源
- bodyParser 解析 post 请求体
- Express 结合模板引擎实现动态网站
- 数据库基础知识
- MySQL 的安装及客户端使用
- SQL 语句及语法、导入和导出 SQL 文件
- Node.js 操作 MySQL

3) 自定义模块

- Node.js 中的模块
- 导入模块、导出模块
- require 的加载机制
- 封装 mysql 操作

4) Express 项目实战

- 项目搭建、入口文件
- 配置路由、模板继承
- 列表页面、删除数据、添加数据、编辑数据
- 第三方模块、路由模块
- 项目起步
- 分类管理、用户管理
- Promise 的介绍和使用
- 用户登录和状态保持
- 文章管理、个人中心、用户端

| | |
|-------|--|
| 匹配岗位 | 高级前端开发工程师（16K~22K） |
| 学习目标 | <ul style="list-style-type: none">- 掌握企业级项目开发流程，了解前端如何快速切入- 了解互联网研发团队组成、职责划分以及协作模式- 掌握什么是敏捷开发模式，如何通过这种模式提高产能- 掌握一般我们在公司使用的团队研发管理工具- 学会如何持续学习不断提升自己在职场中的竞争力- 掌握 BATJ 等大厂的高频面试真题- 培养面试中应具备的能力和技巧- 学会打造一份优秀的、让人无法拒绝的简历 |
| 核心知识点 | 企业级项目开发流程、互联网研发团队协作、敏捷开发模式、团队管理工具、源码管理技巧、大厂高频面试真题解析、面试专项能力快速提升方法、面试技巧、简历打造技巧 |
| 项目案例 | - 基于 Vue.js 的『趣 IT』移动端项目（二次开发） |





| | |
|------------------|---|
| 1) 团队协作二次开发 | <ul style="list-style-type: none">- 企业级项目开发流程- 互联网研发团队协作- 敏捷开发模式- 团队管理工具- 源码管理技巧- 基于 Vue.js 的『趣 IT』移动端项目（二次开发） |
| 2) BATJ 高频面试真题 | <ul style="list-style-type: none">- BATJ 高频原理题解析- BATJ 高频应用题解析 |
| 3) 面试专项能力突击 | <ul style="list-style-type: none">- 一面：编程基础能力考察- 二面：项目经验考察- 三面：方案设计综合能力考察- 四面：HR综合软素质考察 |
| 4) 面试过程发挥应有水平 | <ul style="list-style-type: none">- 学会表达：如何把硬实力表达出来- 扬长避短：如何表现出自己的闪光点- 洞悉套路：面试考察点与答题套路指导- 面试指导：面试前1v1辅导- 面试复盘：面试后1v1复盘 |
| 5) 打造一份让人无法拒绝的简历 | <ul style="list-style-type: none">- 高分简历模板分析- 打造一份高分简历的方法- 模拟训练：重写简历 1v1 指导服务 |

拉勾教育

— 互联网人实战大学 —