

尾声 | 长风破浪会有时，直挂云帆济沧海 | 极客时间

开篇词 | 从今天起，重新理解前端

01 | 明确你的前端学习路线与方法

02 | 列一份前端知识架构图

03 | HTML语义：div和span不是够用了吗？

04 | HTML语义：如何运用语义类标签来呈现Wiki网页？

05 | JavaScript类型：关于类型，有哪些你不知道的细节？

06 | JavaScript对象：面向对象还是基于对象？

07 | JavaScript对象：我们真的需要模拟类吗？

08 | JavaScript对象：你知道全部的对象分类吗？

新年彩蛋 | 2019，有哪些前端技术值得关注？

09 | CSS语法：除了属性和选择器，你还需要知道这些带@的规则

10 | 浏览器：一个浏览器是如何工作的？（阶段一）

11 | 浏览器：一个浏览器是如何工作的？（阶段二）

12 | 浏览器：一个浏览器是如何工作的？（阶段三）

13 | 浏览器：一个浏览器是如何工作的？（阶段四）

14 | 浏览器：一个浏览器是如何工作的？（阶段五）

15 | HTML元信息类标签：你知道head里一共能写哪几种标签吗？

16 | JavaScript执行（一）：Promise里的代码为什么比setTimeout先执行？

17 | JavaScript执行（二）：闭包和执行上下文到底是怎么回事？

18 | JavaScript执行（三）：你知道现在有多少种函数吗？

19 | JavaScript执行（四）：try里面放return，finally还会执行吗？

20 | CSS 选择器：如何选中svg里的a元素？

21 | CSS选择器：伪元素是怎么回事儿？

22 | 浏览器DOM：你知道HTML的节点有哪几种吗？

23 | HTML链接：除了a标签，还有哪些标签叫链接？

24 | CSS排版：从毕升开始，我们就开始用正常流了

25 | 浏览器CSSOM：如何获取一个元素的准确位置

26 | JavaScript词法：为什么12.toString会报错？

27 | (小实验) 理解编译原理：一个四则运算的解释器

28 | JavaScript语法（预备篇）：到底要不要写分号呢？

用户故事 | 那些你与“重学前端”的不解之缘

29 | JavaScript语法（一）：在script标签写export为什么会抛错？

期中答疑 | name(){}与name: function() {}, 两种写法有什么区别吗？

30 | JavaScript语法（二）：你知道哪些JavaScript语句？

31 | JavaScript语法（三）：什么是表达式语句？

32 | JavaScript语法（四）：新加入的**运算符，哪里有些不一样呢？

33 | HTML替换型元素：为什么link一个CSS要用href，而引入js要用src呢？

34 | HTML小实验：用代码分析HTML标准

35 | CSS Flex排版：为什么垂直居中这么难？

36 | 浏览器事件：为什么会有捕获过程和冒泡过程？

37 | 浏览器API（小实验）：动手整理全部API

38 | CSS动画与交互：为什么动画要用贝塞尔曲线这么奇怪的东西？

答疑加餐 | 学了这么多前端的“小众”知识，到底对我有什么帮助？

39 | HTML语言：DTD到底是什么？

40 | CSS渲染：CSS是如何绘制颜色的？

41 | CSS小实验：动手做，用代码挖掘CSS属性

加餐 | 前端与图形学

加餐 | 前端交互基础设施的建设

42 | HTML-ARIA：可访问性是只给盲人用的特性么？

43 | 性能：前端的性能到底对业务数据有多大的影响？

44 | 工具链：什么样的工具链才能提升团队效率？

45 | 持续集成：几十个前端一起工作，如何保证工作质量？

46 | 搭建系统：大量的低价值需求应该如何应对？

47 | 前端架构：前端架构有哪些核心问题？

期末答疑（一）：前端代码单元测试怎么做？

期末答疑（二）：前端架构中，每个逻辑页面如何可以做到独立发布呢？

winter 2019-05-25



你好，我是 winter。

我们一起度过了 **124** 天，共同阅读了 **185,364** 个字
收听了约 **11** 个小时的音频。

愿长风破浪会有时,直挂云帆济沧海。



你好，我是 winter。

感觉特别长的几个月，忽然间就结束了。对我而言，这是一段奇妙的旅程，对你来说，我希望也是。

去年 9 月我离开原来的公司，打定主意要做一些程序员教育的事情，10 月份在 QCon，我遇到了“极客时间”的编辑，收到邀请来做一个专栏，于是就有了“重学前端”。

最初我答应“极客时间”的时候，其实心里想的是：反正我要做程序员教育，做一个专栏就当整理自己的知识也好。

但是从一开始的磨合和存稿阶段，我就发现这个事情没有我想的那么简单，与以前零散地写博客相比，按时按体系去产出课程需要花费成倍的时间和精力。不断查阅资料，整理代码，“重学前端”这个专栏对我自己来说，也是一个自我重塑的过程。

在这个专栏即将结束的时刻，我想借“重学前端”这个课程谈谈我对教育的理解。

教育是知识的展现形式

现在有一个特别火爆的概念叫做知识付费，其实我不太喜欢这个说法。

“重学前端”所讲的知识点，没有一条是我发明或原创的知识，这么说的原因是：知识就在那里，每个人都可以获得它。

你可以在各种文档和标准中找到它们或者它们的变体。有一些工程领域相关的知识，来自我工作中的实践，有一些也算是首创，但是我不认为这些知识属于我，我只是发现了它们。

所以我认为，知识是免费的，承载它们的教育产品才是收费的。

在整个课程写作中，我致力于在知识之间建立关系。我不希望把我的课程变成抄写文档。我从排版追溯到活字印刷，我从渲染追溯到中国画的技法，还有面向对象的起源。在这个过程中，我自己也颇有收获。我期望这样的一种组织形式能够帮助你更好地理解和记忆知识。

我有时候会觉得，这个把“知识变成课程”的过程，特别像前端工程师把数据变成可见的页面的过程，所以，从某种意义上讲，我做的事情还是前端。

教育的重点在于能力提升而不是知识积累

不过，我也不希望“重学前端”这个课程仅仅是“更容易记忆的知识”。因为在我看来，知识量并非衡量一个工程师优秀与否的因素，能力才是。

我的一点小小的奢望是：“重学前端”这个课程，能给你带来一些建立知识体系的能力。

我在“重学前端”的课程中，穿插了大量的“用代码读标准”的实验。这个小系列，算是我的一种“独门武功”。

我刚开始关注前端的 2007 年左右，国内还少有程序员知道 Web 标准这件事，想要找到资料就更难了，我也经历了最初看标准如同天书的过程。

但是我发现，其实标准中佶屈聱牙、晦涩难懂的那些概念，有着内在的联系，又因为标准文本中用词非常严谨，所以尝试用代码去理解，反而是个快捷的路径，还能找出不少边边角角的冷知识。

通过类似这样的线索，我建立了自己最初的知识体系，我开始对前端的体系的全貌有了粗浅的认知，现在，我希望通过“重学前端”课程，让你也来体会一下这个过程。

教育是一种服务，我们做得显然还不够好

作为老师，我希望这个课程能够真正帮助到你们每一个人，但是其实我们可以看到，每一节课都有一些放弃的同学。

可能有人会说，对平台方和老师来说并无所谓呀，反正付过钱了。但是在我和“极客时间”看来，这正是我们做得不够好的地方。今天“重学前端”这个课程，可能以这样的产品形态呈现出来，凭学员的自觉性来完成学习，实属一种无奈。

不论对我，还是对“极客时间”而言，培养更多优秀的工程师，潜在的价格远远高于卖一些课程得到收入。

我认为，教育是一种服务，理想的教育产品卖的应该是服务的结果，而不是服务的过程。

在未来，我会跟极客邦一起，探索一个全新的教育形态。我们很快会推出第一期线下的课程。如果你具备基本的前端技能，渴望通过自己的努力和我的指导，快速成长为一名高级前端工程师，欢迎你关注我在极客大学线下开设的前端训练营。

另外，虽然正稿已经结束，“重学前端”将来也不会变成一个“过去式”的专栏，我想把它变成一个“Ask me anything”的栏目，不论是后来购买的同学还是已经学完的同学，你们可以来这里的评论区与我交流，不限于课程内容，只要是技术和职业相关的问题，都可以来问我，我会回复或者更新答疑文章来为你解答。

专栏的课程到此告一段落，但是对我而言，这是一个起点。长风破浪会有时，直挂云帆济沧海，希望未来能与你一起遇见更好的自己。



winter 程劭非
前手机淘宝前端负责人



不知道在学习过程中，你有哪些体会和评价？
这里有一份专栏调查问卷，邀请你填写。

**在5月30日前提交，
极客时间赠送给你专属优惠券。**

我们一起继续成长！

去提交

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。



bd2star