

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真的讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端？客户端？浏览器？手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：水平一般。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的构成，第二个是如何构成。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是动态语言，或者也没有讲什么是标识符。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好像它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你不懂，也是正常的。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x = x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程序。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认真地讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端? 客户端? 浏览器? 手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是**语言的构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好象它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你不懂，也是正常的。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“`x = x`”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程序。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：



“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真的讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端? 客户端? 浏览器? 手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵

啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的**构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好象它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你**不懂，也是正常的**。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命的原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x = x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认真地讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE？没有。
- HTML、CSS？没有。
- 前端？客户端？浏览器？手机App？……没有，统统没有。



那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。噢，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的**构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好象它不会解释什么是二进

制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你不懂，也是正常的。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“`x = x`”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真的讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端? 客户端? 浏览器? 手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的**构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好象它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你**不懂，也是正常的**。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或

者变化的。对于语言来说，它的生命的原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“`x = x`”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程序。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来个一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真地讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些



什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端？客户端？浏览器？手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：水平一般。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的**构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；

2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好象它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你不懂，也是正常的。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x = x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人的，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥

血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真的讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE？没有。
- HTML、CSS？没有。
- 前端？客户端？浏览器？手机App？……没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概会到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：水平一般。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，

又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的构成，第二个是如何构成。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是动态语言，或者也没有讲什么是标识符。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好像它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你不懂，也是正常的。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x = x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点的不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真地讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？



会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端? 客户端? 浏览器? 手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概会到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是**语言的构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为

ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好像它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你不懂，也是正常的。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论中表现得非常明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x = x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来

形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认真真地讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE？没有。
- HTML、CSS？没有。
- 前端？客户端？浏览器？手机App？……没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：水平一般。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年来的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的**构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好像它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你**不懂，也是正常的**。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论中表现得非常明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“`x = x`”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：噢，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程序。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认真地讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。



而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端？客户端？浏览器？手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概会是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年来的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年来的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

# 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的构成，第二个是如何构成。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是动态语言，或者也没有讲什么是标识符。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好象它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你不懂，也是正常的。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x = x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之

前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真的讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE？没有。
- HTML、CSS？没有。
- 前端？客户端？浏览器？手机App？……没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概会是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。噢，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是**语言的构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好像它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你不懂，也是正常的。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1进](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得非常明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。



然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x=x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真地讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟



不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端? 客户端? 浏览器? 手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概会到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的**构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好象它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你**不懂，也是正常的**。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命的原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x = x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真的讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE？没有。
- HTML、CSS？没有。
- 前端？客户端？浏览器？手机App？……没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概会到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是**语言的构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好象它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你**不懂，也是正常的**。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第](#)



1讲开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x = x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在第3讲里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。



你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真的讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端？客户端？浏览器？手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：水平一般。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的构成，第二个是如何构成。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是动态语言，或者也没有讲什么是标识符。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好像它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你不懂，也是正常的。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得非常明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x = x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程序。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认真地讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端? 客户端? 浏览器? 手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年来的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年来的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是**语言的构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好象它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你不懂，也是正常的。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区中表现得很明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“`x = x`”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程序。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：



“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。

你好，我是周爱民，今天又到了我的《JavaScript核心原理解析》课。

不过与以往不同，今天不上正经课，讲点课外的话。前两天极客时间的编辑老师找我，开了一个会，认认真真的讨论了一下这门课程，认为当务之急，是给同学们开个小灶，要好好讲一下“怎么学这门课程”。因为这才短短地讲了1/3，许多同学就已经跟不上了。

说到开小灶，我第一个就是想到了“吃”。

为什么呢？因为我父亲就是厨师，正正经经地考过厨师证，说不得多有名气，但一个县城里面，能跟他水平看齐的，也就三两个人而已。我的哥哥后来也随了父业，做了厨师，精通京、粤、淮扬多个菜系，更是拿了一级厨师的证书。

而我呢，没学厨，从父亲手上学到的唯一功夫，就是一个“吃”字。

## 什么是“会吃”？

会吃这件事，其实不是下嘴的功夫。这跟绝大多数人想的不同。下嘴就吃，就算有再多的法子，也无非是生啃细嚼；花样再多，也不过是甩动大腮帮子，劳力活而已。

真正会吃，说的是三件事，第一是食材，第二是味道，第三件，就单单一个“懂”字。

吃货世界里的“食材”，搁在我们今天的话题里，就是这二十来讲的课程。开课之前，有同学便过来打探，问我这个课程都讲些什么，看看值不值得一听。我就索性问了问他，你想听些什么呢？

- VUE? 没有。
- HTML、CSS? 没有。
- 前端? 客户端? 浏览器? 手机App? .....没有，统统没有。

那位同学不死心，又悻悻地问了一句：总该讲点Node.js、React Native，又或者是NPM之类的吧？

我索性给他摊了底牌：二十节课程里面，就只算是提到这些名词，大概也不超过五次。

那位同学回了一句：那你让我学什么？

## “食材”的问题

好问题啊！学什么呢？

我记得在豌豆荚的时候，有参与一些招聘的工作。老实说，作为架构师，招聘工作通常不是在第一轮，大概是到第二、三轮的时候，才会轮到我上场。也就是说，我需要面试的，大抵都是其他“面试官”认为“这家伙还有点料”的。

有一次，我的一个面试结束得比较快，在内部的HR系统中填写评论的时候，正经才写了四个字，便被其他面试官打断了，他嚷嚷着，几乎快要被整个公司都听到了，说：“爱民老师，人家十几年的一线开发，大型系统都做过那么多，经验很丰富的！你怎么这么点评呢？”

嗯，我在HR系统中写评论，起头里的四个字是：**水平一般**。

我想了想，停下写评论，给这个面试官聊聊开头这四个字。我说，我其实不太常用这种措辞来评论候选人，但这个也算是例外了。为什么呢？因为这个人真的是能力很强，下手很快，做的产品和项目也多，经验非常丰富，但是他确实就是水平一般。

什么叫“水平一般”呢？

因为他学的东西，别人也都学；会的东西，别人也都会；他强的东西，别人一样也强。即使别人今天不如他学得多、会得多，又或者不比他强多少，但是只要花点时间、下点功夫，也就一样儿也不会比他差。他十几年的的一线开发，把自己做成了熟手、熟练工，东西会得再多，技巧再熟练，也不过是卖油翁的“但手熟尔”。

我在面试里面，确实问了他几个偏向核心的问题，他也确实知道，很清晰、很准确。但再进一步问原理时，他却是一无所知。所以，我又引导、设问，说：“如果现在让你来考虑这个问题，你会从哪儿入手呢？”那位候选人想了好几个招数，中规中矩，然而无一可用。

这就是关键所在。

“核心原理”不是一些招数技法，不是拿来多练多用就行了的。所谓“核心”呢，不见得是大家都知道的——一眼望去，万千条路径之中，找到最正确的那一条，才是核心。但是这个东西可以教，也可以记，下次看见这个路，照着走就是了。所以，大公司里有所谓的“核心团队”，新人进去，不消半年工夫，功力就大增了，出来能带一个团队了。咦，带团队做什么呢？冲锋陷阵

啊，杀敌交人头啊！反正，和上面说的“吃”一样，还是劳力活。

所以，就算是在“核心团队”里，他们也只是帮你指出核心之所在，最多教你一些套路，让人净增功力。但是这样的核心只是“死东西”，不懂得核心的原理，就如同上面说过的那位面试候选人一样，出的都是些中规中矩的招数。

“中规中矩”有什么不好么？

确实，有些时候就是不好。“中规中矩”就是所谓的“一般”。所谓的“一般”不是指你能力一般，而是指你眼界一般、思想一般，对你面前这个事物的理解程度，也是一般。

所以，所谓“水平一般”，是因为学的是“一般”的东西，再怎么学，还是一般。

汉语中，“一般”这个词，是指跟别的一样，例如“一般无二”，而不仅仅是指“普通”，也并非贬义。

## 这门课到底讲什么呢？

《JavaScript核心原理解析》这门课，核心不是在讲JavaScript如何用，或者如何学习JavaScript。我在开这门课，列出这二十多行代码作为标题时，就说过，我要讲的是语言。

语言最核心的部分也有两个，第一个是语言的**构成**，第二个是**如何构成**。我在这门课里，主要还是讲“如何构成”的，因为ECMAScript就是以“如何构成（也就是如何按照ECMAScript规范来实现一个JavaScript）”为制订规范的主要目标和方向的。

在内容上面，这门课跳过了对“构成一门语言的那些基础元素（也就是语言的构成）”的讨论，而是直接进入到了“如何将基础元素组织起来，成为JavaScript”。比如，我就没讲什么是**动态语言**，或者也没有讲什么是**标识符**。这里有两点是很不幸的：

1. ECMAScript中没有明确地写“我是动态语言”；
2. ECMAScript里面明确地使用了“标识符”这个名词。

为什么“这两点”都是很不幸的呢？

因为无论有没有明确地使用这些名词，在ECMAScript中，这些概念都没有规范性质的定义。这是因为ECMAScript是直接面向“语言/引擎的编写者”，因此这个规范就默认它的读者是了解这些基础的或者学术的概念的。就好象它不会解释什么是二进制，什么是位运算一样，因为它默认是在计算机这个领域里的。

然而，我想你应该不是“计算机语言/编程语言学”的专家，因此那些所谓“构成一门语言的基础元素”，其实你**不懂，也是正常的**。在听这门课的大多数同学，其实也都一样的不懂。

然而，注意这里的这个词——“一样的”。这也就意味着，只有了解了那些“不一样的”“不懂的”东西，你才会成为“不一般的”。

而这门课程的目的，也正是要让你成为那个“不一般的”。

所以，你需要放下以前你认为你懂的那些东西。如同开篇词中所说的，它们正是阻碍了你前进的东西。比如说，有同学就从[第1讲](#)开始，就一直被“引用”这个概念困扰，因为他所理解的JavaScript的概念与我所讲的完全不同，而且混淆不清。这种状况在评论区内表现得非常明显，大多数同学都是在旧有的概念中构建新的概念集合，如同浮沙高塔，原本基础就搭得不对，你再怎么努力，也是上不了这个台阶的。

然而，又如同评论区里的“海绵薇薇”同学一样，只要突破了“引用”这个概念，把旧的东西扔掉，基于这个新的东西来理解，那么再看前几讲的东西，就豁然开朗了。

我所讲的东西，原本并不是什么特别高明的、高深的技巧。问题只是，你要把原本依赖的那些概念、想法、设定，或者你认为正确的那些逻辑一一扔掉，你才能“看到”这些新的东西。

这个“扔掉”的过程太难。很多年前，我在给一个Borland的纪念网页上留下的题字就是：

所见即所得，所见即所得。

Borland是“所见即所得”这一开发理念的大成者；而最终，他也是死于对这一概念的固执坚守。

关于“引用”这个概念的突破，我想如果有机会，我会请极客时间的编辑们约请“海绵薇薇”同学来讲一下他如何理解这个东西，以及在理解这个概念前后，对JavaScript有哪些不一样的认识。

回头来说这门课程。语言这个东西，其实是你最终要在这门课中看到的“真相的全部”。

我希望你通过这门课程，能真正地了解语言：语言是什么，它长得怎么样，它为什么长成这个样子，它如何成长、长大，又或者变化的。对于语言来说，它的生命的原点在哪里？源动力又在哪里？

举一例来说，在第一模块（也就是前五讲）中，如果你理解了“ $x = x$ ”表达的意思是“将值赋给引用”（当然前提是，你需要知道并接受“这里的引用和值，不是JavaScript中的，而是ECMAScript中的概念”）。那么，你就几乎能贯通整个JavaScript语言的构造过程，了解所有语句、词法，以及引擎内核结构的设计原理与应用原则。

所以，你现在再看看第一模块的总标题：“从零开始：JavaScript语言是如何构建起来的（5讲）”。

## 马钰曾经说过

说到这里，可能就会有一些同学听出不同的意思来了：咦，爱民老师好像是在说，他的课程是屠龙秘籍，所以不是一般人学的，或者是一般人学不会的。

当然，绝不是这样的。相反，我对屠龙术与杀鸡刀向来没有偏见，我只是在这里要强调一个东西：这个课程讲的东西，跟平常你听到的并不一样。

尺长寸短，但只要是用在合适的地方，就都是好东西。

相反的，你非得像公孙大娘舞剑一般，去百万军中杀敌，那么你大概就是最先中了黑箭的那一个。关于这一点不同，我在之前写过的一篇名为[《前端要给力之：代码可以有多烂？》](#)的博客文章里面也讲过，那篇文章从一个简单题目一直谈到了模板范型，最后呢，我还是建议大家用最保守的版本来编程。

为什么？

研究是研究，实战是实战，不要把二者混为一谈。就好比在[第3讲](#)里说到过的那个面试题，最终我给出了一个应用场景：用来形成链表。然而，如果你真的要在工程项目中这样去写链表，那么大概第二天就被开除了。将这个代码写入jQuery的那个提交者有没有被开除，我可不知道，反正我见到这样的提交者，是要打板子的。

说回正题。我仅仅告诉你这个东西“与众不同”是不够的。毕竟，你听这堂课的目的，还是要学懂它，对吧。

然而你可知道，历史上有一个人的，正好是怎么学都学不会东西的。他有七个师父，个个是江湖知名的大侠，个个也都呕心沥血，然而却教出来一个弱鸡，还没行走江湖，初遇敌手就被人抓住了脚给扔出去了。这个历史人物来自金庸大师的《射雕英雄传》，就是郭靖。你看，我在这里给你说这个故事，可不是闲扯，因为书中有人说了一句话，道出了这个郭靖“学不会”的实质。书中第32回写道，那道人问道：

“你这六位师父都是武林中顶尖儿的人物，……你又不是不用功，为什么十年来进益不多，你可知道什么原因？”

郭靖回答说：

“那是因为弟子太笨，师父们再用心教也教不会。”

这时候那位道人就笑了，说了一句古今以来，求学问道最核心的要义，这句话，原话就是：

“那未必尽然，这是教而不明其法，学而不得其道。”

所以，关于我们今天说的这一门课程，“学不好、或者学不会”，其关键就在这位道人——马钰（不是马云）——说过的这句话：教者要有其法，而学者要得其道。