

# 前端课程体系

王红元 coderwhy

# 主要内容



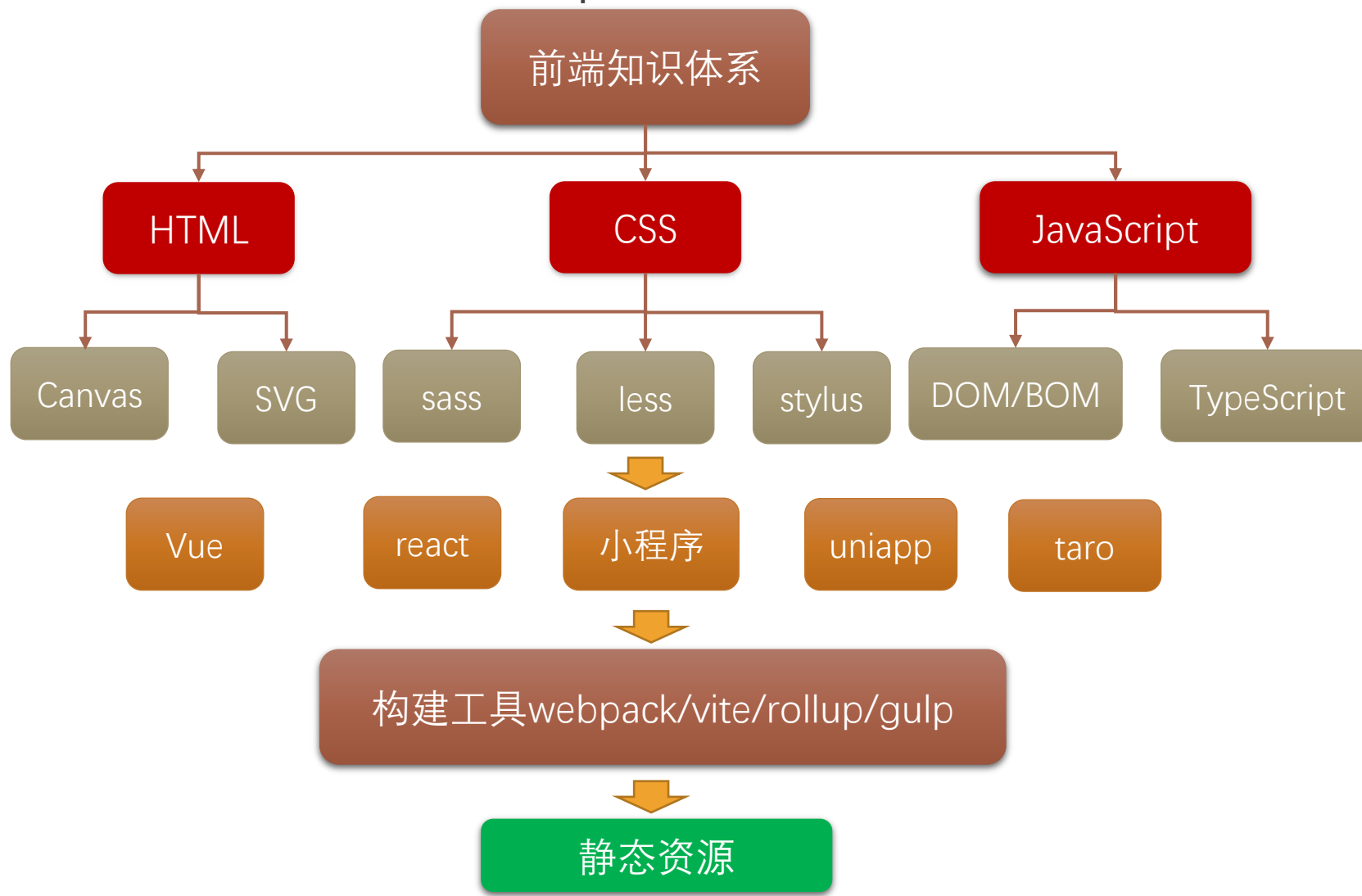
**1** 实践单位情况

**2** 实践的主要内容

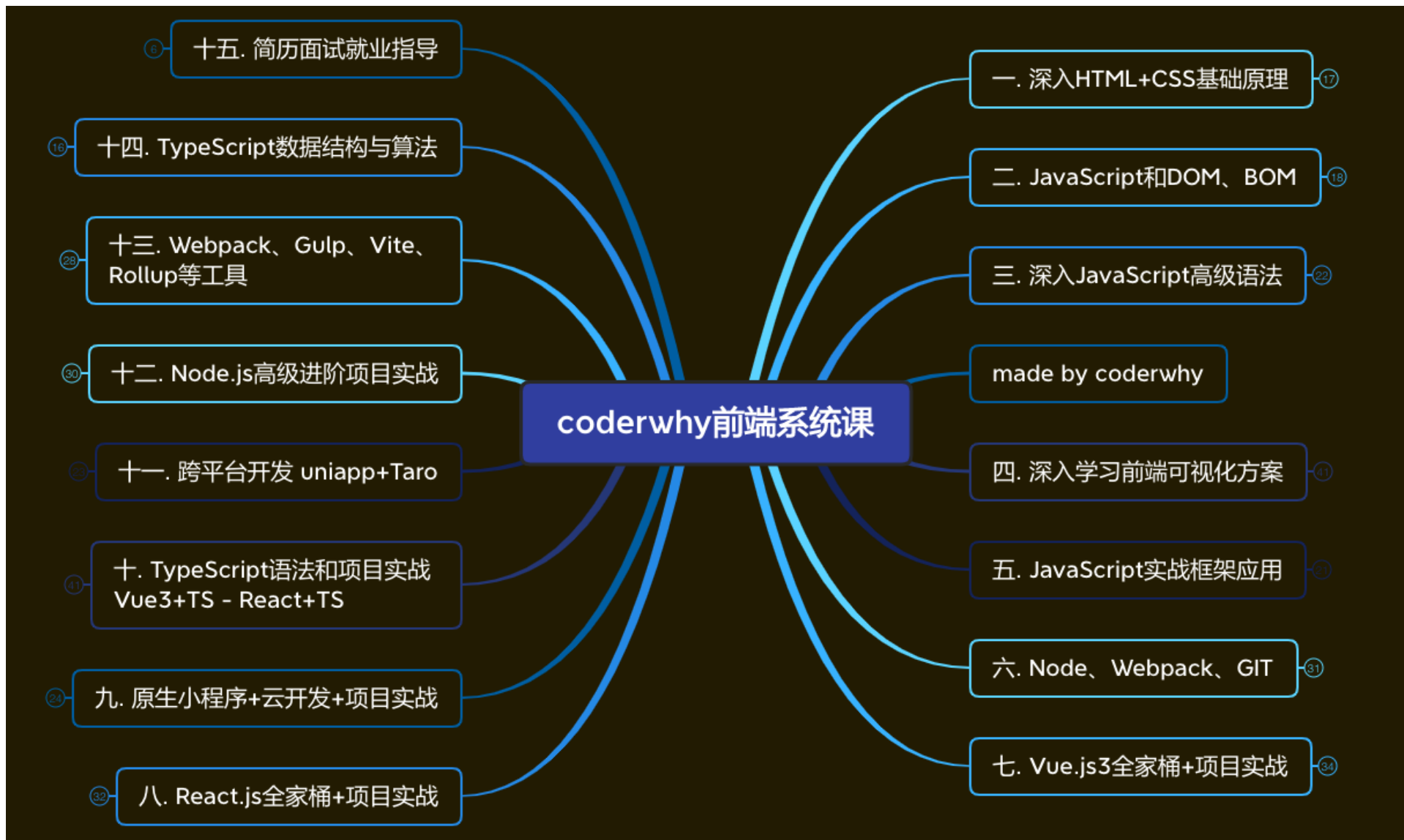
**3** 实践单位意见

**4** 自我总结

- 前端知识体系的核心：HTML + CSS + JavaScript。



# 前端课程大纲（薪资结构）



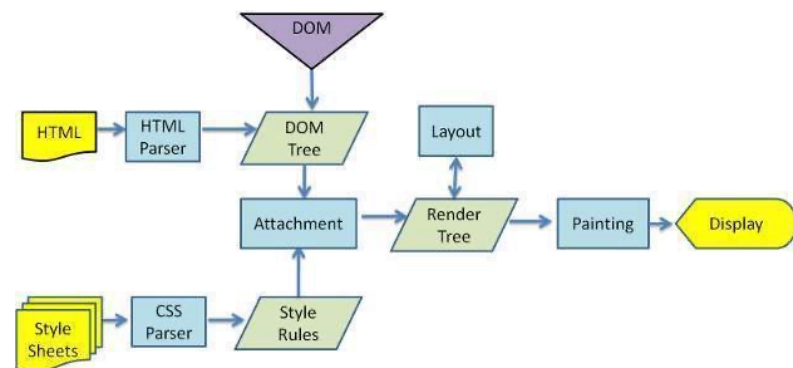
# 相互尊重、共同进步

## ■ 1.所有的内容是从零开始的

- 比如包括什么是**软件开发**，什么是**前端工程师**；
- 比如一个网页从你开始**编写的那一刻**到**用户浏览器**上可以显示，它到底**经历了什么样的一段旅程**；
- 比如什么是HTML、什么是CSS，这些东西给到浏览器，**浏览器到底是如何给你显示的**；
- 比如**JavaScript**是如何被浏览器解析的；

## ■ 2.从零开始，也是**干货满满**，比如CSS

- 比较不好理解的知识：vertical-align、line-height、base-line、viewport；
- 比如div的width的100%和auto之间的区别；
- 比如css移动端适配vw、vh、rem；
- 比如BFC到底是什么，有什么作用；
- 等等...



Juozas Kaziukėnas  
@juokaz



Follow

“44 years ago we put a man on the moon, but we still can’t vertically center things in CSS.”

RETWEETS

328

LIKES

74



2:16 PM - 21 Jul 2013





# 开发规范、知识补充

## ■ 开发规范：

- 目录组织、文件结构；
- 文件夹命名、文件命名、类、函数、变量等等命名规范；
- 代码缩进问题（2个空格、4个空格）、注释规范；
- 代码抽取、封装、设计思想；

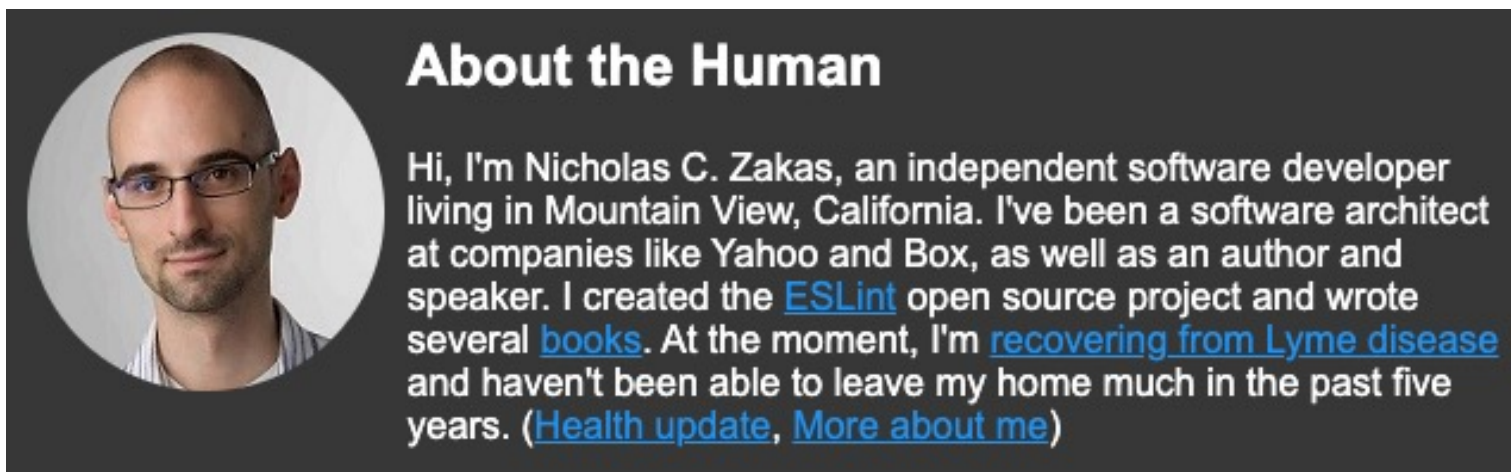
## ■ 知识补充：

- 比如SEO优化 – 搜索引擎工作原理、爬虫、SSR，优缺点、来龙去脉，我们都会讲到；
- 某些知识讲的时候会额外补充很多其他知识，比如网络协议、服务器知识；
- 甚至是为了让大家深层次理解一些知识，会涉及到操作系统、编译原理的知识；
- 数据结构、算法、设计模式的知识；

# 学习方法、学习建议

## ■ 如何学习新的知识？

- 《JavaScript高级程序设计》的作者Nicholas C.Zakas，也是ESLint的作者；



- 在他的博客有说到：想要全面的掌握JavaScript，关键在于弄清楚它的历史、局限性和本质；
- 我个人非常认同这个观点，并且我认为**不仅仅局限于JavaScript**，而是**对任何技术都是适用的**。

# 如何学习新技术？

## ■ 技术的出现是为了解决问题的，不是为了让大家学技术的

- 比如为什么HTML存在*i*、**strong**这些标签，同时还会有对应的css属性可以实现斜体、加粗呢？
- 比如为什么JavaScript刚开始使用var定义变量，后面又多了let、const？
- 比如为什么Vue、React主要是做SPA页面的，又出现了SSR的框架？uniapp、taro的意义是什么？
- 比如webpack已经那么强大了，为什么还会出现vite，它解决了webpack什么痛点，如何解决的？
- 比如为什么存在浏览器的兼容性问题？

## ■ 甚至很多组织、机构的出现都是有原因的。

- 比如W3C这个组织为什么会出现？ECMA和W3C分别是做什么的？
- 一些标准到底是谁制定的，比如CSS标准、DOM标准、JavaScript语言规范；
- W3C的官网是什么？ECMA的官网是什么？

## ■ 学习技术也是为了解决问题的。



# 知识体系的分类

- 那我能不能直接先进公司，到公司做项目，遇到问题再学呢？
  - 不可以，因为公司不给你这个时间；
  - 就算可以，公司给你这个时间，你在没有任何储备的情况下学得会吗？

- 那么知识点这么多，能全部掌握吗？

- 将知识进行分类：



- 每个小视频标注：（掌握）（理解）（了解）

# 优秀是一种习惯

“我们每一个人都是由自己一再重复的行为所锻造而成，因而优秀不是一种行为，而是一种习惯。”

——亚里士多德

