

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做 C/C++ 开发的，到了 2009 年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗

憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 **Vue**、**React**、**Node** 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 **Cookie** 共享的过程中，我发现有一个例外，那就是设置了 **HttpOnly** 属性的 **Cookie** 是无法被共享的。

为了解决这个问题，我就去查 **HttpOnly** 相关概念，发现这和 **Web** 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、**Web** 网络、事件循环机制、**JavaScript** 核心、**V8** 的内存管理、浏览器的渲染流程、**Web** 安全、**CSS**、**React**、**Vue**、**Node**、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 **Vue**、**React**、**Node** 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 **Cookie** 共享的过程中，我发现有一个例外，那就是设置了 **HttpOnly** 属性的 **Cookie** 是无法被共享的。

为了解决这个问题，我就去查 **HttpOnly** 相关概念，发现这和 **Web** 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、**Web** 网络、事件循环机制、**JavaScript** 核心、**V8** 的内存管理、浏览器的渲染流程、**Web** 安全、**CSS**、**React**、**Vue**、**Node**、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 **Vue**、**React**、**Node** 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的是，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的是，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来

解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关

的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器

的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做 C/C++ 开发的，到了 2009 年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 **Vue**、**React**、**Node** 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 **Cookie** 共享的过程中，我发现有一个例外，那就是设置了 **HttpOnly** 属性的 **Cookie** 是无法被共享的。

为了解决这个问题，我就去查 **HttpOnly** 相关概念，发现这和 **Web** 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、**Web** 网络、事件循环机制、**JavaScript** 核心、**V8** 的内存管理、浏览器的渲染流程、**Web** 安全、**CSS**、**React**、**Vue**、**Node**、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 **Vue**、**React**、**Node** 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶

函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。

你好，我是李兵。

这是我们这个系列课程的最后一讲，今天我们不谈技术，我想跟你聊聊这么多年我学习前端踩过的一些坑和学习心得。

其实我早些年主要是做C/C++开发的，到了2009年，接触到了浏览器研发，这个时候我对前端并不是太熟悉，但是在开发浏览器的过程中，需要了解大量的前端知识。我当时的学习策略是，只有当遇到了与前端有关的问题时，才会去学习相应的知识来解决。这样做本意是想要节省时间，结果却反而让我花费了更多不必要的时间。

比如在开发双核浏览器的 Cookie 共享的过程中，我发现有一个例外，那就是设置了 HttpOnly 属性的 Cookie 是无法被共享的。

为了解决这个问题，我就去查 HttpOnly 相关概念，发现这和 Web 安全体系有关系，然后我就去补安全相关的知识。之后我又发现，浏览器多进程架构和安全系统息息相关，也就是说，这些知识是环环相扣的。

因为我当时对前端并没有做整体的了解，所以对浏览器中涉及到的安全相关的设计也没有过多关注。每次遇到一些和前端相关的细节问题时，我都需要花费很多时间去补充相关知识，然后再去解决问题，这种方式白白地浪费了我非常多的时间。更加遗憾的是，我之前积累的很多前端知识，都是这么缝缝补补学习过来的。

所以，当你要开发一个新项目或者学习一门手艺之前，应该将其所涉及到的知识做一个全方位的了解。我就觉得发明“技术栈”这个词的人是一个天才，“技术栈”非常形象地表达了学习一门手艺所需要的是哪些知识，以及应该按照什么顺序来学。

比如学习前端这门手艺，栈底到栈顶依次是浏览器架构、Web 网络、事件循环机制、JavaScript 核心、V8 的内存管理、浏览器的渲染流程、Web 安全、CSS、React、Vue、Node、构建工具链等，我们可以从栈底往栈顶一步步循序渐进地学习。

学习技术栈的每一层都不难，难的是当你抛开底层栈的内容去理解上层栈的内容，那么就有可能陷入知识点的盲区。

其实，我们在学习一门手艺的时候，即便没有完善的技术栈资料，只要肯花更多的时间，也可以克服困难。甚至我们的学习过程可以像一个无头苍蝇，横冲直撞，某一天也会突然顿悟整个知识体系，俗话说，在一个领域耕耘十年，必定能成为该领域的专家。

但是，三年可以搞定的事，为什么非要花上十年呢？

系统性学习一门技术，花费的时间也是最短的，也可以说是性价比最高的，因为系统性地、循序渐进地学习，那么学习到每个知识点时，其实并没有其他的知识盲区，这样学习起来是最轻松、简单的。

虽然技术栈的概念已经提出来很久了，但是我发现大家对技术栈的理解却不同。比如我们在招聘前端工程师时，经常会看到有人在简历上提到自己的技术栈包含了 Vue、React、Node 等这些偏应用的知识，在我看来，这并不是技术栈，这只不过是栈顶函数的上下文而已，而对于那些在技术栈底层上下文相关的知识，在简历中却很少有提及。所谓技术栈，应该是在某一领域，从底层的基础知识到上层的应用技术有一个完整体系。

那么当我们进入一个领域时，应该如何建立适合自己的技术栈呢？

如果你进入的是一个成熟的领域，那么一般都有比较完整的技术栈的资料，我们需要花些时间分析资料，系统性地了解这一领域知识的宏观架构、它的过往历史、它的优缺点，然后结合现有资料和我们自身的特点来建立我们自己的技术栈。如果你所在

的领域还在高速发展中，并没有人总结出完整的技术栈，那么为了更好地理解技术的发展脉络，我们需要花更多一些的时间去整理出该领域的技术栈。

所以后面我就吸取了经验，比如我学习神经网络的过程中，就采取了先构建技术栈的方式，先系统性复习了微积分、概率论、线性代数，然后结合实际项目完成了数据清洗、构建模型、模型训练，最后到实际应用。整个过程我只花了半年时间，最后我对神经网络也有了非常深的认知，现在我们正在打算将神经网络运用到我们的新项目中。

采用了同样的方法，我在学习区块链和数字货币时，整个过程也是非常轻松的，先分析区块链要解决什么问题，然后整理技术栈，再通过技术栈来一步步学习，不到两周，我基本就把整个数字货币的来龙去脉分析的非常清楚了。

好了，到了这里我们的课程也要告一段落了。最后我想跟你说的，学习一门手艺其实并不难，难的是如何充分利用时间来高效学习，因为这世界最值钱的就是我们的专注力和时间，花三年干成一件事和花十年干成一件事的代价是不一样的，希望你都能够找到适合自己的方法来高效学习，也都能学有所成。

另外，如果你有时间，也请帮我填个[课程问卷](#)，我和编辑大茹想收集更多的反馈来优化课程。今天虽然是结课，但我们的工作并没有结束，未来我还会回来加餐、回复留言。

再见。