行业资深大牛，在实际面试百八十号的程序员工作中，深度总结什么样子的人，比较容易拿到offer！绝对值得一看哦

## 1. 题要做满

一般来说，程序员面试都会先由HR给一张面试题。面试题一般包含：逻辑题、代码规范、数据库理论、简单算法、高级函数应用、服务器基础。大多数公司出的题目，百度上都有。如果面试的岗位是初级开发，大胆去百度吧，只要填满就能获得好感度。如果面试的是中级开发，这些题只需要写重点就行，对做题时间会有要求。如果面试的是高级开发，在中级开发的要求上，把字写得漂亮些。

## 2. 面试问答

第一面，基本是技术主管/经理。可能是因为营造优越感，一定会问高级问题，即使需要的岗位是初级岗位。

提问的第一波是面试题上的细节，确定是百度的还是真的懂的。第二波，会随机提问一些技术问题。

PHP面试最喜欢问的就是常用框架和框架之间的差别。

Java面试最喜欢问关于函数构建、算法加密等问题。

前端面试，基本就拿着其他公司的界面设计问多久可以实现。

会涉及一些数据库，基本都是问增改删查的应用、数据库设计的规范。其他的可能会随机选取一些业务场景，让当场做程序设计。

整体的提问过程，能回答上一半就可以进入下一个面试节点。回答的时候态度一定要好，不知道的就回答“以前工作中没怎么用到”，千万不能胡说八道。知道的，也谦虚一点，回答“我过往是如此运用的”。

## 3. 面试沟通

可能是第二面，也可能是第三面，会有一个看上去和蔼可亲的人来面试。这个人基本上是某个项目负责人，面试的内容是沟通能力。

一开始也会带一点基础技术问题，基本上还是根据面试题扩展的。

然后就会开始问是否可以加班、如果需求发生变更怎么处理、对过往的工作从业务上理解多少、响应Bug会具体怎么做。回答不能啰嗦，要肯定句，比如：可以加班。而不要，模棱两可或者带有场景的回答，比如：不排斥加班，但是无意义的加班不太能接受。

## 4. 面试情怀

一般部门总监会谈这个问题。这部分，每个人都不同，但是未来规划一定要明确。一般程序员就两个发展：管理和技术专家。一般问这个问题，就是看短期内会不会离职，是不是一个有目标的人。

还会问一些兴趣爱好，看未来能不能融入团队氛围。

接下来就问薪资和待遇要求了。这部分，请根据自己的心走。不要觉得不好意思开口，面试是双向选择的过程。如果面试感觉比较好，当然也可以适当的降低一点要求。切记不要说出让自己下不了台的薪资。

## 说几个有意思的对话，在以往面试中遇见过的

注意啊，都是面试失败案例，千万别重蹈覆辙

**程序员李某面试：**

* Q：以前的工作是否用过敏捷开发？
* A：核心就是以人为本。开发设计功能跟着心走。

结果：回答完，面试就结束了。

**程序员王某面试：**

* Q：你当初做保险行业的，怎么处理用户名、密码、银行卡入库加密这块？
* A：为什么要加密？明文存就好了啊。

结果：认为明文存是理所当然不假思索的，Pass。

**程序员马某面试：**

* Q：PHP的常用框架了解的有哪些？
* A：ThinkPHP和Yii。
* Q：他们的区别是什么？或者说优缺点？
* A：（沉默了3分钟）其实我没有接触过Yii。

结果：后面的面试也基本是这种一知半解的节奏，Pass。

**程序员刘某面试：**

* Q：如果产品经理安排了一个任务给你，然后当天就需求变更了，你怎么办？
* A：这种事情绝对不允许发生第二次。

结果：应该会很难融入团队吧，Pass。

## 程序员面试中的5个杀手锏问题

也许你是个JavaScript巨星，为了防止被那些烦人的猎头骚扰，不得不删除你在LinkedIn上的个人资料。又或者，也许你是一个普通、可靠的合作伙伴，一年到头也只会收到2到3次的面试邀请。

不管你去面试的频率如何，下面这五个问题是每个软件工程师都应该问的——将有助于你确定自己在这家公司长期工作是否会合作愉快。

lego-job-interview-software-engineer

lego-job-interview-software-engineer

### 你们的企业文化是什么？

你每天将会有10至12个小时需要与同事的信仰、价值观和行为打交道。企业文化重视技术吗？尊重软件工程师吗？软件工程师在产品开发上有发言权吗？企业有没有提供便利以便于软件工程师将工作做到最好？

为了找到答案，可以问问企业从开发到测试都喜欢什么工具，Luca Bonmassar，Gild公司的联合创始人和首席技术官建议说（Gild是一个用于查找评估和招聘技术人才的SaaS平台）。如果面试官不能回答，Bonmassar说，“这通常是一个坏兆头”，说明该公司对你重视的技术并没有给予足够的重视。

他还建议询问开发流程：“开发人员的投入有多少会进入到产品？项目经理是否决定了进度的每一个细节？需要构建什么，或者工程团队有没有发言权，有多少发言权？“

company-culture

company-culture

询问工程和其他团队之间的关系。Doug Schade，WinterWyman公司软件技术搜索部门的合作伙伴和招聘人员，建议问“在应对项目时，你们公司会给开发人员什么级别的自主性？” Bonmassar说，对软件工程师的反馈缺乏任何机制是一个“危险信号”。

### 如何衡量我？

你的雇主如何定义你的“成功”与给你的工资和津贴等各种福利息息相关。但是，不同公司的评判标准不同，要满足你觉得不舒服的目标会让你的生活苦不堪言。

hengliang

hengliang

有些公司衡量软件工程师看的是他们的努力，比如他们工作了多少小时，提交了多少代码，Ari Weil，Yottaa公司的产品副总裁说（Yottaa是一家自适应的内容分发网络提供商）。也有的用结果来评估软件工程师，如因缺陷而需要召回的代码数量，或在规定时间和预算范围内，小组完成的项目数量。

### 有什么成长计划？

TonyaShtarkman，Riviera Partners的首席技术招聘人员说（Riviera Partners是一家总部位于旧金山的猎头公司），很多软件工程师觉得“他们在当前公司已经不可能有多大发展了。“她建议软件工程师在面试时要询问是否有一个针对软件工程师的成长计划——允许他们继续晋升，并且有机会让他们参加会议和研讨会来建立新的产品和功能，并受到辅导。

许多软件工程师希望雇主会告知他们最新、最好的技术工具，使他们能够保与时俱进。但Bonmassar警告说，“它通常是一个不好的兆头”，当公司坚持某个极其特殊的技能，并要求能迅速改变的时候，可能要不了多久该公司就会开始找人来代替你。如果说需要更匹配的长期合作，他说，那么可能这家公司现在需要的是“聪明，但不必知道工具和技术每一个细节的人”。

他还建议询问一下，多少外部聘请vs公司内部晋升。这答案能说明很多关于随着企业发展你的成长之路会怎么样的趋势。

### 你们的发展计划？

如果你正在考虑去创业公司工作，那么你需要了解他们的发展计划：“加入创业公司，总是涉及着一定程度的风险水平，然而创业公司的工程师往往比大企业的工程师不怕风险， “Shtarkman说。 “不过，将风险控制在一定的稳定范围内是必需的。”

第一个步骤是调查。Shtarkman建议可以问这样的问题，如“你们的资金消耗率（公司的负现金流）是多少？” ，以便于了解公司在没有其他资金和不盈利的情况下能维持多久。Jim Barnett，Glint公司的首席执行官（Glint是一个用于跟踪可以影响保留趋势的网络平台），建议在签署保密协议前可得仔细看清楚。

### 我会喜欢你们的人吗？

聊到目前的团队成员，“我碰到过一些工程师之所以接受创业公司的offer，纯粹是因为他们与团队融合得非常好——有时候甚至是因为某个人的魅力，”Shtarkman说。 “说来说去，公司是由人组成的，如果你不能与你的队友和睦共处，那么当作长期的职业生涯几乎是不可能的。”

试着和公司的内部人士聊天，以便于知道“公司内部管理人员大致的情形，”Barnett说。 “他们好合作吗，他们做事征求意见吗，他们提供反馈吗，他或她投资团队成员并帮助他们成长吗？”

试着和团队中你共事的人进行非正式的交谈。问问他们工作中最让他们沮丧的是什么。比起面试官，他们更可能现实地回答你，Shtarkman说。

底线：挖掘得更深一点以了解今后你每天需要共事的人，和你每天要经历的工作流程，而不要只关注薪水。

## 程序员面试技巧总结

**闲聊**

在深入代码之前，大多数面试官喜欢聊聊你的背景。

img

img

**他们想知道：**

你对编码认知。你是否知道如何编写好代码？

个人能力/领导力。你是否经历过整个工作流程？你是否修复过并不怎么正确的东西，即使你并不需要这么去做？

沟通。和你交流技术问题是有用的还是痛苦的？

**你应该至少说明以下中的一个：**

你曾解决的一个有趣的技术问题

你曾克服的一个人际冲突

显示领导力或个人能力的例子

你曾在以往项目中做出的贡献

最喜欢的语言的一些琐事，对这种语言你做了什么，以及你不喜欢它哪里

有关公司产品/业务的问题

关于该公司的工程策略（测试，Scrum，等等）

热爱技术。表达你对你所做的一切感到骄傲，你对自己的选择充满自信，你对语言和工作流有着自己的看法。

**沟通**

涉及到编码问题的时候，沟通是关键。一个在工作时需要帮助却能和人正确沟通的求职者比那些能轻松解决问题的求职者甚至更好。

**了解这是哪种问题。**

**有两种类型的问题：**

**编码。**面试官希望你能针对问题写出简洁高效的代码。

**闲聊。**面试官希望能和你聊一聊。话题通常是（1）高水平的系统设计（“如何克隆Twitter？”）或（2）琐事（“Javascript中的hoisting是什么意思？”）。有时候这些琐事中也会引入“实际”问题，例如，“如何迅速排序整数列？好的，如果不是整数，是其他类型的呢…… ”。

如果你开始编写代码，并且面试官并不想多说废话，只想尽快过渡到“实际”问题，那么如果你罗哩叭嗦太多的话，她可能会觉得厌烦。不妨直接问，“是不是为这个问题写代码？”

**让人感觉你有团队精神。**面试官想知道和你一起工作是什么感觉，会有什么问题，所以要让他们看到你的团队合作性。使用“我们”来代替“我”，例如，“如果那个时候我们做广度优先搜索的话，就能及时/准时得到解决方案。”如果让你选择在纸上还是在白板上编码的话，选白板。这样，你就可以接近面试官，直接面对他提出的问题（而不是和她在桌子两边遥遥相望）。

**把自己的想法大声说出来。**不是开玩笑，比如说：“我不知道这样做是否有效——但请让我试一试。”如果你不知道怎么办，不知道这个问题该如何解决，那么就说一说你现在的想法。说一说你认为怎么做可能会有效。说一说你认为哪些会有用，以及为什么没用的原因。这同样适用于琐碎的闲聊问题。当面试官要求你解释Javascript闭包的时候，“这与范围有关，不妨把它放到一个函数中”可能会让你得到90％的分数。

**不知为不知。**如果正在谈论的话题（例如，具体的语言事务，具体的琐事，运行时分析）的确是你不曾涉猎的内容，那么不要不懂装懂。相反，你可以直接说：“我不知道，但我猜$thing，因为……”，因为后面可以通过分析排除其他选项，还可以拿其他语言或问题做例子。

**说话不要不经大脑。**不要自信地将答案脱口而出。如果是正确的，那么你还是需要时间来考虑如何解释，如果是错的，那会显得你冲动鲁莽。你不是在和人比速度，而且你这么做更有可能因为打断她的话或者妄下结论而惹恼她。

img

img

**摆脱困境**

**有时候你会陷入僵局。**放松。这并不意味着你已经失败了。请记住，面试官通常更在乎的，是你能否巧妙地从几个不同的角度去揭示问题，而不是一根筋走到底地坚持正确答案。

**画图。**不要浪费时间在脑袋里思考，可以画到板上。画出几个不同的测试输入。画出你如何手动如愿得到所需的输出。然后想想将你的方法转换成代码。

**解决问题的简单版本。**不知道如何找到集合中的第4大条目？那么想想如何找到第1大条目，然后试试能否沿用这种方法。

写一个简洁低效的解决方案，然后对其进行优化。竭尽全力。尽一切可能的方法得到某种答案。

**讲讲自己的思路。**讲一讲你知道什么。讲一讲你认为什么可能工作以及为什么无效的原因。你可能突然会意识到它实际上是可以工作的，或修改版本是有效的。也有可能，你会得到提示。

**等待提示。**不要用期待的眼光盯着面试官，但可以有短暂的“思考”时间——面试官或许已经决定给你个提示也说不定呢，等待她的提示以免打断她。

**考虑空间和运行时的界限。**如果你不知道你是否可以优化解决方案，那么就说出来。

**例如：**

“我必须至少看看所有的条目，我做不到时间复杂度比O(n)还好的了。”

“蛮力方法才能检验所有的可能性。”

“答案将包含n2数据项，所以我必须至少花费N2的时间。”

**写下你的思路想法**

**凭空地想很容易自我矛盾。**把你的想法写下来，然后再去考虑细节。

**调用帮助函数，继续前进。**如果你不能或多或少地马上想出如何实现算法，那就跳过它。写一个命名合理的调用函数，例如：“this will do X”，然后继续下一步骤。如果帮助函数非常微不足道，你甚至可以将它忽略。

**不要担心语法。**不妨一笑而过。如果你非要考虑语法，那就还原到英语。只要向面试官说明稍后会回来整理即可。

**预备足够的空间。**你可能后面会想要在代码行之间添加代码或笔记。从白板的顶部开始写，并在每一行之间留一条空白。

**最后写一个重头检查的标志。**不要担心你写的for循环是否应该有“<”或“<=”。在代码的最后画个勾选提醒自己最后再检查一遍。先按自己的思路走。

**使用描述性的变量名。**想名字需要时间，但可以防止你忘记自己写某段代码的目的。使用names\_to\_phone\_nums\_map而不是nums。在名称中说明类型。返回布尔值的函数应该以“is\_ \*”，保存列表的Vars应该以“s”结尾。标准化很有意义。

img

img

**完成之后的整理**

**浏览解决方案，大声地讲，输入一个例子。**当程序运行时记录下变量保存的值——如果你只是记在脑子里，不会让你赢得任何加分。这有助于你发现bug和消除面试官的困惑。

**寻找差一错误。**你的for循环是不是应该使用“<=”来代替“<”？

**测试边缘情况。**措施包括空集合，单项目集合或负数。加分点：提一提单元测试！

**不要惹人厌烦。**有的面试官可能并不在意这些整理步骤。如果你不确定，可以这样说，“我通常会检测一些边缘情况——那么我们接下来是不是做这个呢？“

**实践**

最后，运行实践问题是没有捷径的。

**好记性不如烂笔头。**对自己诚实。用笔写可能一开始会让你觉得别扭。但是如果你现在就能克服这个难题，那么当面试的时候，你就不会觉得笨拙和不顺手了。

本文中的实践问题只是提供了每个面试过程的线索要点，没有真正的金科玉律，在真正面试时还需实际问题实际解决。**最后，祝大家面试成功。**