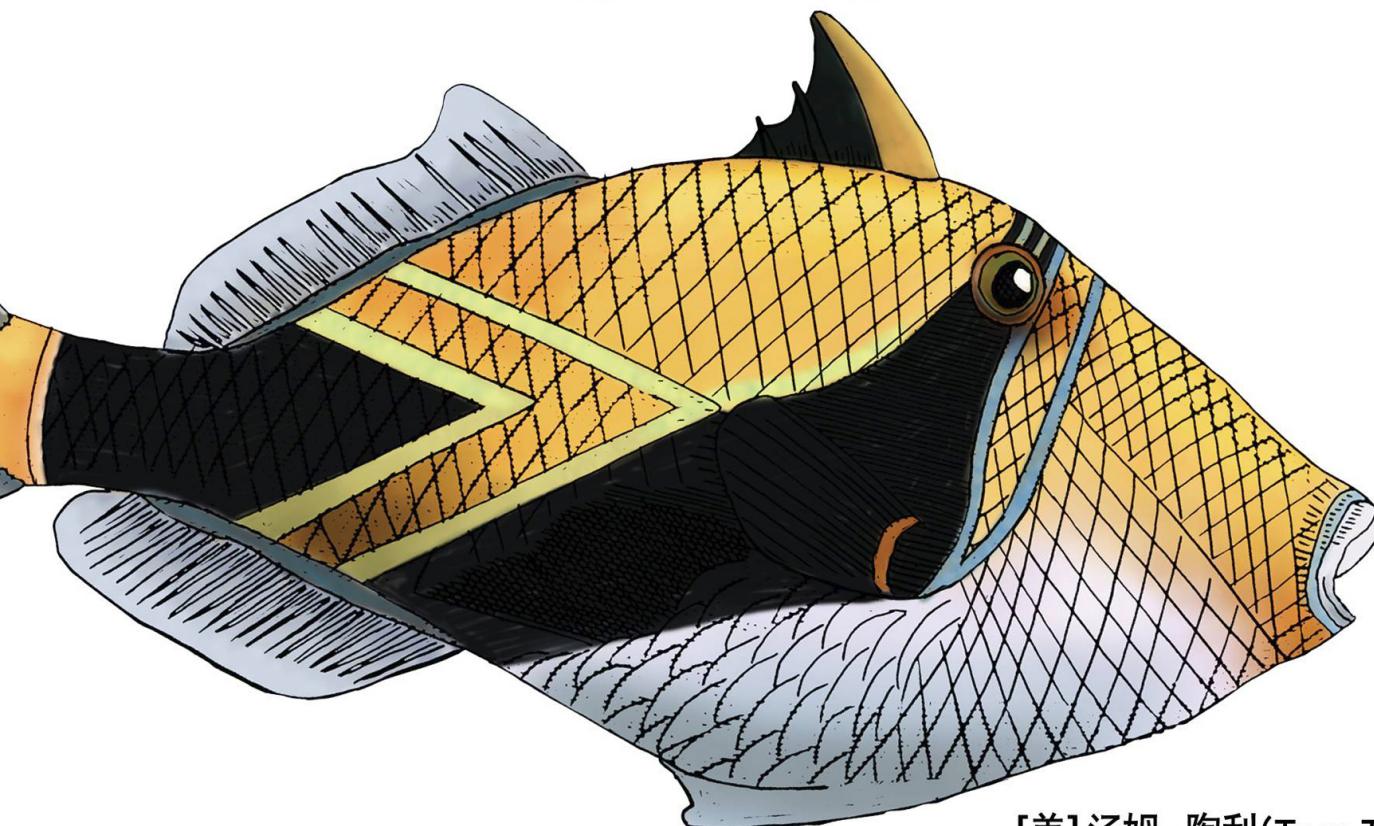


O'REILLY®

异步图书  
www.epubit.com

# AI辅助 编程实战

AI-Assisted Programming



[美] 汤姆·陶利(Tom Taulli) 著  
成海霞 译  
刘轶煜 审校



中国工信出版集团



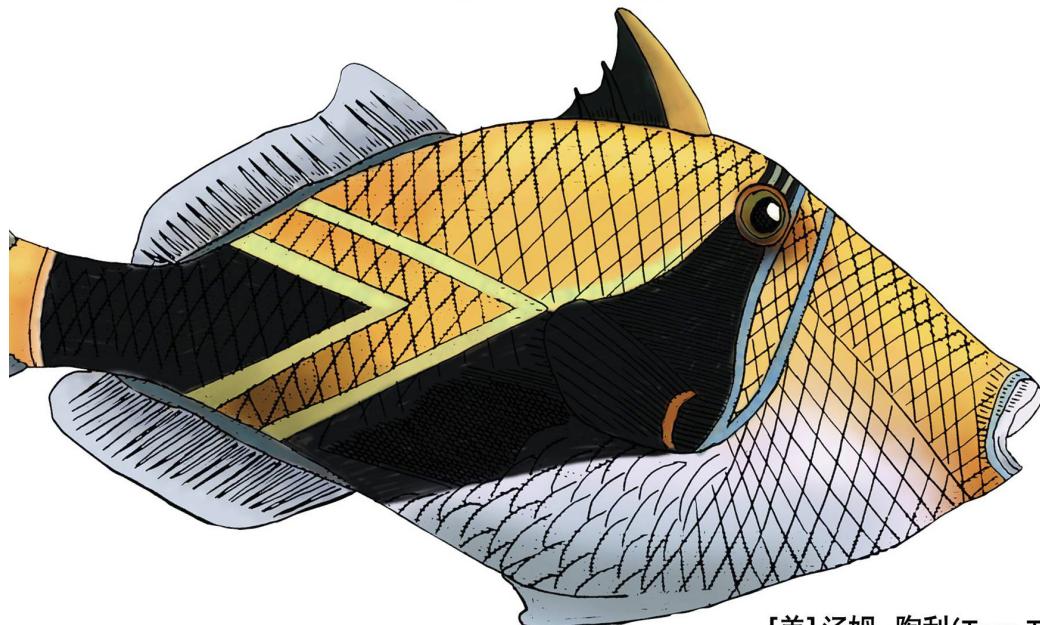
人民邮电出版社  
POSTS & TELECOM PRESS

O'REILLY®

异步图书  
www.epubit.com

# AI辅助 编程实战

AI-Assisted Programming



[美] 汤姆·陶利(Tom Taulli) 著

成海霞 译

刘轶煜 审校



中国工信出版集团



人民邮电出版社

POSTS & TELECOM PRESS

## 对本书的赞誉

《AI辅助编程实战》是一本专为编程领域设计的实用指南，它全面整合了前沿的AI技术，并将其应用于多种软件开发场景，从而显著提升开发效率。从自动化测试到代码智能补全，这本书的每一章都深入阐释了高效编程的理念。通过独特的视角，这本书重新审视了软件开发生命周期，并深入探讨了如何有效地将AI技术整合到开发流程中。它不仅拓展了读者对AI在编程领域应用的理解，还提供了一套全新的方法论来提升开发效率。对于软件开发者，这本书是一份宝贵的参考与指导资源，是他们不可或缺的实战宝典。

——吴晨

微软AI事业部首席应用科学家

在编程的广阔海洋中，AI既是指引方向的灯塔，也是推动船只前行的帆。《AI辅助编程实战》一书以独特的视角，将AI技术与编程实践紧密结合，不仅传授编程技巧，更教授读者如何运用AI技术实现高效编程。这正体现了“授之以渔”的深层含义。这本书不仅适合编程新手，也适合那些希望提升编程效率的资深开发者。通过阅读这本书，你将学会如何让AI成为你的编程助手，共同开启编程世界的无限可能。

——黄正行

浙江大学计算机科学与技术学院教授

随着大语言模型的兴起，越来越多的开发者开始利用AI辅助编程。这本书深入浅出地覆盖了AI辅助编程的多个方面。无论是广受欢迎的GitHub Copilot，还是其他主流AI辅助编程工具，读者都可以在这本书中找到相关的学习资源。精通这些工具，开发者将能够显著提升开发效率。

——韩骏

VS Code中文社区创始人，《Visual Studio Code权威指南》作者

当我们在6周内为Cassandra引入向量搜索功能时，使用GitHub Copilot和ChatGPT是我们在截止日期前完成任务的关键。然而，许多开发者尚未充分发掘AI开发工具的潜力。Tom的书作为一本实用的入门指南，将助你一臂之力，为你节省试错时间。

——Jonathan Ellis

DataStax联合创始人兼首席技术官

这本书是一部展示Tom的专业技能的优秀作品。它不仅为读者揭示了当今软件开发领域的新进展，而且激发了每个人编程的潜能。

——Justin Dorfman

Sourcegraph公司开源社区经理

AI正在迅速改变开发者构建软件的方式。从代码编辑器到终端等，AI辅助编程工具正变得无处不在。对于任何试图在日常开发工作中最大限度地利用AI的开发者，这本书无疑是推荐读物。

——Zach Lloyd

Warp首席执行官兼联合创始人

Tom Taulli的这本书是一次精心构建的探索之旅，展示了像ChatGPT这样的AI工具是如何改变开发者（无论是初出茅庐的新手还是经验丰富的专家）的工作方式的。书中分享了AI驱动软件开发各个阶段的实用经验和独特见解。这本书非常适合希望将AI纳入其工具箱的开发者阅读。

——Titus Capilnean

Private Market Labs联合创始人

AI正在重塑软件开发领域的游戏规则。掌握如何利用像 GitHub Copilot这样的工具，对每位开发者而言都至关重要。Tom 在这本书中细致地介绍了AI辅助编程工具的使用方法，为读者提供了宝贵的指导。

——Muddu Sudhakar

Aisera首席执行官兼联合创始人

## 内容提要

本书深入探讨了AI（人工智能）如何革新软件开发领域，从AI辅助编程的演变、优势与挑战到具体的工具与技术实现，为开发者打开了一个全新的世界。全书共10章，第1章介绍了编程的历史演变和生成式AI的崛起；第2章深入技术细节，解释了AI辅助编程工具的主要功能和工作原理；第3章讨论了提示工程的重要性，以及如何与AI工具进行有效沟通；第4章和第5章通过GitHub Copilot和其他AI编程工具的案例，展示了AI在实际开发中的应用；第6章扩展到通用大语言模型，如ChatGPT和GPT-4，探讨了它们在日常任务处理中的应用；第7章和第8章覆盖了软件开发的早期阶段和代码编写过程，展示了AI如何辅助决策、提高学习效率和代码质量；第9章讨论了调试、测试和部署过程中AI的作用；第10章总结了AI辅助编程的一些启示，包括学习曲线、优势、不足及对未来工作的影响。

本书适合软件开发者、编程爱好者、技术项目经理、高等院校计算机科学专业的学生及对AI在编程领域的应用感兴趣的专业人士阅读。

# O'Reilly Media, Inc. 介绍

O'Reilly以“分享创新知识、改变世界”为己任。40多年来我们一直向企业、个人提供成功所必需之技能及思想，激励他们创新并做得更好。

O'Reilly业务的核心是独特的专家及创新者网络，众多专家及创新者通过我们分享知识。我们的在线学习(Online Learning)平台提供独家的直播培训、互动学习、认证体验、图书、视频等，使客户更容易获取业务成功所需的专业知识。几十年来O'Reilly图书一直被视为学习开创未来之技术的权威资料。我们所做的一切是为了帮助各领域的专业人士学习最佳实践，发现并塑造科技行业未来的新趋势。

我们的客户渴望做出推动世界前进的创新之举，我们希望能助他们一臂之力。

## 业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列非凡想法（真希望当初我也想到了）建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的领域，并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，那就走小路。’回顾过去，Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

# 资源与支持

## 资源获取

本书提供如下资源：

- 源代码及资源包；
- 本书思维导图；
- 异步社区7天VIP会员。

要获得以上资源，您可以扫描下方二维码，根据指引领取。



## 提交勘误信息

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区（<https://www.epubit.com>），按书名搜索，进入本书页面，单击“发表勘误”，输入勘误信息，单击“提交勘误”按钮即可（见下页图）。本书的作者和编辑会对您提交的勘误信息进行审核，确认并接受后，您将获赠异步社区的100积分。积分可用于在异步社区兑换优惠券、样书或奖品。

图书勘误

发表勘误

页码: 1 页内位置 (行数) : 1 勘误印次: 1

图书类型:  纸书  电子书

添加勘误图片 (最多可上传4张图片)

+ 提交勘误

## 与我们联系

我们的联系邮箱是contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们。

如果您所在的学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

## 关于异步社区和异步图书

**“异步社区”**是由人民邮电出版社创办的IT专业图书社区，于2015年8月上线运营，致力于优质内容的出版和分享，为读者提供高品质的学习内容，为作译者提供专业的出版服务，实现作者与读者在线交流互动，以及传统出版与数字出版的融合发展。

**“异步图书”**是异步社区策划出版的精品IT图书的品牌，依托于人民邮电出版社在计算机图书领域四十余年的发展与积淀。异步图书面向各行业信息技术用户。

## 推荐序1

在数字化进程加速的当下，编程已不再是一项孤立的技能，而是成为推动创新和技术进步的核心力量。《AI辅助编程实战》一书的出版，正逢其时，为我们提供了探索人工智能如何重塑编程领域的宝贵视角。通过作者Tom的精心编著和海霞的精准翻译，这本书不仅是一本技术指南，更是一份探索未来编程世界的导览。

AI辅助编程的兴起，无疑是近年来最令人振奋的技术变革之一。它不仅改变了我们编写代码的方式，更重要的是，它改变了我们思考和解决问题的方式。《AI辅助编程实战》一书以深入浅出的方式介绍了AI辅助编程的原理、优势、挑战，以及如何在实际开发中应用这些技术，为读者打开了通往未来编程世界的大门。

本书之所以独树一帜，不仅因为它全面覆盖了AI辅助编程的多个方面，从生成式AI的基础原理到具体工具的应用，从编程技术的深层机制到软件开发的初期规划，每一章都提供了丰富的信息和深刻的见解。更为关键的是，作者结合真实案例和个人经验，将抽象概念具体化，让读者能够直观地理解AI辅助编程在现实世界中的应用。

我特别推崇这本书对提示工程的深入探讨，这不仅因为它是AI辅助编程中的一项核心技术，更因为它展现了编程艺术和科学的完美融合。通过精心设计的提示，开发者能够引导AI生成更加准确和有效的代码，这无疑为软件开发带来了革命性的变革。

无论你是AI辅助编程的新手，还是希望深入探索这一领域的专家，《AI辅助编程实战》都将是你不可多得的良师益友。我相信，通过阅读这本书，你不仅能够获得宝贵的知识和技能，还能够激发对编程艺术的热情和创意。

在此，我要向译者表达深深的敬意和感谢，感谢她的辛勤工作和卓越贡献。正是她的努力，使众多中文读者得以接触并了解这项激动人心

的技术。同时，我也要向每一位选择阅读本书的读者致以诚挚的谢意，因为正是你们的好奇心和求知欲，推动了技术的发展和创新。愿这本书成为你们探索AI辅助编程世界的灯塔，照亮你们前行的道路。

黄寓凡

微软亚太研发中心开发者事业部数据与AI团队首席研发经理

## 推荐序2

在过去的几十年中，编程领域经历了翻天覆地的变革。从早期的低级编程语言发展到今天的高级编程语言，编程技术的进步极大地提升了软件开发的效率和便捷性。人工智能的融入为编程领域开启了全新的篇章。

《AI辅助编程实战》在这样的大背景下应运而生。这本书不仅是一本编程指南，更是一本向读者展示如何利用人工智能工具和技术，从项目构想到最终实现，全方位优化编程过程的指南。

在这本书中，作者深入浅出地介绍了生成式AI如何转变开发者的工作方式，对GitHub Copilot、CodeWhisperer、Duet AI等流行AI辅助编程工具进行了全面剖析，并探讨了如何根据专有代码库的需求对这些工具进行个性化定制。书中详细讲解了如何利用AI工具进行高效的错误修复、代码审查、单元测试编写等，以确保软件的质量与稳定性。无论你是编程新手还是经验丰富的专家，都能从这本书中获益。书中的实例和实践建议将帮助你在实际工作中迅速应用所学，提升编程效率，降低错误率。

特别值得一提的是，作者不仅关注技术的应用，还深入探讨了人工智能如何改变编程的思维方式。这种思维方式的转变，不仅能让开发者在技术层面上取得突破，更能在理念和方法上获得新的启发。

在此，我还要特别感谢这本书的译者。我与译者相识已久，深知其作风认真严谨。她在翻译过程中付出了巨大的努力和心血，不仅精准地翻译了书中的技术细节，还确保了语言的流畅性和可读性，使得更多的中文读者能够领略这本佳作的精髓。她的辛勤工作，让这本书在中文世界里得以传递其价值和理念。

总之，这本书是一本不可多得的佳作。它不仅能帮助你掌握最新的编程技术，更能引领你进入人工智能时代的编程新世界。我相信，这本

书将成为你编程生涯中的得力助手，让你在编程的道路上走得更远、更稳。

希望你能像我一样，从这本书中获得启发和力量。

王利

深圳大学大数据系统计算技术国家工程实验室副研究员

# 序言

20世纪90年代初，当我从加州大学洛杉矶分校毕业时，我曾憧憬未来能成为一名教授，投身于学术殿堂。然而，当IBM向我抛出橄榄枝时，我很快爱上了科技行业。自那时起，我便踏上了一条充满挑战与机遇的道路，从未回头。我先后在VMware、Pivotal、EMC及SGI等企业留下了坚实的足迹。

随后，我踏上了创业之路，成功创办了几家初创公司。其中，我最新的创业成果是Aisera，这是一家专注于为企业提供生成式AI（Generative Artificial Intelligence）解决方案的公司。我们的平台致力于在客户服务、员工体验、企业搜索、IT服务及AIOps等领域提供帮助。

在职业生涯中，我见证了许多创新，包括互联网的飞速发展、云计算的兴起、移动设备的普及，以及深度学习的突破。然而，尽管Aisera在众多领域取得了长足的进步，但在软件开发这一关键技术领域，其发展步伐却相对缓慢。

身为公司创始人，这无疑成了我所要面临的一个重大挑战。我们怀揣诸多创意与愿景，却时常因工程团队的限制而难以付诸实践。然而，人工智能（Artificial Intelligence，AI）辅助编程的引入彻底改变了这一局面，为Aisera注入了全新的活力。它让我们的团队变得更加敏捷与灵活，成为推动公司发展的强大倍增器。

如今，当我考虑聘用一名开发者时，我很想知道他如何巧妙地运用AI。我关心的是他如何利用这一前沿技术来简化烦琐的开发流程，以及他如何利用AI来提高代码编写的效率与速度。

一言以蔽之，我坚信，熟练掌握AI辅助编程工具已成为现代开发者的必备技能。正如微软首席执行官Satya Nadella所言，这些工具“正在成为每位开发者的标配……”。

然而，要真正理解并驾驭AI辅助编程，我们需要采取一种迥异的策略。这项技能与传统的开发方式截然不同。关键在于学会如何运用生成式AI系统。这正是Tom所著的这本书的价值所在。他精心地为渴望在AI辅助编程领域精通技能的开发者编写了一本实用的操作指南。书中不仅深入探讨了关键主题，还辅以丰富的实例，使得读者能够轻松掌握。更难能可贵的是，Tom以一种轻松有趣的方式呈现这些内容，让这一技术变得更易掌握且引人入胜。

毫无疑问，我们正身处一个充满无限可能与机遇的时代。在这个时代，职业发展的道路宽广而多样，而学习并掌握AI辅助编程技术无疑是你可以迈出的最具前瞻性且最有价值的一步。

Muddu Sudhakar

Aisera首席执行官兼联合创始人

# 前言

早在20世纪80年代初，我就踏上了编程的旅程。我的第一台设备是雅达利400，它配备薄膜键盘，内存仅有8KB，而程序则需要通过磁带加载。在那个时代，我学习了BASIC语言，并编写了简单的游戏和实用程序。

随着时间的推移，我的设备自然而然地升级到功能更为强大的机器。在这个过程中，我逐步接触并掌握了诸如Pascal、C和C++等编程语言。然而，除了引入语法高亮和增强调试功能以外，集成开发环境（Integrated Development Environment，IDE）在结构上并未发生显著的变化。

随着GitHub Copilot和ChatGPT的登场，整个编程领域经历了翻天覆地的变化。当我首次尝试这些工具时，感觉就如同我第一次拿起iPhone，那种颠覆性的体验令人难以忘怀。

我用自然语言让ChatGPT为我编写代码。在VS Code中，只需要输入一个函数片段，GitHub Copilot便能够生成完整的代码块，这种默契的协作仿佛每一次都正中靶心。更令人惊奇的是，我还尝试使用ChatGPT将图片转换为代码，这一创新功能为我带来了全新的编程体验。

然而，AI辅助编程工具真正的强大之处在于它们能够轻松处理开发者日常面临的诸多烦琐任务。谁愿意深陷于复杂的regex语句、费尽心思拼凑bash命令或调试GitHub Action呢？我当然也不喜欢。但有了这些AI辅助编程工具，那些曾经令人头疼的问题，如今能像享用早餐一样轻松解决。

事实证明，AI工具在编程领域的便捷性令人瞩目，而且其用途远不止于此。我开始利用ChatGPT来激发创意思维，构思应用程序的新颖想法，起草详尽的需求文档，甚至编写和执行单元测试。

很快，我便坚信不疑，AI辅助编程将成为未来开发者不可或缺的核心技能之一。

确实，我深刻感受到编写一本关于AI辅助编程图书的迫切需求。因此，我精心制定了一个大纲，并满怀信心地将其提交给O'Reilly出版社。令我欣喜的是，他们迅速认可了我的潜力和这本书的价值。

撰写本书的过程不仅充满了乐趣，更是一次宝贵的学习之旅。我有幸采访了许多才华横溢的开发者，他们不仅为我提供了丰富的见解和独特的观点，还分享了许多实用的编程技巧和宝贵的经验。

然而，AI辅助编程的发展日新月异，为了紧跟这一领域的最新动态，我特意为本书创建了一个配套的GitHub仓库（<https://github.com/ttaulli/AI-Assisted-Programming-Book>）。这个仓库不仅是我对本书内容进行持续更新和完善的地方，更是一个汇聚了AI辅助编程领域最新发展和重要突破的平台。

感谢你选择阅读本书。我衷心希望这本书不仅能为你带来丰富的知识和见解，更能成为你人生旅程中不可或缺的宝贵指南，引领你深入探索AI辅助编程的广阔天地。

## 内容概览

每一章的主要内容如下。

第1章深入探讨了生成式AI如何彻底颠覆开发者的工作模式。该章详细阐述了先进的AI工具如何协助开发者更加专注于全局性的战略思考，从而减轻他们在编程细节上的负担。此外，该章还追溯了编程语言的发展历程，并对GPT-4等前沿AI技术进行了详尽的介绍和解析。

第2章清晰阐述了生成式AI以及为何Transformer模型和大语言模型（Large Language Model，LLM）在编程领域备受推崇。更为关键的是，该章通过实际操作OpenAI的Playground平台，展示了如何灵活运用AI模型，并根据个人编程需求进行定制和调整。

第3章的内容对想要高效利用AI辅助编程工具的用户来说至关重要。该章详细解析了一系列实用技巧，旨在帮助用户处理冗长或令人困惑的提示，并有效防止AI产生不准确的输出。同时，该章还深入剖析了提示的关键组成部分，并提供了关于如何巧妙运用提示来优化编程体验的建议。

第4章将带你领略一款强大工具——GitHub Copilot的魅力。该章详细介绍了GitHub Copilot的核心功能，包括生成带注释的代码、聊天体验以及使用AI驱动的命令行界面。此外，该章还探讨了如何根据专有代码库的需求，对GitHub Copilot进行个性化定制，从而充分发挥其在软件开发中的巨大潜力。

第5章全面解析了其他常见的AI辅助编程工具，包括Amazon的CodeWhisperer、Google的Duet AI及Replit等。

第6章深入探讨了如何利用ChatGPT与其他通用LLM来处理正则表达式、快速启动代码编写以及优化GitHub Action等关键任务。

第7章的重点是使用聊天机器人启动软件项目，其中涉及头脑风暴、市场调研、竞争分析、需求分析和测试驱动开发等主题。

第8章深入剖析了AI辅助编程工具在软件开发过程中的典型应用场景，涵盖了API的使用、模块化编程策略、代码重构技巧等内容。同时，该章还探讨了AI工具在函数处理及面向对象编程方面的实践。

第9章聚焦软件开发中不可或缺但往往被忽视的调试、测试和部署环节。该章探讨了如何利用AI辅助编程工具有效修复错误、进行高效的代码审查、编写可靠的单元测试，以及编写优雅的拉取请求描述，以确保软件的质量与稳定性。

作为本书的结语，第10章总结了全书的核心要点，并强调AI辅助编程工具在软件开发中的重要作用和潜在价值。

## 本书特色

软件开发者以确定性为基石。当我们为程序输入特定的内容时，它始终如一地输出相同的结果，这种严格的逻辑确定性一直是软件的核心与精髓所在。

然而，当涉足AI辅助编程工具的世界时，事情开始反其道而行。获取结果仿佛成了掷骰游戏，因为一切都建立在概率之上。即使你多次使用相同的提示来引导AI工具编写代码，每次的输出也很可能不尽相同。初尝此道，或许会令人感到困惑和挠头，但一旦你掌握其中的窍门，就会发现这一切都是值得的。因此，本书特别设置了一章来深入解析提示工程，旨在帮助读者驾驭这种全新的编程方法。

## 谁应该阅读这本书

本书是为所有开发者量身打造的，无论是初出茅庐的新手，还是已在软件行业摸爬滚打多年的资深专家，本书都值得阅读。

## 本书所使用的印刷约定

下面是本书所使用的印刷约定。

**斜体 ( *Italic* )**

表示新的术语、URL、电子邮件地址、文件名和文件扩展名。

**等宽字体 ( Constant Width )**

用于代码清单，以及在段落中引用程序元素，如变量或函数名称、数据库、数据类型、环境变量、语句和关键字。

**等宽粗体 ( Constant Width bold )**

表示应由用户提供的值或由上下文确定的值替换的文本。



这表示一个一般性的注释。

## 如何使用示例代码

本书的补充材料（如代码示例、练习等）可从  
<https://github.com/ttaulli/AI-Assisted-Programming-Book> 下载。

如果读者在使用代码示例时遇到问题，可以通过异步社区中的本书页面与我们联系。

本书是为了帮助读者更好地完成工作。一般而言，对于本书所提供的代码示例，读者可以在自己的程序和文档中使用它们，并不需要联系我们以获得许可，除非读者复制了大块的代码。例如，在编写程序时使用了本书的几段代码并不需要许可，但是，销售或发布O'Reilly图书的代码示例则需要得到我们的许可；引用本书以及书中的代码示例并不需要获得许可，但是把本书的大量代码示例复制到自己产品的文档中需要获得许可。

我们赞赏注明出处，但一般情况下并不要求这么做。出处注明通常包括书名、作者、出版商和ISBN。例如，“*AI-Assisted Programming* by Tom Taulli (O'Reilly). Copyright 2024 Tom Taulli, 978-1-098-16456-0”。

如果读者觉得自己对本书的示例代码的使用超出了正常范围或者上面的许可，可以通过permissions@oreilly.com与我们联系。

## O'Reilly在线学习平台（O'Reilly Online Learning）

近40年来，O'Reilly Media致力于提供技术和商业培训、知识和卓越见解，来帮助众多公司取得成功。

O'REILLY®

我们拥有独一无二的专家和革新者组成的庞大网络，他们通过图书、文章、会议和我们的在线学习平台分享知识和经验。O'Reilly的在线学习平台允许你按需访问现场培训课程、深入的学习路径、交互式编程环境，以及O'Reilly和200多家其他出版商提供的大量文本和视频资源。更多相关信息请访问<http://oreilly.com>。

## 如何联系我们

如果读者对本书有任何的评论或疑问，可以与出版社联系。

美国：

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

中国：

北京市西城区西直门南大街2号成铭大厦C座807室（100035）

奥莱利技术咨询（北京）有限公司

我们为本书提供了一个网页，其中包含勘误表、示例和其他信息，读者可以通过<https://oreil.ly/AI-assisted-programming>进行访问。

如果你对本书有什么评论或技术上的建议，请发送电子邮件到  
[errata@oreilly.com.cn](mailto:errata@oreilly.com.cn)。

关于我们的图片和课程的新闻与信息，可以访问  
<http://oreilly.com>。

致谢

衷心感谢O'Reilly的工作人员，特别是Nicole Butterfield、Shira Evans和Brian Guerin，他们的信任与辛勤付出使这本书得以问世。同时，我也要向杰出的技术审稿人表达我的感激之情，他们分别是Roja Boina、Abraham Borg、Sarah Kim、Ebubechukwu (Nnenna) Oguaju-Dike和Gaurav Deshmukh，他们的专业指导和宝贵意见对本书质量的提升起到了关键作用。

## 第1章

# AI打开开发者的新世界

2011—2016年，Andrej Karpathy不仅在斯坦福大学深耕密集神经网络架构和像素处理计算机视觉的研究，同时还在Google做兼职工作。在Google期间，他持续优化YouTube视频系统，并成功开发出一套先进的特征学习系统。随后，他选择加入OpenAI，成为其创始成员之一。紧接着，他加入特斯拉担任人工智能高级总监，领导团队成功构建了自动驾驶系统。

Karpathy无疑是世界上顶尖的开发者之一，不仅如此，他还擅长文字创作，其推特账户上拥有近80万粉丝。当ChatGPT这一创新技术问世时，他发表了一条引人注目的推文：“最热门的新型编程语言是英语”（The hottest new programming language is English）。他的话绝非戏言，这既是对编程艺术的深情颂歌，也是对未来的瑰丽憧憬。展望未来，只须通过自然语言输入简单的提示，就能轻松调用任意编程语言的代码。你的计算机仿佛被嵌入了一个精通双语的“精灵”，它能够随时将你的英语指令转换为精确的代码命令，实现编程愿景。

随后，他的另一条推文道出了许多开发者的心声：“Copilot极大地提升了我的编程效率，让我不想回到过去‘手动编程’的日子。尽管我还在不断熟悉和学习如何充分利用Copilot的功能，但它已经为我自动生成了大约80%的代码，且准确率高达80%。我感觉自己不是在编程，而更像是在与Copilot进行高效的协作，我只需要提供必要的提示和进行精细的编辑即可。”

Karpathy向微软的GitHub Copilot致敬，这一创新举措标志着AI辅助编程新时代的到来。紧接着，诸多类似工具如雨后春笋迅速涌现，创新的步伐之快令人惊叹。

对于广大开发者，眼前呈现的是一片生机勃勃、充满活力的编程丛林。这片丛林中的AI工具为我们打开了一个怎样的新世界？它们有哪些闪光点让人惊艳，又有哪些挑战需要我们正视？你该如何在这片丛林中披荆斩棘，成为一名游刃有余的AI辅助编程专家？

本书将为你揭晓这些疑问的答案，甚至更多。我们将深入探讨如何利用这些智能工具加速编程过程，使代码编写更加高效、智能且充满乐趣。现在就让我们挽起袖子，共同踏上这场充满挑战与机遇的AI辅助编程之旅吧！

## 1.1 编程的演变与革命

编程语言发展的核心驱动力之一是抽象化，这是一种独特的方法，它极大地简化了开发者与系统之间的交互。抽象化通过后台处理了烦琐的细节，使开发者能够更专注于核心任务的实现。这一进步一直是推动技术创新的关键力量，为互联网、云计算、移动互联网及AI等领域的突破性发展提供了坚实的支持。

图1-1直观地展示了近几十年来抽象化概念在编程语言中的演变过程。

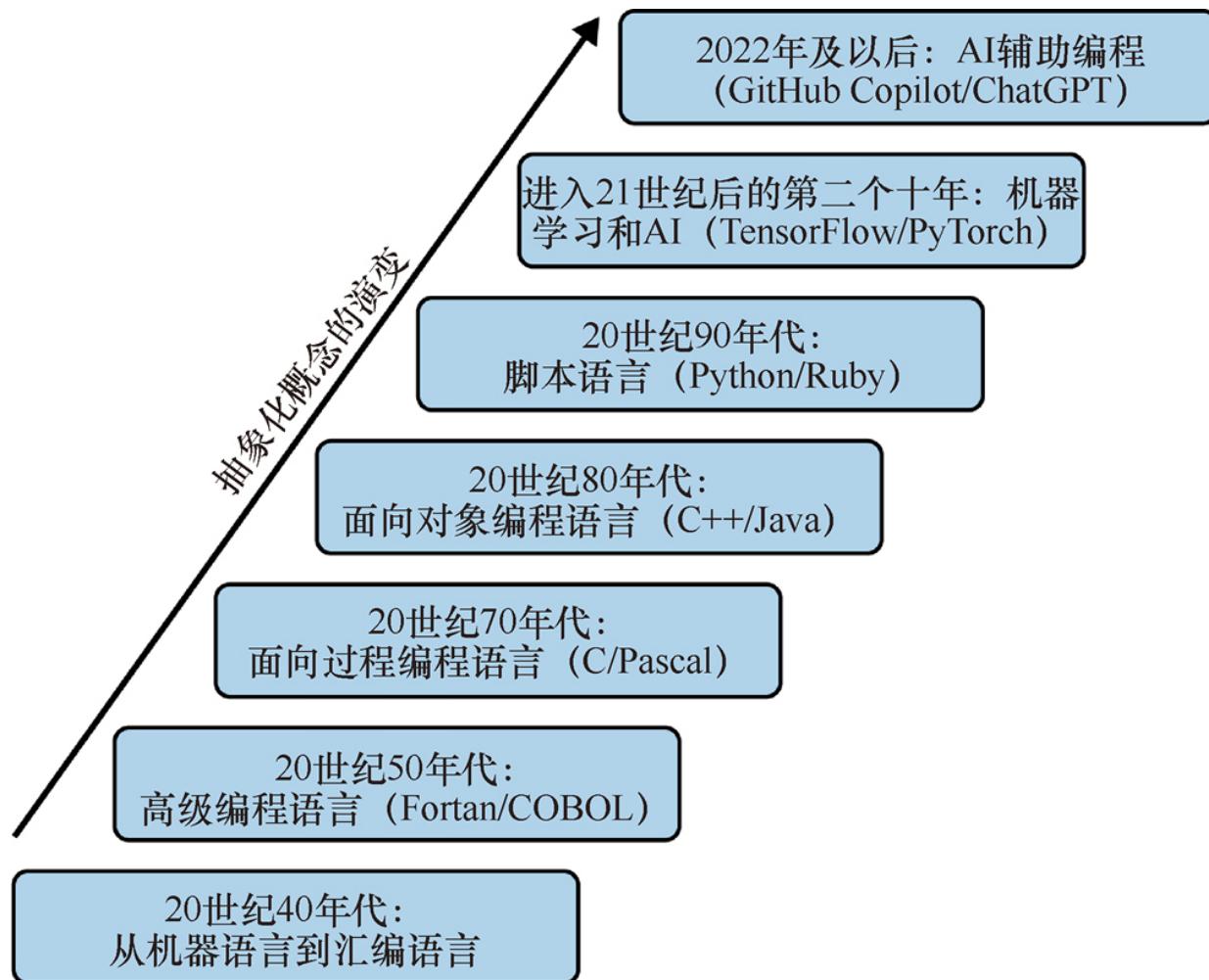


图1-1 近几十年来抽象化概念在编程语言中的演变过程

我们从20世纪40年代开始详细回顾一下。

### 从机器语言到汇编语言

在计算机时代的早期，开发者不得不与0和1搏斗，让机器按照他们的意志运转。后来，汇编语言出现了。它提供了字母与数字指令，使编程变得更容易，也显著降低了出错的可能性。

### 高级编程语言

20世纪50年代，Fortran和COBOL这两种高级编程语言诞生，它们允许程序员通过DISPLAY、READ、WRITE和IF/THEN/ELSE等

简单的英语指令编写代码。编译器则负责将这些代码转换成计算机能够理解的二进制代码（0和1）。同时，这些高级编程语言使得没有技术背景的人也能较好地阅读代码，理解程序的工作流程。高级编程语言的出现无疑极大地推动了计算机革命的进程。

## 面向过程编程语言

随着C和Pascal等编程语言的兴起，面向过程编程的概念开始普及。这些编程语言通过引入函数的概念，将复杂的编程任务分解为一系列整齐的小模块。这种抽象化的编程方式不仅提高了代码的可重用性和可维护性，还使得管理庞大的软件项目变得得心应手。

## 面向对象编程语言

在面向对象编程领域，C++和Java等编程语言脱颖而出。这些编程语言引入了全新的抽象化，允许开发者使用类和对象来模拟现实世界中的实体，并封装数据和行为。面向对象编程极大地促进了软件开发的模块化，使得解决问题变得更加直观。

## 脚本语言

Python、Ruby和JavaScript等脚本语言的出现，为编程领域带来了新的活力。它们通过抽象许多与编程相关的底层任务，提供了大量的库和内置数据结构，使得常见的编程任务变得更为简单。这些编程语言极大地减少了完成任务所需的代码量。

## 机器学习和AI

随着机器学习和AI技术的快速发展，TensorFlow、PyTorch等框架和专业库应运而生。这些框架和库通过抽象出许多复杂的数学编程细节，使得开发人员能够专注于模型架构的设计和训练过程的优化。

## AI辅助编程

最新的抽象化里程碑无疑是AI辅助编程。GPT-4和其他大语言模型（Large Language Model，LLM）如同智能助手，随时准备在开发者的命令下生成代码。

让我们来看一个简单的例子。在这个例子中，我们使用具有强大编程能力的ChatGPT。我们将使用一段提示语来引导ChatGPT完成特定的编程任务。假设我们给出的提示如下。

**提示：**用Python编写一个程序，检查给定的整数是偶数还是奇数，并打印结果。

图1-2展示了来自ChatGPT的响应。

```
python Copy code
def check_even_or_odd(number):
    """
    检查给定的整数是偶数还是奇数，并打印结果。
    参数:
        number (int): 要检查的整数
    返回值:
        None
    """
    # 检查数字是否为偶数
    if number % 2 == 0:
        print(f"{number} 是偶数")
    else:
        print(f"{number} 是奇数")

    # 从用户输入获取一个整数
    number = int(input("请输入一个整数: "))

    # 调用函数并传递用户输入的整数
    check_even_or_odd(number)
```

### 代码解释：

#### 1. 定义函数 `check\_even\_or\_odd`：

- 该函数接收一个整型参数 `number`。
- 文档字符串解释了函数的作用、参数和返回值。

图1-2 当ChatGPT被要求创建代码时，它的回复不仅包括代码，还包括代码注释

我们成功地获取了ChatGPT生成的代码，这段代码不仅包含了实用的代码注释，还附带了关于程序运行机制的详细说明。只需要单击图1-2右上角的“Copy code”（复制代码）按钮，即可将这段代码轻松粘贴

到集成开发环境（ Integrated Development Environment , IDE ）中，并加以运行和测试。

## 1.2 生成式AI

在深入剖析AI辅助编程工具的工作原理之前，我们首先需要对其基石——生成式AI（ Generative AI , 或称为GenAI ）有一个基本的了解。生成式AI作为人工智能领域的一个重要分支，专注于生成创新且独一无二的内容。为了更直观地理解这一概念，请参考图1-3，它清晰地描绘了AI系统中各部分之间的关系。



图1-3 以生成式AI和LLM为中心，用嵌套子集的形式表示不同类型的  
人工智能

人工智能是一个十分广泛的领域，它涵盖了所有能够模拟人类智能以完成任务的系统。在这一宏大领域中，机器学习（Machine Learning, ML）占据了至关重要的位置。机器学习系统通过对海量数据进行分析来提炼深层次的见解。它通常以复杂的算法为基础，不需要硬编码即可进行预测或决策。

深度学习（Deep Learning, DL）作为一种特殊的机器学习技术，通过构建多层次的神经网络（这正是“深度”一词的来源），实现了对数据更深层次的抽象和表示。深度学习在图像识别、语音识别等领域取得了令人瞩目的成果。

生成式AI通过学习和分析训练数据，能够创造出反映这些数据特征的新数据。

生成式AI中最引人注目的则是LLM，如GPT-4、Gemini、Claude和LLaMA 2等。这些被称为“基础模型”的强大工具，基于先进的算法和海量数据的训练，能够生成与人类语言高度相似的文本。

然而，生成式AI并不仅限于LLM。生成式AI还具备多模态能力，即能够生成图像、音频和视频等多种类型的数据。

第2章将深入剖析生成式AI的工作原理和机制。在此之前，我们先了解一下AI辅助编程工具的优缺点。

### 1.3 AI辅助编程工具的优点

AI辅助编程工具旨在为开发者赋能，使他们能够摆脱单调任务和复杂代码细节的束缚，从而专注于解决高级问题和实现创新。GitHub选择“Copilot”（副驾驶）这一术语，恰如其分地描绘了这些工具的角色——它们如同驾驶舱中值得信赖的伙伴，为复杂且枯燥的编程工作提供导航，让开发者能够专注于真正重要的事情。

在接下来的内容中，我们将深入探讨这些强大的AI辅助编程工具的独特优势以及它们在现实世界中的广泛应用。

### 1.3.1 最小化搜索

开发者时常会陷入数字迷宫，追踪棘手的软件错误或解密复杂的代码逻辑。面对困境，他们通常会求助于Google或Stack Overflow，输入代码片段进行搜索，然后迅速返回IDE中继续工作。

但有些时候，搜索过程仿佛一场漫长的马拉松。Stack Overflow上的讨论可能令人一头雾水，即使不断搜索，似乎也找不到正确的答案。然而，偶尔会有某个讨论给予启发，于是深入研究相关主题，甚至可能在YouTube上观看教学视频。经过30多分钟的苦苦思索，最终找到解决方案。

所有开发者都经历过这种挑战。事实上，Stack Overflow在2022年的《开发者调查报告》（涵盖70000多名开发者）中强调了搜索答案的挫败感。该报告显示，62%的受访者在搜索答案上每天花费超过30分钟，而25%的受访者更是超过1小时。对于一个拥有50名成员的开发团队，每周因搜索答案/解决方案而损失的时间达到333 ~ 651小时。

现在，如果有一种方法可以穿越耗时的搜索，迅速找到解决方案，那会怎样呢？我们的闪亮算法骑士——AI辅助编程登场。微软的研究证明：90%以上使用GitHub Copilot的开发者能以更快的速度完成任务。

为了验证这一点，微软进行了一次编程实验。他们招募了95名专业开发者，并将这些开发者分为两组，要求他们用JavaScript开发HTTP服务器。结果显示，使用GitHub Copilot的开发者完成任务的速度比不使用该工具的开发者快55%。

不仅是微软，麦肯锡公司也进行了一项研究。他们调查了来自美洲和亚洲的40多名开发者，这些开发者的经验和背景各不相同。在数周内，他们完成了3项常见的软件任务——代码生成、重构和文档编写。

研究结果表明，在保持文档和代码的整洁性方面，AI辅助编程工具表现出色，所需时间减少了近一半。而在编写新代码和重构代码方面，使用AI辅助编程工具所花费的时间与传统方法几乎相同。然而，对于特别复杂的任务，AI辅助编程工具的优势并不明显，仅缩短了不到10%的时间。

有趣的是，研究还发现，尽管AI辅助编程工具缩短了完成任务的时间，但并未对代码的整体质量（如错误率、可读性和可维护性）产生负面影响。实际上，随着开发者逐渐熟悉并使用这些工具，代码质量还得到了边际提升。

麦肯锡公司的研究为我们揭示了AI辅助编程的潜力，并为未来开发提供了一些宝贵的启示。

## 优化日常任务流程

AI辅助编程工具极大地简化了琐碎任务的处理，如自动填充代码、实时提供代码建议以及自动生成代码文档。通过这些方式，它们为开发者释放了宝贵的时间，使他们能够更专注于解决复杂的业务问题，并加速软件功能的部署。

## 激发代码创作的灵感

面对空白的代码编辑器，许多开发者可能会感到无从下手。然而，借助AI生成工具，开发者可以在IDE中，或仅仅通过简单的提示，迅速获得有价值的代码建议，从而激发创造灵感，迅速进入高效的编程状态。

## 高效调整与优化现有代码

通过智能提示，开发者能够更迅速地对现有代码进行调整和优化。他们可以轻松地从在线库中检索代码片段，输入提示框中，并基于特定标准请求AI进行迭代性调整。

## 增强对新挑战的适应力

当面对新的编程环境或编程语言时，AI辅助编程工具就像一位经验丰富的导师，可以帮助开发者快速熟悉并掌握新知识。它们能够清晰地解释新概念，分析各种代码库，并提供全面的框架使用指南。

## 整合多种工具以提升效率

研究表明，结合使用多种工具能够显著提升工作效率。例如，开发者可以使用一款工具进行代码提示或聊天交流，同时将另一款工具作为代码库的一部分，由其提供智能自动完成选项和建议。这些工具在协作中展现出惊人的能力。在代码重构过程中，第一款工具因其灵活的对话特性，在处理查询时表现出色；而第二款工具则在生成新代码方面表现高效，并能与IDE无缝对接。当这些工具协同工作时，开发者观察到的时间效率提升高达1.5 ~ 2.5倍。

### 1.3.2 充当你的顾问

通过ChatGPT，你可以就不同场景下的开发活动寻求建议。下面是一个实用的提示。

**提示：**为了在编程时最大限度地减少搜索时间并提升工作效率，请提供详细的技巧和最佳实践。要求涵盖与代码组织、文档、工具和思维方式相关的策略。

ChatGPT的响应如图1-4所示。

## 代码组织

### 1. 模块化编程

- 将代码分割成小的、可重用的模块或函数，每个模块或函数只做一件事。
- 使用明确的模块结构和目录结构来组织代码，使代码库更易于导航。

### 2. 命名规范

- 使用有意义且一致的命名规范，以便从名称就能理解变量、函数、类的作用。
- 遵循命名约定，如驼峰命名法（CamelCase）用于类名，蛇形命名法（snake\_case）用于函数名和变量名。

### 3. 代码注释

- 为复杂的代码段添加注释，解释代码的功能和逻辑。
- 使用块注释（/\* ... \*/）和行注释（//）来区分不同类型的注释。

### 4. 代码格式化

- 遵循一致的代码风格和格式化规则，使用工具如 Prettier 或 Black 自动格式化代码。
- 使用空行、缩进和空格来增加代码的可读性。

## 文档

### 1. 文档字符串

- 在每个模块、类和函数的开头使用文档字符串，解释其用途、参数和返回值。
- 遵循标准的文档字符串格式，如 Google Style、NumPy/SciPy Style 或 Sphinx Style。

### 2. 自动化文档生成

- 使用工具如 Sphinx、Doxygen 或 JSDoc 从代码中自动生成文档。
- 保持文档和代码同步，确保文档始终是最新的。

### 3. README 文件

- 在每个项目的根目录中包含一个详细的 README 文件，解释项目的目的、安装步骤、用法和示例。
- 使用 Markdown 格式编写 README 文件，使其易于阅读和维护。

## 工具

### IDE

- 使用功能强大的 IDE，如 Visual Studio Code、PyCharm 或 IntelliJ，利用其代码自动补全、错误检查和调试功能。
- 配置 IDE 插件，如 linter（如 ESLint、Pylint）和代码格式化工具。

图1-4 ChatGPT给出的有关编程任务的建议

### 1.3.3 与IDE集成

与IDE的无缝集成对于AI辅助编程具有举足轻重的作用。这种集成确保了开发流程的流畅进行，避免了掌握新平台所带来的额外学习负担，从而节约宝贵的时间，使开发者能将更多精力专注于编写代码，减少在不同平台或工具之间切换的次数，显著降低操作的摩擦，让开发体验变得更加流畅。

实时反馈机制更是与IDE集成的亮点之一。当开发者编写或修改代码时，集成工具能够即时指出潜在错误，提供修正建议或推荐更优的编程方法。这种即时反馈如同一位时刻陪伴在侧的友好教练，引导开发者编写出更为简洁、高效的代码，而无须进行烦琐的人工审核或外部检查。

此外，AI辅助系统还能通过深入洞察更广泛的编程领域来优化IDE的使用体验。它能够理解变量类型、方法签名及项目结构的深层逻辑，从而为开发者提供与项目高度相关的代码建议。这凸显了AI在编程辅助中不仅具有强大的代码生成作用，更在于其能够深入理解并优化整个开发过程。

表1-1展示了流行的一些AI辅助编程工具及其支持的IDE。

表1-1 一些流行的AI辅助编程工具及其支持的IDE

AI辅助编程工具	IDE
GitHub Copilot	Visual Studio Code Visual Studio Vim Neovim JetBrains suite Azure Data Studio
Tabnine	Visual Studio Code WebStorm PyCharm Eclipse IntelliJ Platform PhpStorm CLion Neovim JupyterLab Rider DataGrip AppCode Visual Studio 2022 Android Studio GoLand RubyMine Emacs Vim Sublime Text Atom AI Jupyter Notebook
CodiumAI	Visual Studio Code JetBrains (IntelliJ PyCharm CLion WebStorm)
CodeWhisperer	Visual Studio Code IntelliJ IDEA AWS Cloud9 AWS Lambda console JupyterLab Amazon SageMaker Studio JetBrains (IntelliJ PyCharm CLion, GoLand WebStorm Rider PhpStorm RubyMine DataGrip)



根据微软的一项深入调查研究，高达88%的GitHub Copilot用户反馈，在使用此工具后，他们的挫败感显著降低，同时注意力也更为集中。其中一个尤为显著的原因是，GitHub Copilot帮助他们在IDE中减少了搜索时间，这极大地促进了开发者持续处于高效的“心流”状态。

### 1.3.4 适配你的代码库

在编程领域，一些先进的AI辅助编程工具被精心打造，旨在与特定的开发环境无缝对接。这些工具不仅具备强大的适应性，而且能根据开发者的微调，深入理解项目的内部库、API（Application Programming Interface，应用程序接口）、最佳实践及架构蓝图。通过这种方式，它们提供的代码建议不仅在技术上无可挑剔，而且能精准满足项目的需求。

定制功能进一步确保了生成的代码建议与组织既定的编程标准、质量指标和安全协议高度一致。在追求高质量代码的道路上，这样的工具能够帮助团队避免使用过时或不被推荐的代码片段，从而保障代码库的鲁棒性和可维护性。

此外，对于新加入开发团队的成员，这些能够适配代码库的智能工具更是一大福音。传统上，新人需要花费大量时间适应新的代码库，包括浏览代码、查阅文档以及学习编程规范。然而，借助AI辅助编程工具，这一过程将大幅缩短。

### 1.3.5 检查代码完整性

代码完整性是高质量软件开发的基石，它体现了源代码执行预期功能时的稳定性与可信度。它如同一面精密的透镜，透过它可以审视代码的完整性、准确性、一致性和安全性。任何瑕疵都可能为系统埋下安全隐患，导致潜在的漏洞和数据泄漏风险。

影响代码完整性的关键因素众多，包括但不限于代码的精确性、完整性、统一性、安全规定及维护的难易程度。为了提高代码的完整性，开发者可以采用多种方法，如执行单元测试、集成测试、进行同行代码审查、进行静态代码分析和严格的安全评估等。

随着技术的不断进步，越来越多的AI辅助编程工具涌现，它们正致力于强化代码完整性。这些工具能深入剖析代码的每个细节，生成精准且富有洞察力的单元测试和边缘案例。

部分先进工具还配备了“修复”建议功能，这些建议均经过严格审核，确保在提供给开发者之前不会引入新的问题。开发者可以在IDE中直接查看和采纳这些建议。

此外，AI辅助编程工具的一大优势在于它们能够快速分析拉取请求（Pull Request，PR），简洁明了地总结代码变更。它们还能自动生成发布说明，为记录软件版本的演进提供便利。

### 1.3.6 AI驱动的文档生成器

虽然文档常被低估，但其在软件开发过程中扮演着至关重要的角色。它是确保代码库具备高可读性、可维护性和可扩展性的关键因素，尤其在当前团队协作频繁、项目复杂度不断提升的背景下。然而，编写和更新文档往往被视为一项烦琐且耗时的任务，有时甚至被置于次要地位，这在一定程度上削弱了文档应有的价值。

幸运的是，随着人工智能技术的快速发展，AI辅助编程工具为文档编写带来了革命性的改变。这些智能工具如同数字抄写员，能够在极短的时间内高效生成大量文档，且内容质量上乘，表达清晰。之所以能实现这一点，是它们依赖于LLM这一核心技术。LLM在处理自然语言方面展现出卓越的能力。

### 1.3.7 现代化支持

2011年，Marc Andreessen在《华尔街日报》上发表了颇具远见的卓识——“软件正在吞噬世界”。

作为一名杰出的企业家和风险投资家，Andreessen以敏锐的洞察力著称，他总能在技术的浪潮中捕捉到变革的脉搏，创造出一个又一个商业传奇。他的这篇文章无疑是技术史上的一个里程碑。

Andreessen强调，随着基础设施的日臻成熟，全球产业正站在一个变革的临界点。AWS等云平台的崛起与宽带互联网的普及，如同打开了潘多拉的盒子，彻底改变了行业的游戏规则。它们不仅打破了服务

器成本和网络技术的传统壁垒，更为像Uber、Netflix及众多社交媒体平台这样的创新者提供了改写行业规则的舞台。

从Andreessen这篇深具洞察力的文章中，我们不难窥见，创新的车轮正在飞速转动，而它所带来的颠覆性力量，对大型企业来说尤为显著。许多传统巨头依然依赖于昂贵且僵化的传统系统，实现现代化转型无异于一场风险重重的博弈。企业的层级结构往往成为决策的绊脚石，庞大的体量则让变革之路更加崎岖。此外，员工可能难以与日新月异的技术创新保持同步。

IBM敏锐地将这一挑战作为契机，倾注大量资源研发出AI辅助编程工具。2023年10月，IBM正式推出Watsonx Code Assistant for Z，该系统能在大型机系统上将COBOL代码优雅地翻译成Java代码，并输出高质量的面向对象代码。

Watsonx.ai模型作为IBM的得意之作，具备对115种编程语言的深刻理解能力，背后依托的则是1.5万亿个token（分词）的强大支撑。该模型拥有约200亿个参数，无疑是当前代码开发领域的一颗璀璨明珠。

然而，将拥有数千亿行代码的COBOL系统迁移到现代语言绝非易事。像COBOL这种拥有几十年历史的编程语言，往往缺乏或几乎没有文档支持。若转换处理不当，后果将不堪设想。要知道，世界上大部分信用卡处理是通过大型机完成的。

遗憾的是，失败的迁移项目屡见不鲜。例如，加利福尼亚州机动车辆管理局就曾投入2.08亿美元<sup>②</sup>，却在数年后不得不终止项目。

大型机项目的高风险性质导致开发者薪酬普遍较高，但企业在招聘方面仍面临重重挑战。年轻的开发者往往接受的是现代编程语言的培训，对大型机开发持怀疑态度。同时，众多经验丰富的大型机开发者即将步入退休年龄。

IBM深知，要应对大型机开发的严峻挑战，AI的介入不可或缺。尽管传统的代码转换器或翻译器已存在多年，但它们往往只是简单地将COBOL的“面条式代码”转换为Java的“面条式代码”，并未带来实质性改进。因此，这些Java代码仍需要开发者做大量的润色工作，这也是许多项目步履维艰甚至失败的原因。

但IBM表示，通过运用生成式AI，他们能够显著提升项目成果，提升的效率甚至达到10倍。

其他公司也在积极把握现代化的机遇。GitHub的首席执行官Thomas Dohmke在其帖子中提到：“使用COBOL的主框架仍在运行是一个比我们想象的要严重得多的社会问题。”在接受《财富》杂志采访时，他表示，2023年他听到的关于COBOL的消息比过去30年都要多。他还透露，许多公司都在询问如何在迁移项目中使用GitHub Copilot。

ChatGPT也精通传统编程语言。表1-2列出了常见的传统编程语言。

表1-2 常见的传统编程语言

编程语言	描述	使用时期
COBOL	为处理业务数据而开发	20世纪50年代晚期至20世纪60年代早期
Fortran	专为科学和工程计算而设计	20世纪50年代
Pascal	为鼓励良好的软件工程实践而开发	20世纪60年代晚期至20世纪70年代早期
BASIC	为学生和初学者设计的易学编程语言	20世纪60年代中期
ALGOL	影响了后来的编程语言，如Pascal C和Java	20世纪50年代晚期至20世纪60年代早期
汇编语言	与所设计的CPU架构相匹配，可追溯到早期的可编程计算机	计算机时代早期
PL/I	用于科学、工程、商业和系统编程	20世纪60年代早期

下面的例子展示了AI辅助编程对传统编程语言的帮助。假设需要处理以下代码片段。

```

MODULE ComplexModule

IMPLICIT NONE

TYPE :: ComplexType

REAL :: real, imag

CONTAINS

OPERATOR(+) (a, b) RESULT(c)

TYPE(ComplexType), INTENT(IN) :: a, b

TYPE(ComplexType) :: c

```

```
c%real = a%real + b%real  
  
c%imag = a%imag + b%imag  
  
END OPERATOR  
  
END TYPE ComplexType  
  
END MODULE ComplexModule
```

你不知道这是什么编程语言，也不知道它是如何工作的。你通过代码不能直观地理解工作流程。

现在，你访问ChatGPT并输入以下内容。

**提示：**这段代码用什么编程语言编写的？有什么作用？另外，请解释它是如何工作的。

图1-5展示了来自ChatGPT的响应。



这段代码是用Fortran语言编写的。

### 作用：

这段代码定义了一个Fortran模块 (`ComplexModule`)，其中包含一个复数类型 (`ComplexType`)。该模块还定义了一个用于复数加法的重载运算符。

### 如何工作：

#### 1. 模块声明：

- `MODULE ComplexModule` 开始了一个模块的声明。
- `IMPLICIT NONE` 声明了在模块中使用显式变量声明。这意味着所有变量都必须在使用前显式声明，以防止隐式声明。

#### 2. 类型定义：

- `TYPE :: ComplexType` 定义了一个名为 `ComplexType` 的类型。
- `REAL :: real, imag` 声明了 `ComplexType` 类型的成员变量，即实部和虚部。

#### 3. 运算符重载：

- `OPERATOR(+) (a, b) RESULT(c)` 定义了一个重载运算符“+”，用于两个复数的相加。
- `TYPE(ComplexType), INTENT(IN) :: a, b` 指定了输入参数 `a` 和 `b` 的类型为 `ComplexType`。
- `TYPE(ComplexType) :: c` 声明了结果变量 `c` 的类型为 `ComplexType`。
- `c%real = a%real + b%real` 和 `c%imag = a%imag + b%imag` 执行了复数实部和虚部的相加，并将结果存储在 `c` 中。



图1-5 ChatGPT对用传统编程语言编写的代码的解释

ChatGPT能够精确地辨识所提供的代码片段是用Fortran编程语言编写的。它不仅指出这段代码定义了一个名为ComplexModule的模块，还详尽地解释了该模块中的关键组成部分。具体来说，该模块包含一个名为ComplexType的派生类型，用于表示复数，还包含一个重载的加法运算符“+”，该运算符被设计为处理两个复数之间的加法运算。随后，ChatGPT对这段代码进行了逐步深入的解析。

## 1.4 AI辅助编程工具的缺点

当我们探讨AI辅助编程工具的潜力和优点时，AI辅助编程工具尚存的不足同样不可忽视。正如任何新兴技术在其初始阶段都会面临挑战一样，即使是革命性的产品，如第一代iPhone，也曾显得稚嫩。AI辅助编程工具同样存在局限性、问题，以及需要克服的障碍。在创新的征途上，总有一些细节需要精心打磨和持续微调。接下来让我们深入探索AI辅助编程工具的一些不足之处。

### 1.4.1 幻觉

对于LLM，幻觉是指模型输出的数据看似准确，但实际上并不正确，或者与模型所训练的输入数据不符。幻觉给软件开发带来巨大挑战。它会导致不准确的代码建议，生成误导性文档，并产生错误的测试场景。此外，它还会导致调试效率低下，误导初学者，并有可能削弱开发者对AI工具的信任。

然而，值得欣喜的是，业界在减少幻觉现象方面已取得显著进展。众多学术研究机构致力于深入探究幻觉问题，而人工智能公司也积极采用各种有效策略，如通过基于人类反馈的强化学习（Reinforcement Learning from Human Feedback，RLHF）来努力减轻这一现象的影响。

但我们必须意识到，鉴于LLM本身的复杂性及其背后庞大的数据基础，要完全消除幻觉现象似乎是一项艰巨而长期的任务。

此外，在使用AI辅助编程工具时，我们观察到某些编程语言的表现更为出色。例如，Python、JavaScript、TypeScript和Go等编程语言，由于在公共资源库中具有广泛的代表性和丰富的数据集，使得AI在学习这些编程语言时能够提供更准确、更强大的建议。

### 1.4.2 知识产权

Matthew Butterick拥有开发者、设计师和律师的多重身份背景，对字体设计情有独钟。他的人生轨迹涵盖了写作、字体设计以及文档编辑和排版程序的开发。然而，2022年6月，当他接触到GitHub Copilot时，他并未感受到期待中的喜悦，反而激发他撰写一篇题为“This Copilot is Stupid and Wants to Kill Me”的博客文章来表达他的不满。

Butterick的愤怒并未随着这篇博客文章的发表而平息。他的不满很快升级，最终演变为对微软、GitHub和OpenAI提起的集体诉讼。诉讼的核心争议在于GitHub是否违反服务条款和隐私政策，并可能涉及版权侵犯。

这一法律纠纷凸显了AI辅助编程工具在知识产权领域的模糊地带。鉴于其输出结果往往是海量现有代码的混合体，代码的所有权问题变得尤为复杂。

在关于知识产权的探讨中，普遍认可基于“合理使用”的原则。然而，法律界对于AI生成内容的处理方式并不明确，缺乏具体的指导原则。为了解决这个问题，可能需要依赖法院裁决来明确相关权益。

与此同时，微软已经为GitHub Copilot的用户构建了一道法律屏障。微软承诺，在满足一定条件的情况下，将保护用户免受法律索赔的困扰。

AI辅助编程工具与开源软件方法的结合进一步加剧了法律问题的复杂性。Copyleft许可证，如GNU通用公共许可证（General Public License, GPL）的第2版和第3版，要求任何衍生作品都必须遵循原始代码的许可条款。这一要求旨在促进创新，但同时也给开发者带来了挑战，因为它可能剥夺了他们保护自己的应用程序知识产权的权利，甚至要求他们公开整个代码库的源代码。

### 1.4.3 隐私

AI辅助编程工具由于云端部署的特性，引发了广泛的数据隐私和保密性问题。公司内部如何妥善保护数据？这些数据是否会被用作训练数据？

不同供应商在数据处理上可能有着截然不同的做法，这也导致一些开发者对AI辅助编程工具持谨慎态度，甚至选择完全避免使用它们。

Gridspace公司的联合创始人兼联合工程负责人Anthony Scodary就是其中一位坚定走自己道路的开发者。该公司专注于开发能够处理复杂电话对话的语音机器人，其技术基础涵盖语音识别、语音合成、LLM和对话系统等多个领域。

Gridspace公司并没有选择利用现有的AI辅助编程平台，而是选择了一条独特的道路——自主研发AI辅助编程平台。Gridspace公司的AI辅助编程平台基于Kubernetes集群中的Docker服务，以IDE扩展的形式进行部署，并能针对公司的特定代码库进行微调。Scodary表示：“通过自研的AI辅助编程平台，我们能够避免将自己的知识产权和数据发送给其他公司。同时，这也意味着我们拥有了一个更小、更高效、更符合我们工作风格的模型。”

虽然Gridspace公司的做法并非放之四海而皆准，因为每个组织都有自己的观点和偏好，但在评估AI辅助编程工具时，深入了解其对数据隐私的影响至关重要。

#### 1.4.4 安全

在题

为“Security Weaknesses of Copilot Generated Code in GitHub”的研究论文中，Yujia Fu等学者深入探讨了GitHub Copilot在安全方面的隐患。通过对GitHub上435个项目中由AI生成的代码片段进行详尽的审查，他们发现其中有高达35.8%的代码片段存在通用缺陷枚举（Common Weakness Enumeration，CWE）实例。

这些安全问题并非局限于某一特定编程语言，而是跨越了42种不同的CWE类别，呈现出多编程语言错误行为的复杂性。其中，操作系统命令注入、使用不充分的随机值以及异常情况检查或处理不当这3个类别的问题尤为突出。需要注意的是，这些CWE问题中有11个被列入2022年CWE Top 25清单。

然而，这并不意味着AI辅助编程工具本身必然带来巨大的安全风险。事实上，供应商们正在持续努力研究并改进相关防护措施。但无论如何，对于任何代码，保持高度的安全意识都是非常重要的。

### 1.4.5 训练数据

AI辅助编程工具所依赖的LLM训练数据，往往存在显著差距，这些差距会直接影响它们在现实应用中的表现与实用性。以下是对这些差距的详细分析。

#### 表征差异

若编程语言或代码库的特定领域在开源项目中缺乏足够的体现或完全未被涉及，则AI模型可能会因此缺乏相应的知识，导致提供的编程建议不够准确。这凸显了训练数据的质量和范围对AI输出质量的重要性。

#### 质量不一致

LLM中的开源代码质量参差不齐，你无法预知下一个项目会是什么质量。优秀的项目与劣质的代码共存，这导致AI辅助编程工具提供的建议质量也呈现出不稳定性。

#### 知识截止日期

LLM的训练数据往往有明确的截止日期，这相当于一个时间快照。当编程语言或代码库发布新版本、更新或被废弃时，模型的知识库可能无法及时反映这些变化，从而带来挑战。

## 泛化差距

泛化差距即模型在训练数据与未见数据上的表现差异，它是人工智能领域的一个常见问题。根据Rie Johnson和Tong Zhang在论文“Inconsistency, Instability, and Generalization Gap of Deep Neural Network Training”中的研究，缩小这种差距是提高模型性能的关键。

## 上下文理解

AI虽然可以基于经验提供建议，但如果它没有遇到过与当前场景完全相似的情境，则可能会做出错误的判断。因此，在创建提示时，我们不能仅凭假设，而需要更加细致地考虑上下文环境。

### 1.4.6 偏见

开发者难以牢固掌握人工智能伦理，这可能是由于计算机科学课程或强化训练中通常缺乏对这一关键主题的深入探讨。这种知识上的不足可能导致算法在设计中无意间产生偏差，甚至可能引发数据的滥用。

同样，AI辅助编程工具也面临着偏见问题。这些工具在生成代码时，可能会不自觉地延续其训练数据中已有的偏见。例如，当要求生成人名列表时，它们可能会倾向于推荐英文名字，因为训练数据集中包含了大量英文数据。这种偏见有时会导致生成有害或不恰当的输出。

例如，当给出“def race(x):”的代码提示时，AI辅助编程工具自动填写了一组有限且固定的种族类别，这明显反映了其训练数据中的偏见。

## 1.5 开发者的新世界

麦肯锡公司的研究深刻揭示了AI辅助编程工具对软件开发领域的潜在变革。软件开发的成功将越来越依赖于完善的培训、最佳实践的强

调，以及对提示工程、编程标准和质量控制的深入实践。同时，探索并了解生成式AI相关风险也是明智的选择。

对于新手开发者，尤其是工作经验尚浅的群体，建议深入学习编程基础原理的进阶课程，以提升工作效率。在整合AI辅助编程工具至日常工作流程时，得到团队资深专家的指导，积极参与社区交流，保持持续学习的状态至关重要。这包括在专业的在线论坛上交流心得，定期举行团队会议以分享实际案例等。这些活动不仅能够培养团队中持续学习的氛围，还有助于传播最佳实践，并在早期阶段发现潜在问题。

随着开发者工作效率的显著提升，管理者可以灵活调整角色分配，使团队专注于更具价值的工作。同时，技能提升也应纳入日常工作计划，以填补可能存在的知识空白。

需要明确的是，上述建议并非硬性准则。鉴于AI辅助编程领域仍处在快速发展阶段，我们应保持灵活性和开放心态，随时准备应对可能出现的挑战。

### 1.5.1 职业发展

虽然没有确凿证据证明使用AI辅助编程会提升你的职业前景，但一些迹象表明，以下可能会成为就业市场的主要变化。

#### 招聘热潮中的AI技能需求

Indeed等网站的招聘栏开始热闹起来，越来越多的招聘者开始寻找具有AI辅助编程工具经验的求职者。从初级开发者到高级热门人才，所有级别的职位都在招贤纳士。

#### 提升效率，助力职业发展

AI辅助编程工具之所以引起人们的关注，是因为它们在不牺牲软件质量的前提下提升生产效率。对于开发者，这可能是在职场中晋升的一条途径。

## 开发者的广泛认可

开发者普遍认为，AI辅助编程工具已经开始流行。例如，GitHub Copilot在独立软件评测网站G2.com上获得了4.5星（满分5星）的高分。

### 1.5.2 10倍开发者

10倍开发者意味着拥有10个开发者的能力。AI辅助编程工具就像编程界的Usain Bolt<sup>注</sup>，在你说出“修复错误”之前，就能迅速解决问题并给出解决方案。

所以你可能会想：在AI辅助编程工具的帮助下，我能成为10倍开发者吗？很抱歉，可能不会。虽然人工智能技术带来了显著的变化，但它的改进通常不是以数量级为单位的。

此外，“10倍开发者”的概念会激起刻板印象和偏见，让人觉得技术圈是个排外俱乐部。更不用说，成为超级开发者的压力可能会让你直接陷入职业倦怠。因此，虽然成为10倍开发者听起来很棒，但请记住，这可能是一厢情愿的幻想。

### 1.5.3 开发者的技能

根据麦肯锡公司的研究，AI辅助编程工具的有效性往往取决于开发者掌握的专业知识。以下是一些考虑因素。

#### 错误修复

尽管AI生成器可以成为你信赖的伙伴，但它也会出错。此时，发现并纠正错误的担子落在开发者的肩上。一些开发者发现自己需要对AI的结果一轮又一轮地进行修正，以达到最佳的精确度；而另一些开发者则不得不对AI进行“灌输”，让它准确地进行调试。错误修复无疑会耗费大量时间。但经验丰富的开发者知道如何避免走入“兔子洞”。

## 理解组织的要求

AI辅助编程工具在编程方面相当可靠，但它们在处理个别项目的独特性或满足公司的特殊需求时可能会失误。这正是资深开发者的价值所在。他们知道如何引导这些工具获得最符合组织目标、性能目标和安全性的结果。

## 解决棘手问题

AI辅助编程工具可以很好地完成打磨代码等任务，但如果遇到一些复杂的挑战，比如混合不同的编程框架，它们可能就会被自己绊倒。在这种情况下，经验丰富的开发者就需要登场了。

## 1.6 小结

AI辅助编程工具无疑是软件开发领域的一颗璀璨新星。随着人工智能技术的飞速发展，这些工具将极大地提升编程效率，自动处理烦琐的任务，让开发者能够聚焦于更高层次的问题解决和创新。

然而，它们的诞生也伴随着一系列挑战：错综复杂的知识产权问题、开放源代码软件许可的迷宫、潜在的偏见风险及安全隐患等。

在大多数情况下，AI辅助编程工具应被视为我们的得力助手，而不是我们自身知识、技能和经验的替代品。虽然它们可能并非无所不能的“超级英雄”，但它们无疑已经成为开发者工具箱中不可或缺的强大补充。

**[1]**1美元约合7.26元人民币。——编辑注

**[2]**Usain Bolt（尤塞恩·博尔特），牙买加田径运动员，获得2008年北京奥运会男子100米、男子200米、男子4×100米3项赛事的金牌。——译者注

## 第2章

# AI辅助编程技术的工作原理

在本章中，我们将深入探索AI辅助编程工具的内在机制，揭开它们背后的神秘面纱。具体来说，本章将首先带领读者简要回顾这些工具的发展历史；然后探讨Transformer模型和LLM的工作原理，并借助OpenAI Playground进行实践演示；最后提供一系列实用的评估建议，帮助读者了解如何对LLM的性能进行全面评估。

通过本章的学习，读者将能够清晰地了解这项强大技术的优势和局限，从而在实际的软件开发项目中更明智、更有效地利用AI辅助编程工具。

## 2.1 AI辅助编程工具的主要功能

GitHub Copilot、Tabnine、CodiumAI和Amazon CodeWhisperer等AI辅助编程工具在市场上引起了热议。以上每种产品的制造商都试图标榜自己产品的特色。这些工具也有很多共同的常见功能，如表2-1所示。

表2-1 AI辅助编程工具共同的常见功能

功能	描述
代码提示	根据注释和文件上下文提供代码建议，推荐单行代码或整个函数
上下文感知的代码补全	根据全部或部分代码库提供上下文感知的代码补全和编程建议
自动生成测试	分析代码，生成有用的测试、绘制代码行为图并发现边缘情况，以确保交付前软件的可靠性
支持用户与IDE交互	当用户在IDE中输入代码时，系统会自动激活并提供指导；用户可通过聊天与代码互动
代码分析	分析代码片段、文档说明和注释，提供可靠的代码预测并标记可疑代码
缺陷检测和修复	识别代码中的潜在错误并提供修复方法
自动生成代码文档	自动添加文档说明并提高代码文档质量
常规任务自动化	为常规或耗时的任务、不熟悉的应用程序接口或SDK，以及文件操作和图像处理等其他常见场景创建代码
API和SDK用法优化	辅助正确有效地使用API和SDK
开源项目发现和归因	有助于发现和归因开源代码和库

AI辅助编程工具日新月异，表2-1中所列的功能并不是一成不变的。显然，AI辅助编程工具通过提供代码提示和上下文感知的代码补全功能，为开发者带来了极大的便利。

## 2.2 智能代码自动补全与上下文感知的代码补全

智能代码自动补全（也叫自动代码完成或IntelliSense）是许多IDE中的一个神奇功能。它会根据开发者输入的内容，实时地给出代码段的提示、填充和高亮显示以提供帮助。事实上，这项技术最早可以追溯到20世纪50年代末的拼写检查程序。

这项技术的突破发生在20世纪90年代中期。当时，Microsoft Visual Basic 5.0引入了实时建议和自动补全功能，主要侧重于基本语法和函数签名的辅助。这一创新不仅有效减少了代码错误，还极大地提升了开发者的工作效率。

你可能会问：像IntelliSense这样的工具，与AI辅助编程工具有何区别呢？毕竟IntelliSense已经具备一定的人工智能和机器学习技术基础。

两者之间存在一个显著的区别：AI辅助编程工具由生成式AI驱动。它们提供的不仅仅是代码，还有自助式文档、规划文件和帮助指南等。得益于生成式AI，这些工具能够根据给定的上下文，娴熟地生成、调整和理解类人文本（human-like text），从而在翻译、摘要、文本分析、主题建模和查询回答等领域展现出卓越的能力。使用这些工具，我们有时仿佛在与代码进行智能对话。以LLM为核心，它们可以捕捉到用户输入中的上下文和意图变化。

## 2.3 编译器与AI辅助编程工具

了解编译器的工作内容对熟悉AI辅助编程工具很有帮助。以下是编译器执行的主要步骤。

### 词法分析（分词）

编译器宛如一位语言教师，它会将用户的代码拆解成一个个token，以便进一步解析和理解其深层含义。

### 语法分析

在语法分析阶段，编译器会检查token是如何分组的。它不仅验证用户的代码是否由正确的命令构成，还会确保代码具有正确的结构。

### 语义分析（错误检查）

在语义分析阶段，编译器确保用户的代码在编程语言的上下文中具有意义。它不仅涉及语法的正确性，还关系到代码背后所承载语义的正确性。

## 中间代码生成

这是代码转换的起始阶段。编译器将高级代码翻译成中间代码。虽然中间代码还不是最终的机器代码，但已经初具雏形。

## 代码优化

在代码优化阶段，编译器就像代码的私人教练，负责让用户的代码更加精简与高效。它会调整中间代码，使其运行速度更快，占用空间更少。

## 机器代码生成

这是代码转换的终极阶段。编译器将优化后的中间代码进一步转换为CPU可以直接理解的机器代码或汇编代码。

## 链接和加载

链接常常被视为编译过程的一部分，负责将各种代码模块和库文件有机地组合成一个完整的可执行程序。加载则是将这个程序加载到内存中准备执行，让代码从静态的文本变为动态的、能够执行的任务。

AI辅助编程工具（如GitHub Copilot）的工作方式与编译器大相径庭。它们并不具备真正“理解”编程语言的能力，而是通过人工智能技术，基于海量的现有代码进行模式识别并提出代码建议。由于AI辅助编程工具是在玩概率游戏，因此它们提出的建议可能会有很大的差异。然后编译器会将这些代码编译成机器可以运行的程序。

LLM基于预测模式，而不是编译引擎。如果有些知识在训练中不常见，它们可能无法被捕捉到。所以，AI辅助编程工具生成的代码可能

会漏掉一些简单的元素，比如括号。此外，AI辅助编程工具也会令人迷失方向，它们可能会根据情况给复杂代码的建议。

在发现错误方面，AI辅助编程工具虽然普遍有效，但它们仍无法与编译器忍者般的查错技能相提并论。尽管如此，这些工具仍然很强大。例如，它们可以捕捉恼人的语法错误——分号缺失、函数名中有错别字、括号不匹配，并迅速提出正确的修正建议。

AI辅助编程工具还能帮助你避免常见的编程陷阱。它们可能提醒你在打开文件后正确关闭文件，或者建议你以更有效的方式循环使用数组。说到逻辑错误，AI辅助编程工具的洞察力令人惊讶。它们可能无法解决所有复杂问题，但它们往往能提出你可能未曾考虑过的替代方法或解决方案，引导你朝着正确的方向解决问题。

一言以蔽之，AI辅助编程工具使编程更顺畅，但它们无法取代编译器的全面检查和开发者的敏锐洞察力。

上述缺点真正凸显了将AI辅助编程工具的智能与编译器检查的彻底性，以及人类干预相结合的重要性。毕竟你不仅希望代码优秀，还希望它们准确无误。

## 2.4 能力等级

2023年10月，Sourcegraph的首席执行官兼联合创始人Quinn Slack分享了一篇颇有见地的文章。在这篇文章中，他深入探讨了像GitHub Copilot这样的AI辅助编程工具，并提出了一种新颖的思考模式，他称之为“AI代码层级”。Slack的分级框架为我们理解AI辅助编程工具的功能提供了一个清晰的视角，同时他还对那些销售AI辅助编程工具的公司所宣称的夸张说法进行了批判性的审视。图2-1显示了编程系统中AI能力的不同层级。

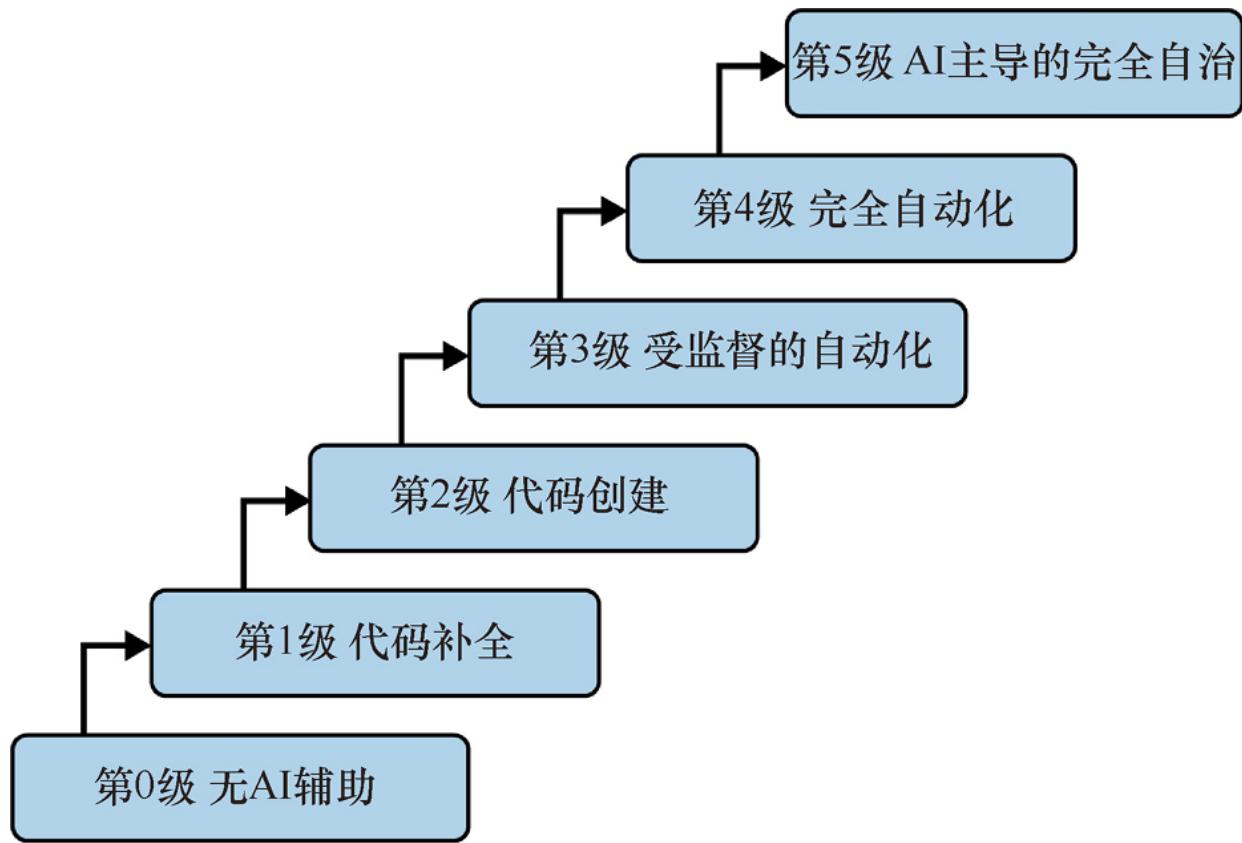


图2-1 编程系统中AI能力的不同层级

在编程的前3个层级中，重点是以人类为主导的编程过程，主要参与者是开发者。首先是第0级，开发者在没有AI的帮助下手工完成所有编程任务。第0级作为基线，为AI的介入奠定了基础。

然后是第1级，即代码补全阶段。在这个层级，AI开始介入并发挥作用，它根据开发环境中的事件来辅助完成单行或多行代码。尽管如此，开发者仍然占据主导地位，负责指导整个程序的开发，AI则作为提高编程效率、完成常规编程任务的一种工具。

在第2级，即代码创建阶段，AI的能力显著增强。此时，AI不仅能够编写更长的代码段，还能够完成更复杂的任务，如设计API，甚至修复现有代码。尽管如此，这些活动仍然需要人类的监督和指导。在这个层级，AI需要对代码库及其运行环境有一个深入的理解，才能生成既正确又适用的代码。

接下来进入第3级，也就是受监督的自动化阶段，我们可以看到AI在编程领域的重大转变。AI可以处理多项任务，实现人类设定的更广泛的目标，而无须在每次完成任务后都进行人工检查。使用这一层级的AI编程，就像将工作委托给一名初级开发者。AI具备足够的智能来排查错误、添加新功能，并将不同的系统集成，同时在这个过程中还会向开发者寻求解释和指导。

第4级标志着完全自动化的实现，AI在此层级真正地增强了自身的能力。它能够独立处理复杂的任务，而不需要人类对代码进行最终审批。想象一下，作为首席执行官或产品经理，你对顶尖工程师的信任程度。第4级所追求的正是这种信任关系。AI不只是被动响应，它还能够主动监控代码，发现并解决问题。

第5级是AI主导的完全自治。与前4个层级相比，在第5级，AI不仅遵循人类的命令，而且能够设定自己的目标。这些目标是基于AI的核心奖赏机制来确定的。可以想象成在一个充满竞争的世界中，AI在玩自己的游戏。虽然这个层级听起来有点像科幻小说，但考虑到技术发展的速度，不难想象，这个层级很可能会成为现实。

到目前为止，像GitHub Copilot这样的工具大致处于第3级。虽然确定其确切层级存在一定难度，但Slack的分级框架为我们提供了一个非常有用的参考，可以帮助我们理解技术及其关键交互。毫无疑问，AI编程技术并没有放慢脚步，而是在以惊人的速度发展。

## 2.5 生成式AI和LLM

采用AI辅助编程工具并不要求你成为生成式AI技术专家。但是，对这项技术有一个宏观的了解，将使你能够更敏锐地评估工具的响应速度、能力范围及潜在局限，进而更灵活和有效地利用这些工具。

在这里，“透明性”（transparency）并不仅仅是一个时髦的词汇，要让一项新技术得到广泛接受和应用，清楚地理解其工作原理至关重要

要。我们是否采纳这项技术，很大程度上取决于我们对它的信任程度。在编程领域，可靠性和责任感不仅仅是附加特性，更是基础，是编程工作的基石。

接下来将简要介绍生成式AI和LLM，以帮助你获得更清晰的认识。

### 2.5.1 技术演变

生成式AI的历史可以追溯到几十年前，早期的代表作之一是20世纪60年代中期由麻省理工学院的教授Joseph Weizenbaum开发的聊天机器人ELIZA。ELIZA的设计灵感源于心理治疗师的对话方式，它通过模仿用户的输入来运行，采用的是一种基于规则的算法。ELIZA的技术水平相对初级，至今你仍然可以在网络上找到它的踪迹。

有趣的是，许多人发现与ELIZA对话比与真正的心理治疗师交谈更加令人愉悦，甚至有人误认为自己正在与一个真实的人进行交流。这种现象被称为“ELIZA效应”，它揭示了人们如何轻松地赋予计算机程序人性化的理解能力。

然而，生成式AI的发展历程并非一帆风顺。它的核心技术进展缓慢，如同蜗牛爬行。直到进入21世纪第二个10年，这个领域才迎来一个转折点。得益于强大的计算能力、GPU

(Graphics Processing Unit，图形处理器)等先进的硬件系统、丰富的数据资源及深度学习模型的微调，生成式AI再次加速发展。随着技术的进步，生成式AI可以分为两大类——变分自编码器和生成对抗网络。

#### 变分自编码器

Diederik P. Kingma和Max Welling通过他们的开创性论文“Auto-Encoding Variational Bayes”在2013年首次提出了变分自编码器(Variational Auto-Encoder, VAE)的概念。VAE的核心是不需要监督的编码器-解码器架构，它能够从复杂、高维的数据中提取出低维的潜在空间表示。当讨论高维数据时，我们指的是那些具有大量特

征的数据集，每个特征都代表一个维度，例如一幅28像素×28像素的图像在784维空间中的表示。低维潜在空间可以看作数据的精简版本，它在保留关键信息的同时，去除了不必要的维度。这种降维技术至关重要，因为它不仅减轻了计算负担，还帮助我们规避了所谓的“维度诅咒”，使得数据更易于可视化和解释。从高维到低维的转换，本质上是对数据的一种简化，它将数据压缩到最基本的形式。与传统的自动编码器不同，在VAE中，编码器不是为每个潜在属性输出一个单一的值，而是输出一个概率分布。随后，解码器从这个概率分布中抽取样本来重建数据。这种在潜在空间中提供一系列可能的数据表示，并不是仅仅提供单一值的策略，而是为生成新的数据或图像提供了无限的可能性。

## 生成对抗网络

生成对抗网络（Generative Adversarial Network，GAN）是一种用于无监督机器学习的AI算法，由Ian Goodfellow及其团队在2014年首次提出。GAN的核心机制涉及两个相互竞争的神经网络——生成器和判别器，它们在一场类似游戏的对抗中相互较量。

生成器的任务是创造新的数据，这些数据看起来与真实数据无异。而判别器的角色则相当于裁判，其工作是区分生成器产生的假数据和真实世界中的数据。在每一轮的对抗中，生成器不断学习并提升自身，生成越来越逼真的数据样本。判别器也在不断进化，以更好地识别数据的真伪。

这种巧妙的设置不仅推动了生成器和判别器技术的进步，而且为人工智能领域带来了新的可能。如今，GAN已被广泛应用于创造逼真的图像、合成语音以及开发各种创新应用。

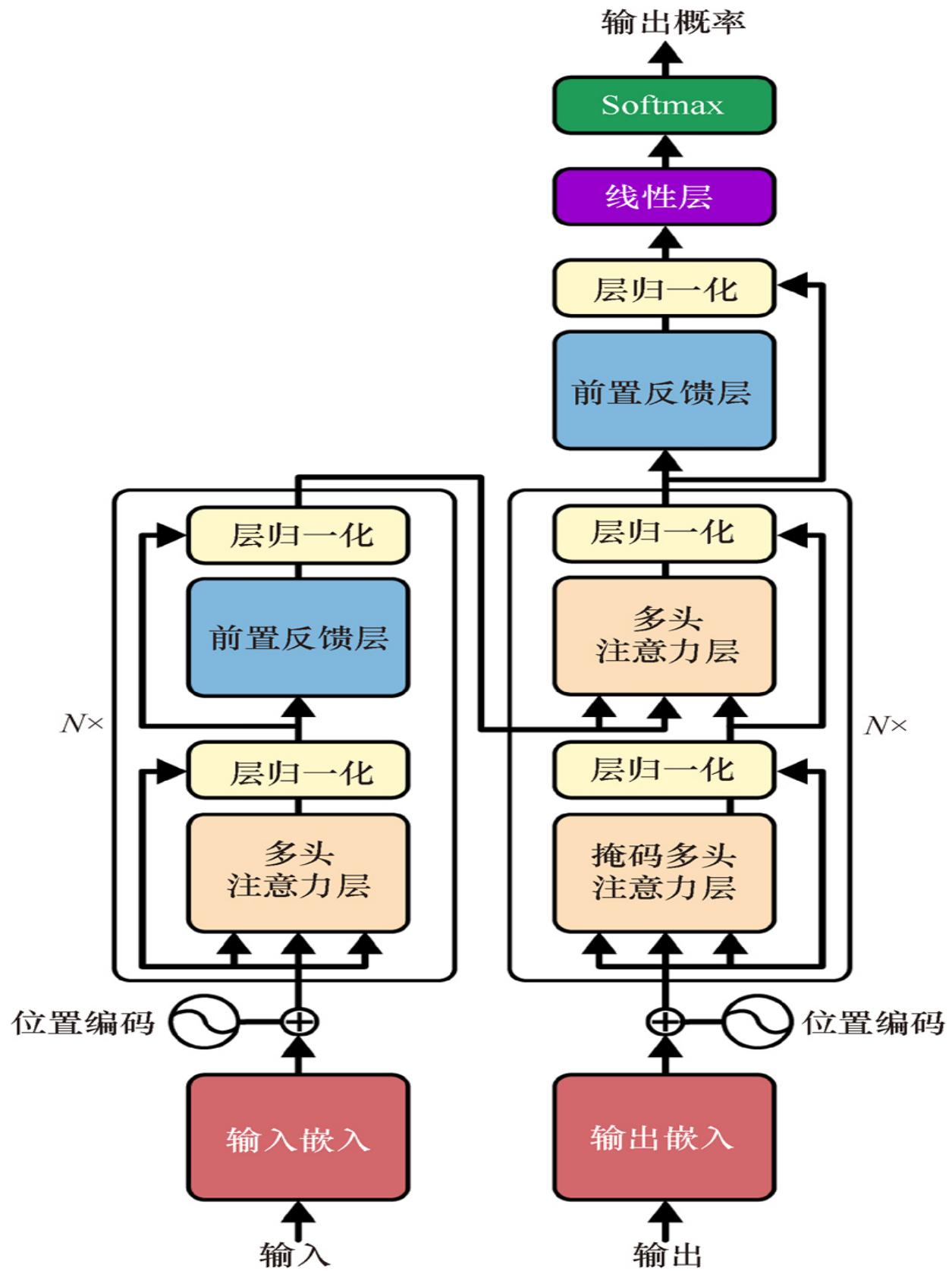
上述类型的生成式AI成为Transformer模型的重要构建模块，这是一次真正意义上的突破，使LLM成为现实。

### 2.5.2 Transformer模型

在Transformer模型成为主流之前，自然语言处理（Natural Language Processing，NLP）的首选技术是循环神经网络（Recurrent Neural Network，RNN）。循环神经网络专为处理序列或时间序列数据而设计。它通过更新隐藏状态来记忆序列中先前步骤的输出，这使得它在语言建模、语音识别和情感分析等任务中非常有用。循环神经网络一次处理序列中的一个片段，根据当前输入和之前处理过的内容更新隐藏状态。但循环神经网络在面对长序列时会出现梯度消失或梯度爆炸的问题。因此，循环神经网络很难跟踪数据中的长期关系。

Transformer模型的出现彻底颠覆了这一局面。Transformer模型不采用循环神经网络的逐步处理方法，而是并行地、轻量级地处理数据，并通过注意力机制来追踪输入序列中不同位置之间的关联，而无论这些位置相隔多远。这种架构使得Transformer模型能够轻松处理长短期序列，同时避免出现梯度问题。此外，Transformer模型的并行处理能力与GPU或TPU（Tensor Processing Unit，张量处理器）等现代计算硬件架构完美契合。

由Ashish Vaswani和他带领的Google研究团队开发的Transformer模型的核心架构，在2017年发表的开创性论文“Attention Is All You Need”中公布。图2-2展示了Transformer模型的主要部分。



## 图2-2 Transformer模型的主要部分

Transformer模型可以视作一位精通语言的大师，擅长揭示语言的复杂性。它的工作主要分为两个阶段——编码和解码。每个阶段都由多层网络构成。在编码阶段，Transformer模型就像理解外语句子的语言学家一样，读取并理解输入文本的含义。在解码阶段，Transformer模型利用编码阶段获得的知识生成新的文本或翻译，类似于语言学家将外语句子译成母语。

Transformer模型的关键在于注意力机制，该机制能够评估句子中每个单词与其他单词的相关性，并为每个单词分配相应的注意力权重。例如，在句子“The cat sat on the mat”中，当模型聚焦于单词“sat”时，“cat”和“mat”可能会获得更高的注意力权重，因为它们与动作“sitting”直接相关。

Transformer模型的一个显著特性是自注意力机制，该机制使得模型能够审视整个句子，理解单词之间的相互关系，并在处理长文本时保持这种理解。这为Transformer模型提供了一种长期记忆的能力，使其能够集中关注所有出现过的单词或标记（即整个单词或单词的一部分），从而更全面地理解上下文。

尽管具备这些能力，但Transformer模型最初并不擅长识别单词在句子中的顺序，而这却是理解句子意义的关键。位置编码（positional encoding）显得至关重要。它相当于为模型提供了一个GPS（Global Positioning System，全球定位系统），指示每个单词在句子中的位置，帮助模型区分类似“The cat chases the mouse”和“The mouse chases the cat”这样的句子。

Transformer模型通过引入更为复杂的多头注意力机制，进一步增强了其处理能力。可以想象，该模型拥有多双“眼睛”，每双“眼睛”都能从独特的视角审视信息，关注不同的方面或词语间的关系。比如，一组注意力可能专注于理解动作，另一组可能聚焦于识别主体，还有一

组可能致力于捕捉地点信息。这种多角度的分析方法赋予了Transformer模型对文本更深层次的理解。

Transformer模型常以预训练模型的形式出现。由于这类模型已经在大规模数据集上完成训练，因此可以立即投入使用或进行进一步的微调。预训练完成后，模型可以通过API被轻松调用，快速集成到各种语言处理任务中。企业和个人可以迅速将Transformer模型集成到自己的系统中，比如集成到AI辅助编程应用中。此外，通过在特定领域的数据集上进行微调，这些经过预训练的LLM可以进一步优化，从而在专业领域（如医疗或法律文本分析）里发挥出色的性能。这种方式避免了从头开始开发复杂的语言模型，节省了大量时间、精力和资源。预训练模型所具备的基础语言理解能力，为开发生成式AI应用提供了一个强有力的观点。



构建和运营LLM的成本是相当高的。根据《华尔街日报》在2023年初的报道，每位用户每月在GitHub Copilot上的花销超过20美元。在一些特定情况下，某些用户每月的花销甚至高达80美元。然而，未来几年随着生成式AI基础设施的规模化发展，预期每位用户的成本将会逐渐降低。

Transformer模型主要分为两大类——GPT（Generative Pretrained Transformer，生成式预训练Transformer）和BERT（Bidirectional Encoder Representations from Transformer，来自Transformer的双向编码器表示）。GPT是由OpenAI开发的一款工具，在文本创作、信息总结和语言转换方面表现出色。GPT基于自回归的LLM架构，这意味着在生成文本时，它会根据已经生成的内容逐字考虑，就像讲故事的人逐字逐句地展开叙述一样。GPT的技能是通过在大量文本数据上进行训练而获得的，它使用解码器来生成内容。

在AI辅助编程应用领域，用到的Transformer模型主要是GPT。实践证明，GPT能够根据开发者提供的上下文信息，有效地预测并自动完成代码，这极大地提升了编程效率。

### 2.5.3 OpenAI Playground

OpenAI Playground是一个生成式AI沙盒，可用于访问OpenAI开发的各种模型。它配备了一个用户友好的图形界面，使用户能够轻松定制模型。

OpenAI Playground极大地简化了我们理解和评估各种LLM的过程，让我们能够快速把握它们的性能和局限。此外，它还允许用户根据不同的参数输入（如温度）实时测试和调整模型的行为。

不过，OpenAI对GPT的使用收取费用。如表2-2所示，费用以token为单位，根据使用量来计算。需要注意的是，价格会定期调整。好消息是，相关费用一直在下降。

表2-2 GPT收费标准

模型	输入费用	输出费用
GPT-4/8K上下文	1000个token 0.03美元	1000个token 0.06美元
GPT-4/32K上下文	1000个token 0.06美元	1000个token 0.12美元
GPT-3.5-Turbo/4K上下文	1000个token 0.0015美元	1000个token 0.002美元
GPT-3.5-Turbo/16K上下文	1000个token 0.003美元	1000个token 0.004美元

例如，假设你使用的是 GPT-4/8K 上下文LLM。你输入的提示包含 1000个token，而模型生成的响应包含2000个token。那么输入费用为0.03美元，输出费用为0.12美元。

首次注册OpenAI账户时，你将获得5美元的积分，可用于 OpenAI Playground，或者用于支付API调用的费用。

## 1. token

接下来我们详细了解一下token。OpenAI提供了一个名为Tokenizer 的工具，如图2-3所示。该工具可以分析我们输入的如下内容。



输入：ChatGPT is unbelievable! I love it.

GPT-3.5 & GPT-4    GPT-3 (Legacy)

ChatGPT is unbelievable! 🎉 I love it.

**Clear**   **Show example**

<b>Tokens</b>	<b>Characters</b>
13	38

ChatGPT is unbelievable! 🎉 I love it.

**Text**   **Token IDs**

图2-3 Tokenizer显示文本的token数量

在分词 ( tokenization ) 过程中，“ChatGPT”这个短语被分成3个 token。具体来说，“Chat”“G”和“PT”中的每个都单独作为一个 token。单词“unbelievable”及其后面的感叹号被分为两个token ——一个表示单词本身，另一个表示标点符号。至于表情符号，则被分成3个token。每个标点符号各占一个token。空格的token则被合并到相邻的单词中。

Tokenizer适用于GPT-3、GPT-3.5和GPT-4。需要注意的是，不同的LLM通常采用不同的分词方法。这意味着相同的输入文本在不同的模型中可能会被分解成不同数量或类型的token。



根据经验，1000个token大致等同于750个单词。

## 2. 使用OpenAI Playground

进入OpenAI Playground后，你将看到图2-4所示的页面。页面中间是我们与LLM交互的主要工作区。

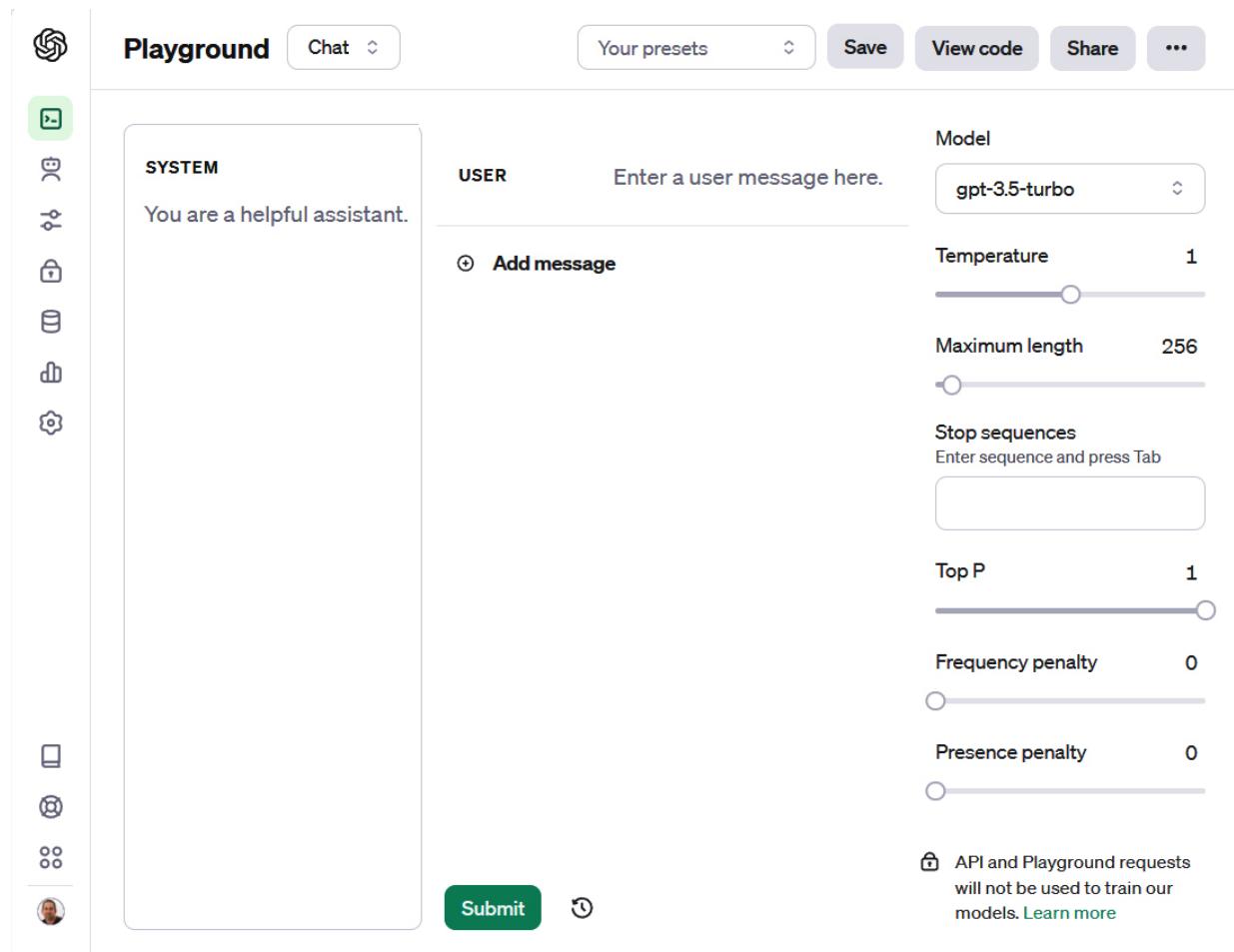


图2-4 包含提示、资源和互动区域的OpenAI Playground首页

## SYSTEM (系统提示)

这是为LLM提供背景信息的地方，例如“你是Python编程方面的专家”。系统提示是会话中的第一条信息，旨在为交互奠定基础。自定义系统提示可以更好地控制模型在对话中的行为，对于确保模型保持在期望的参数或上下文范围内特别有用。

## USER (用户提示)

这是提示的主要指令。例如，你可以在此处要求LLM执行编程任务。

## Add message (添加消息)

在这里，你可以与LLM持续聊天。

假设你正在开发一个Python项目，却不确定如何利用Tkinter（库）来捕获用户输入。在这种情况下，你可以这样询问。

**系统**：你是擅长Tkinter的Python专家。

**用户**：我想用Tkinter创建一个简单的图形用户界面来获取用户的姓名和年龄。请问如何做？

LLM将为你生成相应的代码示例。假设你不仅需要获取输入，还想对输入进行验证，此时你可以进一步询问：“我如何确保输入的年龄是一个数字而不是文本？”LLM将提供相应的代码解决方案，其中可能包括使用try-except语句块来尝试将输入的年龄转换为整数，并处理可能出现的异常。

整个过程与使用ChatGPT类似，但提供了更加结构化和专业的体验。真正强大的是其自定义功能，这些功能可以在图2-4所示页面的右侧找到。

## **Model (模型)**

你可以选择不同的模型，甚至可以使用自己微调的LLM，以确保模型专注于你独特的编程需求。有关微调模型的更多信息，请参阅OpenAI的API文档。

## **Temperature (温度值)**

温度值可以调整所生成内容的随机性或创造性。数值范围为0 ~ 2，数值越低，回答越确定和集中。表2-3列出了不同类型编程任务的建议温度值。

表2-3 不同类型编程任务的建议温度值

任务类型	温度值	描述
代码生成	0.2~0.3	确保代码更加确定、准确，并遵循通用惯例，从而获得可靠、可理解的结果
代码审查	0.2或更低	注重既定的最佳实践和标准，实现精确反馈
错误修复	0.2或更低	为发现的问题提供更准确、更直接的解决方案
创造性地解决问题	0.7~1.0	探索更多可能的解决方案，有助于头脑风暴或创新性地解决问题
学习与实验	0.7~1.0	提供更广泛的示例和解决方案，了解解决问题的不同方法
数据分析和可视化	0.2或更低	生成准确、有意义的可视化或分析结果
任务优化	视情况而定	允许在探索（较高温度值）和开发（较低温度值）之间取得平衡，以获得高效解决方案

但是，如果使用的温度值过高，结果就会变得毫无意义。下面是使用2作为温度值时的提示示例。

**提示：**在Python中，将数据从CSV文件迁移到MySQL数据库的步骤是什么？

如你所见，图2-5展示的输出结果并不具有实际意义。

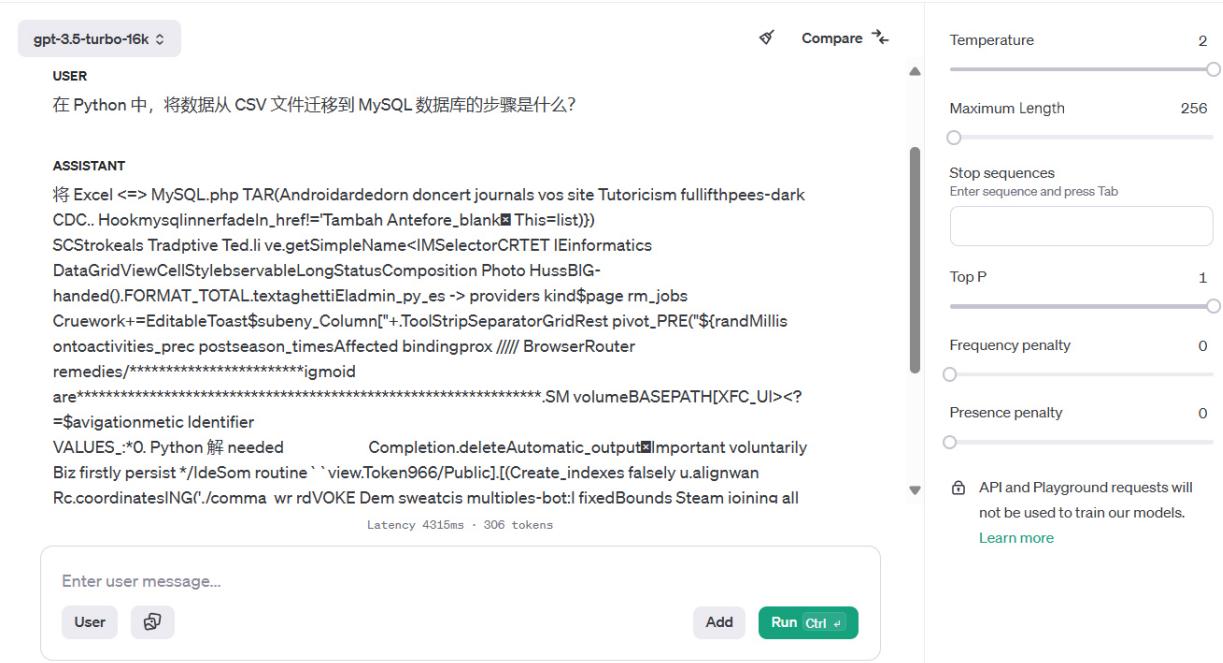


图2-5 温度值为2时，LLM输出的结果大多毫无意义

## Maximum Length (最大长度)

用于设置生成内容时所允许的最大token数量。这个限额包括了提示 (prompt) 和回复 (response) 的token数量总和。内容生成与 token之间的比例关系，取决于选用的具体模型。

## Stop sequence (终止序列)

终止序列是一种信号，用于指示LLM应在何时停止继续生成文本。用户可以设定一个特定的字符串或字符组合，当在生成的文本中遇到这一特定的序列时，模型便会接收到停止生成的命令。

## Top P

核采样是一种根据累积概率阈值（用变量 $p$ 表示，取值范围为0~1）来选择单词的方法。简单来说，LLM在生成文本时，并不是仅仅从概率最高的几个单词中选取，而是根据设定的 $p$ 值考虑一个更宽泛或更狭窄的单词范围。当 $p$ 值设定得较小时，模型可以选择的单词范围较小，

更集中，这有助于生成更可预测且连贯的文本。相反，当 $p$ 值设定得较大时，模型考虑的潜在单词范围更广，能够生成更多样化、更具创造性的文本。

### **Frequency penalty (频率惩罚)**

频率惩罚有助于解决LLM的一个常见问题——重复的短语或句子。取值范围为0~2，值越大，重复的短语或句子越少。但是，当值大于1时，文本生成的结果会变得难以预测，甚至变得没有意义。

### **Presence penalty (存在惩罚)**

存在惩罚的取值范围也是0~2，值越大，LLM包含的分词种类越多，也意味着涉及更广泛的词汇或概念。

凭借频率惩罚、存在惩罚和核采样等工具，OpenAI建议基于实验的方法来选择和调整适合特定任务的策略。鉴于LLM的复杂性，寻找最优解通常没有固定的公式或规则。

## **2.6 评估LLM**

评估LLM是一项极具挑战性的任务。这些复杂的系统往往缺乏透明度，使得外界难以深入了解其工作原理。人工智能领域公司间的竞争进一步加剧了这种不透明性。关于模型训练所用的数据集、微调行为的参数量，以及支持模型运行的硬件设备的具体信息，正变得日益稀缺。

然而，斯坦福大学的研究人员带来了一线希望。他们开发了一个名为Foundation Model Transparency Index（基础模型透明度指数）的评分系统，旨在衡量LLM的透明度。该系统基于100条标准构建，目的是澄清LLM透明度方面的混乱现状。

排名采用百分比形式。表2-4展示了具体的排名情况。遗憾的是，结果并不令人鼓舞。研究人员认为，目前还没有任何一个主流的LLM达到“足够的透明度”，它们的平均透明度仅为37%。

表2-4 主流LLM的透明度<sup>注</sup>

公司	模型	透明度/%
Meta	LLaMa 2	54
BigSdence	BLOOMZ	53
OpenAI	GPT-4	48
Stability.ai	Stable Diffusion 2	47
Google	PaLM 2	40
Anthropic	Claude 2	36
Cohere	Command	34
AI21Labs	Jurassic-2	25
Inflection	Inflection-1	21
Amazon	Titan Text	12

LLM的一个显著优势在于其处理多领域任务的灵活性（例如软件开发），但这同时也增加了评估它们的复杂性。对这些模型的评估需要依赖特定领域的评估指标和基准测试，以确保模型在每个特定应用场景中都能表现出有效性和安全性。

尽管存在这些挑战，但在评估LLM时，仍须考虑以下几个关键的衡量标准。

## BERTScore

BERTScore是一种评估文本生成模型性能的指标，它通过对生成的文本与参考文本，利用预训练的语言模型BERT来进行计算。这种度量方法主要被用于对自然语言文本进行评估，但通过进行适当的扩展或调整，它同样可以用于代码生成任务，特别适合用在那些涉及自然语言注释或代码评论的场景中。

## 困惑度

困惑度（Perplexity）是评估概率模型（如LLM）性能的常用指标。它衡量模型预测的概率分布与实际数据分布之间的一致性。在代码生成的背景下，较低的困惑度意味着模型在预测代码序列中下一个token时具有更高的准确性和优势，表明模型对代码的语言模式有更深刻的理解。

## BLEU

BLEU（BiLingual Evaluation Understudy）最初是为机器翻译任务设计的，现在也被用在代码生成领域。它通过对生成的代码与参考代码进行比较来评估生成代码的质量。具体来说，BLEU通过计算 $n$ -gram的精确度分数来量化生成文本与参考文本之间的相似度，这有助于衡量生成代码的语法准确性。 $n$ -gram的精确度分数越高，说明生成的文本在特定长度的词序列上与参考文本的一致性越好，从而反映出生成代码的质量越高。

## **ROUGE**

ROUGE ( Recall-

Oriented Understudy for Gisting Evaluation )<sup>注</sup> 是从自然语言处理领域借鉴的一个指标，同样适用于评估代码生成模型的性能。

ROUGE主要通过计算生成文本和一组参考文本之间的*n*-gram重叠度来衡量它们的相似性。

## **MBXP**

MBXP ( Most Basic X Programming problems ) 基准测试是专门为评估多语言环境下代码生成模型的性能而设计的。它采用一个可扩展的转换框架，允许提示和测试用例从原始数据集被转换到目标编程语言。这一过程促使对代码生成模型进行全面的多语言评估，从而更准确地衡量模型在不同编程语言中的代码生成能力。

## **HumanEval**

HumanEval是一个专门用于评估LLM代码生成能力的基准测试，它通过衡量模型根据文档字符串合成程序时的准确性来进行评估。这个基准测试对于持续开发和提升AI模型在代码生成方面的能力起到至关重要的作用。尽管不同模型在HumanEval上的表现各异，展示了它们各自的熟练程度，但名为HUMANEVAL+的扩展版本在识别那些由主流LLM生成，但之前未被探测到的错误代码方面发挥了关键作用。

## **Multilingual HumanEval**

Multilingual HumanEval ( HumanEval-X ) 是对原始 HumanEval基准的扩展与升级。这个基准测试旨在全面评估LLM在代码生成和翻译方面的能力，覆盖了超过10种不同的编程语言。

HumanEval-X引入了一个转换框架，该框架能够将Python语言中的提示和测试用例适配并转换成目标编程语言的对应格式。这样的设计为评估多语言环境下的代码生成和翻译提供了一个更为全面的基准。

当评估LLM时，一种常见的方法是考查其参数量，这个数字有时可达到数千亿。但参数越多越好呢？答案并非如此简单。评估这些模型应该采用更为细致和全面的方法。

首先，模型参数的扩展在计算能力和能源消耗方面成本极高，这可能导致LLM在实际应用和商业化方面变得不经济。其次，随着参数数量的增加，模型的复杂性也随之增长，这可能导致出现过拟合现象。过拟合往往发生在模型在训练数据上表现优异，但在未见过的数据上表现不佳的情况下，这会削弱模型的泛化能力。

另外，我们需要大量且多样化的训练数据来满足模型对信息的庞大需求。然而，获取和处理如此庞大的数据集不仅需要大量资源，还可能引发数据隐私和偏差方面的担忧。此外，随着参数数量的增加，对模型的评估也变得更加复杂。评估指标需要更全面和多样化，以准确衡量模型在大量不同任务上的性能。

最后，微调是一种在不显著增加基础LLM参数规模的情况下，提升模型性能的有效方法。

## 2.7 LLM的类型

LLM有多种类型，其中备受瞩目的一类是开源LLM。这类模型可以自由使用、调整或共享。它们的开放透明让你能够洞察模型的运作方式。此外，开源LLM鼓励开发者协作创新，共同开发插件，以便及时修复那些烦人的错误。

开源LLM最吸引人的一点是什么？毫无疑问，它们是免费的。

然而，开源LLM并非完美无缺。它们可能没有专门的团队来解决问题、提供支持，以及定期推出更新。如果在使用过程中遇到难题，则你可能需要自己动手解决，或者在论坛上寻求社区的帮助。

开源模型的质量和性能可能会有波动，有时就像坐过山车一样起伏不定。此外，它们也可能带来一些安全方面的挑战。由于模型的开放性，黑客可能更容易找到植入恶意代码的途径。因此，使用开源模型时需要格外小心。

开源模型的用户指南和文档有时可能不够完善，甚至难以理解，让人感觉像是在解读象形文字。

表2-5列出了一些主流的开源LLM。

表2-5 主流的开源LLM

模型	公司	参数规模/B	值得关注的功能
GPT-NeoX-20B	EleutherAI	20	在ThePile数据集上训练，能够执行各种NLP任务，例如故事生成、聊天机器人和摘要等
LLaMA 2	Meta	7~70	使用2万亿个token进行训练，上下文长度是LLaMA 1的两倍
OPT-175B	Meta	175	一系列模型的一部分，训练时碳足迹低于GPT-3
BLOOM	BigScience	176	在ROOTS语料库上进行训练，公开了训练数据的详细信息和评估方法，以实现透明度
Falcon-40B	Technology Innovation Institute (TII)	40	使用1000B token进行训练
Dolly 2.0	Databricks	12	基于EleutherAI的Pythia模型系列，提供类似ChatGPT的指令跟随交互性
Mistral 7B	Mistral AI	7.3	使用分组查询和滑动窗口注意力机制；经过大量数据集的训练，擅长处理较长的序列
Mistral 8X7B	Mistral AI	46.7	稀疏混合专家模型；像12.9B模型一样进行推理，支持多种语言，在代码生成和推理等任务中表现出色

注：在LLM中，在提到参数规模时，一般用千参数（K）、百万参数（M）、10亿参数（B）表示。

另一类LLM是闭源或专有LLM，它们的保密性更好，通常将代码、训练数据和模型架构视为高度机密。开发这些复杂系统的公司往往拥有雄厚的财力支持。表2-6展示了这些公司在2023年筹集的风险投资金额。

表2-6 顶级LLM公司在2023年筹集的风险投资金额

公司	风险投资金额/ 亿美元
Anthropic	12.5
OpenAI	100
Cohere	2.7
Inflection AI	13

有了充足的资金作为后盾，这些公司便能够招募到世界上最杰出的数据科学家，来打造尖端的基础设施。这使得他们开发的LLM在性能上往往处于行业领先地位，代表着业界最前沿的技术。同时，这些模型也是为了满足规模化运营和企业级严格标准而设计的，尤其是在安全性和隐私保护方面。

然而，闭源模型也存在一些缺点，其中最突出的是信任问题。模型是如何生成响应的？它们如何处理幻觉和偏见？目前，这些问题的答案往往不够透明和详尽。

此外，大型AI服务提供商可能会形成垄断，这导致客户有可能会被限制在一个特定的生态系统内，缺乏灵活性和选择权。

与开源LLM相比，闭源LLM可能更容易陷入创新停滞的状态。这是因为它们无法像开源LLM那样从广泛的社区输入和持续的审查中获益。

## 2.8 评估AI辅助编程工具

选择合适的AI辅助编程工具可能是一项令人头疼的任务。在这个过程中，需要考虑多种因素，包括工具的准确性、交互性、安全性、速度和易用性。有时，选择工具可能归结为寻找一个最适合你的工作方式的选项。然而，如果你的雇主或组织坚持使用特定的工具，那么你的选择可能会受到限制。

Stack Overflow的2023年《开发者调查》是一份快速了解当前热门工具的宝贵资源。该调查汇集了近90000名开发者对当前流行的AI辅助编程工具的看法和偏好，如表2-7所示。

表2-7 流行的AI辅助编程工具排名

AI辅助编程工具	使用比例/%
GitHub Copilot	54.77
Tabnine	12.88
Amazon CodeWhisperer	5.14
Snyk Code	1.33
Codeium	1.25
Wispr AI	1.13
Replit Ghostwriter	0.83
Mintlify	0.52
Adrenaline	0.43
Rubberduck AI	0.37

通过表2-7，你可以对众多可用的AI辅助编程工具有一个全面的了解。在选择特定工具时，一个明智的策略是查看其他开发者是否给予该工具正面的推荐。此外，亲自尝试几款不同的工具也是一个好主意。值得庆幸的是，上述大多数工具提供免费试用版本，这让你有机会先行测试，而不用立即做出全面投入的决定。

另一个需要考虑的重要因素是公司是否拥有稳固的财务支持。你所在的公司是否有进行风险投资的资金支持？资金的充足与否可能直接影响到公司的发展和平台的持续创新能力。如果资金不足，公司可能难以维继，甚至面临停止服务的风险。

一些AI辅助编程工具的供应商可能不得不终止服务，这会给依赖这些工具的开发者带来不便。以Kite为例，它曾是这个领域的先驱之一，自2014年起开始运营。然而到了2022年，Kite公司决定终止该项目。幸运的是，Kite将其大部分代码库开源，为社区提供了持续使用和进一步开发该工具的可能。

## 2.9 小结

在本章中，我们揭开了生成式AI和LLM的神秘面纱。我们回顾了一些引人入胜的历史案例，例如早期的聊天机器人ELIZA。接着，我们深入探讨了人工智能领域的一个重要里程碑——Transformer模型。此外，我们还亲自体验了OpenAI Playground，并学习了如何根据需要定制LLM。

**[3]**根据基础模型研究中心发布的“*Foundation Model Transparency Index Total Scores 2023*”（2023年基础模型透明度指数总分）得到。

**[4]**ROUGE特别关注召回率，即生成文本中包含参考文本的程度。通过这种方式，ROUGE有助于深入了解生成代码与预期输出之间的一致性和准确性。ROUGE分数高表明生成的代码在结构和内容上与参考代码有较高的匹配度，这对于验证代码生成模型的有效性至关重要。——译者注

## 第3章

# 提示工程

提示工程（Prompt Engineering）是机器学习和自然语言处理领域的一个重要分支，它专注于研究如何让计算机更准确地理解和解释人类的语言。这个学科的核心目标是探索与LLM和复杂AI系统有效对话的方法，以便它们能够生成我们所需的答案。这里所指的模型是专门用来处理和生成类似于人类语言的响应的模型。

正如在向他人寻求建议时需要提供背景信息和明确的请求一样，LLM同样需要这样的上下文。设计问题或提示时需要精心构思，有时甚至需要在问题中加入提示或附加信息，以确保LLM能够准确理解你的需求。

向LLM提问并不总是一次性的，有时更像是与LLM进行一系列对话，通过不断交流和调整问题，最终获得有价值的信息。

假设你正在使用AI辅助编程工具开发一个Web应用程序。一开始，你可能会询问如何用JavaScript创建一个简单的用户登录系统。初次的回答中可能包含基础用法，但随着对更高级的功能提出需求，你随后给出了更具体的提示，比如询问如何实现密码加密和安全的数据库连接。每一次与AI辅助编程工具的互动都是对其响应的细化，以使其更好地满足项目的特定需求。

需要注意的是，提示工程师已经成为一个非常受欢迎的职业。根据Willis Towers Watson提供的数据，提示工程师的平均年收入约为13万美元，尽管这个数字可能偏于保守。为了吸引顶尖人才，雇主通常会提供十分吸引人的期权和奖金。

在本章中，我们将深入探讨提示工程，并介绍一些实用的策略和技巧，帮助你更有效地与LLM进行交流。

## 3.1 艺术与科学

提示工程是一门结合了艺术与科学的技艺。一方面，它要求你精心挑选合适的词汇和语调，以引导AI按照你的意图做出反应，这是在塑造对话的流向。你需要一点直觉和创造性思维，来精炼自己的语言并激发出细致的回复。

对于软件开发者，这样的要求可能显得颇具挑战性。传统上，开发者按照一套规则编写代码，结果要么成功，要么由编译器明确指出错误所在。这是一个逻辑性强、可预测的过程。

但是，提示工程则显得更加自由和不可预测。

另一方面，提示工程同样包含大量的科学元素。正如第2章所讨论的，你需要深入理解AI模型的工作原理和细节。除了创造力以外，你还需要精确性、可预测性，以及能够重现结果的能力。这意味着你必须进行实验，尝试不同的提示，分析结果，并不断调整，直至获得期望的响应。

对于提示工程，不要期待会有一个一劳永逸的神奇解决方案。市面上的许多课程、视频和图书声称揭示了提示工程的“秘诀”。对此，你需要保持谨慎，否则可能会感到失望。

此外，人工智能和机器学习正在不断进化，新的模型和技术层出不穷。因此，在提示工程中，期待有一种固定不变的技术显然是不现实的。技术在不断变化，我们也要不断学习和适应。

## 3.2 挑战

有时提示工程可能会令人感到沮丧。尽管LLM采用了先进的基于概率的框架，但即使是对提示语措辞的微小调整，也可能显著改变模型的输出结果。

以下是提示工程面临的一些挑战。

### 啰嗦

有时LLM可能会过于健谈。你给出一个提示，它们会提供一段详尽的回答，而实际上你可能只需要一个简短的答案。但它们往往會附带大量相关信息或事实，导致回答过于冗长。如果希望LLM直截了当地回答，你可以要求它们“简明扼要”。

### 不可移植

在一个LLM中表现出色的提示，在另一个LLM中可能效果不佳。换句话说，如果想要从ChatGPT切换到Gemini或GitHub Copilot，则可能需要调整提示，因为每个LLM都有自己独特的训练、设计和专长。不同的模型是在不同的数据集和算法上训练出来的，对提示的理解和解释也各不相同。

### 长度敏感

LLM可能会对过长的提示感到困惑，开始忽略或误解你输入的部分内容。这就好比模型的注意力开始分散，它们的回答也会变得含糊不清。因此，应避免在提示中提供过多的细节，并尽量将提示内容限制在一页以内。

### 歧义

如果你的提示不够明确，LLM可能会感到困惑，并提供与事实不符或完全虚构的回答。因此，提供清晰的提示至关重要。

尽管存在这些挑战，但仍有一些方法可以改善结果。本章的后续部分将介绍一些改善提示效果的策略和技巧。

## 3.3 提示

提示（prompt）主要由4个部分组成，如图3-1所示。

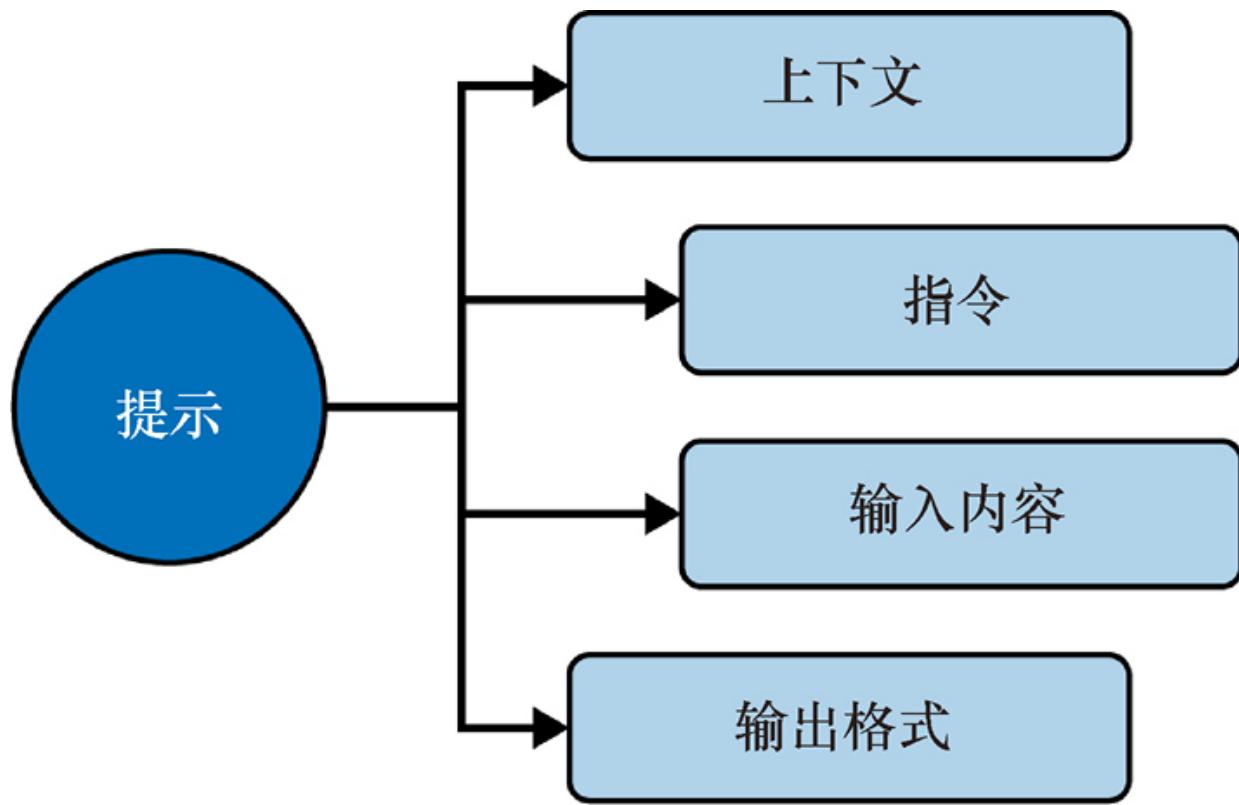


图3-1 提示的4个主要组成部分

在构建有效的提示时，上下文是关键，它定义了LLM在回答问题时所扮演的角色。紧接着是具体的指令，比如要求模型进行总结、翻译或分类。为了使LLM能够更准确地处理并给出恰当的回答，你需要提供相关的输入内容。此外，明确指出你期望的输出格式也很重要。

实际上，并不是上述所有组件（即提示的4个主要组成部分）在每种情况下都是必需的。有时，你可能只提供一个组件——比如一个明确的指令——就能获得满意的答复。但作为一般规则，向LLM提供的细节越具体，得到的回答往往越精准。

接下来的内容将介绍上述每个组件。

### 3.4 上下文

在编写提示时，通常从简洁的一两句话开始，以便为模型提供必要的上下文。这通常包括指定LLM在回答时应该扮演的角色或身份。通过设定上下文，LLM不仅能够提供更精确的回答，而且这些回答将与上下文紧密相关，使得结果更富有意义且具有针对性。

比如，你想调试一段代码，此时可以提供以下上下文。

**提示：**你是一位专注于Java应用程序调试的 경험丰富的软件工程师。

再比如，你想学习某种算法的优化技术，此时可以提供以下上下文。

**提示：**你是一位擅长算法优化的资深软件开发人员。

提供清晰的上下文有助于引导LLM以合适的视角和专业知识来处理提示，从而以正确的方式给出回答。

### 3.5 指令

提示中至少应包含一个明确的指令。虽然可以添加很多指令，但应谨慎行事，因为过多的指令可能会使LLM感到困惑，从而难以获得所需的答案。

我们分析一下为什么会出现这种情况。首先，当存在多个指令时，情况可能变得复杂。如果指令不够清晰或相互矛盾，LLM可能会感到迷茫，不确定应优先考虑哪个指令，或者如何平衡各个指令。

其次，指令越多意味着LLM需要处理的信息量越大。LLM必须理解和解析提示的每个部分，然后确定如何综合这些部分，以形成一个连贯的回答。这不仅需要强大的处理能力，有时还可能导致错误或偏差。

最后，LLM会按顺序阅读指令。指令的排列方式会影响模型对提示的理解，进而影响得到的回答类型。

因此，专业建议是保持简单。与其一次性向LLM提出一系列复杂问题，不如尝试将问题拆分成一系列较小的提示。使查询过程更像是一次轻松的对话，而不是一段冗长的独白。

提示中的指令类型多种多样。接下来的内容将讨论软件开发中常用的一些指令类型。

### 3.5.1 摘要

一种摘要是浓缩，这种摘要可以将较长的文本浓缩成较短的形式，同时保留主要观点和要点。摘要对于快速掌握冗长的文档非常有用。对于软件开发者，在表3-1所列的情况下，摘要是一个特别方便的工具。

表3-1 编程任务的摘要提示

用例	描述	提示范例
代码文档	简明扼要地概述大量文件，重点介绍关键功能、依赖关系和结构	总结以下文档讨论的要点，快速概述代码库
缺陷报告	快速识别大量或冗长的缺陷报告中的主要问题	总结以下缺陷报告中报告的常见问题，找出需要解决的主要问题
研究报告	从冗长的研究论文或技术文章中提取简洁的见解，为用户提供最新的研究或技术信息	概述以下研究论文中讨论的关键结论和技术
变更日志	从冗长的变更日志中了解代码库或工具新版本中的关键变更	在以下1.1.2版本的变更日志中总结关键变更
电子邮件主题	从冗长的电子邮件主题中提取讨论或决策的要点	总结以下电子邮件讨论的要点

另一种摘要是主题建模，这种摘要通过统计模型辨识文档集中出现的抽象“主题”。以下是一些专为开发者设计的主题建模提示。

**提示：**请从以下文本中识别讨论的关键主题：{text}

**提示：**请从以下文本中提取关键词，以推断其主要话题：{text}

**提示：**请根据以下文本的内容，为其推荐标签：{text}

### 3.5.2 文本分类

文本分类是一项让计算机为文本分配标签的任务。情感分析是文本分类的一种应用，LLM能够识别社交媒体帖子中表达的是积极、消极还是中性的情感倾向。对于开发者，情感分析是一款宝贵的工具，可用于衡量用户对其开发的应用程序的反馈。

以下是一些实用的参考提示。

**提示：**请分析这些客户评论，并告诉我整体上是积极、消极还是中性的情感倾向？{text}

**提示：**这是我们用户论坛上讨论最新更新的一些帖子。你能为我总结整体的情感倾向吗？{text}

**提示：**我整理了来自应用商店页面的一系列反馈。你能根据其情感倾向对评论进行分类吗？{text}

**提示：**请评估这些关于我们产品公告的博客帖子评论的情感倾向。大家的共识是什么？{text}

### 3.5.3 建议

你可以指导LLM提供建议，这对于开发者是缩短工作流程的宝贵资源，无论是消除错误、优化代码还是更高效地使用API。

以下是一些实用的参考提示。

**提示：**调用以下代码片段时抛出NullPointerException异常。你能帮助我找出潜在原因并提出修复建议吗？

**提示：**这是我编写的一个函数，用于对整数列表进行排序。你能否提出一些优化建议，使其运行得更快或更具可读性？

LLM的建议可以成为你工作中的强大助力，它不仅能节省大量的时间，还能提供你可能未曾考虑到的想法，特别是在处理复杂或微妙的任务时。

然而，LLM并非无懈可击。一个潜在的问题是，模型的回答可能过于简化，忽略了某些细节。此外，由于模型的知识是截至某个时间点的，它可能无法掌握最新的信息或趋势。

总之，LLM的建议可以是开启一项工作的方式，但你需要自己深入研究和探索，以获得更全面的了解并加以掌握。

### 3.5.4 翻译

本地化本质上是使软件适应特定地区的语言和文化的过程。它让软件能够使用当地语言，并理解当地特有的行为习惯，是扩大市场并与目标受众建立紧密联系的关键。本地化带来的好处是显而易见的：用户感到更加满意，因为他们觉得软件是为他们量身定制的，而用户的满意度直接关系到企业利润的增长。

在竞争激烈的市场中，本地化可以为产品提供竞争优势，尤其当竞品在这方面存在不足或完全没有本地化时。此外，使软件适应当地的习俗和法规，不仅能让软件在市场中占有一席之地，往往还能使其成为当地用户唯一的选择。

然而，本地化并非没有挑战。它既昂贵又耗时，需要严格的质量保证来维持软件在不同语言中的一致性。与此同时，软件开发是一个持续的过程，本地化则是一个不断更新和增加新功能的循环，每个新功能都可能有自己的本地化需求。这种持续性增加了项目的复杂性和成本。

这正是LLM可以发挥作用的地方。这种先进的系统能够在多种语言之间进行翻译，成为开发者工具箱中的一款强大工具。表3-2列出了一些在本地化过程中可能会用到的语言翻译提示，以帮助开发者更高效地进行软件的本地化。

表3-2 语言翻译提示范例

任务类型	描述	提示范例
用户界面文本翻译	翻译按钮、菜单项、错误消息、对话框等	将以下用户界面文本翻译成法语：保存、退出、文件、编辑、帮助
文档翻译	翻译用户指南、帮助文件和其他文档	将用户手册中的以下段落翻译成西班牙语
错误消息翻译	翻译软件可能会生成的错误消息	将以下错误消息翻译成德语：找不到文件、拒绝访问、网络连接丢失
提示框的翻译	翻译当用户将鼠标指针悬停在某个选项时提示框中显示的附加信息	将以下工具提示翻译成日语：单击保存、单击打开新文件、单击打印

我们应当审慎地对待LLM的多语言能力，因为它们并非完美无缺。有时这些模型可能会忽略语言之间的微妙差异、习语表达和独特的文化背景。语言之间的细微差别极其复杂，直接翻译并不总是正确的方法；更重要的是要传达出正确的意义和语境。

处理特定术语或专有名称尤为棘手，特别是在目标语言中没有直接对应词汇的情况下。此外，保持正确的语调和风格也是一个挑战。这不仅仅是单词选择的问题，更关乎如何表达，而这在不同的语言和文化中可能大相径庭。

让语言专家审查LLM的输出结果可以避免许多潜在的问题，确保翻译的准确性和文化适应性。

## 3.6 输入内容

在构建提示时，使用####或""等特殊符号来明确区分指令和希望LLM学习的内容或信息，是一种非常有效的策略。这些符号充当边界或标记，旨在帮助LLM清晰地识别指令的结束点和内容的开始点。

例如，当软件开发者需要从长篇文档中提取有关正在集成的新API的核心要点时，可以采用以下提示结构。

**提示：**从下面的文本中提取出API的关键实施步骤。

文档 : """

{API文档内容}

"""

使用""分隔符将指令与API文档内容明确分开的方法既简洁又直观。它不仅能让LLM更清楚地理解任务要求，还提高了获得清晰、准确的步骤摘要的可能性。此外，分隔符有助于整理提示信息，使其更易于阅读和理解，这对于处理较长或较复杂的文本输入尤其有用。

### 3.7 输出格式

在提示中，可以告诉LLM如何格式化输出。请看下面的例子。

**提示：**创建一个Python函数，接收用户对象列表（每个对象包含一个用户的ID和姓名），并返回一个将用户ID映射到姓名的JSON对象。将输出转换为JSON格式。

ChatGPT的响应如下。

```
{  
    "code": "def map_user_ids_to_names(users):\n        user_map = {user['id']: user['name'] for user in users}\n        return user_map",  
    "example_usage": {  
        "input": [  
            {"id": 1, "name": "Alice"},  
            {"id": 2, "name": "Bob"},  
            {"id": 3, "name": "Charlie"}]  
    }  
}
```

```
{"id": 3, "name": "Charlie"}  
],  
"output": {  
    "1": "Alice",  
    "2": "Bob",  
    "3": "Charlie"  
}  
}  
}
```

其他可以进行格式化输出的提示范例如表3-3所示。

表3-3 格式化输出提示范例

格式类型	提示范例
表	创建一个表格，比较Python Java和C++的语法、性能和用例
列表	用列表列出排除网页加载缓慢故障的步骤
Markdown/HTML	解析HTTP方法GET和POST的区别，用Markdown格式展示
文本层次结构	提供软件开发生命周期的层次结构，包括各阶段及其主要活动
LaTeX格式化	在LaTeX中表示二进制搜索算法的时间复杂度

在构建提示时，可以通过指定期望的回答长度来进一步指导LLM。例如，通过使用“请提供一个简短的概述”或“请写出一个详细的解释”等指令，可以帮助模型理解回答的深度和广度。如果需要更精确地控制回答的长度，可以明确地给出要求，如“请将回答限制在300字以内”。虽然有时LLM可能会超出设定的字数限制，但它通常会在指定的大致范围内给出回答。

### 3.8 优秀实践

接下来我们将探讨一些制作有效提示的优秀实践，以帮助你从LLM中获得所需的答案。然而，这些建议不应视为一成不变的法则，它们更多是基于经验的指导性建议，可能带有主观性，并非不可更改。随着

你与LLM交流的深入，你将逐渐发现一些适合自己的、更为有效的提问技巧。这些提问技巧是提示工程这一领域的重要组成部分。

### 3.8.1 越具体越好

编写一个恰当的提示就像在愉快的对话中找到那个恰到好处的点，这是与文本生成系统达成默契的关键步骤。提示中的信息越详尽、越清晰越好。如果提示过于模糊，LLM就可能做出错误的假设，甚至产生不切实际的幻想。

首先，我们来看一些过于笼统的提示范例。

**提示：**开发一个增强数据安全的功能。

**提示：**你能构建一款自动化流程的工具吗？

**提示：**优化这段代码。

**提示：**我们需要一个处理事务的函数。

相比之下，以下是一些更详细的提示，它们更有可能让你获得满意的结果。

**提示：**开发一个Python函数，用于从字符串中解析日期。该函数应能处理以下日期格式：YYYY-MM-DD、MM/DD/YYYY和Month DD, YYYY。它应返回一个datetime对象。请提供一个脚本，演示该函数正确处理每种日期格式的至少3个示例，并在附带文档中说明任何依赖项、函数中使用的逻辑，以及如何运行该脚本。

**提示：**开发一个SQL查询，从数据库中检索2023年最后一个季度消费超过500美元的客户列表。该查询应返回客户的全名、电子邮件地址、消费总金额和上次购买日期。查询结果应按消费总金额降序排列。请确保该查询已针对性能进行了优化。

### 3.8.2 专业术语与缩略语

在构建提示时，明确专业术语和缩略语至关重要。缩略语在不同的语境中往往有不同的含义，这可能导致模型提供不相关或无效的回答。因此，最好使用专业术语的完整形式，并对所使用的专业术语提供清晰的定义或解释。

假设你正在使用ChatGPT帮助解决数据库连接问题，以下是一个设计不当的提示。

**提示：DB连接出现问题。如何解决？**

在上述提示中，“DB”指代不清，因为它可能指MySQL、PostgreSQL或其他数据库系统，而且也没有说明数据库连接问题的细节。

更有效的提示如下。

**提示：在尝试使用JDBC连接PostgreSQL数据库时，我遇到了连接超时问题。如何解决这个问题？**

这个改进后的提示清楚地指明了正在使用的数据库系统（PostgreSQL）、连接的方法（使用JDBC）以及遇到的具体问题（连接超时）。



马克·吐温说：“恰当的表达和不恰当的表达，它们之间的区别就像闪电和闪电之间的区别。”这句话同样适用于写作提示的编写。一个好的提示能够直击要点，引导出清晰、有力的回答；而一个含糊的提示则可能导致混淆和无效的交流。

### 3.8.3 零样本学习和少样本学习

在零样本学习（zero-shot learning）中，往往只需要提供一个精心构思的提示，就能引导模型给出正确的答案。这种方法在许多情况下都非常有效。然而，考虑到编程语言和框架的复杂性，有时单纯的提示可能不足以让你获得满意的结果，这时就需要采取进一步的措施。

少样本学习（few-shot learning）是解决这一问题的有效途径。少样本学习指的是LLM能够利用极少量的示例或训练数据来理解任务要求并执行任务。与传统的机器学习模型相比，这无疑是一个巨大的优势，因为传统模型通常需要大量的训练数据才能完成复杂的任务。LLM之所以具备这一能力，是因为它在针对特定任务进行微调之前，就已经在广泛的网络文本上进行了大规模的预训练。

以下是一个少样本学习的示例。假设我们需要生成一个函数，用于对给定的数字列表进行归一化处理，使取值区间变为[0, 1]。在指令中，我们提供了几个输入和归一化输出的示例。

**提示：**根据以下示例，将数字列表归一化，使取值区间变为[0, 1]。

输入：[2, 4, 6, 8] 输出：[0, 0.3333, 0.6667, 1]

输入：[5, 10, 15] 输出：[0, 0.5, 1]

输入：[1, 3, 2] 输出：[0, 1, 0.5]

请生成一个Python函数，该函数接受一个数字列表作为输入，并返回另一个归一化的数字列表。

通过这种方式，ChatGPT可以从提供的示例中“学习”并生成所需的代码，如下所示。

```
def normalize(numbers):
    min_value = min(numbers)
    max_value = max(numbers)
```

```
if min_value == max_value:

    return [0] * len(numbers) # Handle the case where

                                # all numbers are the same

else:

    return [(x - min_value) / (max_value - min_value) for x in n
umbers]

# Test the function with one of the examples

print(normalize([2, 4, 6, 8])) # Output: [0.0, 0.3333333333333333,
                                # 0.6666666666666666, 1.0]
```

### 3.8.4 引导词

引导词是一种特殊的关键字或短语，它能够指导LLM生成特定类型的输出。在某些情况下，仅仅一个恰当的引导词就足以激发模型按照预期的方式做出回应。以下是一个示例。

**提示：**

```
# 创建一个简单的Python函数，该函数能够
```

```
# 1. 提示输入华氏温度
```

```
# 2. 将华氏温度转换为摄氏温度
```

```
def
```

这里使用def作为引导词，告知模型接下来应该开始编写一个Python函数。表3-4列出了更多的引导词。

表3-4 引导词

语境	引导词
JavaScript函数	Function
HTML元素	<button
CSS样式	P{
SQL插入查询	INSERT INTO
Java方法创建	public

### 3.8.5 CoT提示

2022年，Google的研究人员在论文“Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”中提出了一种创新的提示技术——思维链（Chain-of-Thought，CoT）<sup>注</sup>提示。CoT提示通过将复杂问题分解为一系列步骤，显著增强了LLM的推理能力。实际上，CoT提示的原理与少样本学习类似，通过向模型展示少量示例来引导其理解和执行任务。

在软件代码生成任务中，CoT提示显得尤为有效。下面用一个示例来说明。假设你打算使用Python语言的Flask框架来开发一个具备用户注册与登录功能的Web应用程序，表3-5展示了如何应用CoT提示来指导模型。

表3-5 CoT提示范例

行为描述	提示范例
理解需求	我需要使用Flask创建一个Web应用程序。该应用程序应具有用户注册和登录功能。我应该从哪里开始？
创建Flask应用程序	让我们从创建一个基本的Flask应用程序开始。我该怎么做呢？
创建用户模型	现在Flask应用程序已经创建好了，我需要创建一个用户模型来处理注册和登录。我应该如何创建这个用户模型？
实现注册功能	有了用户模型，我怎样才能创建一个包含必要字段的注册页面呢？
实现登录功能	现在，让我们继续创建登录页面。如何确保登录安全？
会话管理	用户登录后，我应该如何管理用户会话，使用户在浏览应用程序时保持登录状态？
实现用户退出功能	最后，如何实现注销功能，以安全地退出应用程序？

### 3.8.6 引导性问题

在构建提示时，引导性问题可能导致LLM产生不寻常或不准确的回答。为了获得最佳结果，提示应当保持中立，避免带有任何倾向。同时，清晰和明确的表述对于避免误解和臆断至关重要。

以下提示会产生引导性问题。

**提示：**迁移到微服务架构并不总是能提升系统可扩展性，对吗？

优化后的提示如下。

**提示：**迁移到微服务架构在系统可扩展性方面有哪些优势和潜在挑战？

### 3.8.7 请求类比示例

设想你并不熟悉面向对象编程中继承的概念。打开ChatGPT并输入以下提示。

**提示：**解释面向对象编程中继承的概念。

你将得到一个详细的答复。如果你希望获得更易于理解的解释，一种有效的方法是让LLM使用类比来说明。

**提示：**通过类比解释面向对象编程中继承的概念。

ChatGPT：把继承想象成一棵家谱树，孩子从父母那里继承某些特征和属性，而这些特征和属性也可能是孩子从祖父母那里继承的。

在这里，ChatGPT通过类比的方式提供了一种更加具体且形象的解释，来帮助阐释继承的核心要素。

## 3.9 减少幻觉

在第2章中，我们提到提示LLM时可能会导致其产生幻觉，即生成的内容是虚假或误导性的，但LLM会把内容当成真的来响应。幻觉对于准确性要求高的软件开发来说尤其具有挑战性。

本章中的经验教训无疑可以帮助减少幻觉问题。然而，即便是精心设计的提示，也可能触发幻觉。造成幻觉的原因多种多样，包括如下这些。

### 缺乏基本事实验证

LLM根据从训练数据中获得的知识生成响应，但它无法验证信息的准确性或真实性。

### 过拟合与记忆

模型可能会记住训练数据集中不正确或误导性的信息，尤其是当这些信息重复出现或较为常见时。

## 训练数据的偏见

如果训练数据包含偏见、不准确或虚假的信息，则模型可能会在输出中复制这些内容。

## 推断和推测

模型有时可能会根据数据中的模式进行推断，生成训练数据未充分覆盖的主题或问题的响应。

## 缺乏或误解上下文

LLM可能会误解上下文或缺乏必要的背景知识，无法准确回答某些提问。模型可能不完全理解一些查询的细微差别或含义。

## 俚语和习语

俚语和习语可能会产生歧义，导致模型误解原意，尤其是当模型在训练过程中没有看到足够多的俚语或习语示例时。

为了减少幻觉，首先应避免提出过于开放性的问题，例如下面的提示。

### 提示：优化数据库有哪些不同的方法？

开放性提示可能会鼓励模型进行猜测或过度概括，模型也可能误解问题的意图或所需回答的格式，导致回答偏离主题或包含虚构信息。

一种有效的方法是提供一组预定义的选项，让LLM从中选择。例如，上述提示可以改写成如下形式。

### 提示：以下哪一项是优化数据库的方法，索引、碎片整理还是压缩？

另外，考虑要求LLM对某个结论给出判断。下面是一个有效的提示。

**提示：**在Java中初始化数组时，下列语法是否正确？请回答“是”或“否”。

也可以在提示中包含多个步骤，以引导模型完成结构化流程，降低偏离正轨的可能性。

**提示：**

步骤1 创建斐波那契数列生成器。

步骤2 使用迭代法。

步骤3 编写一个名为generate\_fibonacci的Python函数，将整数n作为参数。

步骤4 该函数以列表形式返回斐波那契数列中的前n个数字。

## 3.10 安全和隐私

在构建提示时，确保安全和隐私至关重要。实际上，应将采取适当预防措施的责任纳入组织的规章制度。在提示中应避免包含任何敏感或个人信息，例如个人身份信息。以下是一个包含个人身份信息的提示。

**提示：**你将如何解决John Doe报告的账号john.doe@example.com的登录问题？

更明智的提示方式是去标识化，例如下面的提示。

**提示：**如何处理用户报告的登录问题？

避免在提示中泄露任何敏感的系统详细信息也是明智之举。应避免构建如下提示。

**提示**：如何修复IP地址为192.168.1.1的生产服务器上的数据库连接错误？

相反，使用更通用的问题往往更安全，例如下面的提示。

**提示**：如何修复通用数据库连接错误？

此外，要确保提示不会在无意中诱发不正当的行为。从安全角度来看，如下提示是恰当的。

**提示**：如何检测和防止SQL注入？

但下面的提示可能会诱导一些不良意图。

**提示**：如何利用网站的SQL漏洞？

除了遵守安全和隐私规则以外，在构建提示时，接受多样性和包容性同样重要。理解和控制偏见（通常反映在训练数据中）至关重要。最好使用中性和包容性的语言，避免在提示中使用任何歧视性或排他性的措辞。此外，从不同人群中获取对提示的反馈也很有帮助，这不仅能提高与LLM互动时的公平性和包容性，还有助于更准确、更全面地理解主题。

### 3.11 自治的AI Agent

到目前为止，我们已经了解逐步引导LLM的整个过程，这是软件代码生成的核心所在。

然而，AI Agent能够提供更进一步的功能。它们不仅可以按照提示执行任务，还能与LLM协同工作，发挥创造力，为你提出的任何目标制

订策略和计划。这些Agent能够利用Pinecone和Chroma DB等专业数据库，处理模型所能够理解的复杂词嵌入。

自治的AI Agent是学术研究的成果，通常作为开源项目的一部分使用。它们真正的强大之处在于其自动化能力。接下来我们通过一个示例来解释它们的工作方式。假设你的目标如下。

**提示：**创建一个简单的带有用户登录系统的天气应用程序。

表3-6列举了自治的AI Agent可能经历的阶段。

表3-6 自治的AI Agent可能经历的阶段

阶段	操作
创建任务	设计用户界面 草绘页面的基本布局 选择配色方案和字体 设计图标和其他图形元素
与天气数据API集成	在互联网上搜索可靠的天气数据API 确定要显示的数据点 编写获取和更新天气数据的代码
地点选择功能	创建搜索栏或下拉菜单，供用户选择地点 将位置信息连接到API代码
错误处理	处理API调用失败或位置条目无效等错误
确定任务的优先级	优先集成天气数据API 关注用户界面 开发地点选择功能和错误处理功能
迭代	查看生成的代码和天气页面的当前状态。确定执行过程中出现的剩余任务或新任务。重复创建步骤并进行优先级排序

自治的AI Agent技术正站在行业的前沿，并展现出巨大的发展潜力。当然，挑战也在所难免，具体如下。

### **资源密集型**

AI Agent可能会消耗巨大的计算资源，对处理器和数据库造成压力，这可能导致等待时间变长、可靠性降低，以及性能随着时间推移下降。

### **无限循环**

在某些情况下，由于缺乏进展或重复的奖励机制，AI Agent可能会陷入循环，无法取得实际效果。

### **实验性质**

作为新兴技术，AI Agent可能存在一些粗糙边缘，包括漏洞或意外行为，因而可能还不足以承担大型任务，这取决于它们的具体用途。

### **记忆问题**

AI Agent可能会遗漏某些步骤或指令。

### **多任务处理困难**

面对繁重的工作量，AI Agent可能会不知所措。

### **被细节分散注意力**

AI Agent可能会被一些无关紧要的细节困扰，这可能导致它们在选择工具或方法时误入歧途。

支持LLM的另一项创新是检索增强生成（Retrieval-Augmented Generation，RAG）。通过RAG，生成式AI应用（例如使用LangChain框架构建的应用）可以访问外部数据源，通常是矢

量数据库。这为模型提供了更多的具体知识，从而增强LLM的响应能力。

在处理复杂的软件开发任务时，针对以下情况，RAG尤其有用。

## 解决软件缺陷和故障

当开发者遇到软件缺陷和故障时，可以使用RAG技术从网络论坛、缺陷数据库等资源中挖掘出修复方法和解决方案。RAG技术可以根据具体问题提供定制的解决方案或代码补丁。

## 为代码审查增添价值

RAG技术可以获取公司内部的所有编程最佳实践、标准及必须遵守的规则。这意味着它可以帮助简化代码审查过程，为代码提供有用的技巧，让你的代码更加出色。

## 加速测试

当需要对代码进行测试时，RAG技术可以充当虚拟测试人员。它能找到各种测试场景和模式，并根据项目需求进行调整，帮助你更快地开发出测试用例或脚本。



在某些情况下，LLM会指出你的提示有问题。例如，假设你提供以下提示。

**提示：**使用汇编语言编写RESTful API。

ChatGPT的响应如下。

ChatGPT：由于汇编语言的低级特性，使用汇编语言编写RESTful API是非常规的且复杂的。汇编语言通常用于需要对硬件进

行细粒度控制的任务，或者用于性能要求很高的应用场景。

### 3.12 小结

重申一下，完美的提示是科学与艺术的结合。我们需要找到合适的方法——运用创造力、直觉和条理清晰的技巧来构建提示，以使LLM提供我们所期望的答案。在这个过程中，并没有一成不变的神奇公式。如果能够清晰地表达提示，确保逻辑连贯，并且提供一些示例，则更有可能得到满意的答案。

这是一个不断尝试和调整的过程。你需要尝试不同的提示，观察它们的成效，然后根据反馈进行调整，再继续尝试。就像其他技能一样，你在不同的主题和任务上练习得越多，你的技巧就会越娴熟。

**[5]**为了更好地还原两词在书写和含义之间的区别，这里直译lightning bug为闪电虫，实际应译为萤火虫。马克·吐温原意是想表达

**[6]**Chain-of-Thought Prompting通常译为思维链提示或思想链提示，本书统一译为思维链提示。——译者注

## 第4章

# GitHub Copilot

本章将深入探讨GitHub Copilot，以下简称Copilot。作为AI辅助编程领域的领军产品，Copilot被誉为生成式AI的“杀手级应用”。它为开发者提供了高级的代码建议、代码创建和代码解释功能，极大地提升了开发效率。Copilot通过理解代码上下文和预测编程模式，不仅节约了开发者的编程时间，还助力他们编写更高效、更可靠的代码。

在本章中，我们将细致了解Copilot的各个方面，包括它的定价策略、配置方法，以及它所提供的所有创新功能。同时，我们也会坦诚地讨论它的局限性。为了帮助读者更好地利用Copilot，本章还会提供一些实用的Copilot使用技巧。此外，我们还将介绍Copilot如何通过合作伙伴计划与其他企业进行协作。总而言之，本章旨在全面而深入地介绍Copilot的功能和应用。

## 4.1 GitHub Copilot简介

2021年6月，微软推出了GitHub Copilot。GitHub首席执行官Nat Friedman在一篇博文中将Copilot形容为“AI编程助手”，Copilot旨在协助开发者更高效地完成编程任务，进而加快开发者的进度。

Copilot是微软与OpenAI合作的成果，最初基于Codex构建而来，Codex是为编程专门定制的GPT-3的一个变体。为了增强Codex理解代码上下文的能力，微软引入了一种名为中间填充（Fill-In-the-Middle，FIM）的新方法，从而提高了代码建议的质量和响应速度。随着更先进的GPT-3.5 Turbo的加入，FIM带来的改进进一步提升了代码建议的接受率。

为了增强安全性，Codex结合了AI系统，能够即时识别并阻止潜在的风险编程模式，并特别关注那些常见的漏洞，如硬编码凭证和SQL注入等，从而保障代码的安全性。

### 4.1.1 版本和价格

Copilot提供30天免费试用服务。试用期结束后，用户有以下3种选择  
注。

#### 个人版Copilot

个人版Copilot月费为10美元，年费为100美元。个人版Copilot提供了一系列功能，如多行函数建议，旨在让编程工作更加流畅。一个特别突出的功能是加速测试生成，这对于确保代码的可靠性和稳定性至关重要。此外，个人版Copilot还具备筛查易受攻击编程模式的能力，并避免提供任何可能涉及版权问题的公共代码建议，从而确保所生成的代码是独特和原创的。

#### 商业版Copilot

商业版Copilot提供的功能更全面，每位用户每月收费19美元。商业版Copilot包括个人版Copilot的所有功能，并增添了一些专为企业设计的额外功能。商业版Copilot简化了许可证管理，这对公司管理订阅和访问权限非常有利。商业版Copilot还具备管理组织政策的功能，有助于统一管理所有项目，保持项目间的一致性。商业版Copilot的另一大亮点是其顶尖的隐私保护措施，这对于确保敏感业务数据的安全至关重要。此外，商业版Copilot还支持企业代理，以确保整个公司网络的连接安全和顺畅。

#### 企业版Copilot

企业版Copilot月费为39美元。企业版Copilot的主要特点是允许在公司内部代码库上对系统进行训练，从而提供更有针对性的代码建议，显著提升开发者的工作效率。企业版Copilot根据开发者的内部编程习

惯、偏好使用的API、框架和软件包进行量身定制，有助于推广最佳实践，同时遵守企业安全规则。

企业版Copilot的另一个优势是可以用COBOL和Fortran等传统编程语言进行训练。由于这些编程语言通常缺乏公共资源库中的训练数据，通用LLM可能在这些编程语言上表现不佳。当然，通用LLM也无法了解特定公司的专有编程语言。

GitHub系统会持续扫描公司的软件仓库，以增强企业版Copilot的能力。例如，GitHub系统关注最近的拉取请求、合并操作，以及“赞成”和“反对”等反馈，这些都有助于突出公司正在使用的最新方法和策略。

定制模型有助于在整个组织内传播专业知识。AI可以捕捉并分享代码中隐含的微妙知识。通过持续训练，AI可以跟上代码库的不断变化，使其提供的帮助随着时间的推移变得更加精确。然而，企业在使用AI辅助编程工具的过程中，必须注意处理好隐私和知识产权风险。



2023年，Gartner的研究表明，仅有不到10%的大型企业开始使用AI辅助编程工具。企业犹豫不决的态度部分出于对安全性和精确性的担忧。然而，随着AI技术的飞速发展，预计在可预见的未来会有越来越多的企业开始采纳这些工具。总之，AI辅助编程工具带来的好处不容忽视。

#### 4.1.2 案例：AMD公司的硬件编程

在AMD公司的一项有趣案例研究中，我们看到了定制模型的实际应用。成立于1969年的AMD公司是CPU领域的先驱，并且如今已成为数据中心、嵌入式系统、游戏平台和个人计算机半导体领域的领导者。

在深入了解这个案例之前，让我们先回顾一下硬件系统开发的基础知识。硬件系统开发与Web应用程序等软件的开发有着本质上的不同。开发者需要对硬件系统有深刻的理解。与运行在通用计算机上的普通软件不同，固件直接与硬件交互，这要求更高的精确性和兼容性。

在固件中，精确性至关重要，因为固件开发中的任何失误都可能导致严重且代价高昂的后果。一个错误不仅意味着数百万美元的经济损失，还关系到时间因素。修复固件问题通常需要重新检查制造流程，这可能会增加几个月的时间。这种延迟不仅会推迟产品的发布，还会影响产品在市场上的竞争力。

显然，敏捷软件开发中流行的“快速行动，打破常规”的思维模式并不适用于硬件编程。这种方法的风险太大。因此，固件开发者必须投入大量时间和精力来制订详细计划并进行广泛测试。保持谨慎可以确保固件在与硬件配对之前尽可能稳固和无误。

2023年，AMD公司在对Copilot进行研究的过程中设定了非常高的标准，产生了相当多的怀疑，但所有怀疑都是合理的。在一个试点项目中，AMD公司为Verilog和SystemVerilog等各种HDL（Hardware Description Language，硬件描述语言）创建了一个定制版的Copilot。HDL是一种特殊的编程语言，专门用于概述电子电路（尤其是数字逻辑电路）的架构、设计和功能。HDL对不同抽象层次的电子系统建模和仿真至关重要。

实验结果超出预期。令人惊讶的是，Copilot生成的代码实际上比AMD公司的开发者编写的代码更符合AMD公司的标准。如此显著的改进，促使一些开发者从Vim（一种高度可定制的文本编辑器）转而采用Visual Studio Code作为IDE。

### 4.1.3 案例：Shopify电商平台

另一个有趣的案例发生在Shopify公司。Shopify是一个电商平台，旨在让客户轻松建立在线商店。该平台在美国市场占有约10%的

份额，在欧洲市场则占有约6%的份额。

显然，为了支持其广泛的业务，Shopify需要庞大的基础设施支持。目前，Shopify公司拥有大约300个公共代码仓库和5000个私有代码仓库。此外，Shopify公司每天对其代码进行大约1500次部署。

Shopify公司是首批拥抱Copilot的公司之一，这一工具极大地改变了该公司的开发流程，提升了开发者的工作效率。目前，Shopify公司有近2000名开发者正在使用Copilot。最引人注目的是，其中70%的开发者认为它非常有用，75%的开发者经常使用它，而且Copilot提供的代码建议约有26%被采纳。

尽管一些功能（比如与命令行界面（Command Line Interface，CLI）的集成）尚未普及，但许多开发者已经开始在日常工作中使用代码自动补全和聊天功能。

以下是一些有趣的总结。

## 代码建议的价值

即使开发者不采用给定的建议，也不会造成完全的损失。任何建议都能激发开发者编写更好代码的灵感。

## 使用率

随着时间的推移，Copilot的使用率通常会增加，这是意料之中的。开发者需要时间来调整他们的日常工作流程并适应新功能。Copilot也有一个学习曲线。

## 高级开发者的态度

早期采用Copilot时，经验丰富的开发者并不热衷于使用它，他们倾向于将其视为一款玩具而非严肃的工具。但随着时间的推移，当看到其他开发者取得的实际效果后，高级开发者对Copilot的使用开始变得更加活跃。

## 学习和强化

Shopify公司注意到，Copilot擅长引导开发者尝试新的编程语言或框架。例如，Rust的采用率明显上升。

Shopify公司的代码库中有大约100万行代码是用Copilot编写的，这表明Copilot对该公司的业务具有重大意义。

### 4.1.4 案例：埃森哲咨询公司

埃森哲咨询公司是一个大型专业服务机构，其通过创新的技术和系统来帮助客户改善运营和实现增长。该公司在全球120多个国家拥有733000多名员工。

2023年，埃森哲咨询公司对内部450名开发者进行了Copilot测试。该公司没有设定任何具体的任务或目标。经理们只是要求每个人像往常一样工作。

埃森哲咨询公司对Copilot的试用长达6个月，结果如下。

- 在编程方面，Copilot建议的接受率为35%，其中88%的修改甚至在经过代码审查后仍然有效。
- 生产力得到显著增强。拉取请求增加了50%，合并率提高了15%。
- 效率得到大幅提升，构建次数增加了50%，成功率提高了45%。
- 开发者非常满意：高达96%的开发者认为他们从第一天起就取得了成功。

看到这些结果后，埃森哲咨询公司呼吁在全球范围内向50000名开发者推出Copilot。

### 4.1.5 安全

Gartner的一项调查研究了2000多名首席信息官（Cheif Information Officer，CIO），发现其中66%的CIO计划在2023年将网络和信息安全作为主要投资领域。这一趋势已经持续了多年。

在当今世界，网络威胁日益复杂和普遍，安全漏洞可能导致巨大的经济损失、声誉损害、法律纠纷和运营中断。此外，随着数据隐私法规变得日益严格，CIO必须保持对合规性的警觉，以避免罚款并维护客户的信任。

因此，GitHub将安全性作为Copilot计划的一个重点。GitHub开发了一套基于LLM的系统，能够在不安全的编程模式出现时立即检测并修复它们。

此外，GitHub还推出了Copilot辅助的高级安全功能，具体如下。

### **扫描代码**

实时扫描安全漏洞和编程错误。

### **扫描敏感数据**

根除已提交到私有存储库的密钥和令牌等敏感信息。

### **审查依赖**

展示依赖项变更的影响，并在合并拉取请求时提供漏洞的详细信息。



对于学生、教师和流行开源项目的维护者，Copilot是免费的，但他们需要完成身份验证过程。

## **4.2 使用Copilot**

在开始使用Copilot之前，首先需要创建GitHub账户。GitHub是一个支持版本控制和协同软件开发的在线服务。它建立在Git的基础上，是一款提供代码审查和项目管理功能的工具。

设置完GitHub账户后，单击屏幕右上方的个人照片，你会看到一个下拉菜单，如图4-1所示。单击Your Copilot命令以启用你的Copilot。你可以选择计划类型，然后单击Continue按钮。如果不符免费账户的条件，则需要提供付款详情。



ttaulli  
Tom Taulli

X

🏡 Working from home

---

👤 Your profile

👤+ Add account

---

💻 Your repositories

🚧 Your projects

🏢 Your Copilot

🏢 Your organizations

🌐 Your enterprises

⭐ Your stars

❤️ Your sponsors

leftrightarrow Your gists

---

⬆️ Upgrade

🌐 Try Enterprise

👤 Feature preview

⚙️ Settings

---

📖 GitHub Docs

👤 GitHub Support

---

Sign out

图4-1 单击个人照片后弹出的下拉菜单

### 4.2.1 Codespaces与VS Code

Copilot有两种使用方法。一种方法是通过Codespaces进行访问。这是一种基于云的开发环境，运行在Visual Studio Code（简称VS Code）上，可直接在GitHub网站上使用。

另一种方法是使用VS Code桌面版，当然，也可以选择VS Code Copilot扩展。本书将重点介绍如何使用VS Code Copilot扩展。

VS Code本身是免费的。图4-2显示了VS Code的主界面。VS Code可以运行在各种平台上，包括Windows操作系统（如Windows 7、8、10和11）、macOS和各种Linux发行版（如Ubuntu、Debian、Fedora等）。

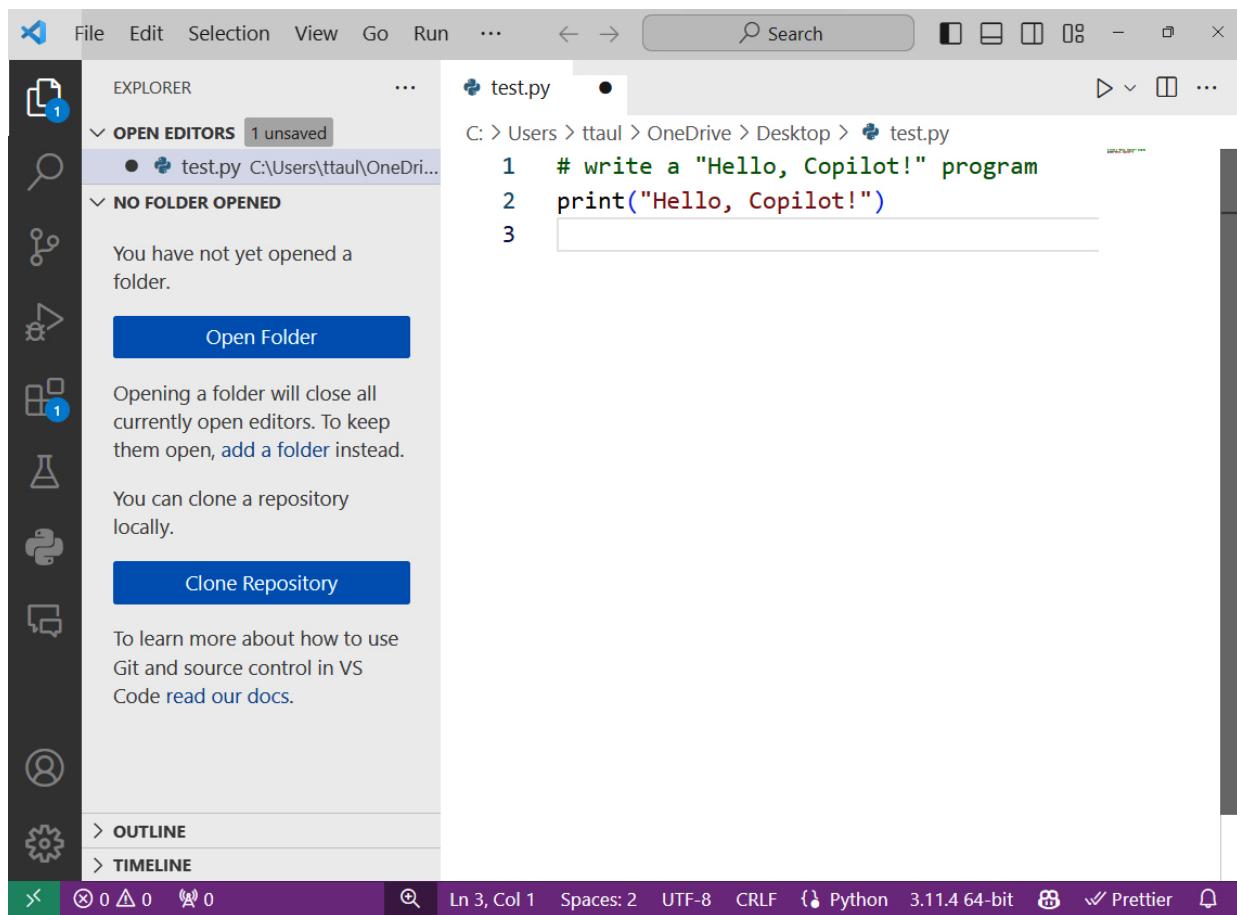


图4-2 VS Code的主界面

VS Code主界面的左侧是活动栏，里面有一堆图标。你可以使用它们来加载文件和文件夹。你还可以选择其中的第5个图标，进入安装扩展的区域，如图4-3所示。

你将看到输出“Hello, Copilot!”的代码。这是使用以下提示创建的，Copilot会将其转换为代码。

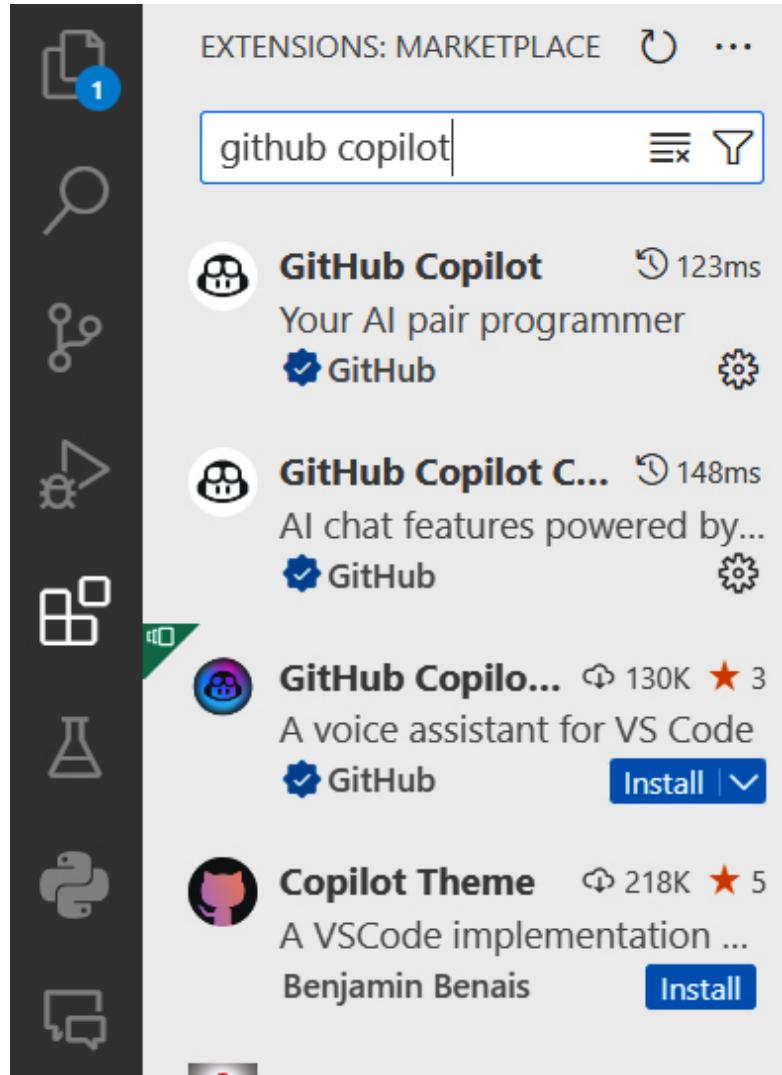


图4-3 可以在VS Code中安装扩展，例如Copilot扩展

**提示：**#写一个“Hello, Copilot!”的程序。

字符#声明了这是一条注释。这是指示Copilot生成代码的方法之一。

值得关注的是，代码是用Python语言编写的。这是如何实现的呢？原因在于文件的扩展名为.py。Copilot根据文件的扩展名来判断并采用相应的编程语言。

VS Code主界面的右上方有一个运行程序的图标。单击它，VS Code将启动终端。你将看到输出到终端的信息。

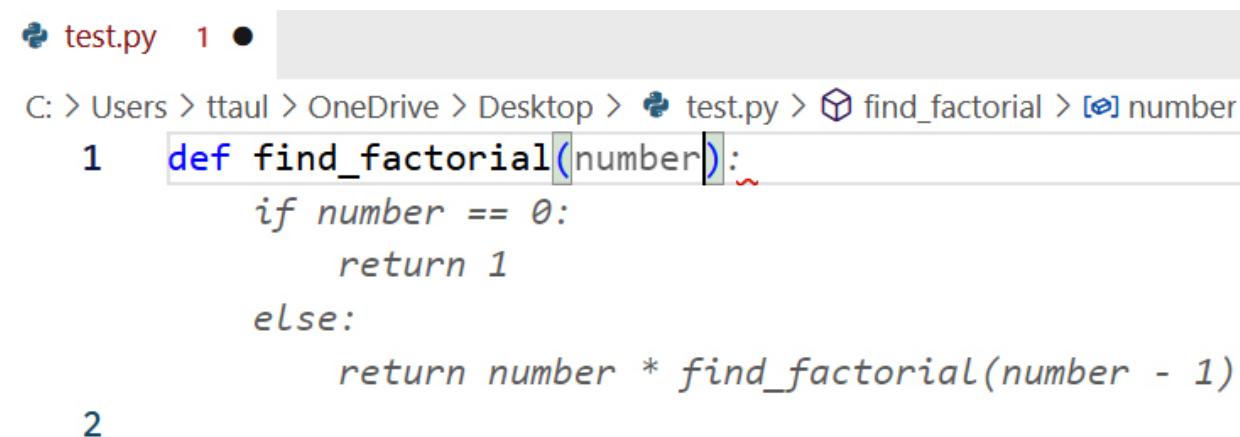
## 4.2.2 代码建议

在VS Code中使用Copilot的方法非常简单。当你开始输入代码时，Copilot就会启动，为你提供代码建议，并根据你的输入生成代码片段。

假设你在Python中输入如下函数头。

```
def find_factorial(number):
```

Copilot会立即提供一个完整的函数体，如图4-4所示。生成的编码称为幽灵文本（ghost text），以灰色高亮显示。



The screenshot shows a Python file named 'test.py' in VS Code. The cursor is at the start of a function definition. The code completion suggestion for the function body is shown in a light gray box:

```
C: > Users > ttaul > OneDrive > Desktop > test.py > find_factorial > number
1 def find_factorial(number):
    if number == 0:
        return 1
    else:
        return number * find_factorial(number - 1)
```

The line numbers '1' and '2' are visible on the left. The entire code block is highlighted in light gray, indicating it is generated by Copilot.

图4-4 当用户输入函数头时，Copilot建议的代码

Copilot建议的代码在你的机器上可能有些不同，这很正常，因为底层LLM是根据一组复杂的概率来工作的。

GitHub知道你想写一个基于参数计算阶乘的函数，因此，它向你推荐了获得正确结果所需的if-else结构。

你可以按Tab键接受代码建议。但如果Copilot建议的代码不是你所期望的，你可以按Esc键跳过它们。

将鼠标指针悬停在生成的代码上，你会看到更多选项，如图4-5所示。

```
test.py 1
C: > Users > ttaul > OneDrive > Desktop >
1 def find_factorial(number):|
    if number == 0:
        return 1
    return number * find_factorial(number - 1)
```

图4-5 将鼠标指针悬停在生成的代码上，右上方会出现更多选项

你可以单击Accept（接受）按钮接受代码建议，就像按Tab键一样；也可以单击Accept Word（接受单词）按钮，一次选择代码中的一个单词。如果单击上面有3个点的按钮，就会出现一个菜单，你可以选择接受一行或始终显示工具栏。

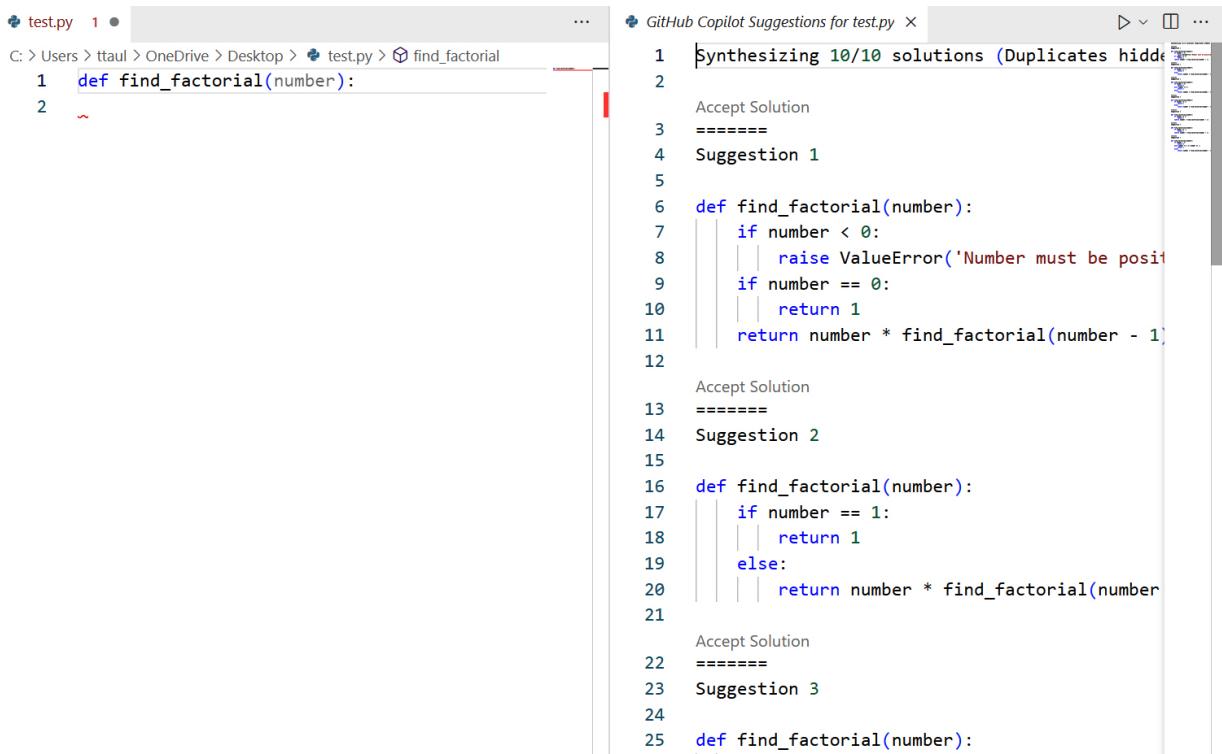
通过单击工具栏左侧的“1/2”按钮，你可以浏览Copilot为所生成的代码提供的备选方案。如图4-5所示，存在两个备选方案，当前展示的是第一个方案。单击“>”按钮将切换至第二个方案，而单击“<”按钮则会返回至第一个方案。

你还可以使用表4-1所列的快捷键来导航代码建议。

表4-1 用于浏览代码建议的快捷键

操作系统	下一个代码建议（快捷键）	上一个代码建议（快捷键）
macOS	Option (⌥) o或Alt + ]	Option (⌥) o或Alt + [
Windows	Alt + ]	Alt + [
Linux	Alt + ]	Alt + [

如果按Ctrl + Enter快捷键，VS Code将创建一个新的选项卡并显示10个备选方案（见图4-6）。



The screenshot shows the VS Code interface with two tabs open. The left tab is 'test.py' containing the code:

```

1 def find_factorial(number):
2     ~

```

The right tab is 'GitHub Copilot Suggestions for test.py' showing 10 synthesized solutions for the factorial function. The first few lines of the suggestions are:

```

1 synthesizing 10/10 solutions (Duplicates hidden)
2
3 =====
4 Suggestion 1
5
6 def find_factorial(number):
7     if number < 0:
8         raise ValueError('Number must be positive')
9     if number == 0:
10        return 1
11    return number * find_factorial(number - 1)
12
13 =====
14 Suggestion 2
15
16 def find_factorial(number):
17     if number == 1:
18         return 1
19     else:
20         return number * find_factorial(number - 1)
21
22 =====
23 Suggestion 3
24
25 def find_factorial(number):

```

图4-6 VS Code显示了10个备选方案

要选择其中一个备选方案，请单击相应代码上方的Accept Solution（接受方案），否则关闭选项卡以拒绝所有的代码建议。

### 4.2.3 注释

4.2.1节介绍了如何使用一个或多个注释来提示Copilot生成代码。但我们需要用当前使用的编程语言的语法来指明注释才能提示Copilot生成代码。

此外，你还可以做得更多。你可以使用注释与AI系统交互，请看下面的示例。

**提示：**#q:在面向对象编程中，什么是“类”？

**Copilot：**#a:类是创建对象的蓝图。对象具有与之相关的属性和方法（函数）。在Python中，几乎所有东西都是对象。

如你所见，答案与所使用的编程语言有关。现在，假设你向Copilot询问系统是如何工作的。

**提示：**#q:接受内联建议的快捷方式是什么？

**Copilot：**#a:Tab键。

你可以向Copilot询问VS Code的功能。

**提示：**#q:VS Code中用于用户设置的快捷方式是什么？

**Copilot：**#a:Ctrl+。

你还可以向Copilot询问其他IDE的快捷键。

**提示：**#q:Atom编辑器中创建新文件的快捷键是什么？

**Copilot：**#a:Ctrl+N。

需要注意的是，Copilot能够识别出我是在Windows操作系统上进行开发的，并据此提供了适用于Windows操作系统环境的代码建议。

#### 4.2.4 聊天

Copilot Chat类似于ChatGPT，是专为编程环境设计的对话工具。无论是需要澄清语法问题、解决软件缺陷、梳理测试用例，还是仅仅想要探讨不同的编程概念，Copilot Chat都能方便地与你进行代码相关的讨论。Copilot Chat由OpenAI的GPT-4和GPT-3.5 Turbo提供支持，并结合了微软自研的LLM。

你可以从GitHub网站直接访问Copilot Chat。它非常适合于代码分析、处理拉取请求、审查文档或解答编程相关问题。此外，GitHub应用程序的iOS和Android版本都已经集成了聊天功能，使得编程交流更加便捷，随时随地都能进行。

接下来，我们将重点讨论如何在VS Code中启用和使用Copilot Chat。这涉及添加一个新的扩展。要在VS Code中使用Copilot Chat，你需要打开扩展面板，搜索并找到“GitHub Copilot Chat”。安装完成后，你会在屏幕左侧看到一个新的图标，如图4-7所示，这表示Copilot Chat已经准备好与你进行互动了。

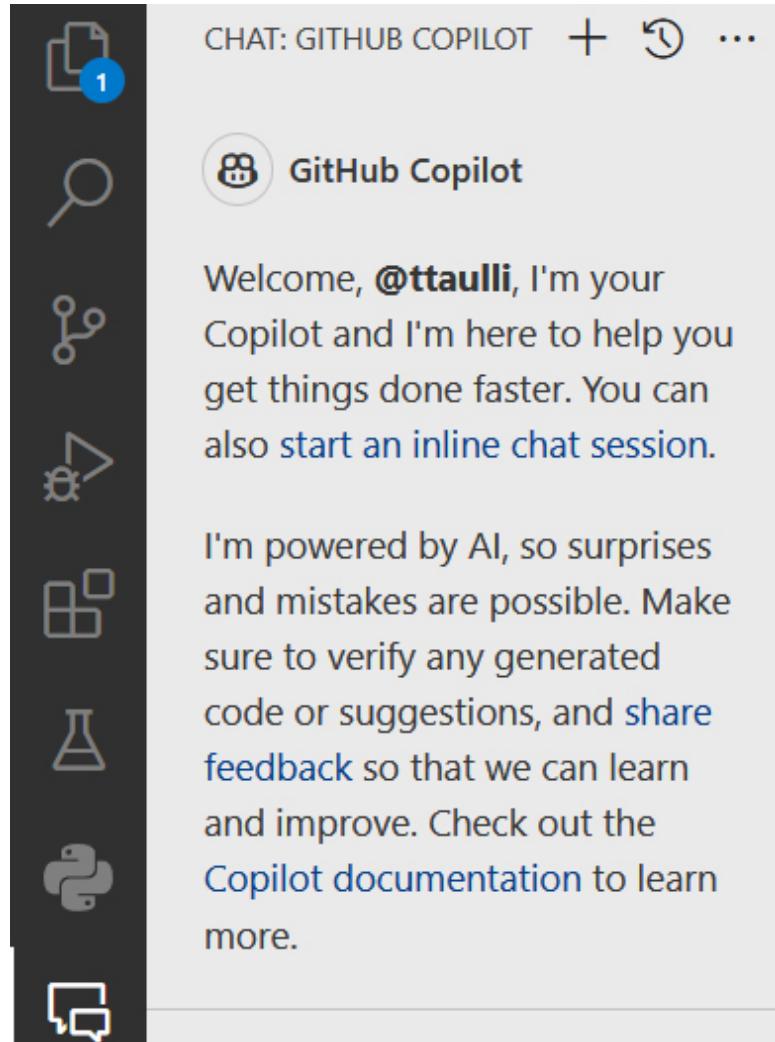


图4-7 屏幕左侧出现聊天图标

在聊天面板的顶部有一组图标。其中的第一个图标用于清除所有聊天内容。这个功能与对话上下文紧密相关。当你输入一个提示时，LLM会参考之前的对话历史来生成回答。如果聊天内容涉及多个杂乱的主题，则可能导致模型提供不太有用的答案。因此，保持每次聊天专注于单一主题，并在需要讨论新主题时开始新的聊天会更为有效。

第二个图标则允许你直接发送关于系统的反馈。紧接着是一个历史按钮，它采用时钟图标的设计，可以让你回顾之前的聊天主题。

单击上面有3个点的图标，便可以在编辑器内部访问和使用聊天功能，这里提供了进行深入交流的更宽敞的空间。

聊天系统还配备了一系列快捷方式来提升效率。例如，快捷方式“@workspace”指向工作区中当前打开的文件。通过使用这些快捷方式，你可以更快速、更精确地与系统交互。下面是一个示例。

如果已打开一个Python计算器程序，可以输入如下提示。

**提示：**@workspace /explain

图4-8显示聊天系统已经分析了程序文件并提供了详细解释。

你可以只要求聊天系统解释部分代码。选中感兴趣的代码，然后右击，将弹出一个菜单，如图4-9所示。

CHAT: GITHUB COPILOT + ⏱ ...

ttaulli

@workspace /explain

GitHub Copilot used @workspace

Used 1 reference >

This Python code is a simple command-line calculator program that can perform four basic arithmetic operations: addition, subtraction, multiplication, and division.

The program defines four functions, each corresponding to one of the arithmetic operations. The `add(x, y)` function takes two arguments and returns their sum. The `subtract(x, y)` function returns the difference of the two arguments. The `multiply(x, y)` function returns the product of the arguments. and the `divide(x, y)`

❖ How can I handle floating-point precision issues when performing calculations in Python?

Ask Copilot or type / for coi ▶

test.py X

C: > Users > ttaul > OneDrive > Desktop > test.py > ...

```
1 def add(x, y):
2     return x + y
3
4 def subtract(x, y):
5     return x - y
6
7 def multiply(x, y):
8     return x * y
9
10 def divide(x, y):
11     if y == 0:
12         return "Error! Division by zero."
13     else:
14         return x / y
15
16 while True:
17     print("Select operation:")
18     print("1.Add")
19     print("2.Subtract")
20     print("3.Multiply")
21     print("4.Divide")
22     print("5.Exit")
23
24     choice = input("Enter choice(1/2/3/4/5): ")
25
26     if choice == '5':
27         print("Exiting the program")
28         break
29
```

▲图4-8 Copilot Chat可用于解释代码

The screenshot shows a code editor window with a file named 'test.py'. The code implements a simple calculator program. A context menu is open over the line of code 'if choice == '1':'. The menu includes options like 'Go to Definition', 'Copilot' (which is highlighted in blue), and 'Explain This' (also highlighted in blue). Other visible options include 'Generate Docstring', 'Cut', and 'Run in Interactive window'.

```
C: > Users > ttaul > OneDrive > Desktop > test.py > ...
26     if choice == '1':
27         print("Exiting the program")
28         break
29
30     num1 = float(input("Enter first number: "))
31     num2 = float(input("Enter second number: "))
32
33     if choice == '1':
34         result = add(num1, num2)
35         print("The result is " + str(result))
36
37     elif choice == '2':
38         result = subtract(num1, num2)
39         print("The result is " + str(result))
40
41     elif choice == '3':
42         result = multiply(num1, num2)
43         print("The result is " + str(result))
44
45     elif choice == '4':
46         result = divide(num1, num2)
47         print("The result is " + str(result))
48
49     else:
50         print("Invalid choice")
```

▲图4-9 选中代码并右击，即可将聊天焦点集中在选中的代码上

当聊天系统只关注部分代码时，它不仅可以解释代码，还可以修复代码、生成文档或创建测试。

另一个有用的快捷方式是“/new”，用于为新的项目或功能创建“脚手架”。例如下面的提示。

**提示：**/new创建一个Python脚本，用于从网站上抓取数据。

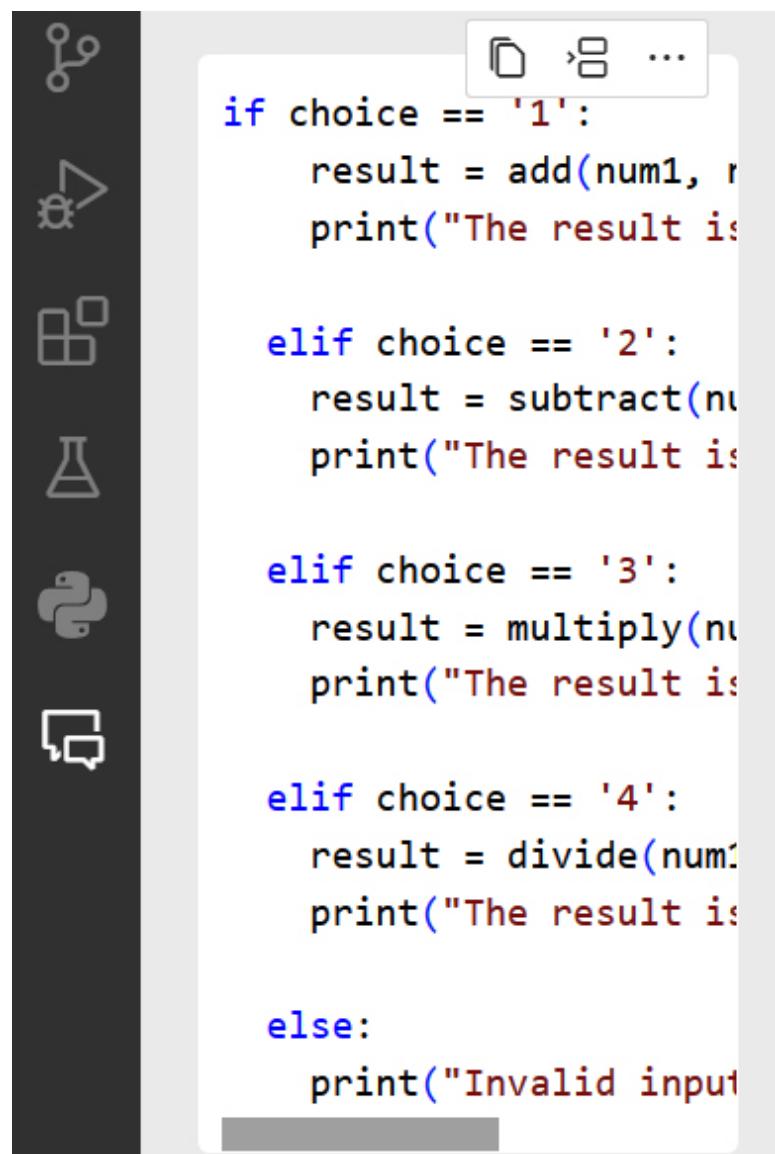
**提示：**/new为学生生成一个Java类，包含姓名、年龄和年级字段。

**提示**：/new使用Flask构建简单的RESTful API。

**提示**：/new创建一个JavaScript函数，对数字数组进行升序排列。

**提示**：/new为博客设计SQL数据库模式，其中包含用户、帖子和评论。

根据这些提示，聊天系统将显示主要步骤，然后提供代码清单。你还可以选择如何处理代码。如图4-10所示，代码清单的顶部有一些关于如何处理代码的选项。



The screenshot shows a code editor interface with a dark theme. On the left is a vertical toolbar with icons for file operations (New, Open, Save, etc.) and file types (Python, JSON, etc.). The main area contains the following Python code:

```
if choice == '1':
    result = add(num1, num2)
    print("The result is " + str(result))

elif choice == '2':
    result = subtract(num1, num2)
    print("The result is " + str(result))

elif choice == '3':
    result = multiply(num1, num2)
    print("The result is " + str(result))

elif choice == '4':
    result = divide(num1, num2)
    print("The result is " + str(result))

else:
    print("Invalid input")
```

图4-10 在聊天系统的代码清单的顶部有一些关于如何处理代码的选项

其中的第一个图标用于复制代码；第二个图标用于将代码插入文件中光标所在的位置（也可以按Ctrl+Enter快捷键）。如果单击上面有3个点的图标，代码将被插入新的文件或输出到终端。

聊天系统提供了创建Jupyter Notebook的快捷方式“/newNotebook”。但你需要说明希望Jupyter Notebook做什么。

以下是一些提示范例。

**提示：**/newNotebook创建一个Jupyter Notebook，使用matplotlib将CSV文件中的数据可视化。

**提示：**/newNotebook创建一个Jupyter Notebook，预处理和清理机器学习数据集。

**提示：**/newNotebook创建一个Jupyter Notebook，使用scikit-learn实现简单的线性回归模型。

**提示：**/newNotebook创建一个Jupyter Notebook，用于分析时间序列数据。

**提示：**/newNotebook创建一个Jupyter Notebook，用于从网站上抓取数据并进行分析。

聊天系统还提供了“/terminal”快捷方式。有了它，你可以就终端或命令行操作提问或寻求帮助，例如导航目录、运行脚本或安装软件包。

以下是一些提示范例。

**提示：**/terminal列出所有环境变量的命令是什么？

**提示：**/terminal如何使用grep在多个文件中查找特定文本？

**提示：**/terminal如何使用find命令按权限搜索文件？

**提示：**/terminal如何将命令输出重定向到文件？

**提示：**/terminal如何使用awk处理文本文件？

接下来介绍快捷方式“@vscode”。有了它，你可以就VS Code的某些方面（如功能、导航、配置或扩展）提出问题或寻求帮助。

以下是一些提示范例。

**提示：**@vscode如何将编辑器分成多个窗口？

**提示：**@vscode如何定制键盘快捷键？

**提示：**@vscode如何设置多根工作区（multi-root workspace）？

**提示：**@vscode如何在VS Code中配置任务自动化？

**提示：**@vscode如何在VS Code中设置和使用Docker？

最后介绍快捷方式“/api”。有了它，你可以就API的开发或使用（包括创建、测试和集成API）提出问题或寻求帮助。

以下是一些提示范例。

**提示：**/api如何在Node.js中通过JSON数据发出POST请求？

**提示：**/api如何处理Express.js API中的CORS问题？

**提示：**/api如何在Django RESTful API中对用户进行身份验证？

**提示：**/api如何在Rails API中对结果进行分页？

**提示：**/api如何在ASP.NET Core中对API进行版本控制？

你还可以在VS Code上下文中使用快捷方式“/api”。

以下是一些提示范例。

**提示**：@vscode /api如何在VS Code扩展中创建新命令？

**提示**：@vscode /api如何读取和写入VS Code扩展中的设置？

**提示**：@vscode /api如何在VS Code扩展中创建新的webview面板？

**提示**：@vscode /api如何与VS Code扩展中的文件系统交互？

**提示**：@vscode /api如何将新的选项添加到VS Code扩展的活动栏中？

如果你想清除聊天内容，可以使用快捷方式“/clear”。

#### 4.2.5 内联聊天

在代码清单中，你可以便捷地使用聊天功能。首先高亮选择想要讨论的代码片段，然后在Windows操作系统中按Ctrl+I快捷键，或者在macOS中按Cmd+I快捷键，随后将弹出一个对话框，如图4-11所示。

The screenshot shows a GitHub Copilot Chat interface integrated into a code editor. At the top, there's a tab labeled 'test.py'. Below it, a file path is displayed: 'C: > Users > ttaul > OneDrive > Desktop > test.py > add'. A prominent feature is a floating 'Ask Copilot to edit code...' input field with a blue border. Below this, a message says 'You can also type / for commands'. The main code area shows three lines of Python code:

```
1 def add(x, y):  
2     return x + y  
3
```

图4-11 内联聊天功能允许你在代码清单中聊天

GitHub称，内联聊天已经成为一项广受欢迎的功能，因为它使开发者能够在工作流程中直接与Copilot Chat进行沟通。

利用前面提到的快捷方式，开发者可以就代码提出问题，而相关的解释和讨论将会直接显示在聊天面板上，开发者的交互体验更好。

需要注意的是，代码的旁边会有一个橙色闪光的图标。单击这个图标后，就会展开几个实用的选项，它们提供了修复代码、解释代码或重写代码的功能，如图4-12所示。

This screenshot is similar to Figure 4-11, showing the GitHub Copilot Chat interface. A horizontal line with an arrow points from the text '橙色闪光的图标' to the orange glowing icon located next to the second line of code ('return x + y'). The code itself is identical to Figure 4-11.

```
1 def add(x, y):  
2     return x + y  
3  
4 def subtract(x, y):  
5     return x - y  
6
```

图4-12 单击橙色闪光的图标即可直接与代码交互

下面我们尝试使用内联聊天功能。你可以使用以下提示来开始对话。

**提示：创建一个冒泡排序函数。**

这将指导聊天系统根据提示创建一个冒泡排序函数。现在，假设你想要探索是否存在更高效的排序方法。你可以通过选择已生成的冒泡排序函数，然后激活内联聊天功能来提出问题，如图4-13所示。

```
1 # 创建一个冒泡排序函数
2 def bubble_sort(arr):
3     n = len(arr)
4     for i in range(n):
5         for j in range(0, n-i-1):
6             if arr[j] > arr[j+1]:
7                 arr[j], arr[j+1] = arr[j+1], arr[j]
8
9
10
```

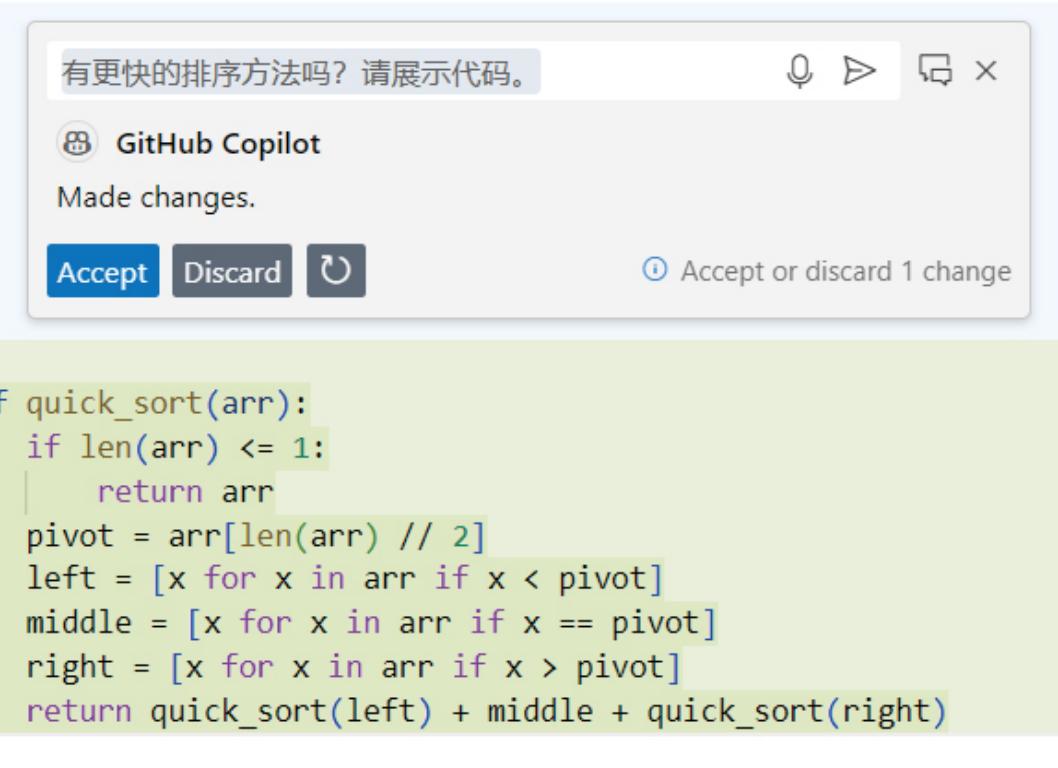


图4-13 内联聊天功能响应了你的请求，找到了比冒泡排序算法更快的排序算法

聊天系统提出了快速排序（Quick Sort）算法作为替代，并以差异对比模式（diff mode）展示了新旧代码之间的差异。你既可以接受全部新代码，也可以只接受部分代码。

#### 4.2.6 打开的标签页

在开发标准的软件项目时，IDE往往会涉及众多文件，每个文件都在应用程序的架构中承担着特定的功能。以一个采用React的Web项目为例，你可能会看到多个用于不同组件的.jsx文件和.js文件，以及用于页面结构的HTML文件和用于样式设计的CSS文件。此外，还有许多常见的配置文件，如.json文件和.env文件。

文件角色的区分为Copilot的工作提供了很大的帮助。Copilot对环境的理解越深入，它的表现就越出色。它会综合考虑当前编辑文件中的所有内容，并提供相应的辅助。至于其他打开的文件，LLM不会全部扫描，而是专注于与当前工作直接相关的部分。此外，Copilot 不会浏览项目中的其余文件（出于对隐私的尊重）。

#### 4.2.7 命令行界面

Copilot可以在CLI中使用。CLI的主要功能有两个——解释命令和提供建议。

为此，你需要先安装GitHub CLI，然后登录GitHub账户。

```
gh auth login
```

接下来安装Copilot。

```
gh extension install github/gh-copilot
```

也可以升级扩展Copilot。

```
gh extension upgrade gh-copilot
```

以下是请求Copilot解释CLI命令的提示。

**提示**：gh copilot explain xcopy

Copilot的响应如图4-14所示。

以下是让Copilot提供建议的提示。

**提示**：gh copilot suggest

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\ttaul> gh copilot explain xcopy
```

```
Welcome to GitHub Copilot in the CLI!
version 0.5.3-beta (2023-11-09)
```

```
I'm powered by AI, so surprises and mistakes are possible.
Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve.
```

#### **Explanation:**

- **xcopy** is a command used in Windows to copy files and directories.
  - It has various options and parameters that can be used to control the behavior of the copy operation.
  - Some commonly used options include:
    - **/s** - Copies directories and subdirectories, excluding empty ones.
    - **/e** - Copies directories and subdirectories, including empty ones.

图4-14 Copilot在CLI中解释了xcopy命令

接下来，Copilot会询问你需要哪一种命令的帮助，如通用shell命令、gh命令或git命令。然后，它会询问你希望该命令执行什么操作。下面是一个示例。

**提示：查看和设置系统环境变量的命令有哪些？**

图4-15显示了Copilot的响应，其中包括使用命令的建议。你可以复制提示或向Copilot寻求更多帮助，例如解释命令或修改命令。

? What kind of command can I help you with?  
> generic shell command

? What would you like the shell command to do?  
> 查看和设置系统环境变量的命令有哪些?

Suggestion:

查看系统环境变量的命令：  
- Linux/macOS: echo \$变量名  
- Windows: echo %变量名%

设置系统环境变量的命令：  
- Linux/macOS: export 变量名=值  
- Windows: setx 变量名 值

? Select an option [Use arrows to move, type to filter]  
> Copy command to clipboard  
Explain command  
Execute command  
Revise command  
Rate response  
Exit

图4-15 Copilot的响应中包含了使用命令的建议

## 4.3 Copilot合作伙伴计划

开发者经常利用第三方工具来丰富应用程序的功能，并从数据中提取更多洞见。以Splunk为例，这是一款广受欢迎的日志挖掘和数据分析工具。它为开发者提供了强大的搜索与报告功能，旨在帮助他们监控应用程序并迅速定位问题。

另一个例子是New Relic，它能确保应用程序顺畅运行。New Relic提供实时的洞察和诊断功能，使开发者能够发现并解决性能问题，从而提升应用程序的运行效率，优化用户体验。

Datadog也是一款关键工具，专注于监控基础设施和应用程序。通过其广泛的集成能力，Datadog使开发者能够从多个来源收集、聚合并展示数据，辅助他们在性能优化和资源管理方面做出明智的决策。

鉴于第三方系统的重要性，GitHub推出了Copilot合作伙伴计划。该计划的特点是不断扩大与流行集成平台的合作，允许所有平台通过扩展进行访问。

DataStax公司作为该计划的早期合作伙伴之一，是Astra DB的开发商。Astra DB是一个专为构建高级AI应用而设计的向量数据库。只需要通过一个扩展，开发者就可以利用Copilot Chat加快数据库的创建过程。例如，Copilot Chat能够根据编程标准为数据库的结构和元数据提供代码建议，使得开发者编写的代码更加整洁、更易于维护。

在Copilot Chat的辅助下，开发者编写的代码将变得更加简洁、规范，同时也更易于维护和理解。

可以使用快捷方式“@datastax”提出问题，以下是一些实用的提示。

**提示：**告诉我chat\_prod数据库的信息。

**提示：**告诉我数据表的模式。

**提示：**编写SQL查询以从聊天表中读取数据。



Mary-Brenda Akoda，一位AI研究员，在眼科人工智能领域拥有专利，并且是Copilot的忠实拥趸。她分享了自己使用Copilot的体验：“曾经有一次，我利用Copilot不到1小时就编写出整个模型开发流程的代码。这通常是一项需要投入更多时间和精力才能完成的任务。有趣的是，当我不得不在一个不支持Copilot的在线IDE中工作时，我才真正意识到没有Copilot，开发过程会是多么枯燥和缓慢。”

## 4.4 小结

Copilot迅速崛起，成为编程领域不可或缺的AI工具，显著提升了开发者的工作效率和代码质量。本章深入探讨了Copilot的核心功能：从智能建议复杂代码块的能力到可靠的安全特性，再到令人印象深刻的性能表现，以及Copilot在VS Code中的便捷使用体验。Copilot还引入了聊天和内联聊天功能，允许开发者在编程过程中与AI进行互动交流。

尽管Copilot仍须人工指导，但它无疑标志着AI驱动编程新时代的到来。

**[7]** 原英文图书从编写到出版已有一段时间，其间，像Copilot这样的工具可能已经进行过多次更新，其版本和费用政策也可能随之发生变化。作者在书中提供的版本详情和费用信息可能与当前最新的实际情况不完全一致。此处保留了作者提供的信息。——译者注

## 第5章

# 其他AI辅助编程工具

尽管GitHub Copilot在AI辅助编程领域处于领先地位，但市场上还有许多其他优秀的工具可供开发者选择。在日常开发工作中，开发者常常会结合使用多种AI编程辅助工具，以适应不同的编程需求和场景。

幸运的是，许多AI辅助编程工具受到GitHub Copilot的启发，采用了类似的设计，使得它们易于上手和使用。这些工具通常支持通过注释来触发命令，允许开发者使用快捷键迅速获得编程建议。此外，它们也普遍具备实用的聊天功能。

## 5.1 CodeWhisperer

Amazon的CodeWhisperer是一款多功能的扩展工具，它与VS Code、PyCharm及AWS Lambda等IDE高度集成，就像一把“瑞士军刀”。支持CodeWhisperer的LLM经过大量代码的训练，能够理解15种编程语言。Amazon在引入新的编程语言时非常注重准确性和安全性，而不仅仅追求上市速度。

CodeWhisperer有如下两个版本。

- 个人版：免费，可以使用AWS Builder ID或Amazon登录名注册。个人版提供代码建议，跟踪开源代码的来源，并进行安全检查。
- 专业版：每位用户每月收费19美元。专业版包含个人版的所有功能，并为大型企业提供额外的功能。专业版拥有权限管理系统，可以控制谁可以访问哪些内容，甚至可以管理所使用的开源数据类型。

专业版最引人注目的特性是其定制功能。企业可以根据自己的代码库、API和框架调整CodeWhisperer，以便提供更相关的代码建议。

经过微调，CodeWhisperer提供的代码质量更高，减少了在过时代码上的时间浪费。专业版允许用户设置最多8种不同的自定义功能。

CodeWhisperer的两个功能如下。

- Reference Tracker（引用跟踪器）：引用跟踪器密切关注代码，并与开源列表中的代码进行对比，显示开源代码在发布时所使用的许可证。这个功能有利于遵守法律和尊重知识产权。引用跟踪器可以显示项目资源库的注释、文件参考和许可证信息，有助于开发者根据企业的合规要求决定是否采纳代码建议。
- Security Scan（安全扫描）：安全扫描在IDE中运行，旨在检查包括开放式Web应用程序安全项目（Open Web Application Security Project，OWASP）的十大威胁和通用缺陷枚举（Common Weakness Enumeration，CWE）列表在内的漏洞。它甚至会遵循加密库的最佳实践。安全扫描不仅能发现漏洞，还会提供修复方法。它既能加快开发速度，又能降低成本。鉴于AI辅助编程工具生成的代码可能存在一些难以预料的问题，拥有这一内置的安全功能无疑是一个巨大的优势。

CodeWhisperer的另一个显著优点是能够与AWS服务无缝集成，这意味着开发者可以在不阅读大量文档的情况下快速成为AWS服务专家。

接下来，我们将通过几个示例来展示CodeWhisperer的能力。

**提示：**# 编写一个Python函数，用于将文件上传到S3存储桶。

**提示：**# 构造一个Python Lambda处理程序，用于处理来自Kinesis流的记录。

**提示：**# 编写一个Java方法，通过排序键从DynamoDB表中查询条目。

**提示：**# 编写一个Bash脚本，使用AWS CLI通过实例ID停止一个EC2实例。

**提示：**# 在TypeScript中通过AWS CDK创建带有PostgreSQL引擎的RDS实例。

Amazon对CodeWhisperer的用户进行了一项调研，结果表明，与不使用该工具的开发者相比，使用CodeWhisperer的开发者在任务成功完成率上平均提高了27%，同时完成任务的速度也平均提高了58%。

根据埃森哲咨询公司的调查，在AI项目中采用CodeWhisperer，开发时间显著缩短，节省的时间高达30%。数据预处理等任务往往既耗时又烦琐，而CodeWhisperer的应用大幅提升了这些工作的效率。开发者只须使用恰当的提示，便能显著提升工作效率。例如下面的提示。

**提示：**# 创建机器学习数据的预处理数据类脚本。

Persistent Systems是一家全球性的数字工程公司。通常，新入职该公司的开发者需要至少4个月的培训，才能充分准备好参与项目工作。然而，借助CodeWhisperer，这一培训时间被显著缩短到仅1个月。

## 5.2 Duet AI

Google是Duet AI的创造者，这是一款为开发者设计的AI辅助编程工具，支持包括VS Code、IntelliJ、PyCharm、GoLand、WebStorm、Cloud Workstations和Cloud Shell在内的多种IDE，并且兼容超过20种编程语言。Duet AI建立在Gemini（Google自研的先进LLM）之上。

Duet AI具备AI辅助编程系统的标准功能，例如内置的聊天功能。此外，Duet AI还提供智能快捷操作，可以快速生成代码解释和单元测试

等。

在安全性方面，Duet AI同样表现出色。它配备了包括专用端点（如 Private Google Access）、VPC服务控制和企业级访问控制在内的一系列加固保护措施。

Duet AI的治理规则确保了用户输入的代码不会被用于训练任何共享模型或用于构建产品，以保障用户对自己的数据和知识产权拥有完全的控制权。此外，Duet AI还能够识别出你从代码库中直接复制的代码实例。

Google Duet战略的一个关键部分是构建一个广泛的合作伙伴生态系统。一些合作伙伴在Google云上提供特定环境的支持。与合作伙伴的协作有助于Duet AI提供更精确的代码建议、文档和知识源。

下面是一些合作示例。

## **Elastic**

作为搜索分析领域的领导者，Elastic与Duet AI集成，使开发者能够在开发环境中直接查询、测试和处理Elastic数据。

## **HashiCorp**

HashiCorp是云基础设施自动化软件的顶级供应商。HashiCorp建立了一个系统，以方便开发者使用Terraform和HashiCorp配置语言（HashiCorp Configuration Language，HCL）编写配置和自动化脚本。

## **MongoDB**

作为NoSQL数据库的领军者，MongoDB与Duet AI的结合可以帮助开发者获得最佳实践，加速应用程序的开发。

## **Neo4j**

作为图数据库的领军者，Neo4j与Duet AI的结合可以提供Cypher查询编程语言的建议，并帮助在复杂的数据集中发现隐藏的关系和模式。

Duet AI已经内置于Google云控制台中，因而只需要单击Google云控制台右上方的激活按钮即可启用Duet AI。之后将弹出一个聊天面板，如图5-1所示，在这里可以输入创建脚本、分析日志或深入研究JSON等任务的提示。



-

## How do I connect to my data in Looker Studio?

To connect to your data in Looker Studio, you can follow these steps:

1. **Create a new report.** Click the "Create" button in the top right corner of the Looker Studio homepage.
2. **Select a data source.** In the "Data sources" panel, click the "Add data source" button. A list of available data sources will appear.
3. **Find your data source.** Scroll down the list of data sources and select the one you want to connect to.
4. **Enter your credentials.** You will be prompted to enter your credentials for the data source.
5. **Click "Connect."** Once you have entered your credentials, click the "Connect" button.
6. **Your data will now be imported into Looker Studio.** You can now start creating visualizations and reports.

Here are some additional tips for connecting to your data in Looker Studio:

- If you are having trouble connecting to your data source, you can check the Looker Studio documentation for more information.
- You can also contact Looker Studio support for help.
- If you are using a custom data source, you may need to create a custom connector.

图5-1 单击激活按钮后，可以在Duet AI聊天面板中处理相关任务

聊天功能则被集成到Google日志资源管理器

( Google Log Explorer ) 中。当你遇到难以理解的日志条目时，只需要选中它并单击Explain this Log ( 解释此日志 ) 按钮，该日志条目将自动跳转到Duet AI聊天面板，你可以在那里请求解释，甚至可以得到一些解决问题的建议。

Duet AI的收费标准为每位用户每月19美元，按年支付。Google云的新用户可以获得300美元的免费积分。



AI技术服务公司Turing AI一直在使用Duet AI，该公司开发团队的工作效率大幅提升了33%。

### 5.3 Tabnine

Tabnine是AI辅助编程领域的先驱。这家公司由Dror Weiss和Eran Yahav在2013年成立。自20世纪90年代以来，他们就开始涉足软件行业，专注于代码分析和仿真。然而，在亲身体验了复杂的软件开发工作后，他们开始思考：如果AI能够解决他们所面临的问题，软件开发的世界会变成什么样？

事实上，AI确实给他们提供了帮助。在那个还没有Transformer模型的时代，他们不得不亲自动手，构建属于自己的模型。这段经历让他们深刻理解了AI在软件开发中的潜力。让我们把时间快进到近几年，Tabnine已经采用了先进的Transformer模型。

Tabnine投入巨资打造了一个安全系统。该系统的工作原理是：当你输入代码时，每个字符都会被标记、加密，然后发送到Tabnine的推理服务器（这些服务器可以位于云端，也可以在本地）。Tabnine不

会存储客户的任何数据，且所有员工都无法访问客户数据。此外，Tabnine还符合SOC-2标准。

Tabnine有如下3个版本。

- 入门版：免费提供，主要面向个人用户，提供短代码补全和社区支持等基本功能。
- 专业版：每位用户每月12美元，专为专业开发者和小型团队设计，包括整行代码自动补全、自然语言到代码的自动补全等高级功能，并提供标准支持。
- 企业版：为需要全面安全、控制和定制的大型企业提供服务，提供无限用户支持、私有和自定义的AI模型、私有部署选项和高级支持等功能，收费标准根据需求而定。

Tabnine拥有庞大的用户基础，每月有超过100万用户，有时甚至每天有数十万用户使用Tabnine。值得一提的是，Tabnine的知名客户涵盖了Google、Amazon、Netflix和Atlassian等科技巨头。

## 5.4 Replit

Replit是一种基于Web的多功能IDE，支持多种编程语言，并且允许用户托管应用程序。Replit提供了类似Google文档的丰富协作功能，使得多用户可以实时协同工作。为了扩大覆盖范围，Replit还推出了适用于macOS、Windows、Linux、Android和iOS等操作系统的桌面版本。Replit拥有一个庞大且活跃的开发者社区，大约有2300万用户。

Replit公司由Amjad Masad、Faris Masad和Haya Odeh在2016年共同创立。Amjad在Replit公司成立十多年前就开始构思Replit的理念。在创立Replit公司之前，Amjad曾在雅虎和Facebook（现Meta）担任工程师，专注于开发工具的构建。他还参与创办了著名的

编程教育平台Codecademy，这进一步证明了他对创新技术和编程教育的热爱。

Replit有如下3个版本。

- 免费版：提供无限的公共REPL（Read-Eval-Print Loop）和10GB存储空间。
- 黑客版：每位用户每月7美元，提供无限制的私人REPL。黑客版还根据用户需求提供不同级别的内存和存储空间。
- 专业版：每位用户每月20美元，包含黑客版的所有功能。专业版用户还可以享受最强大的AI模型和高级功能，以及更高效且功能丰富的代码开发体验。

Replit的营销和设计副总裁David Hoang说：“Replit是一个一体化的软件创作平台。”他还提到：“Replit的产品体验旨在减少软件开发过程中的摩擦——从开发环境配置、代码编写到在生产环境中部署。正因为如此，Replit能够在代码生成之外的环节集成AI。我们的代码补全模型也是由自研的语言模型驱动的。”

Replit的LLM经过了高达1万亿个token的训练，能够理解30种编程语言。当你在Replit上工作时，你在公共编程环境中编写的代码（包括按键操作，即keystroke<sup>注</sup>）可能会被用于进一步训练Replit AI模型。如果你不希望自己的工作成果被用于训练，可以使用私人REPL。需要注意的是，在使用Replit时，你对代码的所有权不会改变。正如Replit的许可信息所示，公共编程环境中的代码自动采用MIT许可证。

图5-2显示了Replit的主界面：左上角是文件树和搜索框，中间是编辑器，输出（如控制台）显示在右侧，左下方是可用工具栏。如果选择AI选项卡，聊天功能就会出现在右侧面板中。

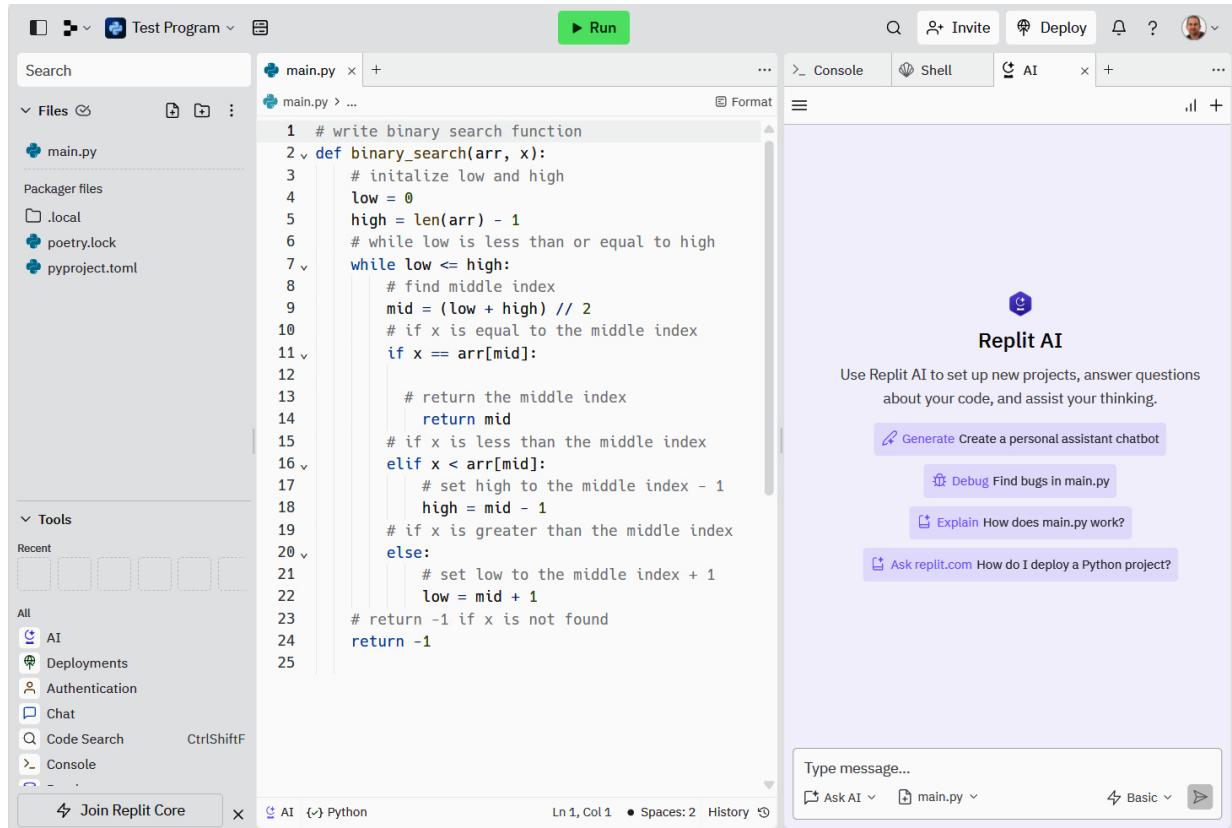


图5-2 Replit的主界面

在使用Replit时，如果选中一些代码并右击，将会出现以下选项。

## Explain（解释代码）

Replit提供了解释功能，以帮助你理解代码。如果首次解释得不够详尽，你可以选择“重试”，让Replit再解释一次。此外，如果你有具体的问题要咨询，则可以从Explain（解释代码）菜单顶部的下拉列表中选择Ask Chat（智能聊天）选项。Replit还提供了其他辅助开发者编程的有用功能，如询问文档、编辑代码和生成代码等。

## Modify（修改代码）

在Replit中，你可以提示Replit对代码进行修改。例如，如果你选中了一段二进制搜索的代码，并希望将其转换为JavaScript函数，则可以编写一个提示给Replit。你也可以修改代码的实现逻辑，比如要求

将代码修改为使用递归方法。Replit的代码修改功能提供了一种灵活且交互性强的方式来帮助开发者完善代码。

以下两个有趣的故事展示了Replit的易用性和强大功能。

## 血糖水平监测应用

当伦敦的iOS开发者Marwan Elwaraki得知弟弟被诊断出糖尿病时，他决心开发一款用于监测血糖水平的应用。他和妻子Salwa Al Alami希望联手打造一个实用而独特的应用。Marwan说：“在飞机上，我制作了一个锁屏小部件原型，将这个小部件与血糖跟踪器的API连接，就能看到我弟弟的血糖水平，这种体验比打开应用查看最新数据要好得多。”

他们又添加了发送信息的按钮，并改进了短信功能。然后，他们在iOS应用商店公开发布了这个应用。虽然Marwan是一名经验丰富的移动前端开发者，但他在Python或后台开发方面的经验不足，而Salwa是一名没有编程经验的产品经理。起初，他们使用ChatGPT辅助编程，在引入Replit后，情况发生了巨大变化。Marwan说：“AI辅助编程工具不仅帮助我们生成了部分代码，而且仔细检查了几乎所有的后台代码。我听说过开发者在AI的辅助下工作效率得到大幅提升的故事。现在，通过亲身体验，我可以证实这个观点。”

## 从AI黑客马拉松到初创公司

Priyaa Kalyanaraman令人印象深刻，她曾在微软、Snapchat和Waymo担任产品经理。尽管缺乏技术背景，但她并没有因此退缩。她参加了Craft Ventures的AI黑客马拉松，并利用Replit开发了一个旨在简化内容创作并为创作增添乐趣的应用。她赢得了10000美元的大奖，她的努力得到丰厚的回报。她利用自身在产品管理方面的专业知识，精心准备了详细的规格说明，并合乎逻辑地构建了应用。她获奖的那个应用后来成为她的初创公司Lica的基础，她还因此成功获得了种子轮融资。

## 5.5 CodeGPT

CodeGPT是专为VS Code设计的扩展，其提供了一系列定价选项以满足不同用户的需求，涵盖从基础的免费版到功能丰富的高级版，高级版的收费标准为每位用户每月49.99美元。此外，CodeGPT还提供10天的免费试用期，用户有机会在决定是否订购前先体验一下服务。

设置CodeGPT扩展的方法是：打开VS Code，单击File（文件）菜单，导航至Preference（首选项）并选择Settings（设置）命令。在打开的设置窗口中，单击左侧的Extension（扩展）菜单，然后选择CodeGPT进行配置。

CodeGPT的一个显著特点就是与多种LLM集成的能力。CodeGPT支持与OpenAI、Cohere、AI21、Anthropic等公司的模型进行集成。用户只需要在Hugging Face上创建一个API账户，便能够访问广泛的开源模型，从而增强CodeGPT的能力。

选择模型后，用户可以自定义以下配置。

- 提示和响应的最大token数。
- 温度值，用于控制所生成文本的创造性。
- 聊天窗口的内存容量，即一个聊天中过往对话的数量。

CodeGPT还提供了一个有趣的功能——API服务。API对于需要理解和生成类人文本的系统（如聊天机器人、虚拟助手等应用）非常实用。CodeGPT的RESTful API设计确保了CodeGPT与其他各种平台的广泛兼容性和集成能力。此外，CodeGPT还提供Python和JavaScript的SDK，增强了实施的灵活性。CodeGPT的API是更广泛的编程流程变革计划的一部分，旨在为各种开发项目提供先进的AI辅助编程工具。

## 5.6 Cody

2013年，Quinn Slack和Beyang Liu共同创立了Sourcegraph公司，专注于开发代码搜索工具。Beyang的构想源自他在Google的工作经验，他在那里亲眼见证了Google内部平台在代码智能和洞察方面的强大优势，特别是在处理庞大的代码库时。

Sourcegraph公司的核心目标是打造能够处理“海量代码”的先进工具。该领域面临的一个重大挑战是复杂性。Sourcegraph公司所做的一项调查显示：约77%的开发者表示，他们的代码库在3年内增长了5倍。更有甚者，72%的受访者表示担心，不断增长的代码规模可能会威胁到公司的创新能力和竞争力。

AI在Sourcegraph公司的战略中扮演着至关重要的角色。该公司开发了一个名为Cody的AI驱动的代码生成系统。Cody依赖于Anthropic和OpenAI的LLM，并结合了Starcoder的能力。

Beyang指出：“我们利用自身的代码搜索能力来打造Cody。开发者80%的时间用来阅读和理解代码，而不是编写代码。Cody不仅能扫描整个代码库，还能轻松地引入其他库和框架，因此开发者能得到更好的结果。”

以下是在Cody平台上使用的提示范例。

**提示：**存储库的结构是怎样的？

**提示：**这个文件有什么作用？

**提示：**X组件在哪里定义？

Cody有3个版本——免费版、专业版和企业版。

**免费版：**提供代码自动补全、聊天和上下文感知等功能。

专业版：每位用户每月9美元，相应功能如下。

- 无限制的代码自动补全、消息和命令。
- 大型代码库的个性化。
- 多种LLM的聊天选择。
- 更高级的支持。

企业版：可以针对大型组织的需求进行定制，每位用户每月19美元，相应功能如下。

- 用户管理。
- 单租户部署。
- 审计日志。
- 组织使用情况汇总。
- 测试期间的每日费率限制。
- Web和API访问支持。
- 可配置的LLM。

为了更好地了解Cody的潜力，让我们来看看Cody的全栈开发者和开源贡献者Deepak Kumar的使用体验。他说：“我最喜欢的两个功能是聊天和命令。我使用聊天功能来咨询所有与编程相关的问题，比如寻求启动项目的初始想法、修复错误的方法等。命令是预定义的提示，它们在完成编写文档和添加测试等任务方面非常有用。”

2021年7月，Sourcegraph公司宣布在D轮融资中以26.25亿美元的估值获得1.25亿美元的投资，投资者包括Andreessen Horowitz、Insight Partners和Geodesic Capital。

## 5.7 CodeWP

WordPress是一个非常流行的开放源代码内容管理系统（Content Management System，CMS），以简洁性、多功能性、丰富的主题和插件选择而闻名。最初作为博客平台推出的WordPress，目前已经扩展应用到各种类型的网站。WordPress的吸引力在于其友好的用户界面和高度可定制化的特点。根据W3Techs的报告，互联网上约有45.8%的网站由WordPress提供支持。

AI辅助编程为WordPress开发带来了巨大的机遇，CodeWP充分把握住了这个机遇。CodeWP由James LePage创立，他在高中时就已经创建了一家WordPress代理公司。虽然公司业务迅速发展，但工作强度较大。在引入AI辅助编程技术后，工作效率得到了显著提升。James对WordPress开发中的常见用例有深刻的理解，他的洞察力进一步提升了AI的效率。

CodeWP是一个网络平台，它有3个版本——免费版、专业版和代理版。专业版和代理版按年付费可享受最高33%的折扣。

- 专业版：每位用户每月18美元，提供额外的AI事务、项目和支持。
- 代理版：面向更大规模的需求，每位用户每月48美元。代理版面向专业网站开发和营销公司，提供无限项目和协作功能。

使用CodeWP的流程如下。

- (1) 访问CodeWP网站并创建账户。
- (2) 单击Create New按钮，然后选择New Snippet命令来创建代码。
- (3) 如果想咨询问题，则可以单击New Chat按钮。

CodeWP的界面布局如下。

- 界面左侧包含用于存储代码片段、跟踪对话和列出项目的选项。
- 中间是编辑器，用于显示正在处理的文件内容。如果要创建新文件，则可以单击New File按钮。
- 界面右侧是聊天功能区。

下面是一些提示范例。

**提示：**注册一个名为“book”的自定义帖子类型，标签为“Books”。

**提示：**使用WP\_Widget类创建一个名为“My Widget”的WordPress小部件。

**提示：**为register\_activation\_hook函数创建自定义插件激活钩子。

CodeWP不仅支持WordPress生态系统中的关键组件，同时也考虑到与WooCommerce、WPSimplePay、Gravity Forms、SearchWP和Contact Form 7等主要工具的兼容性。

CodeWP具有实时预览选项，可即时查看代码的运行情况，简化测试和调试过程。此外，在开发WordPress项目时，CodeWP还能简化部署过程。CodeWP的集成工具可以轻松地将代码传输到WordPress网站，从而更高效地在实际环境中部署和测试代码。

## 5.8 Warp

CLI起源于20世纪60年代，并随着Multics和UNIX等操作系统的发  
展而流行。尽管图形用户界面(Graphical User Interface, GUI)  
后来以美观和操作便捷居上，但CLI依然保持其简洁本质。CLI的核心  
理念是简单性：通过输入纯文本命令获取所需结果，不附加任何多余  
功能。

CLI因其简单性而广受欢迎。它们对于管理云系统、处理计算机上的文件和程序，以及构建高效的工作流程至关重要。CLI擅长的领域包括自动化任务、数据处理、网络管理和软件开发等。

掌握CLI不仅能加快工作进度，还能提高准确性。CLI的一个显著优点是能够组合不同的命令来处理复杂任务。此外，一旦学会了命令行编程，开发者就有更大的灵活性来根据个人喜好调整程序和自动化任务。这意味着开发者可以根据个人需求定制任务，使工作流程更加顺畅和高效。

CLI尽管功能强大，但它并非没有缺点。CLI不适合协作，广泛使用CLI可能会影响工作效率和团队合作。此外，一旦关闭终端会话，所有内容都会丢失。CLI通常受限于单一机器，跨设备切换非常不便。有时，面对复杂任务，使用CLI可能达不到理想效果。而且对于许多人，熟练掌握CLI命令及其语法并非易事。

因此，CLI领域存在创新空间。Warp公司就是一家引领创新的初创企业。

Warp公司由创始人兼首席执行官Zach Lloyd于2020年创立。在创立Warp公司之前，Lloyd曾在Google担任Google Sheets的首席工程师。Lloyd表示：“作为一个有着二十多年经验的开发者，我深刻体会到CLI的强大功能。它对开发者非常有用，但仍有改进的空间。”

Warp是使用Rust构建的。Rust是一门以快速和高效著称的编程语言。与传统的JavaScript不同，Warp利用Rust的优势，即使在浏览器中直接运行，也能提供类似本地应用程序的体验。对Rust的巧妙应用意味着用户在使用Warp时能获得更流畅的体验，从而改善与整个应用程序的交互。

想象一下，Warp是一个超级终端。它具有先进的文本输入功能，如选择、光标定位和自动完成菜单，允许用户无缝浏览命令历史记录，这

消除了滚动浏览长文本的不便。此外，Warp还使复制输出变得轻而易举。

在Warp中，每个命令及其输出都构成一个独立的块。你可以毫不费力地将鼠标指针悬停在这样的块上，回顾过去的命令。选择一个块，右击并在弹出菜单中选择Warp AI，即可获得针对错误的解释和解决方案。Warp与传统的CLI形成了鲜明对比，传统CLI提供的指导非常有限，主要依赖用户的专业知识来解读错误信息和排除故障。

Warp还具有聊天功能。你可以通过以#开头的提示来使用它。假设你想了解以下信息。

**提示：# 如何使用grep在递归搜索中排除目录？**

你可以将Warp给出的答案复制到终端。

值得一提的是，Lloyd也是Copilot的忠实用户，他也使用Copilot辅助开发。他说：“我在使用Warp的同时也使用Copilot。两者配合得非常好。”

Instagram的联合创始人Mike Krieger是Warp的忠实粉丝。他说：“我每天上班都在使用Warp。我最喜欢的是它的运行速度。我尤其喜欢它出色的超前输入和搜索功能。Warp将终端带入了现代，我迫不及待地想看终端技术会将Wrap带到何方。”

Warp可以运行在Linux和Windows操作系统及macOS上。它有3个版本——免费版、团队版和企业版。

- 免费版**：每天可提供多达20个AI请求。
- 团队版**：每位用户每月12美元，每天最多提供100个AI请求。
- 企业版**：面向有特殊要求的大型机构，收费标准根据客户需求而定。

## 5.9 Bito AI

2006年，Amar Goel和Anand Das共同创立了在线广告公司PubMatic，该公司于2020年成功上市。他们在创业过程中，深刻认识到AI在提升编程效率方面的潜力，这激发了他们创立Bito AI的灵感。Mukesh Agarwal，曾在微软和安永担任产品负责人，也加入了Bito AI的创始团队。

Bito AI具备多语言支持能力，能够支持20种语言，包括独特的混合语言，例如印地语和英语的混合语言Hinglish。

Bito AI拥有多个显著的功能：允许用户为常用的编程结构创建自定义提示模板，这极大地简化了开发流程；支持进行安全和性能检查，并提供优化建议，以提升代码的安全性和效率。

Bito AI拥有一个庞大的上下文窗口，能够处理约240000个token，这增强了其理解和分析大型代码库的能力。Bito AI在用户的机器上使用本地向量数据库，以确保生成的结果更加相关和准确。这一特性显著提升了Bito AI在管理大型编程项目时的效率和有效性。

Bito AI还能很好地处理内存泄漏。当程序占用计算机的部分内存却忘记释放时，就会发生内存泄漏。如果内存泄漏持续发生，就会令人非常头疼。程序不断运行，内存不断泄漏，一直侵蚀可用内存总量。持续的内存泄漏会导致系统运行速度变慢，因为物理内存不够用，系统开始使用交换空间。如果内存泄漏情况非常严重，甚至有可能导致程序或整台计算机因内存耗尽而崩溃。

内存泄漏的发现和修复都很棘手，因为它们可能在程序运行一段时间后才显现。这种延迟使得找出并解决它们变得更加困难。Bito AI能够检测出潜在的内存泄漏。方法很简单：使用IDE中的Insert Code Selected选项，插入如下提示。

**提示：**找出代码中的任何问题。

使用这个提示后，Bito AI将不仅能够标记出问题，还能提出修复建议。

在隐私方面，Bito AI不存储任何实际代码。但是，Bito AI会保留期间生成的元数据。

Bito AI有两个版本——免费版和付费版。

·免费版：针对个人用户。

·付费版：每位用户每月15美元。付费版提供无限制的AI代码补全和访问高达240000个token的上下文窗口等功能，从而增强了整体编程体验。



根据Bito AI的内部分析，使用Bito AI的开发者表示编程效率提升了31%。这些用户平均每月使用Bito AI近200次。目前，Bito AI已经拥有大约10万名活跃用户。

## 5.10 Cursor

Anysphere公司开发了创新性编程工具Cursor。这家公司对其产品有着宏伟的愿景：

未来几年，我们希望打造一个前所未有的代码编辑器——一款更有用、更令人愉悦、更有趣的工具。Cursor是一款不可能写出错误的工具。它是一个只需要50行伪代码就能编写出2000行拉取请求的编辑器。它还是一个能立即解答任何代码库问题的智能系统，甚至是一个可能让源代码变得不那么必要的界面……

Cursor以其强大的功能吸引了包括OpenAI在内的投资者的关注。2023年10月，Anysphere宣布成功融资800万美元，这家公司的总

融资额达到1100万美元。

Cursor支持GPT-4和GPT-3.5，并能够通过AnySphere的专有模型来增强能力。这两个大模型拥有14亿个向量和对15万个代码库的理解，还采用了Merkle树等先进AI技术，从而进一步提升效率。

作为VS Code的一个分支，Cursor兼容Windows、Linux操作系统及macOS，这使得用户无须学习新的IDE，就能够方便地导入现有的VS Code扩展。此外，你也可以在CLI中直接利用Cursor的AI能力，将高级功能无缝集成到用户熟悉的编程环境中。

例如，在聊天区域，你可以轻松导入文档并提出问题。

**提示：**@python docs

你也可以使用@来处理特定文件。

**提示：**@the\_app.ts有什么作用？

Cursor支持在终端中直接调试代码。它的AI系统会筛选文件，并委托一个逻辑进程来定位并尝试解决问题，从而简化故障排除过程。

Cursor的一位用户Jeffrey Biles，一位全栈网络开发者，分享了他对Cursor的体验：

我非常喜欢通过按键将应用程序上下文快速加载到AI系统中。它在回答问题、记忆语法、重构代码片段和编写模板时非常有效。尽管当涉及理解跨多个文件的大型数据模型时，它可能达到极限。但随着模型不断改进，这些极限将被不断突破，例如使用更大的上下文窗口。

Cursor的另一位用户Tosh Velaga，一名软件工程师，同时是多家AI公司的创始人，如Typeblock和STBL（STBL开发了Stable Diffusion），也分享了他对Cursor的体验：

我非常喜欢的一个功能是Command+L快捷键，它允许我询问有关代码的问题。当我在新的代码库中使用不熟悉的后端编程语言时，这个功能极其有用，它能帮助我更快地上手。另外，我可以使用Cursor自带的API密钥来节省成本，并能够在GPT-3.5和GPT-4之间进行切换。

## 5.11 Code Llama

开源项目在AI辅助编程领域受到的关注度日益增加，这在学术界同样成了一个热门议题。Meta作为该领域的领军企业之一，自2023年8月推出Code Llama以来，不断引领着行业潮流。

Code Llama建立在LLaMA 2的基础上，LLaMA 2经过了惊人的2万亿个token的训练，拥有4096个token的上下文长度。LLaMA 2包含一些经过微调的模型，这些模型在超过100万个人类标注的训练下得到加强，其有效性和可靠性均得到提高。

LLaMA 2有多个特殊版本，如LLaMA 2-Chat，它是专门为聊天应用设计的，在关键的基准测试中，它在有效性和安全性方面优于其他开源聊天模型。此外，LLaMA 2还与Hugging Face等平台进行集成，并与微软和Amazon等公司建立了重要的云服务部署合作伙伴关系。

Code Llama支持包括Python、C++、Java、PHP、JavaScript、C#和Bash在内的多种主流编程语言。它提供了3种不同规模的模型——7B模型、13B模型和34B模型，分别拥有70亿、130亿和340亿个参数，所有这些模型都是在5000亿行代码上训练出来的。7B模型和13B模型的特别之处在于它们具有中间填充能力，能够将代码插入现有脚本中，特别适用于代码补全等任务。7B模型可以在单个GPU上运行，通常速度更快。而34B模型则在性能上具有巨大的冲击力。

Code Llama的功能不仅限于LLaMA 2基本模型。Code Llama的所有模型都能处理高达100000个token的上下文，这对编写较长的程序非常有用。更大的上下文窗口使得Code Llama成为调试的有力助手。

Code Llama有两个特殊版本。首先是Python版本，它通过涵盖1000亿个token的代码进行了优化。其次是指令版本，它接受了额外的自然语言训练，以提高其在自然语言应用中的表现。

34B模型的性能给人留下了深刻印象。它在HumanEval上获得了53.7%的高分，在MBPP上获得了56.2%的分数，这些成绩都优于其他开源模型，并且与ChatGPT的表现相近。

此外，Code Llama还通过了安全测试。Meta指出，Code Llama的响应甚至比ChatGPT更安全。

## 5.12 其他开源模型

在开源代码生成领域，近期涌现出一些令人瞩目的创新。以下是几个表现突出的平台。

### 5.12.1 StableCode

Stability AI公司是备受瞩目的文生图系统Stable Diffusion的开发者，该公司还推出了StableCode，这是一款基于开源项目BigCode训练的代码生成工具。BigCode是由Hugging Face和ServiceNow Research合作开发的一个项目，它们创建了一个名为The Stack的庞大数据集，这个6.4TB的数据集包含了独特且可免费使用的GitHub代码，非常适合用来训练AI模型。

StableCode采用了一种独特的技术，它使用旋转位置嵌入（Roary Position Embedding，RoPE）技术，而不是传统的线性偏置注意力（Attention with Linear Biases，ALiBi）方法。

StableCode经过了大规模的微调，包括数据清洗，已成为一款多用途的工具，支持包括Python、Go、Java、JavaScript、C、Markdown和C++在内的多种编程语言。

StableCode有如下3个版本。

### **StableCode-Completion-Alpha-3B-4k**

这是一个仅包含30亿个参数的解码器，适合处理多种编程语言，能够处理长达4000个token的上下文。

### **StableCode-Instruct-Alpha-3B**

这个版本包含30亿个参数，针对指令进行了优化。

### **StableCode-Completion-Alpha-3B**

这个版本包含30亿个参数，在执行大量代码补全方面表现出色，可处理长达16000个token的上下文。

## **5.12.2 AlphaCode**

DeepMind，Google旗下的AI公司，开发了名为AlphaCode的先进AI系统。AlphaCode在编程竞赛领域取得了十分优异的成绩，在由Codeforces（一个知名的编程竞赛平台）举办的编程竞赛中，AlphaCode的得分排在前54%。AlphaCode能够处理各种艰巨的任务，处理这些任务需要批判性思维、逻辑推理、算法知识、编程技巧及对自然语言的理解。AlphaCode的能力甚至获得《科学》杂志的称赞。

编程竞赛与日常编程任务不同，前者通常包含复杂的算法挑战和理论问题。以AlphaCode面临的一个名为Backspace的难题为例，它要求参赛者在两个字符串s和t上工作，通过使用退格键将字符串s转换为字符串t。这个问题要求参赛者提出巧妙的算法，展示了竞技编程所需的高级问题解决技能。

DeepMind的目标是通过AlphaCode实现完全自动化的代码生成，这标志着AI在编程领域的技术进步达到一个新的里程碑。

### 5.12.3 PolyCoder

PolyCoder是一款专为加速C语言编程而设计的AI工具，尤其受到游戏开发者的青睐。它不仅能够提升C程序员的编程效率，还擅长对多种编程语言的代码进行总结和理解。PolyCoder在多个不同的数据集上接受了训练，能够精准捕捉C、C++、Java、Python和JavaScript等多种编程语言中的常见模式和结构。

在高达249GB的数据集上经过密集训练后，PolyCoder掌握了12种不同编程语言的精髓。此外，它拥有27亿个参数，这为其提供了强大的能力，以理解和生成高质量的代码。

### 5.12.4 CodeT5

CodeT5采用了一体化的预训练编码器-解码器模型架构，该架构赋予了它处理多样化编程任务的能力。这种设计让它不仅能够识别代码中的错误和检测代码副本，还能够处理新结构带来的挑战，例如将编程语言转换为纯英文，或进行反向操作，甚至在不同的编程语言之间进行转换。

CodeT5的设计融入了一些特殊的技术，这些技术能够充分利用编程语言的深层结构，从而增强CodeT5对代码的正确理解和输出能力。

在CodeXGLUE ( CODE通用语言理解评估基准 ) 的14个不同挑战中，CodeT5展现出卓越的性能，它在所有生成任务上都超越了PLBART等先前的模型，包括代码总结、将文本转换为代码、代码翻译以及代码美化。同时，在理解任务方面，CodeT5在查找错误和检测代码副本方面同样表现出色。

### 5.12.5 软件公司开发的模型

一些大的软件公司（如SAP、ServiceNow和Salesforce）一直在积极开发自己的代码生成系统。这些公司拥有显著的优势：庞大的客户基础、丰富的资源、广泛的分销网络和值得信赖的品牌。此外，他们拥有庞大的人才库，并且在市场上处于领导地位，这使得他们能够引领潮流和制定行业标准。

通过将新技术与现有产品相结合，这些公司能够提供全面的解决方案，吸引并保留客户。他们已经具备的基础设施和全球业务布局，使得新技术能够迅速扩展，满足当前和未来市场的需求。

以下是一些由这些公司开发的代码生成系统。

## **Salesforce Code Builder**

Salesforce Code Builder使开发者能够轻松定制Salesforce中的客户关系管理功能及其他小工具，支持使用最新的Salesforce语言和框架，可以确保开发过程顺畅。此外，开发者可以直接从开放的VSX市场获取第三方插件。VSX是VS Code扩展的开放平台。更加方便的是，这些插件无须安装或配置就可以立即使用。

## **SAP Build Code**

SAP Build Code专为Java和JavaScript用户设计。此外，它还能够与SAP Joule集成，并支持用户利用SAP的高性能分析设备（High-performance ANalytic Appliance，HANA）等先进的数据库系统。

## **StarCoder**

ServiceNow和Hugging Face合作完成了开源项目StarCoder。StarCoder功能强大，拥有150亿个参数，支持80多种编程语言，并经过了万亿级别的token训练。StarCoder在管理、安全性和合规性方面非常严格，只训练经过授权的代码。

## 5.13 小结

AI辅助编程工具正引领着软件开发领域的一场变革，为开发者带来了一系列创新功能。科技巨头（如Amazon、Google、Meta和Salesforce等）都积极投身于这一变革。与此同时，我们也不应忽视那些小型但富有创新精神的初创公司，它们以精巧的功能和亲民的价格赢得了市场的关注。

开源项目同样在这一领域发挥着重要作用，它们提供了顶尖的AI辅助编程工具，为整个行业的发展贡献力量。

随着AI模型的不断壮大和智能化，它们在代码编写、解释和优化方面的能力也在显著增强。展望未来，我们有理由相信，那些能够成功将AI工具无缝整合到开发者工作流程中的企业，将在市场上占据优势，推动软件开发不断创新和进步。

**[8]**在Replit中，keystroke是指在键盘上按键，将文本或命令输入Replit编辑器或终端的行为。例如，当你在Python文件中输入“print('Hello, world!')”时，需要多次击键才能输入每个字符和符号。——译者注

## 第6章

# ChatGPT和其他通用LLM

本章将深入探讨一些被广泛采用的LLM，它们同样具备编写代码的能力。虽然这些模型可能不具备专业辅助编程模型的所有功能，例如与IDE的集成、全面的安全保障措施，以及用于防止模型训练代码泄露的保护机制，但它们在编程领域的应用仍然具有划时代的意义。此外，这些模型在软件规划和头脑风暴方面也显示出巨大的潜力。

本章将详细介绍ChatGPT、Gemini和Claude等知名的聊天机器人，包括它们的设置方法、成本结构、所拥有的创新功能，以及如何有效利用这些工具来最大化它们的效用。

## 6.1 ChatGPT

ChatGPT ( Chat Generative Pretrained Transformer ) 于2022年11月30日公开发布。尽管OpenAI作为背后推手并没有在营销上进行大规模宣传，但ChatGPT凭借卓越的性能迅速赢得广泛关注。实际上，OpenAI没有必要进行额外宣传。ChatGPT强大的实力足以让它在短时间内声名鹊起。

在推出后的短短5天，ChatGPT就吸引了100万用户。仅仅几个月的时间，ChatGPT的用户规模就急剧增长至1亿，成为互联网历史上用户规模增长最快的平台。

OpenAI的策略制定者Sandhini Agarwal指出：“对我们所有人来说，有这么多人开始使用ChatGPT确实是一个惊喜。我们在模型上投入了大量的精力，但有时会忘记它们给外界带来的震撼。”

截至2023年9月，ChatGPT的访问量已超过15亿次，收入也从2022年的2800万美元激增至12亿美元。然而，这一增长并不仅仅由ChatGPT自身驱动，OpenAI的API业务也在快速增长。

接下来，我们将简要介绍OpenAI的成长背景。OpenAI成立于2015年，其得到了硅谷重量级人物的支持，包括Elon Musk、Sam Altman、Greg Brockman、Ilya Sutskever、John Schulman和Wojciech Zaremba。他们计划投入高达10亿美元的资金来推动OpenAI的发展。

OpenAI最初是一个非营利组织，拥有浓厚的学术氛围，员工大多是数据科学和人工智能领域的博士。OpenAI的使命是实现通用人工智能（Artificial General Intelligence，AGI），也就是让机器在大多数重要任务中超越人类的智能。OpenAI的目标包含利他主义元素：利用AGI造福更广泛的人群，并避免任何可能造成伤害或赋予少数人过多权力的应用。

早期的OpenAI专注于研究并分享其研究成果。OpenAI曾与其他机构合作，公开其专利和代码。然而，开发尖端的生成式AI系统的成本是巨大的。

为了维持其宏伟愿景并吸引合适的人才，OpenAI在2019年转型为一家营利性公司。这使他们能够合法地从风险投资基金和战略投资者那里筹集资金，同时也能够为员工提供激励。同年，微软公司投资了10亿美元。在随后的几轮投资中，微软公司的总投资额达到约30亿美元。2023年4月，微软公司进一步加大投资力度，对OpenAI投入高达100亿美元的资金。

## 6.2 GPT-4

OpenAI的GPT-4在代码生成方面表现出卓越的多功能性，这得益于其广泛而多样化的训练数据。这种广泛的训练让GPT-4在多个专业和

学术基准测试中的表现达到与人类相媲美的水平，而且在大多数编程语言中，GPT-4的表现都优于其前身GPT-3和GPT-3.5。这一成就主要归功于GPT-4在理解和执行自然语言中的复杂指令，以及生成更深入的技术或创意作品方面的能力得到显著增强。此外，GPT-4拥有的32K上下文窗口也是其成功的关键因素之一。

特别值得一提的是，在针对特定代码生成基准的评估中，集成了Reflexion框架的增强版GPT-4在HumanEval基准测试上的通过率高达88%，这一成绩明显超过基础版本的GPT-4，后者的通过率为67%。这一结果表明，经过适当的改进，GPT-4完全有能力在代码生成任务中展现出顶尖水平。

在与Codeforces平台上的竞技编程场景进行比较时，GPT-4的得分为392分，相较于GPT-3.5的260分有所提高。尽管GPT-4在竞技编程领域取得了显著进步，但其得分仍然位于倒数5%的行列。这说明GPT-4虽然在编程竞赛方面已经取得长足的进步，但要达到与人类技能相匹敌的水平，仍有很长的路要走。

## 6.3 访问ChatGPT

本节将向你展示ChatGPT的高级版本——ChatGPT Plus，它是一个每月需要付费20美元的订阅服务。ChatGPT Plus具有如下功能。

### 始终在线

即使在高峰时段，所有用户都在使用ChatGPT时，ChatGPT Plus用户也能获得优先访问权。

### 快速回复

对于ChatGPT Plus用户，所有的响应都变得更迅速。这意味着可以更快地获得答案，享受更流畅、更愉快的聊天体验。

## **优先体验新功能**

当有令人兴奋的新功能或升级推出时，ChatGPT Plus用户将有机会率先尝试。

## **访问最新模型**

ChatGPT Plus用户可以访问OpenAI最新的模型，从而始终处在技术进步的最前沿。

图6-1展示了ChatGPT的界面。

在界面的左上方，将鼠标指针悬停在下拉菜单上，便可以选择所需的模型或访问插件商店。

在界面的左侧，用户可以通过单击按钮快速创建新的聊天会话。

界面的底部展示了用户配置文件。在这里，用户可以查看ChatGPT计划、更改个人设置和添加自定义说明。

用户还可以根据个人喜好更改界面的主题，选项包括系统默认及深色和浅色模式。

在底部的输入框中输入提示，即可与ChatGPT进行互动。

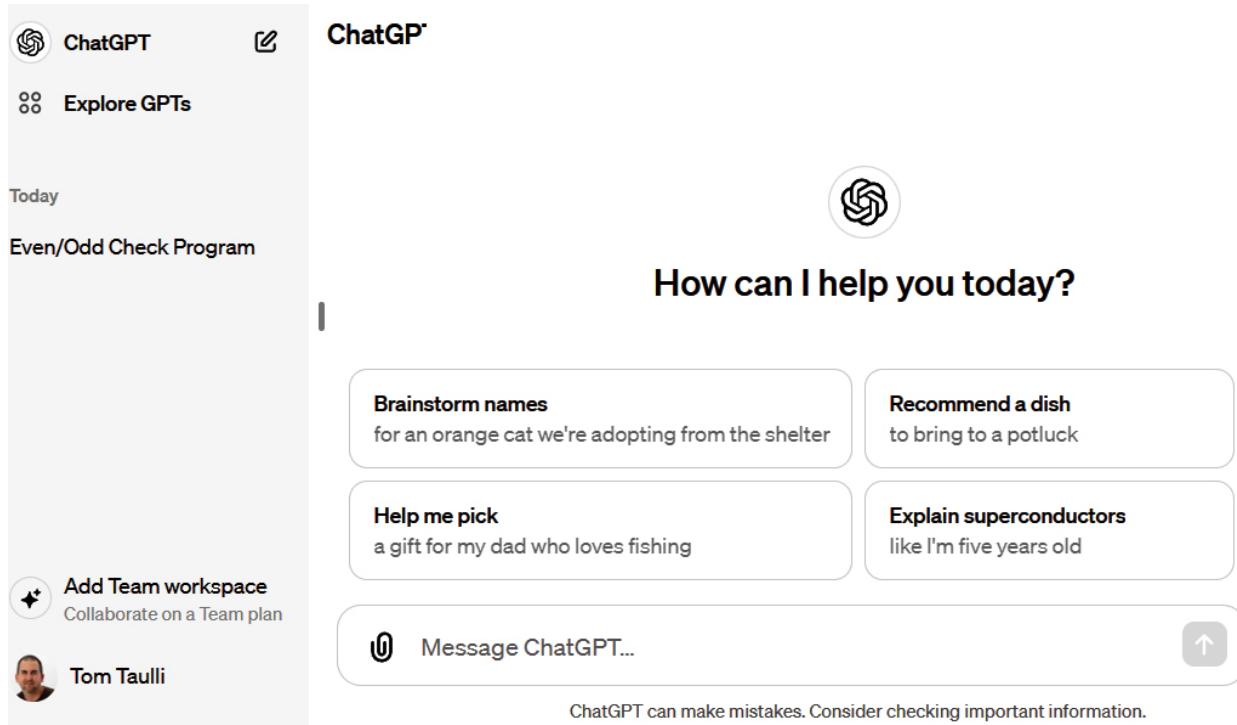


图6-1 ChatGPT的界面

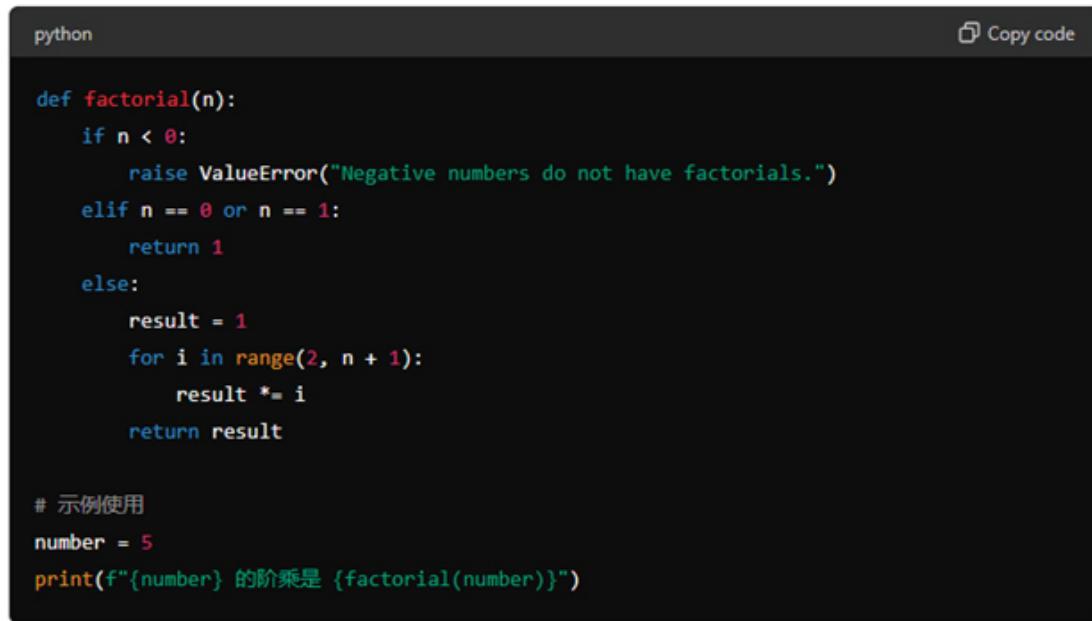
接下来我们尝试一下ChatGPT的功能。首先，输入以下提示来创建代码。

**提示：**编写一个计算数字阶乘的Python代码片段。

ChatGPT的响应如图6-2所示。



下面是一个计算数字阶乘的 Python 代码片段：



```
python
Copy code

def factorial(n):
    if n < 0:
        raise ValueError("Negative numbers do not have factorials.")
    elif n == 0 or n == 1:
        return 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i
        return result

# 示例使用
number = 5
print(f"{number} 的阶乘是 {factorial(number)}")
```

这段代码定义了一个 `factorial` 函数，计算输入数字 `n` 的阶乘。函数首先检查 `n` 是否为负数，如果是则抛出一个 `ValueError`。如果 `n` 为 0 或 1，函数返回 1。否则，函数使用一个循环计算阶乘。



图6-2 ChatGPT创建了一个Python代码片段来响应提示

然后输入以下提示询问是否有其他可以创建这段代码的方法。

**提示：创建这段代码的另一种方法是什么？**

ChatGPT提供了另一种代码建议，即迭代法。如果你不知道迭代法，可以继续聊天。如图6-3所示，聊天会话记录在界面左上方。

ChatGPT将聊天标记为“计算数字阶乘”。如果需要，可以单击旁边的图标进行重命名。当然，还有一个删除图标用于删除会话。

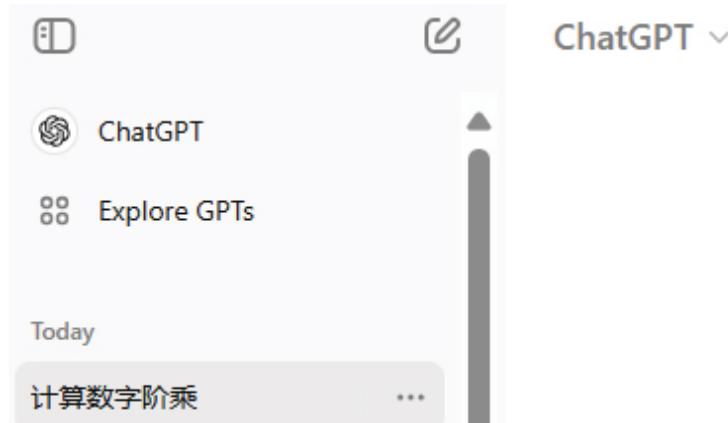


图6-3 ChatGPT在界面左上角记录聊天会话

如果想要创建一个新的会话，单击界面顶部的铅笔图标，就会出现一个空白的聊天界面。

请记住，每个会话都是相互独立的。如果你在后续添加的会话中向ChatGPT询问它为阶乘计算创建的代码，ChatGPT将不知道具体是什么内容。

另外一个重要注意的地方是聊天会话的上下文窗口。上下文窗口决定了模型能“记住”多少之前的对话或文本内容，并用于生成连贯且与上下文相关的回复。对于GPT-3.5，上下文窗口为16K；但对于GPT-4，上下文窗口为32K。



如果想删除所有聊天主题，可以单击界面左下角的个人资料，选择Settings & Beta（设置和测试版），然后单击Delete all chats（清除所有聊天）旁边的Delete all（清除所有）按钮。

在ChatGPT中，想访问和管理自己的数据，你需要单击界面左下角的个人资料，然后选择Settings & Beta，你将看到关于ChatGPT数据的多种管理选项。

## 关闭聊天记录和模型训练

如果关闭聊天记录，则新的聊天记录不会被用于训练AI模型，也不会显示在侧边面板的历史记录中。OpenAI会将所有聊天记录保留30天，之后永久删除。

## 分享聊天记录

你可以将有趣的话题转为链接并与其他分享。

## 下载聊天数据

选择这个选项后，你将收到一封包含所有ChatGPT对话的电子邮件。



有时，ChatGPT会在生成文本的过程中停止。当出现这种情况时，通常会弹出Continue（继续）按钮。单击该按钮将继续生成文本。如果没有按钮弹出，你可以输入提示Continue Output（继续输出）。如果生成的文本没有达到你的要求，可以单击Stop Generating（停止生成）按钮。

### 6.3.1 移动版ChatGPT

ChatGPT在iOS和Android设备上都得到了支持，这使得用户在移动平台上也能享受到与网页版ChatGPT相似的体验。虽然移动版ChatGPT目前尚未集成插件等高级功能，但它依然保留了核心的聊天功能。

对于那些不喜欢在小屏幕上输入文字的用户，移动版ChatGPT提供了一个便捷的解决方案——语音聊天。用户可以在设置中找到并打开New Features（新功能）选项，选择加入语音聊天功能。一旦启用，用户只需要单击主屏幕右上角的耳机图标，就可以开始与ChatGPT进行语音交流。

移动版ChatGPT的语音系统非常先进，它支持5种不同的语音选项，用户可以根据个人喜好选择其中一种进行对话。这一功能的实现归功于一个名为Whisper的生成式AI文本转语音模型，它能将文本转换为自然流畅的语音并输出。

此外，移动版ChatGPT还提供了一个创新的功能——照片分享。用户可以随时随地拍摄照片，无论是物体、地标还是代码，并通过应用与ChatGPT分享，从而丰富聊天内容。移动版ChatGPT内置的绘图工具还能让用户在聊天时突出显示图片的特定部分。

### 6.3.2 自定义指令

在ChatGPT中，自定义指令功能让用户能够根据特定的需求来调整机器人的回复方式。这个功能的存在意味着用户不再需要在每次与机器人交流时都重新微调回复或构思新的提示，从而大大节省时间。

设置自定义指令时，需要回答以下两个问题。

**ChatGPT**：为了提供更好的回复，你希望ChatGPT了解你的哪些信息？

**ChatGPT**：你希望ChatGPT如何回复？

下面是示例。

**ChatGPT**：为了提供更好的回复，你希望ChatGPT了解你的哪些信息？

**开发者**：我是一名开发者，我所在的团队遵循Python代码PEP 8风格指南。我们重视代码的简洁性、可读性，并坚持最佳实践。我通常从事数据处理和分析工作，在编写高效、结构合理的代码方面经常需要帮助。

**ChatGPT**：你希望ChatGPT如何回复？

**开发者**：我希望ChatGPT提供符合PEP 8风格指南的Python代码片段。ChatGPT应优先考虑代码的可读性和最佳实践。在提出解决方案时，我希望ChatGPT能解释为什么推荐某种特定的方法，以及这种方法是如何与PEP 8风格指南保持一致的。如果ChatGPT能指出与当前任务相关的常见陷阱或错误，并提供规避这些缺陷和错误的提示，我将不胜感激。

## 6.4 使用必应浏览功能

ChatGPT的训练数据截至2023年4月，这对开发者来说可能存在一定的局限性。由于代码库、框架和工具的更新换代速度极快，2023年4月之后推出的新特性或功能可能不会被ChatGPT知晓。

在使用ChatGPT时，一个实用的技巧是使用必应浏览功能。通过这个功能，你可以实时搜索互联网，获取最新的信息和答案。

以尝试新框架LangChain为例，自2022年年底推出以来，LangChain已经得到显著的发展和改进。LangChain旨在简化LLM在应用开发中的集成过程，这有可能非常适合你的需求。

为了增强ChatGPT对LangChain当前状态的了解，你可以激活“使用必应浏览”功能，并在提示中加入“Use Internet for this response”（使用互联网进行回复）的请求。以下是一些提示范例，用于获取LangChain的一般信息。

**提示**：介绍LangChain框架及其主要功能。使用互联网进行回复。

**提示**：在互联网上查找有关LangChain入门的介绍性资源或文档。

**提示**：在互联网上查找使用LangChain框架构建的应用示例。

**提示**：查找与LangChain框架相关的最新更新或版本。使用互联网进行回复。

在回答上述提示时，ChatGPT确实提供了相当有用的答案。例如，对于第一个提示，ChatGPT给出了LangChain的5个主要功能和模块，如图6-4所示。



Searched 6 sites

LangChain 是一个专门用于开发由大型语言模型 驱动的应用程序的框架。其设计目标是通过将语言模型与其他数据源连接起来，并允许它们与环境进行交互，从而开发出更强大和多样化的应用  
( Langchain) ([腾讯云 - 产业智变 云启未来](#))。

## LangChain 的主要功能和模块

### 1. 模型 (Models) :

- LangChain 包含对各种大型语言模型的封装，允许用户利用这些模型的强大功能来理解和生成语言。

### 2. 提示 (Prompts) :

- 提示是自然语言处理应用的核心。LangChain 提供了丰富的提示模板，帮助用户格式化输入，从而引导模型生成所需的输出。

### 3. 链 (Chains) :

- 链是将多个步骤串联起来完成复杂任务的过程。例如，可以从特定 URL 获取数据，对返回的文本进行摘要，然后用生成的摘要来回答问题。

### 4. 代理 (Agents) :

- 代理使模型能够与外部环境互动，执行例如 API 调用等操作。代理能够根据输入做出决策，并采取行动，直到完成任务 ([GitHub](#)) ([Bilibili](#))。

### 5. 文档加载器和工具:



图6-4 使用必应浏览功能帮助ChatGPT研究LangChain框架的主要功能和模块<sup>注</sup>

你还可以更改响应的输出格式，例如下面的提示。

**提示：**请用一个表格来展示回答。

ChatGPT的响应如图6-5所示。

当使用必应浏览功能时，ChatGPT可能需要几秒到1分钟甚至更长时间来生成回复。在某些情况下，ChatGPT可能暂时无法响应并显示错误信息，这时你可能需要尝试好几次才能获得答案。这种不稳定性是你在使用ChatGPT时需要注意的问题。

请用一个表格来展示回答。



以下是关于 LangChain 框架及其主要功能的详细表格展示：

功能模块	描述	作用	参考资料
模型 (Models)	LangChain 封装了各种大型语言模型。	利用模型的理解和生成语言的能力。	<a href="#">LangChain中文网</a> , <a href="#">腾讯云</a>
提示 (Prompts)	提供丰富的提示模板。	帮助用户格式化输入，引导模型输出。	<a href="#">LangChain中文网</a> , <a href="#">腾讯云</a>
链 (Chains)	将多个步骤串联起来完成复杂任务。	例如，从 URL 获取数据，生成摘要，并回答问题。	<a href="#">LangChain中文网</a> , <a href="#">腾讯云</a>
代理 (Agents)	使模型与外部环境互动，执行操作。	根据输入做决策，并采取行动直到完成任务。	<a href="#">GitHub</a> , <a href="#">腾讯云</a>
文档加载器和工具	连接到数据源并进行计算的工具。	适用于需要底层系统交互的应用。	<a href="#">LangChain中文网</a> , <a href="#">GitHub</a>
索引 (Indexes)	创建和管理文档索引，特别是向量存储。	用于高效检索相关文档和信息。	<a href="#">LangChain中文网</a> , <a href="#">Bilibili</a>

图6-5 ChatGPT将回复转换为表格形式

在使用必应浏览功能时，输出结果通常会附带一个指向信息源的网络链接。单击该链接以验证信息的真实性和准确性是一种明智的做法。你应该审慎对待ChatGPT提供的所有内容，特别是当回复的内容可能来自社交媒体帖子等非可靠信息源时。

幸运的是，你可以指示ChatGPT专注于从特定类型的信息源获取信息。表6-1提供了一些有用的提示和建议，以帮助你更有效地使用这一功能。

表6-1 指示ChatGPT使用特定信息源的方法

方法	描述
域名限制	将搜索限制为信誉良好的域（例如.edu或.gov）或特定组织
使用学术关键字	包括同行评审、期刊、研究等
使用发布日期过滤器	按最近的发布日期过滤搜索，获取最新的相关信息
专业或学术协会	指定从专业或学术协会寻求权威出版物或报告

下面的示例展示了ChatGPT为LangChain创建代码的能力。这是一个针对简单程序的提示。

**提示：**编写一个使用LangChain框架的Python程序。它需要具备以下功能。

1. 接收用户关于特定主题的输入。
2. 允许用户选择要创建的内容类型：博客文章或社交媒体帖子。
3. 利用LangChain访问OpenAI API，并根据用户输入的主题生成博客文章或社交媒体帖子。

然而，当笔者尝试在ChatGPT中实现这一功能时，结果并不尽如人意。虽然ChatGPT能够提供基本的Python代码结构，这是意料之中的，但它在处理LangChain的代码时却出现了问题。ChatGPT在导

入LangChain库时犯了错误，并且遗漏了一些必要的步骤。此外，ChatGPT未能展示如何正确调用OpenAI API。

尽管如此，使用必应浏览功能对于解决日常问题和获取一般信息仍然非常方便。

## 6.5 处理乏味的任务

软件开发是一个包含多个步骤的复杂过程，涵盖规划、编程和测试等环节。在这一过程中，重复性的编程任务尤为令人头疼——它们不仅枯燥乏味，还极其耗时。开发者常常发现自己陷入了编写常规代码、调试错误和审查代码的循环中，这些工作虽然重要，却并非软件开发中最具乐趣和创造性的部分。

AI工具（如ChatGPT）能够为这些单调乏味的软件开发问题提供解决方案。在接下来的内容中，我们将深入探讨这些工具在软件开发中的一些应用。

### 6.5.1 正则表达式

正则表达式（Regular Expression，简称regex）是一种强大的文本处理工具。你可以把它们想象为一种高级的查找和替换功能，用于识别字符串中的特定模式。但挑战在于，学习regex就像学习一种全新的语言，它由各种符号和命令组成，这可能会让人觉得相当棘手。要记住所有复杂的斜杠、点、括号及其他看似神秘的组合，确实会让人感到困惑。有时，你可能觉得自己需要一个解码器才能理解这些表达式。

例如，考虑如下正则表达式。

```
^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$
```

这个正则表达式用于验证强密码，确保密码至少包含8个字符，且包括至少一个大写字母、一个小写字母、一个数字和一个特殊字符。

编写这样的正则表达式可能看起来并不那么有趣。但是，ChatGPT可以在这方面提供帮助，以下是一些常见示例。

**提示：**请生成一个用于验证标准电子邮件地址的正则表达式。电子邮件地址应该以字母或数字开头，之后可以跟点号、连字符或下划线。紧接着是@符号，其后应有字母或数字字符或点号，最后是一个长度为2~6个字符的域名。

**regex模式：**`^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$`

**提示：**请生成一个正则表达式以验证URL。有效的URL应起始于http或https，随后紧跟“://”。域名部分应由字母、数字字符及点号组成。域名之后可以跟一个可选的路径，该路径以“/”开头，并且可以包含字母或数字字符、斜杠、点号或连字符。此外，URL可以选择性地以“/”结束。

**regex模式：**`^(https?:\/\/)?([\da-z.-]+)\.([a-z.]{2,6})([\//\w.-]*)*\/?$/`

**提示：**请生成一个正则表达式以验证符合MM/DD/YYYY格式的日期。应确保月份在1~12的范围内，日期在1~31的范围内，年份则在1900~2000的范围内。

**regex模式：**`^(0[1-9]|1[0-2])/(0[1-9]|1[2][0-9]|3[01])/(19|20)\d\d$`

**提示：**请生成一个正则表达式以验证IPv4地址。应确保每个8位组是0和255之间的数字，并且各8位组通过点号分隔。

**regex模式：**`^(25[0-5]|2[0-4]\d|[0-1]?\d\d?)\.(25[0-5]|2[0-4]\d|[0-1]?\d\d?)\.(25[0-5]|2[0-4]\d|[0-1]?\d\d?)\.(25[0-5]|2[0-4]\d|[0-1]?\d\d?)$`

ChatGPT还可以解读正则表达式。例如下面的提示。

**提示：**下面这个正则表达式有什么用？`^((\d{3})\s?|\d{3}[-.])\d{3}[-.]\d{4}$`

ChatGPT的回应是：这个正则表达式用于匹配北美电话号码的格式。

## 6.5.2 入门代码

入门代码（Starter Code）是指最初的代码片段或模板，它为项目提供了一个稳固的起点。它不仅可以显著节省开发时间，还能作为一种有效的学习工具。对于编程初学者，研究和探索入门代码有助于理解编程概念和工作机制。此外，入门代码还有助于维持项目中代码的一致性，这在需要团队成员遵循统一编码标准的大团队中尤为重要。

例如下面的代码。

```
const express = require('express');

const app = express();

const port = 3000;

app.get('/', (req, res) => {

  res.send('Hello, World!');

});

app.listen(port, () => {

  console.log('Server running at http://localhost:${port}');
});
```

上述代码展示了如何使用Express.js构建一台基础的Web服务器，这是一个清晰而简洁的示例。Express.js作为一个流行的Web框架，以

简洁著称。这段代码虽然简单，但涵盖了创建基本Web服务器所需的核心元素，非常适合作为入门代码。表6-2列举了一些入门代码的提示范例。

表6-2 入门代码的提示范例

编程语言/ 框架	提示范例
Node.js/Express.js	提供一个基本的Express.js服务器示例
React (JavaScript)	如何创建一个简单的React组件？
JavaScript	如何在JavaScript中定义一个简单的函数？
Android (Java)	Java中的基本Android Activity的入门代码是什么？
Python/Flask	创建一个基本的Flask应用程序示例
Vue.js (JavaScript)	展示如何设置基本的Vue实例
Django (Python)	如何使用简单的视图启动一个新的Django项目？
Swift (iOS)	Swift中简单UIViewController的入门代码是什么？

### 6.5.3 README文件

README文件本质上是项目的介绍信。通过这个文件，你可以快速了解项目的核心内容：项目的用途、工作原理以及如何使用项目。

README文件通常命名为README.md，使用Markdown格式编写，以确保文件既美观又易于阅读。在项目的仓库主页上，README文件总是被显眼地展示出来。

一个写得好的README文件至关重要，它能够帮助所有访问项目的人迅速把握项目概览。无论是想要使用项目的用户还是希望为之贡献代码的开发者，README文件都为他们理解项目奠定了基础，使项目更加易于理解和上手。同时，它还规定了项目的规则和期望，为参与者提供了明确指导。

编写README文件并非易事。你必须在提供所有必要细节与保持内容简洁之间找到平衡。专注于技术的开发者可能会忽略一些重要部分，因为他们可能没有意识到，对他们来说显而易见的细节对其他人来说可能并不那么明显。

编写一个好的README文件确实需要一定的写作技巧，而这并不是开发者的强项。随着项目的不断发展和变化，保持README文件不断更新也是一个挑战，这就是所谓的“既要……又要……”的情况。

ChatGPT不仅简化了编写README文件的过程，还能提供内容丰富的文本。假设你刚刚创建了一个搜索食谱的应用程序，你可以使用以下提示来生成README文件。

**提示：**为我的“食谱搜索器”项目撰写README文件。该项目采用Vue.js进行开发，旨在根据用户手头的食材帮助他们找到相应的食谱。用户需要输入他们手头的食材，应用程序将返回一个他们能用这些食材制作的食谱列表。要启动应用程序，只需要复制源码仓库，执行命令npm install，在.env文件中插入API密钥，然后执行npm start命令即可。

根据这个提示，ChatGPT会创建一个包含项目介绍的README文件，其中包括功能介绍、入门指南、使用说明、贡献方式、许可证和致谢等部分。

对于README文件中的某些内容，ChatGPT会做出一些假设，比如假定项目采用MIT许可证。你可以通过提供更多的项目细节来引导ChatGPT，以生成更符合项目需求的README文件。



如果你不知道使用哪种开源软件许可证，请查看choosealicense网站。该网站可帮助你选择合适的开源软件许可证。

## 6.6 跨浏览器兼容性

跨浏览器兼容性是指确保网站或Web应用程序能够在各种不同的浏览器中正常工作的能力。这一能力至关重要，因为用户访问网站或应用程序时可能会使用Chrome、Firefox、Safari、Edge等浏览器，而这些浏览器各自有着不同的版本和特性。

表6-3列出了一些关于浏览器兼容性的提示，以应对跨浏览器兼容性方面的挑战。

表6-3 关于浏览器兼容性的提示范例

类别	提示
HTML5和CSS3功能	我想在输入字段中使用占位符属性，但它在旧版本的Internet Explorer中不起作用。我怎样才能创建一个兼容属性？
CSS前缀	我想在CSS中使用过渡属性，但我想确保它与所有浏览器兼容。我该怎么做？
JavaScript功能	我正在使用JavaScript中的fetch函数进行HTTP请求，但Internet Explorer不支持该函数。我该如何处理？

## 6.7 bash命令

在终端，与计算机系统交互的指令被称为bash命令。这些命令可以执行多种任务，包括移动文件、查看系统状态等。

对于UNIX/Linux操作系统的新手，掌握bash命令可能会显得有些困难。某些命令的语法可能难以理解，需要时间来熟悉。要精通bash命令，应持续学习和实践。此外，由于bash命令种类繁多，每种命令都带有自己的选项和变体，这进一步增加了学习的复杂性。尽管互联网上的许多资源可以提供帮助，但即使是经验丰富的专业人士，也可能难以记住那些不常用或新学的命令。

不过，ChatGPT可以作为你的bash命令虚拟助手，帮助你更有效地使用这些命令。以下是一些提示范例。

**提示：**如何使用bash命令按修改时间排序列出目录中的所有文件？

**提示：**计算文件内行数的bash命令是什么？

**提示：**如何使用一个命令创建一个目录并导航到该目录？

**提示：**如何将命令输出重定向到文件？

**提示：**搜索系统中正在运行的特定进程的命令是什么？

**提示：**如何使用bash命令将目录压缩成ZIP文件？

**提示：**使用什么bash命令来更改文件的权限？

## 6.8 GitHub Actions

GitHub Actions是GitHub内置的一个持续集成和持续部署  
( Continuous Integration and Continuous Deployment ,

CI/CD ) 的平台，它让开发者能够在自己的软件仓库中直接配置、定制并执行软件开发工作流程。本质上，GitHub Actions的工作流程是一系列定义在工作流文件中的自动化任务。这些自动化任务由不同的作业 (job) 构成，每个作业执行特定的任务，比如执行命令、设置环境或执行特定的操作。它们使用YAML语法编写，并在特定的触发事件（如代码推送、问题创建或定时调度）发生时自动启动。

通过GitHub Actions，开发者可以在GitHub上直接进行代码构建、测试和部署，这使得软件开发过程更加高效和自动化。此外，GitHub Actions支持共享和重用action，甚至允许定制其他开发者创建的action，这对于团队协作和共享自动化软件开发的最佳实践非常有帮助。

ChatGPT可以帮助你创建GitHub Actions工作流模板。以下是一些提示范例。

**提示：**创建GitHub Actions工作流模板，使用Hugo构建和部署一个静态网站，然后将这个静态网站部署到GitHub Pages。

**提示：**为一个使用Maven的Java项目生成GitHub Actions工作流模板，其中包含编译代码、运行测试和打包应用程序的步骤。

**提示：**创建GitHub Actions工作流模板，用于容器化应用程序，其中包括构建Docker镜像，以及将Docker镜像推送到Docker Hub并部署到Kubernetes集群的步骤。

**提示：**为React Native移动应用程序生成GitHub Actions工作流模板，其中包括安装依赖项、构建应用程序，以及将APK上传到指定Google Drive文件夹的步骤。

## 6.9 ChatGPT插件

ChatGPT插件是利用了OpenAI LLM功能的迷你应用程序，它们为用户打开了与外部数据库和应用程序交互的大门，让用户在iOS或Google Play上拥有了自己的小型应用程序商店。

在开始使用ChatGPT插件之前，你需要确保已经启用了这一功能。首先进入“个人资料”界面，然后选择Settings（设置）菜单，最后选择Beta Features（测试版功能）命令。

将鼠标指针悬停在界面顶部的GPT-4图标上，从下拉菜单中选择Plugins（插件）命令，便可以浏览和访问众多可用的插件。单击“插件”按钮后，你将看到类似于图6-6所示的界面，你可以根据需要选择和激活不同的插件来扩展ChatGPT的功能。

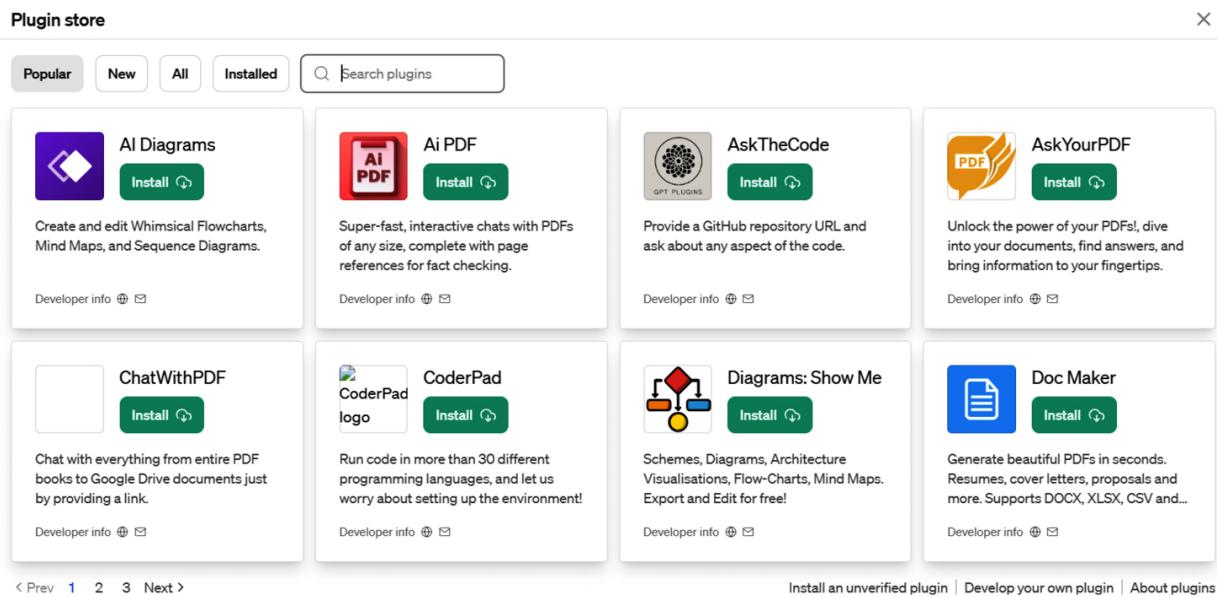


图6-6 ChatGPT插件商店

### 6.9.1 Codecademy插件

下面我们尝试使用Codecademy插件的功能。首先单击“安装”按钮进行Codecademy插件的安装。安装完成后，移至界面顶部，单击向下箭头按钮，然后单击Codecademy图标以激活Codecademy插件。

Codecademy插件提供如下两大核心功能。

- 个性化课程推荐：Codecademy插件允许用户根据自己的学习目标和经验水平搜索特定的课程或学习路径。例如，对AI和ChatGPT感兴趣的用户可以向Codecademy插件询问相关的课程推荐。  
Codecademy插件将根据用户的需求提供一份课程列表，里面包括课程描述以及课程是免费还是付费的信息。

- 技术文档快速索引：Codecademy插件作为技术文档的快速索引工具，为用户提供了更详细的文档和文章链接。

为了测试Codecademy插件的文档功能，可以使用以下提示。

**提示：**解释JavaScript数组的最佳文档或文章是什么？

ChatGPT的响应如图6-7所示。顶部的图标表明系统正在访问Codecademy插件，随后的文字显示了关于该主题的多份资源。紧接着ChatGPT提供了一些最相关的文档链接，以帮助用户快速找到所需的信息。

Used Codecademy ▾

Here are some excellent resources for understanding arrays in JavaScript:

1. [Arrays](#): This article serves as a reference sheet for JavaScript arrays, providing a comprehensive overview.
2. [Javascript Guide: Arrays](#): This guide offers detailed insights into interacting with JavaScript arrays.
3. [Using Arrays to Group Data](#): Learn how to use arrays to organize your data effectively.
4. [Arrays in JavaScript \(Documentation\)](#): This documentation provides a detailed explanation of arrays, including their properties and methods.

These resources should give you a solid understanding of arrays in JavaScript. Feel free to explore them to deepen your knowledge.

The screenshot shows a list of four resources from Codecademy. Each card has a dark blue header with the 'codecademy' logo and a light gray footer. The first card is titled 'Arrays | Codecademy'. The second is 'Javascript Guide: Arrays | Codecademy'. The third is 'Using Arrays to Group Data | Codecademy'. Below the cards is a horizontal navigation bar with icons for back, forward, and search.

图6-7 Codecademy插件响应关于JavaScript数组资源的请求

## 6.9.2 AskYourDatabase插件

Sheldon Niu是AskYourDatabase插件的创意发起人。他经常利用ChatGPT编写SQL语句，但这个过程往往烦琐且效率低——每次都需要向ChatGPT解释整个数据库的架构，并复制和粘贴输出结果到终端后再执行。这激发他产生了一个想法：“如果ChatGPT能够直接与数据库进行交互，那将会怎样？”正是这个想法促使他开发了AskYourDatabase插件。

借助AskYourDatabase插件，用户可以便捷地通过ChatGPT创建数据库模式原型。它简化了数据库查询过程，使得用户能够轻松地与数据库进行交互。更棒的是，用户可以绕过那些需要复杂设置的传统商业智能工具，直接通过自然语言与数据库进行沟通，极大地提升了工作效率。

### 6.9.3 Recombinant AI插件

Mark Zahm是一位开发者，他成立了自己的AI咨询公司。为了帮助开发者在使用GitHub和GitLab仓库时提升效率，他开发了Recombinant AI插件。这个插件使得ChatGPT能够把握整个程序的核心要点，并过滤不必要的细节。如此用户可以更有效地掌握自己的代码，无论是进行调整、分析还是将新想法融入软件中。Mark将这个插件形容为“会话式IDE”，这一描述非常准确地概括了其功能。

以下是Recombinant AI插件的一些应用示例。

- 执行传统任务，例如存储代码片段或编程提示。
- 利用项目和文件系统，创建复杂的任务列表和CoT提示，帮助用户组织和规划开发工作。
- 保存关于最新代码库的重要信息，确保用户能够及时获取和应用新技术。

## 6.10 自定义GPT

可以对ChatGPT进行个性化定制，这个过程称为自定义GPT。定制过程非常简便快捷，通常只需要几分钟即可完成。

例如，可以为软件开发风格指南创建一个GPT，其中包含一些基本原则。例如，如何命名变量以确保每个人都能理解和使用正确的代码缩进方法，以及团队应遵循的特定编程模式或实践。

创建这样一个模板的目的是保持代码库的统一性和整洁性，使每个人都能更容易地阅读和理解代码。在大型项目中，统一的编码风格可能是确保项目可维护性的关键。

统一编码风格可能带来的问题是，新加入的成员可能会感到困惑，尤其当他们已经习惯了不同的编程方式，突然之间又需要适应新的标准时。

GPT在这种情况下可以提供巨大帮助。首先在ChatGPT主界面的左上角选择Explore（探索）菜单，然后选择Create a GPT（创建一个GPT）命令。如图6-8所示，你会看到两个面板。

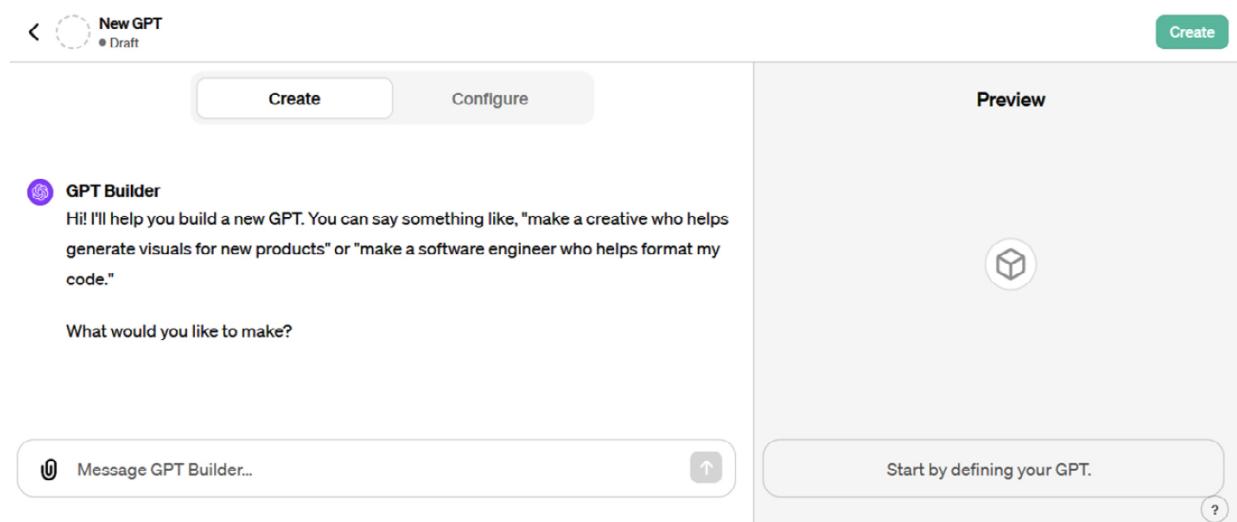


图6-8 使用GPT生成器创建自定义GPT

左侧面板用于创建GPT，而右侧面板是预览区。为了使GPT与软件开发风格指南相匹配，你可以使用以下提示。

**提示：**创建一个强制执行软件开发风格指南的系统。

ChatGPT将引导你开始创建GPT。虽然初始提示可能不够具体，但ChatGPT会通过提出问题来收集更多必要的信息。它会首先建议一个GPT名称，比如“Code Stylist”。

接下来，ChatGPT会利用DALL·E 3来创建GPT的个人资料图像，你也可以选择上传自己的图像。之后，它会询问关于编程语言、最佳实践、指南、规则、应避免的事项等更详尽的信息。例如，你可以指定使用Python语言，并提供一些特定的要求。

在左侧面板的顶部选择Configure（配置）选项卡，上传你的软件开发风格指南文档。这样当你在右侧面板中测试GPT时，如果要求编写代码，GPT就会自动采用Python语言并遵循你设定的规则。

完成设置后，单击右上角的保存图标来保存你的GPT。保存成功后，它将出现在ChatGPT的主界面上。单击GPT图标即可开始使用它。

OpenAI还为自定义GPT创建了应用商店。在ChatGPT主界面的左上方选择Explore GPTs（探索 GPT）命令，即可进入应用商店。

在应用商店中，你会发现很多编程类别。以下是一些可用的GPT示例。

- GPTavern：这是一个创新平台，旨在通过一种称为prompt-gramming的独特方法来教授编程。它提供二十多个快捷键来简化编程，并提供75个入门项目来帮助用户学习编程。该平台是互动式的，允许用户提问、上传图片、访问命令菜单和README文件以获得指导。

- DesignerGPT：由Pietro Schirano开发，专门用于创建和托管美观的网站。

- Screenshot To Code GPT：由godofprompt.ai提供，允许用户上传网站截图，然后转换成简洁的HTML、Tailwind和JavaScript代码。

- Mindmap/Diagram/Chart-PRO BUILDER：由pyxl.ai提供，旨在帮助用户进行代码和数据库的可视化。它允许用户使用拖放编辑器创建流程图、图表和序列，以理解和组织复杂的数据结构。

·Code Guru：由Ryan J. Topps开发，提供一系列服务，包括审查代码、编写拉取请求、生成和优化函数、编写测试以及对现有代码进行注释等。



在ChatGPT的主界面上选择Actions（操作）菜单，即可在你的GPT中包含第三方API。

## 6.11 Gemini

尽管Google在人工智能领域早有涉猎并进行了重大投资，但对于生成式AI的兴起，Google似乎尚未做好准备。Google对于踏入这一新兴技术领域持谨慎态度——担心准确性问题以及由此可能对公司利润丰厚的广告业务造成的影响。然而，到了2023年初，Google的股价遭遇下跌，这促使情况发生变化。Google决定采取行动，推出了自己的生成式AI产品——Bard。起初，Bard遭遇了一些挑战，但它随后逐渐找到自己的节奏，并在Google的多个应用中发挥作用，Google的股价实现反弹。

到了2024年2月，Google将Bard更名为Gemini，并推出了其高级版本——Gemini Advanced。Gemini Advanced提供两个月的免费试用期，之后每月收费19.99美元。Gemini采用了Google最先进的LLM——Ultra 1.0，Google称“它能够帮助你处理更高级的编程场景，成为你思维的共鸣板，并协助你评估不同的编程方法”。

Google还为Android和iOS平台推出了移动版的Gemini。在移动版Gemini中，用户可以通过打字、语音或图像与LLM进行互动。图6-9展示了Gemini的用户界面。

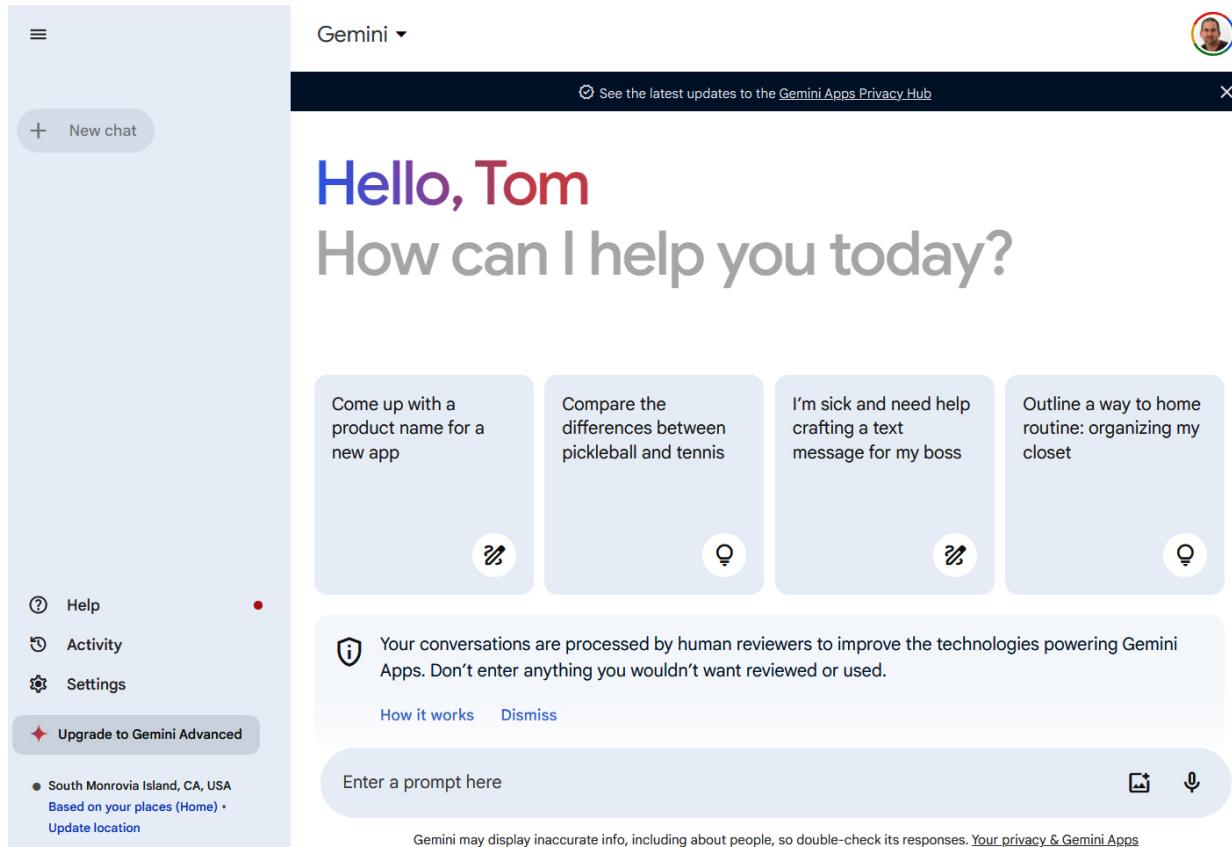


图6-9 Gemini的用户界面

与ChatGPT类似，界面左侧设有一个聊天会话列表。如果用户希望隐藏这个列表，只需要单击界面顶部的汉堡包图标即可。对于任何一个聊天会话，用户都可以通过单击会话旁边的小图标来重命名或删除聊天。另一个非常实用的功能是，用户可以将经常需要参考的聊天记录固定在易于访问的位置。

### 6.11.1 应用

像Excel和Google Sheets这样的生产力增强应用程序，并不仅限于处理电子表格，它们还拥有各自的编程语言来扩展功能。以Excel为例，它配备了Visual Basic for Applications（VBA），这是一个功能强大的编程系统，允许用户编写自己的脚本和程序。VBA非常适合自动化常规任务、创建自定义函数以及进行数据分析。Excel还内置了许多现成的功能，如数据透视表（PivotTables）和

Power Query，并且支持与Power BI集成，以实现高级的数据可视化和分析。

Google Sheets可以与Google Apps Script配合使用，后者可以看作JavaScript的一个变体。Google Apps Script使在Google Sheets中执行与Excel类似的任务成为可能。此外，Google Sheets还能与Google Drive和Gmail等其他Google应用无缝集成。Google Sheets提供了一套即用型功能，用户可以在Google Workspace Marketplace上探索和使用不同的插件以增强其能力。由于Google Sheets完全基于云，因此它非常适合你与他人协作，是Web应用程序和团队项目的可靠选择。

那么，Gemini在这样的背景下能提供什么帮助呢？可以利用Gemini来解释公式——即使是相当复杂的公式也能解释。以下示例展示了如何使用Gemini来解释Excel或Google Sheets中的公式。

**提示：**解释Excel公式`SUM(OFFSET(A1,1,0,COUNT(A:A),1))`。

对于Gemini，理解上述公式完全没有问题。

另一个示例是创建VBA脚本。下面是一些实用的提示范例。

**提示：**如何使用VBA在Excel中创建自定义表单来输入用户登录详情？

**提示：**你能帮我编写一个Excel VBA脚本，用密码保护特定数据吗？

**提示：**我需要用VBA代码来验证Excel用户登录表中的电子邮件地址和日期格式。你能提供帮助吗？

**提示：**能否使用VBA从Excel自动发送电子邮件通知，获取有关用户登录信息的更新情况？

虽然Gemini在响应提示方面表现出色，但在使用电子表格时，理解数据、公式和脚本之间的相互关系至关重要。掌握这些元素之间的关系可以帮助你构建功能强大且稳定的模型。当你更改数据中的某些内容时，这些变化会传递到相关的公式，并可能影响脚本的输出，最终导致结果发生重大变化，而这一切可能在你意识到之前就已经发生。

幸运的是，Gemini支持加载和分析Excel电子表格。你可以针对电子表格向Gemini提出问题，比如逐步探索数据和逻辑，或者强调数据或逻辑的特定方面。

假设你有一个包含图书销售数据的Excel电子表格，以下是一些可能的提示范例。

**提示：**你能展示如何在Excel中找到一位作者赚取的最高版税金额吗？

**提示：**我想在Excel中计算每本书的库存周转率。我应该使用什么公式？

**提示：**我如何在Excel中创建一个库存短缺警报，以根据补货水平提示何时补货？

**提示：**如何在Excel中计算好评（评分达到4分及以上分数）图书的占比？

## 6.11.2 编程

Gemini能够理解20多种计算机编程语言。让我们通过以下提示来看看Gemini的编程能力。

**提示：**生成一个将摄氏温度转换为华氏温度的TypeScript函数。

Gemini的响应如图6-10所示。

Show drafts ▾ 🔊



### TypeScript

```
function celsiusToFahrenheit(celsius: number): number {  
    return (celsius * 9) / 5 + 32;  
}
```

Use code [with caution.](#)



该函数接收一个摄氏温度值（数字）作为参数，并返回相应的华氏温度值（数字）。

以下示例展示了该函数的用法：

### TypeScript

```
const celsius: number = 20;  
const fahrenheit: number = celsiusToFahrenheit(celsius);  
console.log(` ${celsius} 摄氏度等于 ${fahrenheit} 华氏度`);
```

图6-10 Gemini对创建TypeScript函数的请求的响应

在图6-10所示界面的右上角，有一个Show drafts（显示草稿）选项。启用这个功能后，你将能够查看Gemini给出的3个版本的回复。这些回复能够为你完成编程任务提供更丰富的思路和选择。

与ChatGPT一样，Gemini的回复中包括了代码清单，你可以直接复制其中的代码。Gemini还提供了代码解释。

图6-11显示了处理响应的几种方法。

现在，上面的示例代码将输出以下内容：

20 摄氏度等于 68.00 华氏度

我希望这对您有所帮助！



图6-11 单击底部的图标，即可使用不同的选项处理Gemini的响应  
处理选项如下。

## 评价

通过单击大拇指图标进行“赞成”或“反对”，对响应的质量提供反馈。

## 修改回复

单击筛选器图标，可以选择调整回复的长度、复杂度、风格，让回复变得更简洁、更详尽、更通俗易懂或更专业。

## 分享

可以将回复以公开链接方式共享，还可以将回复导出到Google文档或保存为Gmail草稿。

## 仔细检查回复

单击Google公司LOGO，Gemini将审查回复的准确性，还有可能提供支持链接。

## 6.12 Claude

早在2021年，Dario和Daniela兄妹就成立了自己的初创公司 Anthropic。他们曾是OpenAI的核心人物。然而，他们想要创建一个与众不同的生成式AI平台，愿景是让AI更安全，比如尽量减少偏见。他们的想法最终孕育出宪法人工智能（Constitutional AI）的概念。

Anthropic的LLM具有很强的编程技能。它的一个Python编程测试在Codex HumanEval上的得分率高达71.2%。

Claude作为Anthropic旗下的一款产品，一大优势是拥有高达100000个token的上下文窗口，这意味着它可以处理大型代码清单。它还支持上传PDF等文件。

图6-12展示了Claude的初始界面。

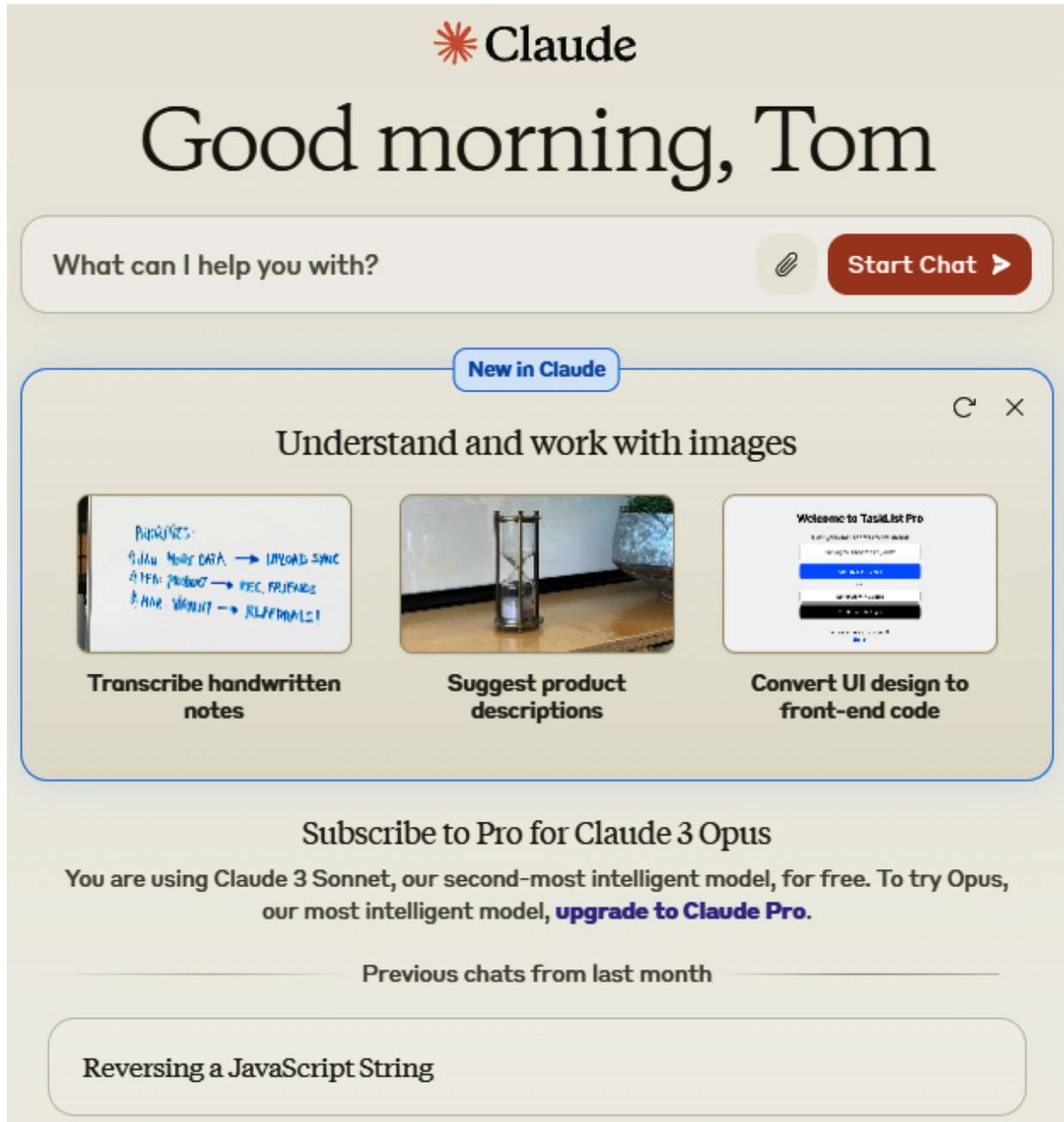


图6-12 Claude的初始界面

在界面的顶部，可以输入提示并且最多上传5个文件。界面的中间是活动列表，按活动发生的天数分组。

假设输入以下提示。

**提示：**你能编写一个反转字符串的JavaScript函数吗？

Claude的响应如图6-13所示。

Can you write a JavaScript function that reverses a string? π

Here is a function that reverses a string in JavaScript:

```
js Copy code  
  
function reverseString(str) {  
    return str.split('').reverse().join('');  
}
```

To break this down:

- The `str.split("")` splits the string into an array of characters
- The `.reverse()` reverses the array
- The `.join("")` joins the array back into a string

So for example:

```
js Copy code  
  
const string = 'hello';  
  
const reversed = reverseString(string);  
  
console.log(reversed); // 'olleh'
```

The `reverseString` function takes a string as a parameter, splits it into an array, reverses the array, and joins it back into a new reversed string.

AI Copy Retry Feedback

图6-13 Claude对编写一个JavaScript函数的请求的响应

## 6.13 小结

本章带领大家探索了一些LLM，如ChatGPT、Gemini和Claude，并展示了它们所具备的一系列令人印象深刻的功能，包括代码生成、插件支持、实时搜索互联网和处理大量信息等。虽然这些模型目前还不能完全替代专业的编程工具，但它们在加速软件开发进度方面已经展现出巨大的潜力。随着LLM的持续优化和升级，我们有理由相信，它们将为AI编程领域带来一波又一波令人瞩目的创新和突破。

**[9]**图6-4中的回答是作者写作此书时LangChain的主要功能和模块。翻译此书时，LangChain已经有很多的更新，译者得到的答案与图6-4并不一致。你在阅读此书时，想必差异更大。如果你希望得到中文回复，可以尝试如下提示。英文提示：Tell me about the LangChain framework and its primary features in Chinese. Use Internet for the response. 中文提示：介绍LangChain框架及其主要功能。使用互联网进行回复。译者使用以上两种提示得到了相同的中文回复。——译者注

## 第7章

# 软件开发的早期阶段：想法、需求和规划

本章将深入探索软件开发的黄金起点——概念形成与初步规划阶段。在这个充满创意火花的初期，所有激动人心的想法开始形成，而精心制订的计划则是确保项目顺利启动的关键。

在本章中，我们将首先利用ChatGPT的强大功能，进行高效的头脑风暴，将规划工作细化到每一个微小的环节。

接下来，我们将借助ChatGPT进行深入的市场调研，洞察行业趋势，分析竞争对手的动态。此外，为了激发团队的创造力和协作精神，我们还将引入一种生动有趣的讨论方式——白板讨论。在这种开放和动态的环境中，团队成员可以自由地提出各种想法，共同推动项目向前发展。

最后，我们将深入研究敏捷开发和Scrum等现代项目管理方法，探讨它们如何帮助我们高效地规划项目，并根据软件项目的特点进行适当的调整。

## 7.1 头脑风暴

在当今这个软件产品层出不穷的时代，想要开发出一款真正独具匠心、能够脱颖而出的产品绝非易事。然而，借助ChatGPT这样的LLM，我们可以汇聚多方智慧。

首先，我们可以利用ChatGPT提供多样化的头脑风暴方法。例如下面的提示。

**提示：**开发新产品前集思广益的技巧有哪些？利用互联网调研这个问题。

ChatGPT提供了10种有用的方法，让我们一窥其中的3种。

**ChatGPT：**

·5个为什么：从问题出发，通过连续提问“为什么”，有助于深入挖掘问题根源。

·坏主意：故意提出一些糟糕的想法，通过讨论它们的不足，反而可能激发出创新的解决方案。

·星爆法：这种方法关注问题而非答案，可以使用星爆图并提出一系列问题，比如谁、是什么、何时、何地、为何以及如何。

如果想深入了解星爆法，则可以通过以下提示获得更多信息。

**提示：**你将如何使用星爆法来提出软件产品的新想法？

ChatGPT针对以软件为主题的“星爆”练习做了分解分析。

**ChatGPT：**

·谁：考虑谁将使用这款软件。

·是什么：了解软件要解决什么问题，以及它与现有软件有何不同。

·何时：确定使用软件、启动软件，以及对软件进行更新或修复的最佳时间。

·何地：考虑可以在哪些平台或设备上使用这款软件。

·为何：深入探讨为什么你的软件不可或缺，以及人们为何要选择它而非其他软件。

**·如何**：思考软件的运行方式、构建过程、销售策略，以及用户需要帮助时的支持方案。

为了更清晰地理解新的软件产品创意，完成ChatGPT建议的过程无疑是一种有效的方法。

现在，当你准备列出一份创意清单时，最好先锁定一个特定的类别，了解目标群体，并明确所要解决的问题。这里有一些提示，可以帮助你起步。

**提示**：我想为老年人开发一个健康监控应用程序。老年人在监控慢性病和用药计划方面面临困难，我想帮助他们解决这个问题。有哪些功能可以使这个应用程序对老年人友好，并帮助他们更有效地管理健康？

**提示**：受众是正在努力理财的大学生。我的预算应用程序应具备哪些功能，才能帮助他们跟踪支出、节省资金并轻松了解财务规划？

**提示**：我想为忙碌的专业人士创建一个学习编程语言的应用程序，因为他们很难抽出时间参加常规课程。你能否推荐一些功能，让我的这个应用程序不仅能适应他们紧张的日程安排，还能提供快速、有效的编程语言学习课程？

ChatGPT给出的创意有多出色？沃顿商学院的几位教授在他们的MBA创新班上进行了测试。他们让学生们提出十几个产品或服务创意，然后利用GPT-4模型对ChatGPT做了同样的测试。在学生们提出的创意中，有一些是适合宿舍使用的厨师套件，还有一些是为教室里的硬座椅准备的舒适坐垫。

教授们使用线上购买意向调查来检验这些创意。调查的问题是：如果这款产品在市场上能买到，你的购买意愿有多大？

平均而言，大约40%的学生创意相当扎实，而ChatGPT的得分略高，为49%。但教授们并没有就此止步。他们把重点放在排名前10%的创

意上，这些是真正有潜力改变游戏规则的创意。你猜结果如何？在40个顶级创意中，只有5个来自学生。

## 7.2 市场调研

在开始软件项目之前，提出如下3个关键性的问题至关重要。

- 是否真的需要这款工具？
- 客户是否愿意为之付费？
- 它的市场潜力究竟有多大？

这些问题构成了市场调研的基础。虽然市场调研并不是能够预测未来的神秘水晶球，但它无疑可以显著降低将宝贵时间和精力投入到可能无法取得成功的项目上的风险。

现实中，许多新产品最终并没有在市场上取得成功。哈佛大学教授兼畅销书作家Clayton Christensen对此进行了深入研究，他发现大约80%的新产品以失败告终。

在软件开发领域，失败的比例可能更高。表7-1列出了一些著名的失败案例。

表7-1 失败的软件产品/服务

软件产品/服务	发布年份	失败的原因
Friendster	2002	技术问题、可扩展性问题、被Facebook赶超
Microsoft Zune	2006	无法与iPod竞争，不断变化的音乐市场
Windows Vista	2007	系统要求高、兼容性问题、安全问题
Google Wave	2009	界面复杂、目的不明确、缺乏用户
Google+	2011	错误较多、数据不准确、导航能力差
Vine	2013	在竞争对手Instagram和Snapchat面前黯然失色
Amazon Fire Phone	2014	价格高、应用程序选择有限、功能不吸引人
Quibi	2020	短内容格式不受欢迎、竞争激烈等

在计算机游戏史上，有一个著名的失败案例——一款根据史蒂文·斯皮尔伯格1982年执导的电影《E.T.外星人》改编的视频游戏。雅达利公司为了获得这款游戏的版权，向环球影业支付了2100万美元，并另外投入了500万美元用于宣传推广。

然而，这款游戏推出后并未达到预期。它混乱不堪，缺乏趣味性，导致仅售出大约150万套。剩余的250万套游戏卡带最终被遗弃在新墨西哥州的垃圾填埋场，成了那个时代的遗憾。

现在，让我们深入了解如何利用ChatGPT进行市场调研。首先，我们需要确定目标受众，思考哪些群体能从软件中获得最大的价值。下面的提示可供参考。

**提示：项目管理软件的潜在目标人群有哪些？**

接下来，让ChatGPT帮助我们完成一项对市场营销至关重要的任务：创建详细的用户角色。可以将这些角色想象成不同类型用户的快照，他们可能是使用产品的不同类型的人。这些角色的细节通常包括年龄、兴趣、工作和生活方式等。通过描绘这些角色，我们可以了解客户是谁，以及他们在寻找什么。

当询问ChatGPT关于项目管理软件的用户角色时，它提供了一些详细的描述，如图7-1所示。

ChatGPT还能协助起草调查问卷，以收集特定数据。下面的提示可供参考。

**提示：制作一份调查问卷，重点了解买家对项目管理软件的需求。**调查应包括如下问题：有助于确定关键功能的问题、可用性偏好、特定行业需求，以及买家希望通过此类软件解决的特定挑战。此外，还应包含一些问题来衡量成本、支持、可扩展性和集成能力等因素的重要性。

在后续章节中，我们将继续介绍在评估软件市场时需要考虑的其他一些关键因素。



项目管理软件的潜在目标人群包括以下几类。

### 1. 企业和初创公司

- **大中型企业**: 需要协调多个部门和团队的工作，项目管理软件可以帮助他们跟踪项目进度、分配任务、管理资源和预算。
- **小型企业和初创公司**: 虽然规模较小，但仍然需要有效的项目管理工具来提高生产力，确保任务按时完成。

### 2. 专业服务公司

- **咨询公司**: 需要管理多个客户和项目，项目管理软件有助于跟踪进度、分配资源并确保按时交付。
- **法律事务所**: 需要管理案件进度和法律文档，项目管理软件可以提高效率和组织性。

### 3. 工程和建筑公司

- **建筑公司**: 项目管理软件可以帮助协调施工进度、管理预算和资源。
- **工程咨询公司**: 需要跟踪复杂项目的各个阶段，确保所有步骤都按计划进行。

### 4. IT和软件开发公司

- **软件开发公司**: 需要管理软件开发过程中的任务分配、代码审查、版本控制和计划发布等。
- **IT服务公司**: 项目管理软件可以帮助管理客户项目、跟踪服务请求和计划维护工作。

### 5. 创意和设计行业

- **广告公司和设计工作室**: 需要协调创意流程，确保设计项目按时完成并满足客户要求。
- **影视制作公司**: 需要跟踪拍摄进度、管理质量和预算。

图7-1 ChatGPT针对项目管理软件分析了用户角色

## 7.2.1 市场趋势

在推出新软件前，深入市场进行调研能让你大有作为。市场调研不仅能够揭示用户真正的追求，而且能让你洞察竞争对手的动向。市场调研可以作为你的秘密战术，它能让你在市场爆发前捕捉到趋势，发现那些可能被他人忽视的机会。此外，它还是一张安全网，可以让你减少盲目猜测，帮助你做出更有可能成功的明智决策。同时，市场调研

还能助力营销策略，对目标受众的深刻理解意味着能够制作出引起共鸣的广告和促销活动，而不是无差别地投放广告，忽视用户黏性。

ChatGPT是进行市场调研的得力助手。虽然它只训练到某一特定日期，但对此有一个巧妙的解决方案：在提示中告诉它“通过互联网进行研究”。

以下是一些提示范例。

**提示：**通过互联网进行研究，找出<特定软件类别，例如手机游戏、教育技术>截至<年份>的趋势。

**提示：**通过互联网进行研究，概述<年份>软件行业日益流行的新技术。

**提示：**通过互联网进行研究，找出当前影响软件市场的消费行为趋势，特别是<目标人群或应用类型>方面的趋势。

**提示：**通过互联网进行研究，找出<年份>软件用户界面和用户体验（UI/UX）设计的最新趋势。

**提示：**通过互联网进行研究，列出<特定领域或类别>的顶级软件，并解释它们在当前市场上取得成功的原因。

## 7.2.2 市场规模

市场规模（Total Addressable Market，TAM）<sup>注</sup>代表了，如果所有潜在客户都采用你的产品，你所能够实现的最大收入。它是一个关键指标，能帮助你确定应集中精力的领域，如何在市场上准确定位产品，以及制订发展计划。TAM是对市场的一次检验，它设定了可达成的目标，即你能占领多少市场份额，以及这将给你带来多少收益。

风险投资人（Venture Capitalist，VC）在考虑投资时会特别关注TAM，因为它揭示了企业的成长潜力。VC总是在寻找能让其投资获

得显著回报的机会，而庞大的TAM意味着存在一个巨大的市场等待挖掘。

通常，风险投资公司会寻找TAM超过10亿美元的投资机会，如果远超这个数字则更佳。毕竟在如此庞大的市场中，即便一家公司只占据一小部分市场份额，潜在利润也相当可观。

在回答类似“估算TAM都有哪些方法”的提问时，ChatGPT可能会这样回答。

### **ChatGPT：**

·自顶向下方法：首先收集行业数据和市场调研信息，了解整个市场的规模。然后逐步细化，聚焦于目标细分市场。

·自底向上方法：估算可能购买产品的人数或企业数量，预测其中实际购买的比例（转化率），并考虑每位客户的平均消费额。虽然自底向上方法需要更多的工作量，但它能提供更具体、更有针对性的洞察。

以一家医疗技术领域的初创公司为例，假设你输入以下提示。

**提示：**SilverConnect Health是一家专注于远程医疗服务的健康科技初创公司，目标市场在美国，重点面向城市地区的老年人。

SilverConnect Health目前的远程医疗服务市场规模有多大？该细分市场未来5~10年的预计增长率是多少？

ChatGPT在搜索互联网后提供了一些有价值的数据。

### **ChatGPT：**

·2022年市场规模：2022年，美国远程医疗服务市场的规模超过350亿美元。另有预测显示，同年全球远程医疗服务市场的总规模为912亿美元。

·预计增长率：一项预测显示，2023年至2032年的复合年均增长率为12.5%；而另一项预测则更为乐观，2023年至2030年的复合年均增长率为22.9%。

### 7.3 竞争分析

竞争分析为软件开发和市场定位提供了如下多重益处。

·市场洞察：竞争分析让你能够深入了解市场现状，观察已有的软件产品，分析哪些功能或服务受到用户的青睐，哪些未能满足市场需求。这有助于你发现市场空缺，确保你的产品能够带来创新和差异化。

·向竞争对手学习：通过研究竞争对手的成功案例和失误，你可以获得宝贵的经验，从而提升自身的竞争力。

·精准营销：通过明确你的软件都有哪些与众不同的特点，帮助你制定更有针对性的营销策略，进而有效吸引并打动目标用户群体。

ChatGPT能够在竞争分析中给你提供帮助。以下是一些有用的提示。

**提示：**列出<特定行业或市场>中<你的软件类型>的主要竞争对手。

**提示：**针对<你的软件>的功能与其前3个竞争对手进行比较。

**提示：**<你的软件>的竞争对手在市场上使用哪些定价模式？

**提示：**主要竞争对手如何在<软件类型>市场中定位自己？

**提示：**分析<竞争对手名称>软件的优势和劣势。

**提示：**总结<竞争对手的软件名称>的客户评论，重点关注用户满意度和他们遇到的问题。

为了使分析结果更加清晰易懂，你可以要求ChatGPT以表格形式呈现信息。

**提示：**概述截至2023年客户关系管理软件市场的主要参与者。请包含公司名称、主要特点、目标市场、独特卖点和市场份额，将结果以表格形式显示<sup>①</sup>。

ChatGPT的响应如图7-2所示。

公司名称	主要特点	目标市场	独特卖点	市场份额
Salesforce	综合CRM解决方案，广泛的定制化功能，强大的生态系统	大型企业，各类行业的中小企业	市场份额领先，广泛的第三方集成	约19.5%
Microsoft	无缝集成Microsoft产品，AI驱动的洞察分析	使用Microsoft生态系统的企，中小企业	最适合Microsoft用户，强大的AI能力	约6.2%
SAP	强大的分析功能，企业级解决方案，集成的业务流程	大型企业，特别是制造业和服务业	与SAP的ERP解决方案集成，强大的分析能力	约4.8%
Oracle	可扩展的解决方案，与Oracle产品深度集成，强大的数据管理能力	大型企业，现有Oracle客户	强大的数据库管理功能，可扩展性适用于大型企业	约3.4%
Adobe	客户旅程分析，营销自动化，AI能力	大型企业和中小企业的营销和销售团队	优越的营销自动化，AI驱动的客户洞察	约3.2%
HubSpot	内向营销，销售自动化，用户友好的界面	小型到中型企业，初创公司	易于使用，非常适合内向营销	约3.0%
Zoho	实惠的价格，广泛的应用程序，定制选项	中小企业，初创公司，各类行业	性价比高，多功能应用	约2.8%
Freshworks	用户友好，多渠道支持，强大的客户参与工具	中小企业，客户支持团队	易于使用，强大的支持功能	约2.6%
SugarCRM	灵活的部署，行业特定解决方案，强大的工作流程自动化	中型到大型企业，行业特定需求	高度可定制，适用于利基行业	约2.3%
Pipedrive	销售管道管理，直观的界面，性能跟踪	中小企业，销售团队	优秀的销售管道管理，直观的设计	约2.1%

## 图7-2 ChatGPT以表格形式提供客户关系管理软件市场的竞争分析结果

ChatGPT甚至可以进行SWOT [S ( Strengths ) 表示优势 , W ( Weaknesses ) 表示劣势 , O ( Opportunities ) 表示机会 , T ( Threats ) 表示威胁] 分析。SWOT可以帮助你发挥自己的强项，弥补或解决薄弱环节，抓住偶然出现的机会，躲避可能的意外。各种类型的企业都在使用它，它对个人职业发展也很有帮助。

## 7.4 需求分析

需求文档在软件开发过程中扮演着至关重要的角色，它们带来的好处如下。

### 指引方向

需求文档清晰地界定了软件的预期功能，确保从开发团队到客户的每一方都能对项目规划有共同的理解，维持工作的有序性。

### 保持目标一致性

需求文档确保项目所有参与者，包括团队成员和客户，对项目预期有清晰的认识，同时减少误解和偏离目标的风险。

### 帮助制订计划

当需要评估完成某项任务所需的时间或成本时，需求文档是规划工作量、时间线和预算的关键资源。

### 促进沟通

在大型团队或成员分散的情况下，需求文档是确保沟通清晰、有效的重要工具。

## **降低风险**

需求文档有助于提前识别潜在问题，避免项目延误，控制成本。

然而，制定软件需求并非易事，这也是软件开发较为复杂的一个重要原因。协调众多团队成员的工作和想法是一项挑战，尤其当团队成员来自不同的国家和时区，拥有不同的见解和需求时。有时，这些想法并不完全一致，需求本身也不明确。

另外，软件需求本身是动态变化的。在快节奏的行业，需求的不断变化尤为常见。因此，要在一个持续变化的目标上实现需求，同时确保需求的可行性，并验证它们是否得到满足，是一项艰巨的任务。

当需求管理偏离轨道时，就有可能导致灾难性的后果，以下是一些典型案例。

### **丹佛国际机场的自动行李系统（1995年）**

系统的目标过于雄心勃勃，需求复杂，导致项目延误16个月，成本大幅超支。最终，机场不得不放弃整个系统。

### **美国联邦调查局的虚拟案件档案系统（2000—2005年）**

美国联邦调查局对自身需求的理解模糊不清，导致系统未能发挥预期作用。尽管投入超过1.7亿美元，项目最终仍被放弃。后来的Sentinel（哨兵）项目由于需求更加具体明确，取得了成功。

### **美国医保网站的推出（2013年）**

美国医保网站未能明确需求，也未进行充分测试，上线后无法承载访问量，功能故障频发。发布失败后，美国政府不得不投入更多资源进行修复。

在软件开发领域，需求分为功能性需求和非功能性需求两大类。功能性需求规定了软件应执行的具体任务或提供的功能，而非功能性需求

则涉及软件的性能、响应速度、安全性和可用性等。

为了深入探讨生成式AI如何在需求管理方面提供帮助，接下来的内容将重点关注两种关键文档——产品需求文档和软件需求规约。

#### 7.4.1 产品需求文档

产品需求文档（Product Requirements Document，PRD）是软件开发的蓝图，它定义了软件的内容和功能。这份文档将应用程序构想成一个引人入胜的故事，并详细描述了它的预期形态、运作方式、目标用户及它所解决的核心问题。优秀的PRD通常非常详尽，覆盖从核心功能到用户体验的每一个细节。

PRD面向的受众广泛而多元。它不仅面向开发者、设计师、项目经理和质量保证团队，为这些关键角色提供指导，而且对高管和其他需要把握产品愿景与目标的利益相关者来说同样至关重要。此外，PRD还能帮助营销团队明确产品的市场定位。总体而言，PRD适用于所有参与产品开发流程、推广和确保产品能够吸引目标用户群体的成员。

如果你尚未接触过PRD，或者需要一些帮助来创建自己的PRD，则ChatGPT可以为你提供帮助。

**提示：**产品需求文档包含哪些主要部分？请创建一个Word文档模板作为参考。

ChatGPT生成的文档如图7-3所示。

# 产品需求文档

## 1. 引言

在本文档中，描述了 [产品名称] 的需求规格和功能。该产品旨在满足 [目标市场] 的需求，并提供 [解决方案概述]。

## 2. 目标市场

描述产品的目标市场，包括主要用户群体、行业类型和地理位置。

## 3. 产品概述

描述产品的基本功能和特性，以及解决的核心问题。

## 4. 功能需求

列出产品的功能需求，包括用户界面、用户操作流程和系统功能。

### 4.1 用户界面

描述产品的用户界面设计要求，包括布局、颜色、字体等。

### 4.2 用户操作流程

描述用户如何使用产品，包括登录、设置、创建项目等操作流程。



### 4.3 系统功能

图7-3 ChatGPT创建了产品需求文档的大纲

## 7.4.2 软件需求规约

软件需求规约 ( Software Requirements Specification , SRS ) 与 PRD 在本质上有着明显的区别。 SRS 可以视作软件项目的技术手册，它深入探讨软件的实现细节，包括软件需要完成什么任务以及如何完成这些任务。与 PRD 从最终用户或客户的角度出发，关注他们的需求和背后的原因不同， SRS 专注于技术层面的深入分析。

对于一个简单的应用程序，SRS可能仅包含数页内容。然而，对于一个更为复杂的项目，如企业级软件实施，SRS的篇幅可以轻松扩展至数百页。

以下是一些有助于你更好地理解和创建SRS的提示。

**提示：**起草一份有效的软件需求规约需要考虑哪些基本要素和最佳实践？

**提示：**概述编写软件需求规约的流程，强调需求收集和分析、利益相关者协作和文档编制所涉及的步骤。

**提示：**为软件项目创建详细的需求规格说明，内容包括项目概述、利益相关者分析、功能性需求和非功能性需求、假设条件和限制条件。<在此处添加你希望包含的每个细节。>

### 7.4.3 访谈

语音识别技术自诞生以来已有数十年历史，但在过去，它往往被视为一种有趣而非必需的工具。传统上，这种技术在日常使用或专业场合中并不总能提供可靠的服务，主要因为它难以准确捕捉和理解自然对话中的语音。

然而，近年来，人工智能的飞速发展使语音识别技术有了革命性的突破。现代的语音识别系统由先进的AI算法驱动，它们不仅能够更精准地识别各种口音和方言，而且能够理解口语中的语境和语言的微妙之处。随着时间的推移，这些系统通过持续学习不断优化自身，显著提高了语音识别的准确率。

语音识别系统在整理需求文档方面展现出巨大的潜力，尤其表现在以下方面。

#### 准确性和回放

现代语音识别系统的准确率已经相当高，而且你可以通过回放功能进行校正，确保记录的准确性。

### **识别发言者**

语音识别系统能够区分会议中的不同发言者，这有助于清晰地记录和理解每个人的需求。

### **识别关键主题**

语音识别系统能够识别并突出显示文字记录中的关键词，让你更容易捕捉和关注讨论中的主要议题或需求。

### **总结要点**

语音识别系统能够自动生成摘要和建议，帮助开发团队快速把握关键需求，并将它们转换为具体的项目任务。

目前市场上一些流行的语音识别工具如下。

· Otter AI。

· Fathom。

· Rev。

· Sonix。

· Notta。

### **7.4.4 白板讨论**

在梳理和汇总软件需求的过程中，白板讨论是一种极具价值的方法。这种方法不仅开启了集体头脑风暴，而且确保了团队成员的广泛参与。白板讨论能够汇聚多样化的想法和视角，这对于明确软件的功能和目标至关重要。此外，这种方法具有极高的实践性，允许参与者在

讨论的同时绘制流程图或图表，将复杂的概念和流程具象化，使理解变得更加直观和容易。

ChatGPT甚至可以解读你上传到系统中的白板图像。假设你设计了一张系统登录草图，如图7-4所示。

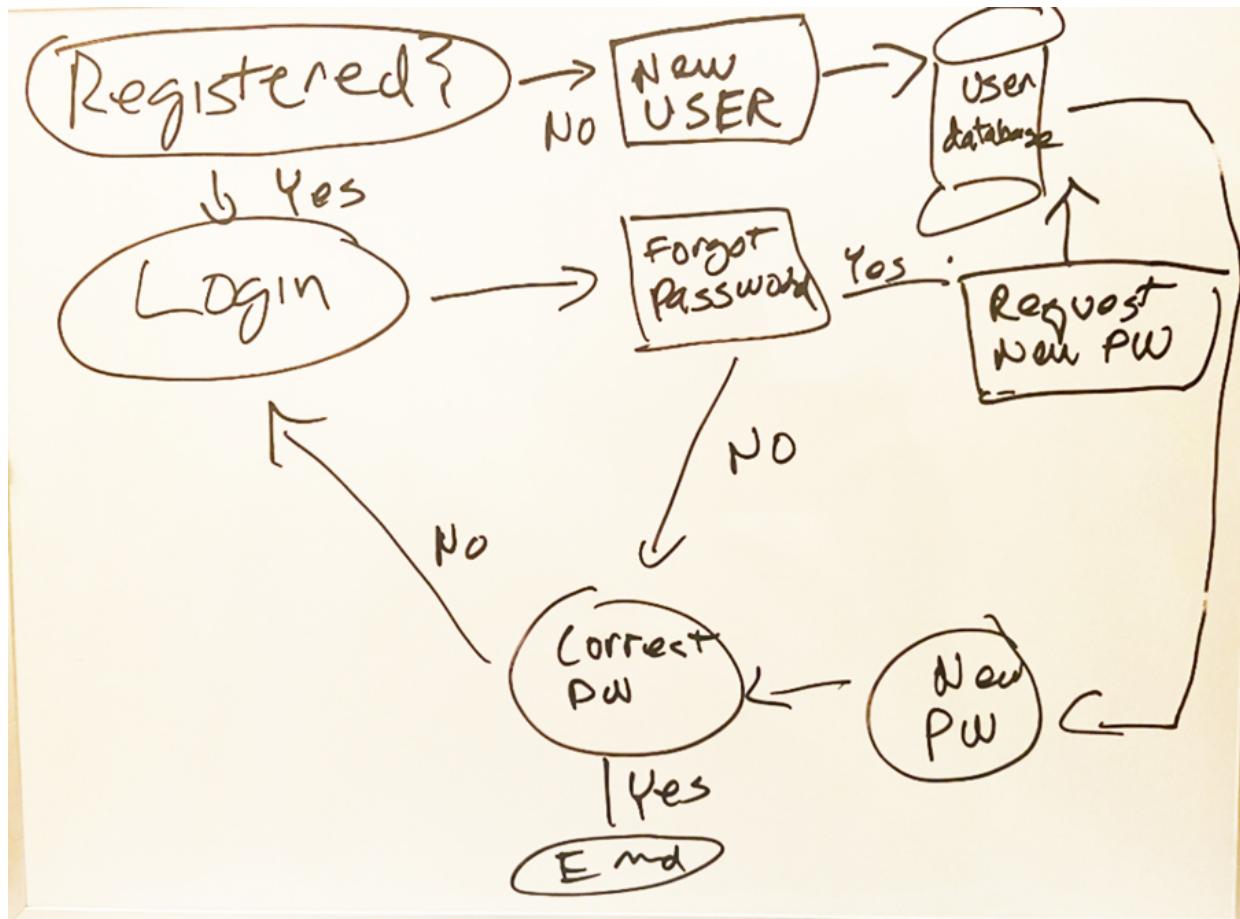


图7-4 系统登录草图

如你所见，草图中的笔迹可能稍显潦草，并且使用了简写形式，例如将password（密码）简写为pw。

让我们看看ChatGPT是如何识别这样的草图的。

**提示：**请分析这张图。

ChatGPT：你上传的图似乎是与用户账户管理有关的流程图，更像是网站或应用程序的登录流程图。

随后，ChatGPT将根据图片内容，提供对整个流程的详细阐述，如图7-5所示。

**1. Registered?**

- The starting point asks if the user is registered.

**2. Yes (Registered User)**

- If the user is registered, they are directed to 'Login'.

**3. No (New User)**

- If the user is not registered, it points them to the 'New User' process where they presumably register and get added to the 'User database'.

**4. Login**

- The user attempts to log in.

**5. Forgot Password**

- If the user has forgotten their password, there is an option to select 'Forgot Password'.

**6. Request New PW (Password)**

- If they have forgotten their password, they can request a new one.

**7. Correct PW (Password)**

- After attempting to log in, the process checks if the password entered is correct.

**8. No (Incorrect Password)**

- If the password is not correct, it loops back, suggesting the user may need to either remember the correct password or go through the 'Forgot Password' process.

**9. Yes (Correct Password)**

- If the password is correct, the user proceeds to what appears to be 'Email', which might imply email verification or simply entering the system that likely involves an email as a username.

图7-5 ChatGPT分析了这张系统登录草图，并给出了易于阅读的细分工作流程

## 7.4.5 文档基调

撰写出色的产品需求文档和软件需求规约堪称一门精密的艺术。确立恰当的文档基调至关重要。这些文档必须达到清晰、简洁、专业的标准。

写作风格的重要性体现在以下几个方面。

### **清晰明了**

产品需求文档和软件需求规约好比项目的指南针。我们当然希望所有内容都精确无误，以确保每个人都对项目的进展有清晰的认识。这有助于避免沟通障碍和开发错误。

### **实事求是**

产品需求文档和软件需求规约不是展示个人风格或华丽辞藻的舞台。你需要像新闻记者一样，坚持用事实说话。只有保持直接和中立，才能确保文档内容专注于产品的实际需求。

### **一致性是关键**

你需要从始至终保持同样的基调。就好比食谱，如果改变配方，饭菜的味道可能会完全改变。一致性使庞大的文档更易于理解和遵循。

### **始终关注产品**

这些文档的核心是产品及其实现的功能。保持直接的语气可以确保你不偏离主题，始终聚焦于产品的功能和需求。

在写作方面，LLM（如ChatGPT）优势明显。它能够显著提高写作质量。你可以通过指导模型来采用正确的基调，例如下面的提示。

**提示：**请起草一份完整的移动银行应用程序技术需求清单，确保写作清晰、结构良好且具有专业性。

市场上有大量的AI写作工具可供选择，每种工具都有自己的特色，但总有一款契合你的写作偏好。例如，Jasper和Writer非常适合撰写内容并保持品牌声音的一致性；Grammarly专注于解决语法问题，提升写作风格；AI21和Copy.ai则让内容创作变得更加轻松。

Jigyasa Grover在她自己的项目中使用了Grammarly。作为机器学习方面的专家，她撰写了一本关于该主题的书——*Sculpting Data for ML*（2021年）。她是这样评价Grammarly的：

Grammarly是另一款帮助我起草技术设计文档的工具。我使用它来获得校对、语法、标点和写作风格方面的建议。它支持与多种网络浏览器和文字处理软件无缝集成。它的语言编排清晰明了，甚至可以根据目标受众和文档目的提供定制化建议。

## 7.5 项目规划方法

在需求被明确提出之后，紧接着的关键步骤是规划项目的执行路径。这相当于为团队描绘出完成软件创造之旅的行军图，涉及设定清晰的里程碑，以及明确任务分工、时间线和执行策略。其核心目标是确保工作流程顺畅，让每一位团队成员都能高效地贡献自己的力量，最终实现产品的预期目标，并赢得所有利益相关者，尤其是客户的满意和认可。

由于软件开发领域充满复杂性，因此团队在规划项目时有多种选择。不同的方法论提供了多样化的策略、规则和步骤，以帮助团队应对任务分配、变化管理、质量保障以及向客户交付卓越产品。无论团队倾向于详尽的事先规划，还是倾向于更灵活、适应性强的计划以随时调整方向，总有一种方法论能够契合团队的需求。

常见的项目规划方法有瀑布模型、敏捷开发、Scrum、看板和极限编程（eXtreme Programming，XP）。选择恰当的方法论至关重

要，因为这可能直接影响项目的成败。方法论的选择取决于诸多因素，如项目规模、复杂性、客户需求、团队协作模式以及项目中存在的确定性和不确定性。有些团队可能坚持使用单一的方法论，而其他团队可能会混合使用多种方法论，以找到最适合其项目需求的策略。

ChatGPT能够在选择合适的方法论方面提供帮助。例如，假设你正在开发一个“待办事项列表”应用程序，并已为其制定了一份简明的需求文档。此时，你可以向ChatGPT提出以下问题。

**提示：**对于这个应用程序，哪种项目规划方法最合适？

ChatGPT可能会建议采用敏捷开发方法，并提供一些理由，如图7-6所示。



对于待办事项列表应用程序，最适宜的项目规划方法是敏捷开发方法。下面是一些理由。

1. **需求的不确定性：**通常情况下，初期制定的需求文档可能无法覆盖所有细节，而且用户需求可能会随着时间的推移而变化。敏捷方法可以更好地适应这种变化，通过迭代开发的方式，不断调整和完善功能，确保最终产品符合用户期望。
2. **用户反馈的重要性：**待办事项列表应用程序属于用户体验至关重要的应用类型。用户对界面设计、功能操作的反馈很重要，而敏捷方法可以在每个迭代中及时收集用户反馈，并根据反馈进行调整和改进。
3. **快速交付功能：**敏捷方法以短周期的迭代开发为特点，可以快速交付部分功能。这意味着用户可以在较短的时间内开始使用应用，并逐步体验新增的功能，而不需要等待所有功能开发完成后再进行发布。
4. **团队协作和透明度：**敏捷方法强调团队协作和透明度，通过每日站会、迭代评审会等方式，团队成员可以及时沟通，协作解决问题，保持项目进度的可见性和透明度。
5. **风险管理：**敏捷方法将风险分散到每个迭代中，而不是将所有风险都集中在项目末期。这样做可以及早发现和解决问题，降低项目失败的风险。

图7-6 ChatGPT解释了为什么敏捷开发方法是开发特定应用程序的最适宜的项目规划方法

下面是一些提示范例。

**提示：**团队如何有效地选择并定制项目规划方法，以适应特定的软件项目？

**提示：**讨论根据软件项目的目标和复杂程度调整项目规划方法的重要性。

**提示：**解释软件项目管理中瀑布模型的顺序阶段。

**提示：**描述软件开发中敏捷开发方法的核心原则。

**提示：**敏捷开发方法是如何提高项目的灵活性和客户参与度的？

**提示：**概述Scrum框架中涉及的角色和仪式（ceremony）。

**提示：**对比Scrum和传统项目规划方法的异同。

**提示：**列出极限编程的主要做法，以及这些做法如何提高客户满意度。

**提示：**讨论将测试驱动开发作为极限编程一部分实施的好处和挑战。

### 7.5.1 测试驱动开发

在开始编写代码之前，制定测试用例是明智的首要步骤。有句古语说得好，“三思而后行”，这一建议在软件开发领域同样适用。在软件开发中，测试驱动开发（Test-Driven Development，TDD）是一种常见的先行策略。

TDD的优势在于，当你深入思考代码需要通过的所有测试时，你会更清晰地理解所需构建的内容。在开始编程前，细致地审视蓝图，将帮助你明确代码的目标，从而避免许多潜在的麻烦。

此外，先行规划测试意味着从用户的角度进行思考，专注于为用户提供高效、易用的软件，避免陷入代码的细节中。尽早发现潜在问题，远胜于事后匆忙修复错误。

在TDD中，测试不仅是开发过程的核心，更是灵魂。在编写代码前先编写测试，要求你始终关注代码的实现目标。TDD促使你带着明确的目的编程，使代码更加简洁高效。

借助TDD，你可以采用如下不同的方法论来构建测试。

### **Given-When-Then ( GWT )**

G ( Given ) 代表设定条件，即操作前的状态；W ( When ) 代表触发事件，即导致变化的动作；T ( Then ) 代表预期结果，即操作后应达到的状态。GWT通过讲故事的方式，阐明代码的执行逻辑。

### **Arrange-Act-Assert ( AAA )**

AAA与GWT类似。但GWT更倾向于以用户为中心讲故事，是开发者与测试人员和非技术利益相关者开展协作的理想选择。AAA则更注重测试的执行。

### **Setup-Exercise-Verify-TearDown ( SEVT )**

SEVT是开展集成和系统测试的关键流程。从设置 ( Setup ) 准备测试环境开始，经过执行 ( Exercise ) 对系统进行测试，再到验证 ( Verify ) 检查结果，最后拆卸 ( Teardown ) 清理环境，为下一次测试做好准备。

以下是使用GWT方法论的TDD提示范例。

**提示：**使用GWT方法论为系统关键功能制定测试策略。系统具体要求如下。

- 开发一个连锁精品酒店的在线预订系统。
- 系统需要从各酒店的内部管理软件提取客房供应数据。
- 允许用户根据日期、价格和设施筛选房间。

- 界面必须直观且兼容桌面及移动浏览器。
- 采用云解决方案以确保系统的可靠性和可扩展性。

TDD还专注于特定场景或用例的测试。以下是一些提示范例。

**提示：**编写测试用例，验证用户通过电子邮件验证以重置密码的功能。

**提示：**编写测试用例，验证数据库连接失败时系统的响应。

**提示：**编写测试用例，确保搜索功能在正常负载条件下2秒内返回结果。

**提示：**编写测试用例，检查所有用户数据在存储时是否已加密。

**提示：**编写测试用例，验证支付网关与订单处理系统之间是否集成良好。

## 7.5.2 Web设计规划

网站或Web应用程序的规划是一个多步骤且可能相当复杂的过程。首先必须清晰地界定网站的目标和目标用户群体；然后绘制网站的蓝图，规划用户的交互体验。Web设计规划通常又称为原型（Prototype）或线框图（Wireframe）。

原型设计完成后，接下来的任务是通过选择色彩、字体和时尚图形来体现品牌精神，提升网页的视觉效果。同时利用引人入胜的文案、引人注目的图片或视频来讲述品牌故事。最后是技术实现阶段，运用HTML、CSS和JavaScript等技术，将所有创意转化为一个真实、可交互的网站。

在Web设计规划阶段，像ChatGPT这样的AI工具可以提供巨大帮助。以下是一些提示范例。

**提示**：我正在开发一个<某种类别的>网站。请提出一些目的或目标。  
目标受众是谁？

**提示**：在线商店一般需要包含哪些页面？

**提示**：我正在为博客主页制作线框图。请建议需要包含的重要元素。

**提示**：我正在为移动应用程序的登录页面制作线框图。我应该确保包含哪些关键部分？

**提示**：对于<某种类别的>网站，有哪些好的内容创意，比如博客文章、视频和信息图表？

**提示**：如何构建网站的导航功能，使其对用户来说直观且友好？

**提示**：列出关键的SEO策略，提高网站在搜索引擎中的可见度。

**提示**：对于<特定类别的>网站，你推荐什么样的调色板？

**提示**：为网站选择适合<特定类别>的字体样式。考虑字体的可读性、品牌一致性，以及对网站整体美感的贡献程度。

此外，还有多种AI工具可以帮助我们创建线框图。Uizard就是其中之一。它与Figma等系统类似，具有设计功能，以及拖放按钮和表单等元素的功能，并支持团队协作。

在Uizard中，AI功能位于界面左侧的工具栏上。单击Magic按钮，即可看到AI功能列表。

当选择Autodesigner来创建线框图时，会弹出一个向导，如图7-7所示。

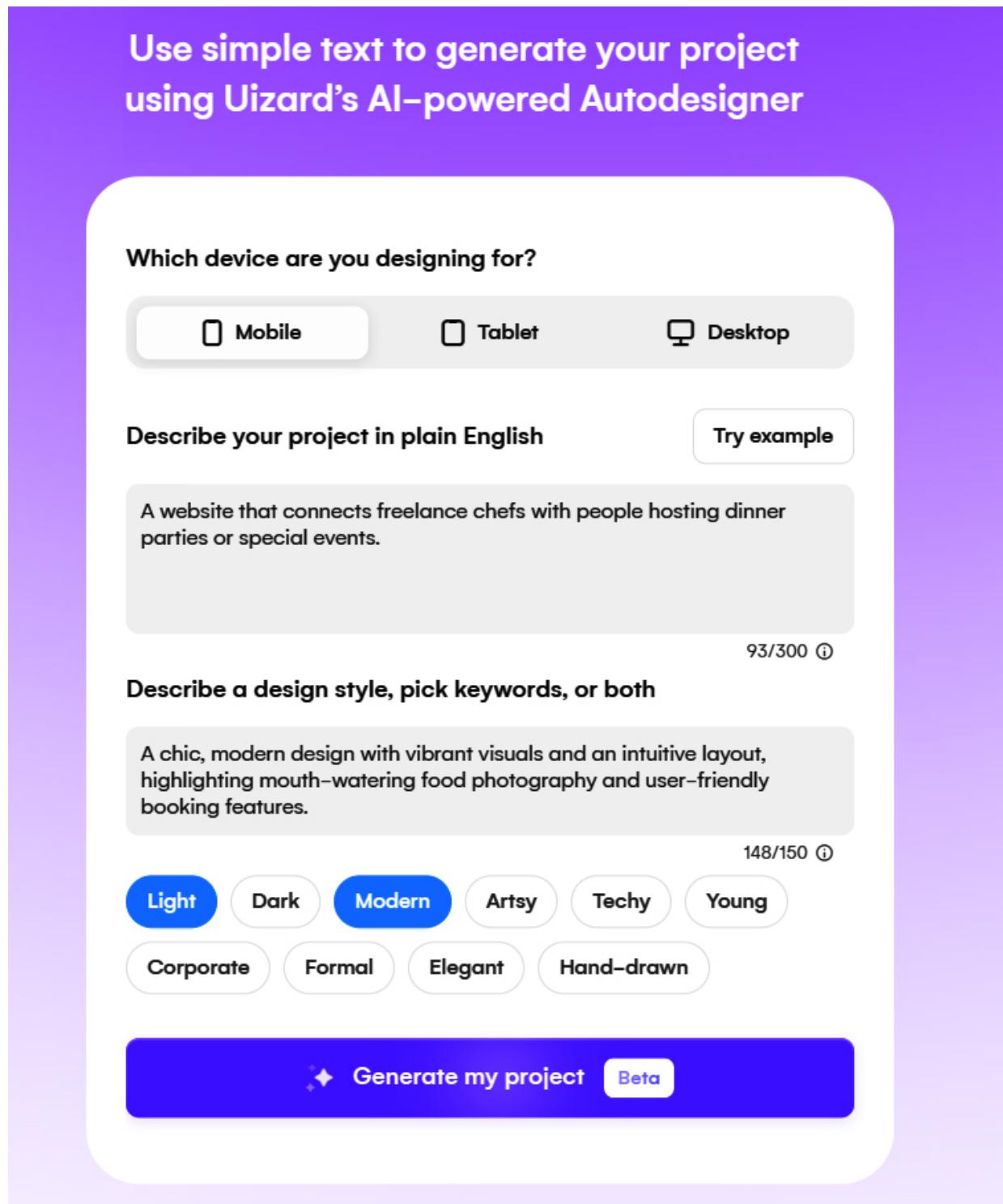


图7-7 Uizard的AI向导，用于为移动应用程序或网站创建线框图

首先，选择设备类型——手机、Pad或网页，然后简要描述项目。下面的提示可供参考。

**提示：**一个连接自由职业的厨师与举办晚宴或特别活动的人士的网站。

最后，定义设计风格。下面的提示可供参考。

**提示：**别致、现代化的设计，具有活力的视觉效果和直观的布局，突出令人垂涎的美食摄影和用户友好的预订功能。

Uizard根据提示为网站生成的线框图如图7-8所示。

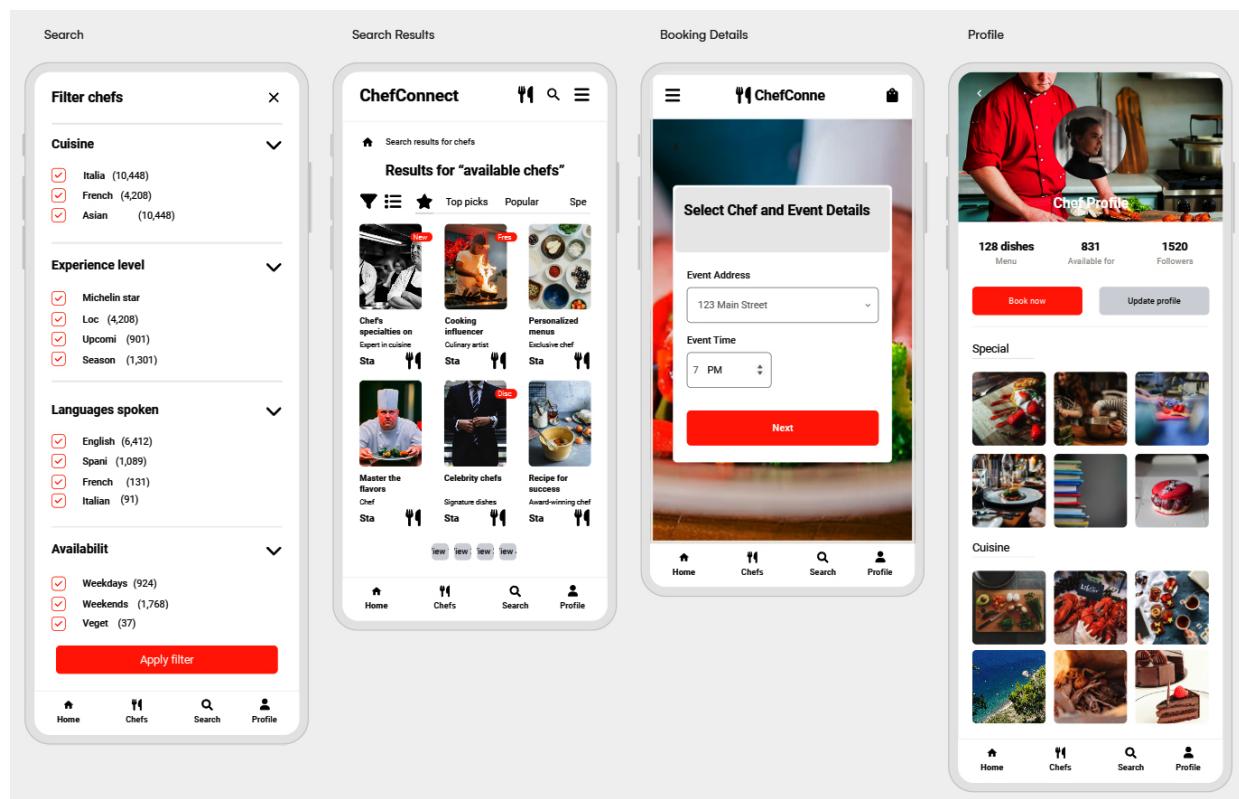


图7-8 Uizard根据提示为网站生成的线框图

## 7.6 小结

本章深入探讨了如何借助ChatGPT来启动和推进软件项目。我们的旅程始于一场头脑风暴，旨在激发创意和集思广益。随后，我们进行了深入的市场调研，以确保项目的方向与市场需求相契合。

本章细致地阐述了需求起草的各个环节，同时特别强调了产品需求文档和软件需求规约的重要性，这两份文档是确保项目目标和技术细节得以实现的关键。

本章还介绍了不同的项目规划方法，从灵活多变的敏捷开发方法到严格有序的瀑布模型，各有千秋。同时，本章也涵盖了白板讨论和测试驱动开发等技术。

通过将ChatGPT这样的先进AI工具与经过时间检验的方法论相结合，可以为软件项目打下一个坚实的基础。

**[10]**Total Addressable Market通常被翻译为市场规模、潜在市场规模、潜在市场范围、潜在市场目标等。本书统一译为“市场规模”。——译者注

**[11]**ChatGPT基于概率算法给出回复，这意味着可能存在一定的不确定性。为了获得更贴近原书内容的答案，建议在提问时明确指定所需的信息，以确保获得更准确和相关的回答。——译者注

**[12]**图7-2所示的表格由GPT-4o提供，读者在实际应用中，可以根据个人需求，提供更为详尽的提示，以定制化获取更精准的信息和答案。——译者注

## 第8章

# 编写代码

本章将结合AI工具深入探索一系列高效的编程技巧：从编程的基本要素开始——深入挖掘函数和类的潜力；接着着手进行代码重构，将那些杂乱无章的代码片段转换为流畅、高效的代码结构。本章还将展示数据的处理与创建方法，末尾将介绍一系列AI工具，这些工具能够在开发者轻松单击之后，自动生成令人印象深刻的前端界面。

## 8.1 现状：AI辅助编程工具能替代开发者吗

AI辅助编程工具的确令人赞叹，它们为编程带来了前所未有的便利。但我们也应该实事求是，它们不是万能钥匙，并不能解决所有的编程难题。这些工具在某种程度上类似于那些仅依赖公开笔记来学习的开发者。它们通过互联网上的代码进行训练，而这些代码的质量参差不齐，并非所有都是实践的典范。因此，由这些工具生成的代码有时可能过于冗长，难以维护。在某些情况下，它们生成的代码可能存在错误，无法满足具体需求。更令人担忧的是，这些代码可能无意中为黑客提供了可乘之机，消耗了宝贵的网络资源。

以下示例揭示了ChatGPT生成的一段不怎么完美的代码。如图8-1所示，这段代码旨在展示如何将两个已排序的列表合并为一个新的列表。这段代码确实正确地合并了两个列表，但它采用了一种烦琐的方法：分别处理空列表的情况，然后使用两个额外的while循环来追加每个列表中剩余的元素。

这正是你的编程智慧真正发挥作用的地方。你必须了解基础知识，并且需要知道在事情不太对劲时应该问什么问题以及对哪里进行探究。

图8-2展示了一种更好的对已排序列表进行合并的方法。

python

 Copy code

```
def merge_lists(list1, list2):
    if not list1:
        return list2
    if not list2:
        return list1

    merged_list = []
    index1, index2 = 0, 0

    while index1 < len(list1) and index2 < len(list2):
        if list1[index1] < list2[index2]:
            merged_list.append(list1[index1])
            index1 += 1
        else:
            merged_list.append(list2[index2])
            index2 += 1

    while index1 < len(list1):
        merged_list.append(list1[index1])
        index1 += 1

    while index2 < len(list2):
        merged_list.append(list2[index2])
        index2 += 1

    return merged_list

# Example usage
print(merge_lists([1, 3, 5], [2, 4, 6]))
```

▲图8-1 当被要求合并两个已排序的列表时，ChatGPT创建了一些混乱的代码

python

Copy code

```
def merge_lists_efficient(list1, list2):
    # Merging and sorting the two lists
    return sorted(list1 + list2)
```

▲图8-2 在开发者的帮助下，AI实现了一种对已排序列表进行合并的更好方法

因此，当你听到那些关于机器人将取代开发者工作的悲观预言时，大可不必感到忧虑。的确，AI模型正在不断地进化和学习，但它们远未达到能够完全掌握编程艺术的程度，也绝不可能成为技术的终结者。开发者仍然是软件开发领域不可或缺的主角。

AI辅助编程工具的效用不容小觑，但它们的作用是辅助而非替代。它们无法取代软件工程师所具备的深厚智慧和精湛技艺。

在与AI辅助编程系统合作时，请时刻记住一点：开发者与AI辅助编程系统之间是伙伴关系，而非进行权力的交接。保持你的敏锐洞察力和无尽好奇心，审慎地审视AI辅助编程工具所提供的每一个建议。它们是我们的助手和合作伙伴，但绝非我们依赖的英雄。

## 8.2 决策：是否采用AI辅助编程工具

在某些情况下，亲自动手编程可能比依赖AI辅助编程工具更为直接和简单。毕竟想让AI准确理解需求，可能需要进行多次尝试和调整。然而，随着对AI辅助编程工具使用的深入，你将逐渐学会判断何时该利用它们的帮助，何时更适合独立完成任务。

以MY.GAMES的首席软件工程师Dmitrii Ivashchenko的亲身体验为例。MY.GAMES是一家总部设立在阿姆斯特丹的游戏公司，其游戏产品在全球的注册用户数量超过10亿。Dmitrii分享了他的见解：

例如，使用提示“用Python编写一个方法，为日期时间对象添加默认时区”。但是，你需要做好心理准备：很多细节可能会被忽略。你可能需要自己处理极端情况，或者与ChatGPT进行较为漫长的沟通，以纠正其错误。在做决策时，一个关键点是初步评估——比较解释一项任务给AI所需的时间与亲自实现它所需的时间。

### 8.3 提高学习效率

利用ChatGPT等工具学习新的编程语言、框架或库，可以极大地提升学习效率。相较于传统的学习方式，如参加培训课程、不断在线搜索或观看大量网络视频，ChatGPT提供的个性化和即时反馈可能更为高效。

可以通过ChatGPT快速掌握代码的基础知识和语法要点。如果倾向于通过实践来学习，可以请求ChatGPT提供代码示例，直观地展示如何综合应用概念。如果已经熟悉某种编程语言，ChatGPT则能通过比较学习法，帮助你更快掌握新的编程语言。例如，如果你精通Python并希望学习JavaScript，ChatGPT将使用Python中的术语来解释两者的主要差异。如果你需要更多的资源，ChatGPT则会在互联网上搜索并向你推荐一些值得信赖的学习材料。

下面是一些实用的提示范例。

**提示：**对于Python初学者，应该首先学习哪些内容？

**提示：**哪些适合初学者的项目可以帮助我提高React技能？

**提示：**推荐一些学习高级SQL查询的资源。

**提示：**如何用Python和JavaScript分别实现一个简单的“待办事项列表”应用程序，以突出这两种编程语言之间的差异？

**提示**：我熟悉如何在C++中使用循环。你能通过比较来告诉我循环在Python中是如何工作的吗？

**提示**：查找5个有助于学习Rust的网络视频。

让我们来看看Ivashchenko是如何利用ChatGPT来加速学习的。Ivashchenko分享了自己的心得：

要想迅速掌握一种新的编程语言或框架，可以直接用ChatGPT创建一个新项目。比如学习Django，可以选择一个Django项目，如构建一个在线商店。首先询问ChatGPT如何用Django开发新项目，它会指导你需要安装哪些依赖项，如何设置项目配置，以及需要掌握哪些技术。然后，逐步深入地提一些更具体的问题，如“如何在Django中组织模型？”“如何在管理面板中添加新部分？”以及“如何进行数据库迁移？”。通过这种方式，我仅用一个周末就高效地掌握了Django。

你还可以让ChatGPT创建有用的学习指南。如下是提示范例。

**提示**：为初学者创建JavaScript学习指南。

然后用以下提示跟进。

**提示**：请为我制定一份学习计划表。我应该学习哪些主题？有哪些练习或测验？遇到编程问题如何处理？有哪些资源链接？

ChatGPT甚至可以帮助你记住新概念（使用缩略语等记忆工具）。例如，假设你想记住DevOps的关键要素，其中包括持续集成、持续交付、微服务和基础设施即代码。以下是ChatGPT的建议。

**ChatGPT**：CI-CD-MI（发音为“Ci-Ci-D-Mi”）。

最后，ChatGPT还能在你准备LeetCode面试题时提供帮助。想象一下，在面试中，你可能需要在白板上或笔记本电脑上解决一个算法或

数据结构问题。这不仅考查你编写代码的能力，更是对你思维速度、解释想法的能力和解决问题风格的综合考量。如下是提示范例。

**提示：**建议3道常见的LeetCode面试题。

ChatGPT随后可能会提出一些挑战性问题，如“两数之和”“无重复字符的最长子串”“有效的括号”。你可以进一步深入研究，获取更长的解释或示例。

## 8.4 高效注释

在紧张的项目截止日期的压力下，开发者有时会忽略编写代码注释，甚至完全不加注释。然而，AI辅助编程工具可以轻松解决这一问题。这些工具能够提供已经包含注释的代码建议，帮助提升代码的可读性和可维护性。

随着AI技术的不断进步，依赖传统的注释方式可能显得有些落伍。面对难以理解的代码段，直接利用工具进行解析和注释，无疑是一种高效的解决方案。

更有趣的是，撰写提示正在演变成一种新的注释形式。许多工具已经开始在注释中嵌入有用的提示，以指导开发者更好地理解和使用代码。

如果想给某些代码添加注释，可以使用如下这个简单的提示。

**提示：**根据最佳编程实践添加清晰的注释。

是否编写注释并没有一成不变的规则，这取决于开发者和团队的具体要求及偏好。但可以肯定的是，利用AI辅助编程工具在代码中添加注释是一项可以轻松完成的任务。

## 8.5 模块化编程

模块化编程是高效软件开发的基石。通过模块化编程，代码将变得更加有序、易于理解，并且能够灵活适应变化。这种编程方式还能简化团队协作，每位成员都能在自己的模块内工作，而相互不干扰。此外，模块的可复用性不仅让开发者节省了大量时间，还有助于保持代码的一致性，降低出错的概率。

模块化编程的优势同样适用于AI辅助编程的实践。

Private Market Labs的联合创始人兼首席产品官Titus Capilnean 分享了自己的体会：

自从采用AI工具，我能够专注于问题本身及其解决方案，而不必深陷于编写代码的细节。面对技术难题，我会先将它分解成更小的部分，并确保每一部分的输入和输出都清晰明确。进行这种分解的原因是，我使用的AI工具的上下文窗口有限，无法一次性提供一个完整的解决方案。我发现，如果只要求模型提供支持简单输入、单一功能、可验证输出的代码片段，调试和构建过程将变得更加顺畅。如果我对中间结果不满意，我会向ChatGPT描述我的问题，并尝试运行它所提供的更新代码。一旦我对解决方案感到满意，我就会添加额外的处理步骤，比如在现有函数中增加逻辑，或者进一步细化前一步的输出，以使其更接近最终目标。

AI工具还极大提升了我的工作效率。在将代码发送给编译器、部署到云函数或检查错误和资源消耗之前，我可以利用这些工具检查代码片段的正确性：只需要将代码粘贴到ChatGPT窗口中，并询问它们是否正确即可。前提是我已经初始化了运行代码的环境或提供了正确的上下文信息，如必要的导入语句、变量、常量和自定义函数等。这样我便能够生成更高效、更优化的解决方案。

## 8.6 快速启动项目

在编程项目的初期，开发者常常会面临所谓的“冷启动问题”或“空白页难题”。想象一下，你眼前是一片空白的屏幕，没有代码，也没有数

据，更没有明确的行动方向。这种情况听起来足以令人感到迷茫。启动项目时，面临的首要难题便是确定项目的架构、设计模式和技术栈。这些决策至关重要，因为你将与它们长期相伴，从项目伊始就应该做出明智的选择。

同时，我们也不能忽视人的因素。要确保整个团队达成共识、找到高效的沟通方式、零基础建立起工作流程——这些沟通上的挑战与技术难题一样考验人。这不仅仅是编写第一行代码那么简单，而是要为项目未来的成功打下坚实的基础。启动一个项目需要精心的规划、扎实的技术知识和团队间的默契合作。

AI辅助编程工具在这一过程中可以发挥巨大的作用。它们能够帮助你快速搭建应用程序的基础框架，并根据你的愿景提供一个定制化的起点。这些工具能够让你绕过烦琐的初始设置阶段，直接进入项目中更富有创造性的部分。

你只需要提出具体的任务需求，然后让LLM为你生成模板或初始代码即可。例如以下提示。

**提示：**为Web应用程序生成启动代码，该应用程序旨在将用户的社交媒体信息从多个平台聚合到一个单一的仪表板界面上。对于编程语言和框架的选择，你有何建议？通用的文件结构又应该是怎样的？

根据以上提示，ChatGPT首先建议使用React来创建流畅的单页面应用程序。它还建议使用Redux来维护强样式且可重复使用的组件。对于后端，它建议使用Node.js运行应用程序，并利用Express.js构建API端点。对于用户认证，它推荐使用Passport.js。而对于数据库和模型的构建，它推荐使用MongoDB和Mongoose。至于通用文件结构，可以参考图8-3。

## General File Structure:

```
bash                                         Copy code

social-dashboard-app/
|
|   └── client/                                # Frontend part of the application
|       ├── public/                             # Static files
|       |   └── index.html                      # Entry point HTML file
|       ├── src/                               # React.js source files
|       |   ├── components/                   # Reusable UI components
|       |   ├── pages/                       # Page components
|       |   ├── app.js                        # Main React application component
|       |   ├── index.js                     # Main JavaScript entry point for React
|       |   └── redux/                        # Redux store and reducers (if using Redux)
|       └── package.json                    # Frontend dependencies and scripts
|
|   └── server/                                 # Backend part of the application
|       ├── config/                          # Configuration files for APIs and DB
|       ├── routes/                         # Express routes
|       ├── models/                          # Database models
|       ├── controllers/                  # Request managers
|       ├── middleware/                 # Express middleware
|       ├── app.js                           # Main server file
|       └── package.json                  # Backend dependencies and scripts
|
└── .env                                         # Environment variables for API keys, DB connections, etc.
└── .gitignore                                  # Specifies intentionally untracked files
└── README.md                                    # Project documentation
```

图8-3 ChatGPT建议的Web应用程序的通用文件结构

## 8.7 自动填充

当你深入代码区，为不同的时间单位逐一设定常量时，像Copilot这样的工具便有了用武之地。Copilot能够提供智能的自动填充功能，简化你的编程工作。

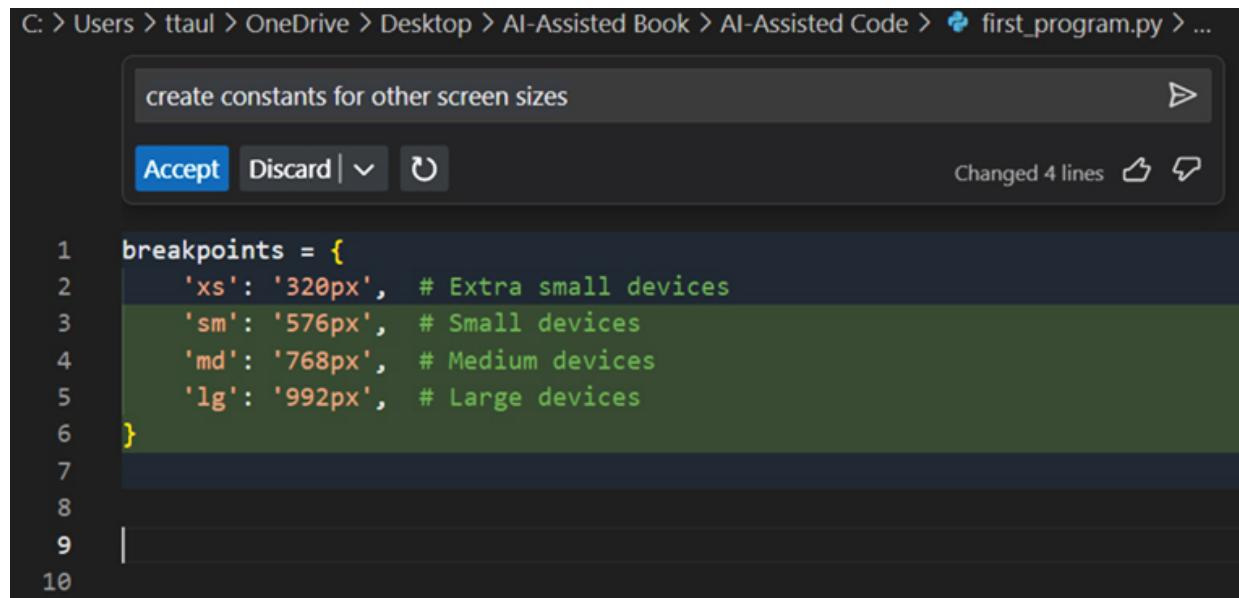
以创建响应式应用程序为例，你需要为CSS-in-JS风格（在JavaScript中编写CSS）的组件库设置断点常量。这通常意味着重复编写多行相似的代码。你可以这样开始。

```
breakpoints={  
  'xs': '320px', #极小屏幕设备
```

接下来在内联聊天中使用以下提示来请求Copilot的协助。

**提示：**为其他屏幕尺寸创建常量。

结果如图8-4所示。Copilot提供了具有类似样式的其他屏幕尺寸和常量。



The screenshot shows a code editor interface with the following details:

- Path: C: > Users > ttaul > OneDrive > Desktop > AI-Assisted Book > AI-Assisted Code > first\_program.py > ...
- Toolbar: "create constants for other screen sizes" button, "Accept" button, "Discard" button, and a refresh icon.
- Status Bar: "Changed 4 lines" and two small icons.
- Code Area:

```
breakpoints = {  
    'xs': '320px', # Extra small devices  
    'sm': '576px', # Small devices  
    'md': '768px', # Medium devices  
    'lg': '992px', # Large devices  
}
```

图8-4 Copilot为不同屏幕尺寸推荐了不同的常量

最后，Copilot可以通过查看项目中打开的文件来帮助进行自动填充。

Capilnean分享了自己的体会：

我发现，当代码的某些部分与数据结构相似时，Copilot会特别管用。如果你已经在另一个文件中定义了一个对象，而该文件又是你正在使用的代码库的一部分，那么，当你试图定义另一个类似的对象时，Copilot就能在你输入时可靠地预先填充代码。例如，Sendgrid电子邮件等常用API调用的对象在这里就很好用。

不过，一不小心，自动填充功能就会出现问题。例如，在导入时，它并不总是能正确地找到文件夹结构，尤其在你设计了自己的代码形状，或者你使用的框架相当新且还没有完善文档的情况下。在这种情况下，必须手动检查组件的来源，以及Copilot建议的目录是否正确。

## 8.8 重构

在软件开发中，重构就如同对代码进行一次大扫除。重构的核心目的不在于增添新功能或修补缺陷，而在于梳理、调整甚至美化现有代码。

通过重构，我们能够维持代码库的简洁与健康，减少混乱，提升直观性。这意味着，当你或其他开发者在未来回顾代码时，将更容易理解代码的逻辑，减少因代码复杂性带来的困扰和沮丧。而且整洁的代码往往更加高效，错误率更低，这使得重构成为一项对所有参与者都有利的活动。

重构是AI辅助编程工具展现其强大功能的重要领域之一。借助简单的提示，这些工具能够深入解析代码的复杂性，去除冗余，提炼出既实用又优雅的代码结构。在AI的辅助下，重构代码变得不再烦琐，而是成为一种保持技术领先的秘密武器。

接下来展示一些具体的重构示例。

### 8.8.1 忍者代码

将忍者代码比作动作电影中的特技场面颇为恰当。它们华丽而复杂，令人不禁惊叹：“这究竟是怎么做到的？”然而，等到最初的敬畏感消退后，理解它们背后的逻辑就会变得异常困难。想象一下，代码如此晦涩难懂、错综复杂，以致连编写它们的开发者在“忍者”阶段过后也难以厘清头绪。虽然完成“忍者”动作的那一刻确实令人震撼，但是，当你或其他开发者需要回头修改代码时，那种兴奋感就会迅速转变为沮丧。

实际上，忍者代码可能展示了一些卓越的编程技巧，但它们往往与良好的编程实践相悖。良好的编程实践应该是保持代码整洁、简洁且易于理解。

幸运的是，忍者代码虽然普遍存在，但AI辅助编程工具可以帮助我们对其进行重构。下面是一个示例。

```
console.log((function(n, a = 0, b = 1) { while (--n) [a, b] = [b, a + b];  
return a; })(10));
```

以上代码的目的可能需要花大量时间才能弄清楚。所以，为什么不让ChatGPT来完成这项工作呢？

**提示：**请逐步解释这段代码的工作原理。另外，你能否以一种更简单、更易于维护的方式重构这段代码？

ChatGPT确认这是一个计算第 $n$ 个斐波那契数的函数。它以6个步骤详细描述了整个过程，然后提供了一个简化的版本，如图8-5所示。

```
javascript
function calculateFibonacci(n) {
    let a = 0;
    let b = 1;
    for (let i = 1; i < n; i++) {
        let next = a + b;
        a = b;
        b = next;
    }
    return a;
}

console.log(calculateFibonacci(10));
```

图8-5 ChatGPT提供的简化版本

**ChatGPT**：在简化版本中，calculateFibonacci函数清晰地展示了斐波那契数列的迭代过程。虽然这个版本可能稍显冗长，但它更易于理解和维护，特别是对未来的其他开发者来说。

## 8.8.2 提炼函数

面对一个冗长的函数或方法，提炼函数是一种常用的重构技巧。这项技术涉及从现有代码中识别出一段执行特定任务的代码，如进行数据验证或特定计算，然后将这段代码抽取出来，形成一个新的函数或方法。

提炼函数的作用远不止于美化代码，它还增强了代码的可读性和可理解性，使得代码更易于维护。当你需要再次执行相同的任务时，便可以简单地调用这个新的函数或方法，而无须重复编写相同的代码。此外，如果出现问题，已经被分解成小块的代码相对更容易定位和修复缺陷。提炼函数不仅简化了开发过程，还有助于保持代码整洁和有序。

以下是提炼函数的一些提示范例。

**提示：**在函数式编程语言中使用提炼函数时，有哪些常见的陷阱应当避免？

**提示：**我附上了一些C#代码。请问哪些部分适合使用提炼函数进行重构？

**提示：**请比较原始函数和使用提炼函数后的重构版本，哪个效率更高？

### 8.8.3 分解条件表达式

分解条件表达式是一种将庞大、错综复杂的if-then-else语句拆解成易于理解的小段代码的技巧。

设想一个if语句，其中包含一个需要检查多个变量或调用其他函数的复杂条件。与其每次都去解读这个复杂的条件，不如将其提炼到一个具有描述性名称的方法中，从而清晰地表达条件检查的目的。例如，可以将

if (user.isActive() && user.hasSubscription() && !user.isBlocked())这样的条件重构为一个名为canUserAccessContent的方法。这样的重构不仅使主方法更加简洁，而且使代码的意图一目了然。

类似地，将if语句中的then和else代码块也分别提炼到不同的方法中，可以让程序的主流程更加清晰易读。读者不必深陷于一行行代码的具体逻辑，而应该从更高层次上把握程序的走向：如果条件成立，则执行一系列操作，否则，执行另一系列操作。程序逻辑的每个部分都被封装在具有清晰名称的方法中，这不仅便于单独测试，也便于修改。

以下是两个提示范例。

**提示：**如何分解Java代码中的条件表达式，从而提升其可读性？

**提示：**如何使用分解条件表达式技术分解复杂的if-then-else语句？

#### 8.8.4 重命名

虽然重命名函数、变量和类看似小事一桩，但是对提升代码的可读性和可维护性至关重要。随着代码的不断演进，原有的名称可能不再能够精确反映函数、变量和类的功能，这时对它们进行重命名显得尤为重要。例如，一个最初被命名为processData的方法，如果其功能逐渐变得更加单一，那么给它起一个更具描述性的名称（如filterInvalidEntries），将能够立即让人理解其实际作用。

以下是一些提示范例。

**提示：**在我的SQL脚本中，用于存储数据库中用户总数的变量应该该怎么命名？

**提示：**下面是我的JavaScript代码片段。请检查变量名和方法名，并提出改进建议。

**提示：**我对我的Java类中的命名不太确定是否足够直观。你能否建议更好的名称，使其更加清晰易懂？

在使用Copilot这样的AI工具时，需要格外小心，因为不当的重命名可能导致现有代码中使用旧名称的部分出现问题。

#### 8.8.5 无用代码

顾名思义，无用代码并不提供任何功能，它们是那些被遗忘的代码。由于功能的废弃或更新，程序的某些部分变得不再必要，最终成为无用代码。

清除项目中的无用代码不仅能让项目变得更加整洁、易于管理，还能减少项目新成员的困惑。当某些代码看起来没有明显作用时，清除它们可以避免无谓的猜测。

以下是一些有助于识别和处理无用代码的提示范例。

**提示：**你能帮我识别JavaScript代码片段中潜在的无用代码吗？

**提示：**这是某Python项目中的一段代码。请指出其中未被使用或看起来多余的代码。

**提示：**请检查以下SQL存储过程，确认是否有代码可以安全移除。

然而，使用AI辅助编程工具来删除无用代码可能伴随着风险。有时，看似被遗忘的代码实际上可能对处理罕见情况至关重要。此外，移除程序的某个部分可能会引发连锁反应，干扰依赖于这部分的其他代码，特别是在涉及复杂逻辑或配置的情况下。因此，务必谨慎行事。

另外，生成式AI可能未能完全理解代码之间的依赖关系。在AI辅助编程工具变得更加成熟之前，应避免使用它们来彻底清除无用代码。

在处理无用代码时，使用专门的工具可能是更好的选择。linter工具就是其中的代表，此类工具是“代码的清洁工”。例如，JavaScript开发者可以使用ESLint，Python开发者可以使用Pylint，而Ruby开发者可以选择RuboCop。这些工具能够高效地识别出恼人的语法错误、潜在的bug，以及那些悄悄存在却无所作为的代码。

此外，还有一类更全面的检查工具——静态代码分析工具，市场上的顶级供应商包括SonarQube、Code Climate和Coverity。这些工具就像侦探一样，不需要运行代码就能通过深入挖掘，发现复杂的、未来可能成为问题的模式，包括无用代码。

## 8.9 编写函数

函数是编程中的面包和黄油，无论使用哪种编程语言，它们在所有类型的计算机程序中都发挥着巨大作用。函数是大段的代码，也是代码复用的利器，它们让开发者能够轻松重用代码段，有助于保持程序的整洁和可读性。此外，函数能够将复杂任务分解为更小、更易于管理的部分。这使得开发者在处理复杂软件时，尤其是在修复错误、进行更新或理解代码逻辑时，能够更加得心应手。

编写函数的关键在于从设计之初就明确函数的功能。确保函数正常运作是基础，但同样重要的是，它们还要能够与周边代码无缝协作。要使函数清晰、易用且高效，应考虑以下要素：如何调用函数以发挥其功能，如何定义函数，如何处理输入输出数据，以及如何应对异常情况。

以下是一些有助于高效使用函数的指导原则。

### **单一职责原则**

单一职责原则有利于我们更简单地理解函数的作用，检查函数是否工作正常，并在函数工作不正常时对其进行修复。

### **命名清晰**

为函数选择一个能够准确描述其功能的名称，比如，当一个函数的功能是计算总价时，可以将其命名为calculateTotalPrice，以增强代码的可读性。

### **简短精悍**

理想的函数应当简短到不需要滚动屏幕即可一览无余。简短精悍的函数更易于管理和维护。

### **参数很关键**

通过参数接收输入，并通过返回值提供输出，确保函数的可预测性和独立性。

## 保持一致

遵循编程语言或项目中的代码规范和风格指南，以维护代码的一致性，便于他人阅读。

基于以上指导原则，我们可以为函数设计一些高效的提示，以请求ChatGPT提供帮助。以下是一些提示范例。

**提示：**编写一个名为calculate\_area的Python函数，它接收长度和宽度两个整型参数，并返回矩形的面积。请包含docstring来解释这个函数的目的，并确保该函数能通过引发TypeError来处理非整数输入。

**提示：**编写一个名为filterAndTransform的JavaScript函数。它的输入为对象数组，其中的每个对象包含姓名（字符串）和年龄（数字）属性。该函数应返回一个新的数组，其中包含18岁及以上年龄人士的大写姓名。请在代码中包含注释，对代码逻辑进行解释。

**提示：**编写一个名为efficientSort的C++函数，用于对整数数组进行升序排列。该函数应针对时间复杂性进行优化。请在该函数中提供注释，解释所选的排序算法及其时间复杂度。

**提示：**编写一个名为safeDivide的Java函数，它接收两个double类型的参数——分别代表分子和分母，返回它们的商。该函数应处理分母为零的情况，并返回自定义的错误提示信息。请在Javadoc注释中解释该函数及其错误处理机制。

## 8.10 面向对象编程

面向对象编程是一种以“对象”为概念来设计和构建计算机程序的方法。可以将面向对象编程想象为创造一系列独立的容器，其中每个容器都装有特定的工具和信息。这些容器在面向对象编程中被称为“类”，它们就像用来塑造不同对象的蓝图。类定义了对象的结构和

行为。基于类，我们可以创建具有独特属性但遵循相同基本架构的独立对象。

探索面向对象编程的世界，就像进入一个由抽象、继承、封装和多态等复杂概念构成的迷宫。这些概念初看可能像是用外星语言写的，令人费解。

此时，ChatGPT的“翻译”功能可以大显身手。它能够将这些复杂的概念分解成易于理解的小块。如果你对封装的真正含义感到困惑，只需要进行简单的提问，就能得到一个清晰易懂的解释，而不是通篇的技术术语。

以下是一些有助于你理解面向对象编程概念的提示范例。

**提示：**能否用<你的首选编程语言>创建一个简单的类来演示封装的概念？

**提示：**在编程中，封装有哪些实际应用案例？

**提示：**用日常生活中的类比来解释面向对象编程中的抽象。

**提示：**你能举例说明编程中的继承是如何工作的吗？

**提示：**在面向对象编程中，继承如何促进代码重用？

**提示：**多态如何增强程序的灵活性？

**提示：**多态在哪些场景下特别有用？请举例说明。

AI辅助编程工具还能帮助开发者设计类的初始结构。以下是一些设计类的提示范例。

**提示：**设计一个包含员工姓名（employeeName）、员工ID（employeeID）和部门等属性的Employee类。请实现一个方法来显示员工的详细信息，并包含一个构造函数来初始化这些属性。

**提示：**请用Java设计一个名为BankAccount的类，其中应包含余额和账户等私有属性。请为存款、取款和查询余额添加相应的方法，并确保它们能够安全地修改或访问这些属性。

**提示：**请用C#创建一个名为Vehicle的类，然后创建一个继承了Vehicle类的Truck类。确保其中包含车轮和燃料类型等属性，并展示不同访问修饰符的用法。

**提示：**在C++中编写一个FileHandler类，在构造函数中打开文件，并在析构函数中关闭文件。此外，请包含写入文件和从文件中读取的方法。

## 8.11 框架和库

在软件开发的世界中，如果没有框架和库，就好比在没有食谱或配料的情况下从头开始制作蛋糕。这样做不仅难度极高，而且耗时漫长。框架和库是让开发者工作更加轻松的法宝。无论是进行网络请求还是操作DOM元素，都可以直接利用现有的工具，而无须重复造轮子。

在探索软件框架和库的过程中，AI辅助编程工具无疑可以提供巨大的帮助。它们不仅可以帮助我们快速掌握基本功能和工作流程，还能指导我们在特定情况下选择最合适的框架或库。

然而，在接受这些工具的帮助时，我们应保持谨慎。Capilnean分享了自己的体会：

鉴于React以及其他框架和库更新频繁，以及我们使用的是特定版本的框架，我必须向我的ChatGPT实例提供最新的系统提示，以优化ChatGPT生成的结果。有时，ChatGPT提供的解决方案可能更偏向于学术性而非生产实践，因此我更倾向于与经验丰富的开发者合作解决更复杂的问题。对于React，我更多的是让ChatGPT检查语法错误和处理特定数据类型的方法，而不是让它构建完整的功能。

Node.js的情况也类似。在向ChatGPT请求某个功能的代码片段之前，我必须考虑其内部API和数据处理方式。如果我能够清晰地描述一个函数的输出，并将其作为正在构建的功能的输入，ChatGPT就很可能提供稳定的代码。如果输出结果不理想或存在错误，我也会采用相同的方法，向ChatGPT说明问题所在，并指导它更新代码，朝着正确的方向前进。

## 8.12 生成样本数据

数据是每个应用程序的核心命脉。数据赋予应用程序生命力，正如血液中的氧气和养分可以维持我们身体的功能一样，数据在应用程序中流动，提供其所需的信息和洞察力，使其发挥效用。

然而，创建样本数据可能是一项棘手的任务。想象一下，当你渴望测试新的应用程序时，你需要大量数据来模拟现实世界的运行情况。你开始手动输入一行行的数据——姓名、日期、数字，无一例外。然后，你一遍又一遍地输入测试数据。

但你必须谨慎行事，以确保测试数据既足够真实以反映有效性，又不至于过于烦琐，避免每个条目都复杂得像一本小说。如果你需要一个庞大的数据集，怎么办？这基本上等同于参与一场复制、粘贴、调整并重复检查的马拉松。

AI辅助编程工具可以巧妙地解决这个问题。首先，你需要选择一种数据库并创建数据库模式和表。然后向这些工具描述表之间的关系。一旦配置和设置完成，AI辅助编程工具就可以提供帮助了。

以下是一些获取数据库评估的提示范例。

**提示：**哪种类型的数据库最适合处理<特定数据类型或功能，如用户交互、产品库存等>？

**提示：**对于预期<高/低>流量且具有<数据类型，如图像、文本、实时数据>的应用程序，我应该考虑哪种数据库？

**提示：**我的预算有限。你能否为小型<应用程序类型，如本地快递服务应用程序>推荐一种经济的数据库解决方案？

**提示：**我对数据库管理不太熟悉。对于初学者，哪些数据库既用户友好又易于维护？

以下是一些获取数据库模式的提示范例。

**提示：**请为<应用程序类型（如在线商店、博客）>设计一个基本的数据库模式。我需要知道应该创建哪些表以及它们之间的主要关系。

**提示：**在关系型数据库中管理<特定类型数据（如客户订单、库存）>的有效表格结构是什么？应该包含哪些字段和数据类型？

**提示：**在<描述应用程序功能，如活动管理、课程注册>应用程序的关系型数据库中，我应该如何定义表之间的关系？具体来说，在理解外键和连接表方面我需要帮助。

**提示：**我正在为<描述项目>设计一个关系型数据库模式。请指导我如何有效设置主键和外键，以确保数据完整性。

**提示：**我在<项目类型，如社交媒体应用程序>中使用了NoSQL数据库。我应该如何设计文档结构来存储<特定数据类型，如用户配置文件、帖子、评论>？

**提示：**在设计数据库模式时，应该考虑哪些索引策略来优化查询性能，尤其是<查询或操作类型，如全文搜索、频繁更新>？

**提示：**我需要将现有数据库迁移到新的数据库模式。在不破坏数据完整性的情况下，重新设计数据库结构的主要考虑因素和步骤是什么？

以下是一些设置初始数据库环境的提示范例。

**提示：**请提供在<特定操作系统，如Windows、Linux>中安装<特定数据库服务器，如MySQL、PostgreSQL>的分步说明。

**提示：**安装<数据库名称>后，为获得最佳性能，应该设置哪些基本配置？

**提示：**保障<特定数据库>服务器安全的最佳做法是什么？我对用户身份验证和保护敏感数据特别感兴趣。

**提示：**如何优化<特定数据库>的性能，使其适用于处理<描述数据性质和预期负载，如大型数据集、高交易量>的应用程序？

下面使用AI创建样本数据。以下是一些提示范例。

**提示：**创建包含100个带有ID和电子邮件地址的演示数据，并将它们保存到CSV文件中。

**提示：**创建包含50款产品的演示数据，每款产品应包含产品ID、名称、价格和类别。

**提示：**创建包含150条订单记录的演示数据集，每条记录中应包含订单ID、客户ID、订单日期和总金额。

**提示：**生成100名员工的样本数据，每名员工应包含员工ID、全名、部门和电子邮件地址。

**提示：**生成80个客户反馈条目的样本数据，每个客户反馈条目应包含反馈ID、客户ID和评论。

有了样本数据，就可以创建SQL语句了。你可以使用下面的提示来处理客户反馈条目。

**提示：**生成SQL插入语句，将数据填充到反馈表中。

如果你是一名开发者，就会知道在XML、SQL、JSON、CSV和TOML等不同格式之间转换数据是很常见的事情。老实说，这是一个乏味的过程，有时还容易出错。当涉及数据转换的琐碎任务时，ChatGPT可以说是开发者的救星，这也是ChatGPT的用武之地。

以下是一些针对数据转换的提示范例。

**提示：**这是CSV文件中的一行数据——“无名氏, 35岁, 纽约”。你能把它转换成XML格式吗？

**提示：**我有一个这样的JSON数组——[{'姓名' : 'Alice', '职位' : '工程师'}, {'姓名' : 'Bob', '职位' : '设计师'}]。我如何用SQL表格来表示这个数组？

**提示：**我需要将TOML配置“title="My Project" owner ="Sara""转换为等效的YAML格式，如何转换？

## 8.13 前端开发

前端开发是打造美观网站与流畅体验的艺术。它的核心在于使用HTML构建网页的基础架构，使用CSS美化网页的外观与布局，以及使用JavaScript实现动态交互。现代前端开发者通常会借助React、Vue等框架，这些框架提供了可复用的组件和实时数据更新等特性，使得构建充满活力和吸引力的网站与Web应用程序成为可能。

即便对于资深开发者，前端开发依旧充满挑战。其中首要的难题是浏览器兼容性和用户交互的不确定性。开发者需要适应多样的浏览器、设备、屏幕尺寸和用户偏好，这就像设计一件能够适合每个人的T恤一样复杂。另外，前端技术日新月异，框架更新迭代快速。

前端开发的另一大特点是，它不仅关乎代码编写，更与卓越的用户体验（User eXperience，UX）和用户界面（User Interface，UI）

设计联系紧密。UX和UI设计关注人们如何与技术互动，如何让设计在视觉上吸引眼球，以及如何在Web应用程序或网站中打造流畅的逻辑流程。这要求前端开发者不仅要有算法和数据结构的才华，还要兼具艺术家的审美和心理学家的洞察力。

不过，AI辅助编程工具的出现，为前端开发者带来了曙光。在接下来的内容中，我们将深入探讨CSS、图形设计以及AI工具如何协助前端开发，提升前端开发者的效率与创造力。

### 8.13.1 CSS

为网站编写CSS无疑是一项充满挑战的任务，特别是那些结构庞大且复杂的网站。确保网站在各种浏览器和设备上都能呈现出一致且吸引人的视觉效果，是一项艰巨的工作。前端开发者需要熟悉各种浏览器特有的细微差别，同时还要小心翼翼地避免在样式层叠中引发冲突，因为这可能会让网站的外观变得混乱不堪。此外，传统的CSS不支持变量和函数，这导致许多样式不得不在代码中反复编写，SASS和LESS等CSS预处理器工具可以有效地解决这一问题。

以下是一些常见CSS任务的提示。

**提示：**能否提供一个CSS代码片段，以便在移动设备上将导航栏转换为汉堡包菜单？

**提示：**如果在使用CSS的弹性盒布局（Flexbox Layout）时遇到对齐问题，能否指出可能导致问题的原因？

**提示：**我需要为网站上的按钮添加悬停效果，能否展示一个CSS示例，使得按钮在悬停时改变颜色？

**提示：**我想在网站的首页上添加淡入动画效果，能否提供相应的CSS代码片段？

### 8.13.2 创建图形

为网站或Web应用程序设计专业级图形通常需要深厚的设计功底以及熟练掌握Photoshop等图像处理工具。然而，当下有众多功能强大的文生图系统，它们能够辅助你创造出卓越的视觉作品。此类系统有Canva、Stable Diffusion、DALL·E 3（ChatGPT内置）、FireFly、Midjourney等。

通过这些系统，你可以创建多种类型的设计资源，包括主图、按钮和徽标等。

以ChatGPT创作徽标为例，提示范例如下。

**提示：**我正在为家族面包店Sweet Whisk设计网站。我需要一个充满温馨感的徽标。我们主营蛋糕和饼干，我希望将这些元素巧妙地融入设计中。我偏好柔和的色调，特别是浅粉色和薄荷绿。徽标应该简洁且具有现代风格，并透露出一丝俏皮。

ChatGPT创作的徽标如图8-6所示。

你可以继续向ChatGPT提出更详细的要求，以优化和调整设计输出。虽然其他AI工具（如Firefly）可能在设计功能上更为丰富，但ChatGPT依然能够轻松制作出吸引眼球的图像。



图8-6 ChatGPT根据提示创作的徽标

### 8.13.3 AI工具

当前市场上存在一些优秀的AI工具，它们能够根据简单的提示或你期望的前端外观图像，快速生成网站。这些工具不仅能处理所有底层代码工作，还能无缝转换至React、Angular和Vue等现代前端框架。更令人印象深刻的是，许多AI工具还支持从Figma等平台导入线框图，以进一步简化开发流程。

一旦代码由AI工具生成，就可以深入代码层进行个性化调整，确保网站达到你心目中的完美状态。AI工具为加速网站构建提供了一条捷径。

可用的AI工具还有TeleportHQ、Anima、Locofy和Vercel v0。

我们重点关注Vercel v0，它的使用方式非常直观。实际上，它的界面设计与ChatGPT相似，易于上手，如图8-7所示。

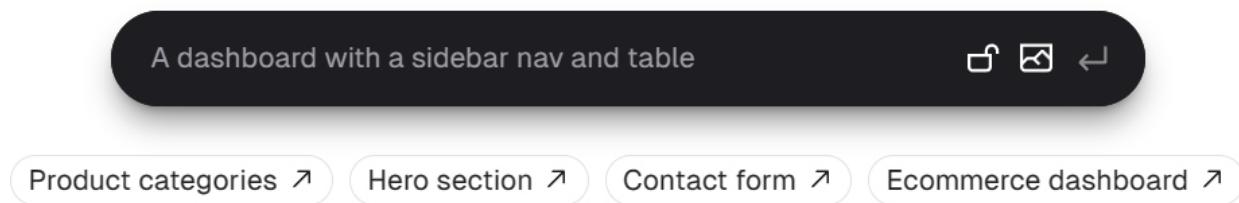


图8-7 Vercel v0的界面

输入期望的页面类型描述，或上传设计好的图像，系统便能根据指示创建页面。以下示例展示了如何使用提示来引导页面设计。

**提示：**我需要一个电子商务仪表板，它具备侧边栏导航功能，还具备表格以展示最近的订单。

Vercel v0将带你进入设计工作室，如图8-8所示。

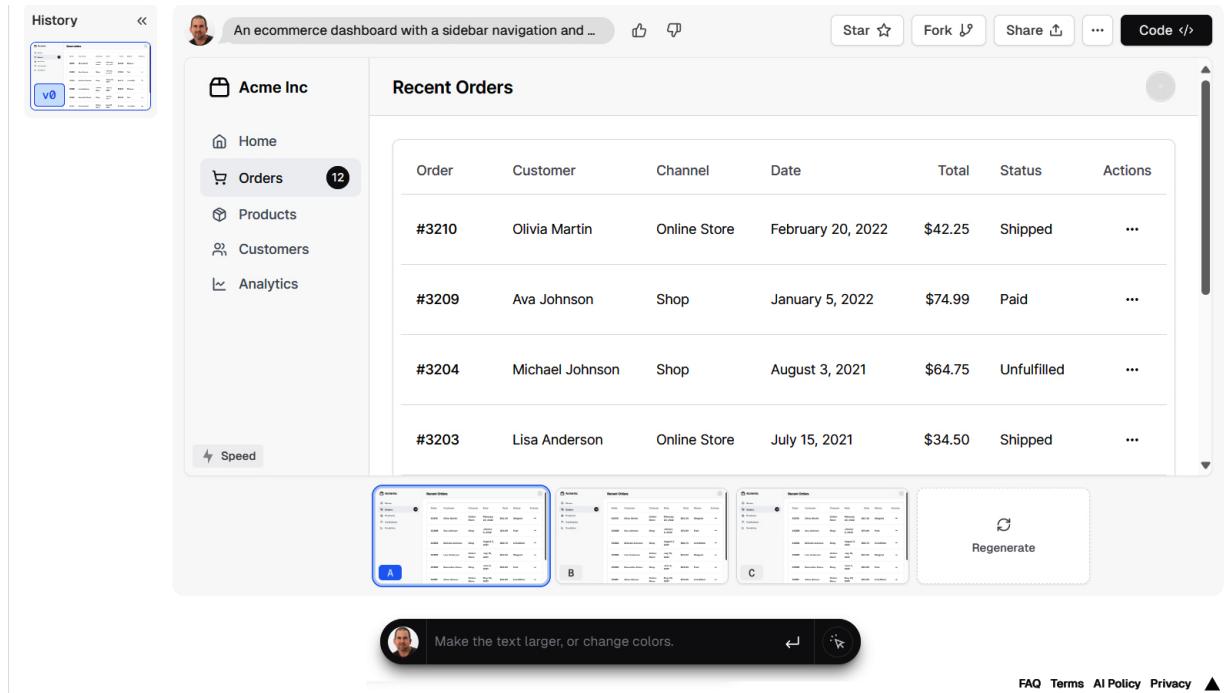


图8-8 输入提示后，Vercel v0便带你进入设计工作室

Vercel v0能够根据需求生成3种不同的仪表板，并且可以利用系统提供的提示功能进行细节调整，比如更改文字的大小、颜色等。

对设计感到满意后，只需要单击界面右上角的Code（代码）按钮，即可获得基于shadcn/ui和Tailwind CSS的简洁React代码清单。

除了Vercel v0以外，还有一些开源AI系统能够将设计图转换为前端代码，例如Screenshot to Code。该系统由Pico的首席执行官兼创始人Abi Raja开发。Raja此前的初创公司在2013年被雅虎收购。

Raja投入了6个月的时间来创建Screenshot to Code的初始代码库。他指出：“前端开发者通常需要将设计图和模型转换为代码，这一过程往往涉及大量重复性工作。但我的软件能够自动化这一转换过程，完成度高达90%。”

目前，Screenshot to Code支持导出到React、Bootstrap和HTML/Tailwind CSS，它在GitHub上获得了超过31000颗星的好

评。

ChatGPT同样具备将图像转换为代码的能力。比如，如果你想创建一个外观类似于iPhone自带的计算器的应用程序，则可以向ChatGPT提供图8-9所示的图像。

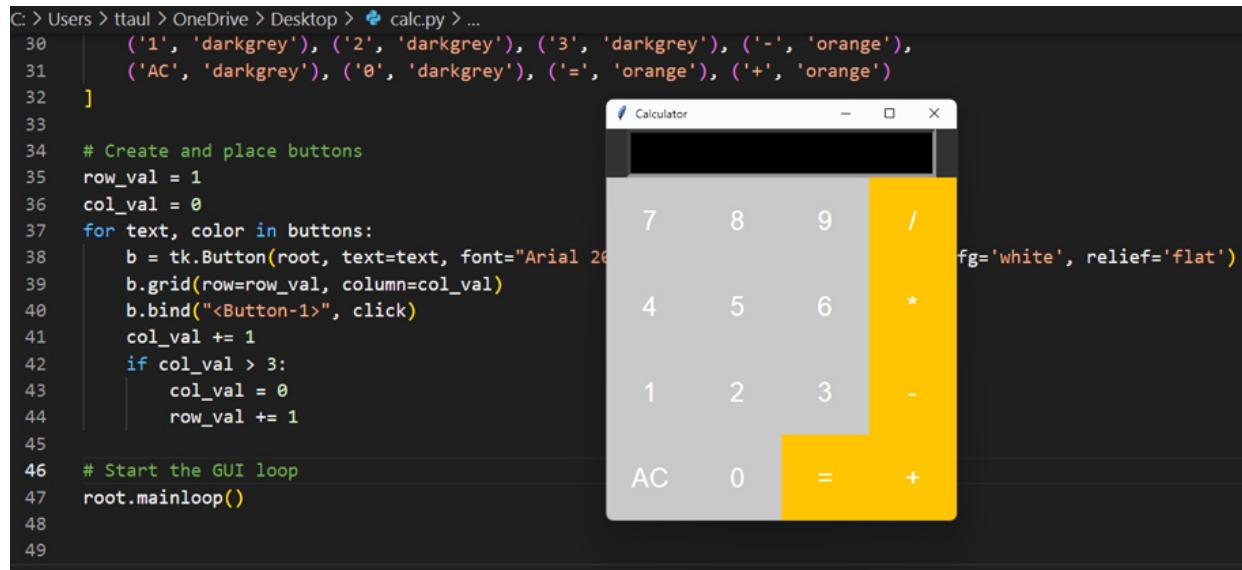


图8-9 将iPhone计算器的图像输入ChatGPT以生成代码

首先请求ChatGPT识别所提供的图像，ChatGPT具备正确识别图像的能力。然后使用以下提示生成代码。

**提示：**根据这幅图像，建议合适的代码实现。

图8-10展示了一个计算器应用程序的界面。ChatGPT根据你提供的图像创建了Python代码，虽然在样式上可能与预期有所偏差，但已经非常接近。如果你希望生成的计算器应用程序在外观上更贴近iPhone计算器的原生设计，则可以在提示中提供更具体的指导，例如指示ChatGPT使用React框架来构建用户界面。



The image shows a split-screen view. On the left is a code editor window displaying Python code for a calculator application. On the right is a screenshot of a Mac OS X-style calculator application window titled "Calculator".

**Code Editor Content:**

```
C: > Users > ttaul > OneDrive > Desktop > calc.py > ...
30     ('1', 'darkgrey'), ('2', 'darkgrey'), ('3', 'darkgrey'), ('-', 'orange'),
31     ('AC', 'darkgrey'), ('0', 'darkgrey'), ('=', 'orange'), ('+', 'orange')
32 ]
33
34 # Create and place buttons
35 row_val = 1
36 col_val = 0
37 for text, color in buttons:
38     b = tk.Button(root, text=text, font="Arial 20")
39     b.grid(row=row_val, column=col_val)
40     b.bind("<Button-1>", click)
41     col_val += 1
42     if col_val > 3:
43         col_val = 0
44         row_val += 1
45
46 # Start the GUI loop
47 root.mainloop()
48
49
```

**Calculator GUI Layout:**

- Buttons are arranged in a grid.
- Row 1: 1, 2, 3, -
- Row 2: AC, 0, =, +
- Row 3: 7, 8, 9, /
- Row 4: 4, 5, 6, \*
- Row 5: 1, 2, 3, -
- Row 6: (empty)

The buttons for operators (-, \*, =, +) and the decimal point (.) are highlighted in yellow. The AC button is also highlighted in yellow.

图8-10 ChatGPT根据提供的图像生成代码

## 8.14 API

掌握新的API往往是一项艰巨的任务。你需要深入研读可能内容繁杂、技术性强的API文档。API文档的质量参差不齐，有的简明易懂，有的却难以理解。此外，获取登录凭证和访问权限也是一项挑战，比如处理API密钥或OAuth令牌。理解API返回的数据结构和响应格式可能会令人感到困惑，更不用说理解API的错误处理，以及还需要持续关注API的更新和变化了。

幸运的是，ChatGPT可以让这些挑战变得更加容易应对。

下面以一个简单的示例来说明：假设你想通过API获取天气信息，你可以输入以下提示来向ChatGPT询问有关天气API的建议。

**提示**：都有哪些优秀的天气API？

ChatGPT可能会推荐你使用OpenWeather API。随后，你可以继续提问。

**提示**：如何使用OpenWeather API？

ChatGPT将描述主要的入门步骤。然后，你可以要求ChatGPT展示如何构建请求。

**提示**：如何使用OpenWeather API构建请求？请演示如何用Python来实现。

ChatGPT将详细介绍以下步骤。

**ChatGPT**：

- 导入requests库。
- 定义API密钥。
- 使用当前天气数据端点创建API请求URL。
- 执行请求并处理响应。
- 检查请求是否成功。

## 8.15 小结

本章深入剖析了模块化编程的精髓、函数的构建艺术及面向对象编程的核心概念。我们不仅探讨了这些编程范式如何帮助我们编写更清晰、更易于维护的代码，还讨论了如何借助AI的力量快速掌握新的编程语言和框架。

此外，本章还涵盖了如何利用AI提升数据处理效率，以及如何使用AI工具来设计引人注目的前端界面。

然而，我们必须再次强调，尽管AI辅助编程工具的功能强大，但它们远非完美。在使用这些工具时，保持警惕并仔细检查它们提供的代码和建议。

## 第9章

# 调试、测试和部署

本章探讨软件开发中那些常被忽视但至关重要的方面——调试、测试和部署。本章将提供全面的指南，从识别不同的软件缺陷到撰写清晰易懂的文档，再到确保代码按预期执行。此外，本章还将详细讨论如何高效地合并代码更改、保障软件的安全稳定运行、倾听用户的反馈，甚至如何在发布软件时制造影响力。

## 9.1 调试

在编写代码的过程中，据调查，开发者会花费35% ~ 50%的时间进行代码的调试。调试不仅耗费了开发者宝贵的时间，还占据了软件项目预算的很大一部分。

代码错误主要分为两大类。第一类是语法错误，这类错误通常发生在代码没有遵循特定编程语言的规则时，例如for循环的末尾遗漏冒号。现代IDE（如VS Code）能够检测并修复许多语法错误。

第二类是逻辑错误，这类错误发生在代码的逻辑出现偏差时，它们往往比语法错误更难发现和修复。假设你创建了一个程序，旨在根据年龄从列表中过滤掉成年人用户。问题是，这个程序并没有排除所有18岁及以上的用户，而是错误地排除了18岁以下的用户，最终得到一个全是成年人用户的列表，这是典型的逻辑错误，代码正在做与你的初衷完全相反的事情。要弄清楚代码为何标记18岁以下的人群，而不是18岁及以上的人群，并不是一件容易的事情。解决方案包括从调整年龄条件判断到彻底重构逻辑，这都有可能。

通常传统的调试方法足以解决代码错误，AI辅助编程工具在调试过程中并不是首选。VS Code等IDE提供了强大的调试功能，可以帮助开

发者设置断点、检查变量的值，甚至逐行执行代码以观察程序的运行情况。在处理大型程序时，这些功能尤其宝贵。

然而，有些代码错误可能会出乎意料地复杂。现代软件就像复杂的拼图，由许多层和部分组成，它们必须协同工作。有时，代码错误可能是软件的不同部分以我们意想不到的方式交互造成的，找出问题根源是一个真正的挑战。此外，文档的缺乏也会增加调试的难度。如果程序没有附带清晰的说明或解释，那么想要理解程序的工作原理以及查找和修复错误将变得更加困难。程序通常依赖外部库或服务，如果这些外部依赖出现问题，也会给程序带来棘手的错误。

当遇到上述问题时，可以求助AI。AI可以帮助你破译难以理解或冗长的错误信息，比如堆栈跟踪信息，即程序崩溃时执行操作的快照。同时，AI还可以提供框架和库之间交互的详细信息。

如果遇到难以理解的错误信息，可以将错误信息复制并粘贴到提示中，并附加如下内容。

**提示：**这段错误信息是什么意思？{错误信息}

如果代码中存在逻辑错误，导致程序没有实现预期的功能，则可以将代码复制并粘贴到提示中，并附加如下内容。

**提示：**程序允许用户上传照片并在图库中显示。但是，当执行该程序时，照片并没有出现。这段程序有什么问题？{代码}

如果ChatGPT发现了问题，它通常会提出解决方案。如果它没有提出解决方案，你可以在提示中加入指令，请求它提供解决方案。

## 9.2 文档

文档是保持项目和团队凝聚在一起的黏合剂。然而，文档常常被排在次要的位置。没有文档的代码在工作时就像在没有地图的迷宫中徘

徊，尤其对于新手或任何试图理解项目的人，这是一次痛苦的经历。

良好的文档能够帮助我们避免不必要的猜测和误解，确保每个人都能站在同一条起跑线上。根据Stack Overflow的一项调查，68%的开发者每周都会遇到由文档引起的知识障碍。此外，GitHub在其2021年的报告中指出，通过文档共享信息可以使团队的工作效率提升55%。

文档不仅能让开发者的工作更轻松，而且可以使开发流程顺利进行。文档就像一张地图，指引你前进的方向和需要注意的事项。没有了文档，开发者就容易盲目行事，这可能导致代价高昂的错误。

ChatGPT支持创建如下类型的文档。

- 用户手册。

- README文件。

- API文档。

- 常见问题解答。

- 故障诊断指南。

在创建有效的文档时，考虑以下因素至关重要。

## 了解文档的目标受众

思考一下文档的目标受众。他们是寻找快速入门指南的新手，还是需要API文档和代码片段等细节的资深技术人员？了解读者的需求是让文档的内容达到要求的关键。

## 保持简洁

避免使用难以理解的行话或技术术语。请保持言简意赅。如果普通人看不明白，文档可能就需要写得更清楚一些。

## 按计划行事

让文档看起来有条理。使用统一的风格、标题、字体等，一致性是关键，就像一首歌需要有好的节奏才能保持流畅。

## 展示而不只是讲述

实际的操作演示能加速人们对内容的理解。因此，请列举现实生活中的例子或场景。如果文档是面向开发者的，代码示例就更加宝贵了。对于其他人，截图或分步演示可能更有帮助。

## 一图胜千言

有时，使用图表、截图甚至视频来表达观点胜过千言万语，尤其是在处理复杂的问题时。

## 原因与方法同样重要

不仅要列出步骤，还要告诉读者为什么这样做。

综合考虑上述所有因素后，可以使用类似下面的提示来请求AI工具编写文档。

**提示：**请你就如何向非技术人员解释Git版本控制的概念提出建议。尽量简单明了，避免使用专业术语。另外，请使用直观的图或表来解释版本控制的重要性。



ChatGPT拥有强大的语言翻译能力，你可以将其运用于文档或任何其他内容。

微软为企业版文档开发了GitHub Copilot for Docs，它不是一款普通的文档挖掘工具。例如，它的搜索结果和回复基于用户的编程背景和经验。它还会根据GitHub代码仓库的最新情况进行更新。它甚至还能添加私人文档。本质上，它是一个高度复杂的知识库，可以极大地促进编程工作。

### 9.3 代码审查

代码审查可以视为代码在被提交到代码库并创建拉取请求之前的一次试运行。这一过程旨在确保一切按预期工作，功能得到实现，并且在执行过程中不会出现故障。

代码审查的目的不只是寻找代码中的不规范之处和小错误；它还是一个团队成员聚集在一起交流意见、相互学习、共同提高的过程。在代码审查中，你可以发现不同的问题解决方法，这有助于你更好地理解整个项目。

同时，代码审查有助于执行组织的编码风格和标准，它在安全检查方面也扮演着重要角色。我们需要认识到，自动化工具可能无法捕捉到所有的安全隐患，有些隐蔽的安全风险需要人工审查才能发现。

ChatGPT也可以成为代码审查过程的一个重要部分。例如，你可以使用以下提示来请求执行代码审查。

**提示：**为下面的代码写一份代码审查报告。重点关注代码的可维护性、潜在的安全问题和性能缺陷。`{代码}`

即使我们故意给ChatGPT一个质量很差的函数，它的审查和评论也做得很好。它提出了许多改进建议，比如建议不要在函数中硬编码数据库连接。它还指出了在SQL查询中直接使用用户ID的问题：因为缺乏用户输入验证，所以存在SQL注入的风险。此外，它还发现了SQL语句结构可能导致的性能问题。

### 9.3.1 单元测试

单元测试可以看作对代码的一小部分（比如几个函数或方法）进行的小型评估。开发者通常会利用一些高效的工具来进行单元测试，例如Java的JUnit、.NET的NUnit或Python的pytest。这些工具不仅帮助开发者编写和执行测试，而且提供测试结果反馈。它们通常还能与开发者正在使用的其他软件工具集成。

单元测试极大地减少了令人烦恼的错误，使后期的调整和问题修复变得更加容易，从而提高软件质量。每个单元测试都专注于一个具体的点，如果出现问题，开发者可以清楚地知道从何处着手。单元测试通常是自动化的，这意味着它们可以快速且频繁地执行，这对于保持软件的顺畅运行和最新状态至关重要。

由于单元测试专注于代码的一小部分，因此它们不会过于复杂且易于编写。此外，单元测试就像软件的使用指南，其他开发者可以通过查看单元测试来理解软件某些部分的工作流程。如果代码发生了变化，单元测试可以确保之前正常工作的部分没有被错误地修改。

假设你创建了一个计算小费的程序，代码如下。

```
def tip_calculator(bill_amount, tip_percentage):
    tip_amount = bill_amount * (tip_percentage / 100)
    total_amount = bill_amount + tip_amount
    return total_amount

bill_amount = float(input("Enter the bill amount: "))

tip_percentage = float(input("Enter the tip percentage (e.g., 20 for 20%): "))
```

```
total_amount = tip_calculator(bill_amount, tip_percentage)

print(f"Total amount including tip: {total_amount:.2f}")
```

这个程序包含一个函数，用于计算账单的总金额（包括小费），参数有两个——bill\_amount（账单金额）和tip\_percentage（小费百分比）。该函数将返回总金额total\_amount。

对于上述代码，单元测试可以检查各种情况。单元测试可以验证该函数能否正确计算一系列输入的总金额（包括小费）。这一点至关重要，因为即使是很小的错误也可能导致较大的差异。单元测试可以覆盖典型的用例，比如标准账单金额和小费百分比，以及边缘情况，如账单金额为零或负值，或小费百分比异常高。单元测试还有助于确保该函数能够妥善处理非数值或空值等异常输入。

以下是一个提示范例。

**提示：**请为该程序提供单元测试建议，以正确计算总金额，并检查典型情况和边缘情况，如账单金额为零或负值，以及小费百分比异常高。还要检查无效输入。对于单元测试，可以使用控制台日志。

图9-1展示了ChatGPT为单元测试生成的代码。

python

 Copy code

```
def run_tests():
    # Typical case
    assert tip_calculator(50, 20) == 60, "Typical case failed"

    # Zero bill amount
    assert tip_calculator(0, 15) == 0, "Zero bill amount case failed"

    # Negative bill amount
    # Depending on your function's intended behavior, update the expect
    assert tip_calculator(-50, 20) == 0, "Negative bill amount case fai

    # High tip percentage
    assert tip_calculator(50, 100) == 100, "High tip percentage case fa

    # Invalid input
    # This requires exception handling in the tip_calculator function
    try:
        tip_calculator("fifty", 20)
        print("Invalid input case failed")
    except ValueError:
        print("Invalid input case passed")

    # Boundary cases
    assert tip_calculator(0.01, 0.01) == 0.01, "Boundary case failed"
    assert tip_calculator(100000, 50) == 150000, "Large amount boundary

    print("All tests passed!")

    # Uncomment below to run the tests
    # run_tests()
```

图9-1 ChatGPT为单元测试生成的代码

如果想采用更有条理、更全面的方法，比如使用测试框架，则可以进一步请求ChatGPT提供帮助。

**提示：**使用测试框架创建单元测试。

ChatGPT建议使用unittest。它展示了如何设置和开发单元测试，并演示了如何执行单元测试。

或者，如果你已经有一个包含单元测试的文件，则可以评估该文件中的单元测试。以下是一个提示范例。

**提示：**下面是一个程序的单元测试文件，`<解释其作用或指向的代码>`。还有其他测试吗？这个单元测试文件还缺少什么？

虽然ChatGPT或AI辅助编程工具可以帮助开发者创建单元测试，但它们并非完美无缺。对于更高级的使用案例或更庞大的代码库，结果可能会有偏差。

David Lee，一位公司创始人、高级工程师兼AWS认证的解决方案架构师，给出了以下建议：

但是，在处理涉及真实数据库交互和Docker的测试时，情况就会发生很大变化。这将成为ChatGPT在某种程度上无法理解的另一个复杂层次，你可能需要先手动编写几个测试，尤其在数据库连接部分，这有助于让ChatGPT学会如何编写其他测试。

### 9.3.2 拉取请求

拉取请求（PR）在编程世界中就好比一张金色入场券，尤其当你与其他开发者合作时。你会将代码打包，推送到像GitHub或GitLab这样的平台上，随后提交一个PR。这不仅将发出一个“请添加我的代码”的请求，还意味着鼓励你的队友审查你的工作，给予认可或提出建议，帮助你的代码更上一层楼。PR确保了代码在被合并到项目的其他部分

时，能够以最佳状态呈现。此外，PR还是追踪谁在何时做了何种更改的有效工具，对于大型项目，这一点尤其宝贵。

撰写一份清晰可靠的PR描述至关重要。你需要阐明代码发生了哪些变化、为何变化以及如何变化。首先概述要解决的问题，然后详细说明提交的代码是如何解决这些问题的。别忘了补充一些细节，例如哪些文件被修改或新增了哪些文件，以及进行了哪些测试。如果对某些具体问题不确定或需要他人的意见，也可以在PR描述中提出。

有效的PR描述是代码审查者的福音。它能够加快审查流程，确保所有相关人员都能迅速理解PR。此外，一份详尽的PR描述也是未来的重要信息资源。

ChatGPT可以成为我们撰写PR描述的得力助手。起草初稿时，只须告诉ChatGPT你所做的工作，它就能帮你梳理思路，抓住重点。如果你已经有了草稿，ChatGPT将检查其清晰度和语法，并提出改进建议。ChatGPT还能帮你找到撰写PR描述的最佳方法，比如先从摘要开始，再逐步深入细节。如果需要简化技术细节，ChatGPT则能让团队中的每个人都更容易理解PR。

以下是一些提示范例。

**提示：**我在应用程序中添加了一个新的搜索功能，旨在根据用户输入的内容过滤搜索结果。你能帮忙编写PR描述吗？

**提示：**我解决了当用户在文本字段中输入特殊字符时应用程序崩溃的问题。我应该如何在PR中对此进行描述？

**提示：**我重构了身份验证模块以提升代码的性能和可读性。在PR描述中应该包含哪些内容？

**提示：**我更新了用户界面，使导航更直观，并添加了新图标。你能帮我起草PR描述吗？

**提示：**在PR描述中表达这些内容的好方法是什么？

**提示：**我为付款处理模块添加了新的单元测试。你能协助我撰写一份突出变更的PR描述吗？

**提示：**我解决了由于主分支最近发生更改而引起的合并冲突。对此我应该在PR描述中如何提及？

微软在Copilot中添加了一个支持创建PR描述的功能，名为Generated Commit Message。要使用此功能，你需要确保你的Copilot与GitHub上的仓库建立了连接，然后只需要单击界面左侧面板中的Sparkle按钮即可，如图9-2所示。

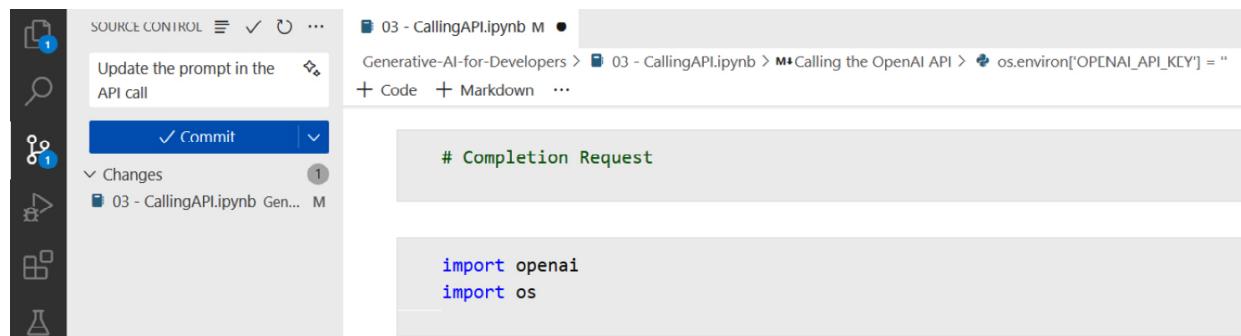


图9-2 Copilot可以为加载到项目中的存储库创建PR描述

随后，Copilot就能写出有用的PR描述。

最后，许多初创公司正在基于LLM技术创建自己的拉取请求系统。其中What The Diff公司的联合创始人兼首席执行官Sebastian Schlein表示：

What The Diff主要有两个功能：其一，以通俗易懂的语言总结拉取请求，让代码审核人员更容易理解PR中的变更；其二，编写完全非技术性的摘要，并发送给其他利益相关者，例如没有GitHub访问权限的产品经理，这样可以让他们轻松了解拉取请求中的变更是否符合要求。

## 9.4 部署

你一直坚持不懈地开发软件，终于迎来了软件正式发布的辉煌时刻。这是一个激动人心的时刻。你将目睹真正的用户如何使用你的软件，他们会告诉你哪些功能令人称赞，哪些地方需要改进，甚至还会提出许多宝贵的意见。对于开发者，用户的反馈无疑是一笔宝贵的财富。

如果你从事软件销售业务，软件的发布无疑是你期待已久的时刻，因为这意味着收益即将源源不断地涌入。

在部署软件的过程中，我们屏息凝神，期待着最佳结果的到来。其中总伴随着一种忐忑不安，担心某些环节可能不尽如人意。

在测试环境中表现完美的软件，一旦进入现实世界，面对不同的硬件配置、不稳定的网络环境或有问题的设置，就有可能变得不稳定。

安全性也是至关重要的。一旦软件上线，你就必须保持警觉，确保将恶意攻击者拒之门外，并严格遵守隐私法规。

此外，软件必须具备足够的韧性，以应对大量用户的访问。无论用户数量如何或业务规模多大，软件都必须保持快速和稳定。

整个CI/CD流程中的每个环节都是为了确保软件部署过程的流畅和自动化，从而减少错误。虽然听起来很理想，但实际上让软件运行得既快速又稳定，确实需要付出一定的努力。

在软件部署和发布方面，ChatGPT也能提供帮助，以下是一些提示范例。

**提示：**你能指导我创建一份供我的团队遵循的部署清单吗？

**提示：**关于使用Docker进行部署，有哪些优秀的入门学习资源？

**提示：**你能否提供Web应用程序中零宕机部署的最佳实践？

**提示**：我在部署软件的过程中遇到“服务器超时”错误。造成这种情况的常见原因和解决方案有哪些？

**提示**：你能帮我编写一个Bash脚本来自动部署Python Web应用程序吗？

**提示**：在生产环境中部署应用程序之前，我应该检查哪些基本配置？

**提示**：如何规划云环境下部署失败的回滚策略？

**提示**：在部署金融应用程序的过程中，应该考虑采用哪些安全措施？

**提示**：如何优化已部署的Node.js应用程序的性能？

虽然ChatGPT并不完美，但它在复杂的DevOps领域确实能给我们提供帮助。以下是Private Market Labs联合创始人兼首席产品官Titus Capilnean的一些见解：

云日志的处理确实有些复杂，尤其是对非开发工程师的我来说。但鉴于我们定期在AWS和Google Cloud上运行进程，我们必须学会处理它们。

有一次，我必须根据大规模函数部署的输出为SQS+Lambda进程设置追踪。函数原本会在日志中打印状态，我计划使用这些打印的状态生成该进程的报告，在我们的案例中，也就是交易的活跃性。我请求ChatGPT提供一个AWS CloudWatch查询脚本，并将其调整到只需要在每个进程运行结束时执行即可获得结果。如果想要通过详细阅读文档来完成这项任务，可能需要5~6小时，所以我选择了ChatGPT的方案，并节省了大量时间。

同样，我必须在Google Cloud中设置一些警报。我与ChatGPT合作创建了一个查询，排除了一些实际上我们并不负责，也不面向用户的系统级错误。ChatGPT不仅节省了我数小时的阅读和工作时间，还给我提供了为整个团队建立有用的警报系统所需的自定义指标。

### 9.4.1 用户反馈

用户反馈是确保软件成功的关键因素。当用户看到他们的意见受到重视并据此改进软件时，他们更有可能成为忠实用户，并对你的产品抱有积极态度。明智的做法是在问题初期就积极解决，而不是等到用户普遍不满时，才去处理。

尽管经过了全面的测试，但软件在现实世界中仍然可能暴露出一些隐蔽的错误。用户就像细心的侦探，他们能够发现那些可能被忽视的问题。

用户常常会以你意想不到的方式使用软件，他们的创新思维可能会激发你开发新功能的灵感，甚至催生出一款全新的软件产品。

提升客户服务的工具箱中包含Zendesk、Freshdesk、Drift和Salesforce等工具。这些工具可以方便你处理各种事务，无论是在实时与客户交流的即时聊天工具中，还是在收集用户反馈的表单中。自动工单系统可以跟踪客户问题，确保没有遗漏。同时，不要忘记客户关系管理系统，它可以将所有客户互动集中在一个地方。这些工具在提供高效、快速响应的客户服务方面发挥着重要作用。

在此基础上，生成式AI无疑能够带来额外的价值。它特别擅长处理非结构化数据，例如用户反馈。

假设你有一个包含大量电子邮件、即时消息和联系表单信息的文件，你可以使用以下提示来访问ChatGPT。

**提示：**识别共同的主题和类别，如可用性、性能、功能、错误和客户服务。同时对该文件进行情感分析，帮助确定哪些错误需要首先修复，或哪些功能值得考虑添加。基于这些信息，创建一份包含图表的报告。

ChatGPT还可以帮助我们起草回复。如果你正在处理用户反馈，它可以帮助你撰写更有效的回复——通过创建常见问题的回复模板。这

样，在与用户或客户聊天时，你就能始终保持准确和专业。

也可以使用ChatGPT起草更具个性化的回复。复制并粘贴用户的电子邮件，然后使用如下提示。

**提示：**给用户写一封回复邮件，语气要平和、善解人意。确保回复友好，并在不使事态升级的情况下缓解用户的担忧。{电子邮件}

在某些情况下，你可以创建自己的基于LLM的应用程序来处理用户反馈。比如，Warp公司的一名开发者花了不到一周的时间就创建了Warp，这还是在他将一半时间投入其他工作的情况下。Warp是使用OpenAI API构建的。

Warp的产品经理Noah Zweben说：“这个应用程序带来了巨大的改变。以前，对收到的用户反馈进行分类和排序是一项艰巨的工作。生成式AI能够非常出色地完成这项工作。”

## 9.4.2 产品发布

在ChatGPT成为万众瞩目的焦点之前，生成式AI就已经在营销等关键领域发挥显著作用了。Jasper公司便是其中的先驱之一。该公司的业务增长速度惊人。实事求是地说，生成式AI在迅速创作吸引人的创意内容方面确实有独到之处。

发布软件并不需要Jasper，ChatGPT就够了。你可以使用ChatGPT制定一些强有力的营销策略。参考以下提示范例。

**提示：**你创建了一个应用程序，旨在帮助用户规划健康膳食。它可以根据用户的饮食习惯、健康目标和营养需求制订个性化的膳食计划。它还可以生成购物清单、提供食谱建议并跟踪营养摄入情况。请为该应用程序制订营销计划。公司目前处于初创阶段，营销预算有限。

ChatGPT首先建议确定目标受众。它推荐的主力用户群体包括“注重健康的人群、健身爱好者、有特殊饮食需求的用户（如无麸质饮食

者、素食者)、忙碌的职场人士等”。接着，ChatGPT介绍了多种营销策略，如利用社交媒体、内容营销、社区参与、电子邮件营销和建立合作伙伴关系等。

以下是其他一些有用的ChatGPT提示范例。

**提示：**撰写一封针对健身教练和健康顾问的有说服力的销售邮件，推广我们的健康规划应用程序。

**提示：**为一个新的健康规划应用程序构思一个易于记忆且能够传达健康理念的名称。

**提示：**编写Facebook和Instagram广告文案，宣传我们的健康规划应用程序，突出它的易用性和个性化功能。

**提示：**制定一个模板，请求健康规划应用程序的早期用户提供推荐信，以便在营销材料中使用。

**提示：**为健康规划应用程序的虚拟发布活动制作一份邀请函，详细说明议程和特邀嘉宾。

## 9.5 小结

本章如同一部幕后纪录片，揭示了软件开发中不太引人注目的事情。诚然，开发新软件的过程充满激情与挑战，但正是调试、测试和文档编写这些日常却关键的环节，最终决定了软件的成败。本章深入探讨了ChatGPT等AI工具如何助力这些幕后工作，让它们变得更加高效。

尽管AI并非万能，但AI在处理海量数据、提供专业建议、撰写文档、发现潜在问题以及加速整个开发流程方面表现出色。有了AI的辅助，开发者得以将精力集中在那些最具挑战性的问题上。从错误修复到产品发布的每个环节，通过运用生成式AI和智能化策略，开发团队将能

够打造出更优质的软件，真正满足用户的需求，同时大幅缩短开发周期。

## 第10章

# AI辅助编程的一些启示

本章将简要介绍本书中关于AI辅助编程的核心要点，为你提供一个迅速掌握该领域知识的概览。

## 10.1 陡峭的学习曲线

尽管AI辅助编程是一项令人兴奋的前沿技术，但其要求开发者在很短的时间内掌握大量的新技能，即使对经验丰富的开发者来说，这也非常具有挑战性。科技行业的快速发展则加剧了这一挑战。想要跟上科技发展的步伐，就像试图徒步赶上一列快速行驶的列车，开发者可能会感到力不从心。

关键挑战之一在于如何适应LLM的细微差别，因为LLM比传统的编程方法复杂得多且难以预测。习惯于传统编程的开发者期待着一个线性过程，其中代码的行为是可预测的并能完全按照编写的方式执行。然而，AI模型具有不可预测性和自主性，这既令人困惑又令人兴奋。这些模型通常会以人类开发者无法立即直观感知的方式处理和响应代码请求。

AI模型带来的变化要求开发者转变思维方式。开发者必须学会预测和解释AI模型经常出人意料的输出，这与他们已经习惯的简单的、有逻辑的做法不同，并且起初可能会令人感到不适。

## 10.2 主要优势

在传统的编程过程中，当遇到困难时，常见的做法是：停止一切工作，疯狂搜索Stack Overflow，或翻阅大量文档，或搜索Google。

但是，AI辅助编程工具就像可以为你提供救生圈的伙伴。它们会在你输入时弹出建议和修正，让你保持头脑清醒，避免在一堆选项卡或应用程序间切换时迷失方向。你可以进入“心流”状态，并一直保持。

有些AI辅助编程工具很聪明，它们能识别你的编码风格及变量、函数和方法的名称。它们甚至能了解代码的整体氛围。因此，当它们给出建议时，它们并不是随意给出的。它们的智慧是为你的项目量身定制的。

现在让我们来谈谈真正的价值主张：AI辅助编程工具将你从无聊的工作中解放出来。这些工作都是让你想打瞌睡的例行任务，如文件处理、数据处理、API调用、用户界面美化、正则表达式处理、代码启动、令人头疼的Bash命令编写，以及与GitHub Actions打交道等，不一而足。

AI为编程带来的价值远不止于此。AI也是理解代码的高手。AI可以自动记录所有代码，并用浅显易懂的语言将所有内容整理出来，而不是让你夜以继日地记下笔记和解释。这一点相当重要，尤其当你把新人带入项目或重温旧代码时。AI可以把一切都安排好。

## 10.3 不足之处

AI辅助编程也有一些棘手的问题有待解决。首先，当AI辅助编程工具生成代码时，就会出现代码所有权的问题。如果你在AI的帮助下做出了很棒的产品，谁可以从中获利？围绕法律的方方面面都很复杂，可能需要一段时间后业界才能达成共识。

安全是另一个需要关注的事情。事实证明，有时AI辅助编程工具生成的代码可能存在安全漏洞。我们无法真正窥探LLM如何变魔术，因为它几乎就是一个神秘的盒子，我们很难确定从中可以得到什么。开发者在拿到AI辅助编程工具生成的代码后，在实际使用之前，必须对它们进行测试和检查。

AI辅助编程工具还有潜在的隐私问题。如果管理不慎，AI辅助编程工具可能会无意中学习私人代码库或专有数据。此外，开发者输入AI模型的数据也可能包含个人信息。

我们面临的挑战是如何确保AI辅助编程工具的训练和使用方式遵循隐私与保密性。这意味着必须实施稳健的数据处理和隐私政策，并确保遵守法规，如《通用数据保护条例》。

需要注意的是，基础训练数据的质量参差不齐等因素有时会造成AI辅助编程工具出错。AI辅助编程工具生成的代码可能远非最佳，或者过于冗长。此外，由于LLM的不可预测性和复杂性，你永远不知道会得到什么。

## 10.4 提示工程是科学与艺术的结合

学习提示工程不是一件容易的事情。提示工程是一门艺术，因为你需要掌握挑选正确词语来引导AI的诀窍。其中涉及如何创造性地提出问题或设置场景。

提示工程也是一门科学。你需要研究AI模型的工作方式，并在不断的尝试中猜测AI将如何响应提示。写提示就像做侦探，进行实验，检查结果，调整提示，直至得到想要的答案。

幸运的是，想要创作出更好的提示，有一些指南可以遵循。例如，提示的长度很关键。如果提示太长，AI可能会感到困惑或忽略重点。提示应该是具体且明确的。资深软件工程师Ankit Anchlia说：“提示必须清晰明确、重点突出，同时必须有足够的上下文；否则，你很可能得不到想要的回应。”

## 10.5 编程之外的能力

AI辅助编程工具的能力远不止于编程。在本书中，它们就像瑞士军刀一样，可以完成各种任务，从集思广益、策划项目到深入市场调研、记录需求。例如，对于快速编写一流的产品需求文档和软件需求规约，AI辅助编程工具可以提供帮助，甚至做得更好。

当你准备推出产品时，不妨猜猜AI能提供什么样的帮助。此时，AI就像装在口袋里的营销大师。你可以使用AI制订营销计划，让产品脱颖而出，引起人们的关注。你还可以使用AI分析用户反馈，不断改进和优化应用程序。

## 10.6 AI不会抢走开发者的工作

当前，人们最担心的是AI会抢走所有人的工作。世界似乎正朝着某种机器接管一切的“天网”方向发展，你为学习编程所付出的努力，会因此白费吗？也不尽然。

事实是，尽管AI辅助编程工具功能强大，但它们不是用来取代开发者的。它们更像是开发者的助手，让开发者变得更优秀。它们并不像真正的开发者那样聪明或独立。

但是，如果你不使用AI辅助编程工具，你可能会落伍。越来越多的雇主希望开发者具备使用AI系统的能力，因为它们带来的好处实在太显著了。我们都需要与时俱进。学会使用AI工具正在成为开发者的一项必备技能，这不是为了防止它们取代开发者（它们做不到），而是因为它们能帮助开发者更好地完成工作。

正如Durable公司创始人兼首席执行官James Clift所说：

AI的进步将导致劳动力市场动态变化。但有一点需要谨记，AI和商业可以更好地结合在一起。不要害怕AI，而是要拥抱AI，让AI支持企业的进一步发展和创造就业机会。AI工具将曾经只有大公司才能使用的资源交到每个普通人的手中。

## 10.7 小结

本书剖析了AI辅助编程的各个方面，然而这仅仅是冰山一角。当前，我们仍处于AI发展的初期阶段，令人鼓舞的是，其未来只会更加璀璨。这项技术正以令人瞠目的速度进步，这预示着将会有更多卓越的工具和策略涌现，助力开发者取得前所未有的成就。我们正身处这场激动人心的竞赛的起点，前方等待着我们的将是层出不穷的无限可能。这是一个激动人心的时代，试想一下，我们将掌握多少新颖的方法来精进技能、优化工作流程，甚至创造出超乎想象的奇迹。AI引领的编程新纪元，未来必定璀璨夺目、潜力无穷。让我们怀揣着对未知的热忱，拭目以待，迎接那即将涌现的令人震撼的科技进步。让我们一同乘上AI创新的巨轮，共同探索，它将引领开发者走向崭新的天地。

## 关于作者

**汤姆·陶利 (Tom Taulli)** 作为一位资深顾问，为包括获得风险投资支持的生成式AI初创公司Aisera在内的多家公司提供专业指导。他凭借深厚的专业知识，撰写了 *AI Basics* 和 *Generative AI* 等多部著作，这些图书涵盖了 ChatGPT、GPT-4 等前沿 LLM 的技术细节和应用前景。此外，汤姆还积极投身于教育事业，在 O'Reilly、加州大学洛杉矶分校和 Pluralsight 等平台或院校教授 IT 课程，传授如何使用 Python 语言构建深度学习和机器学习模型，并深入探讨自然语言处理等主题。

## 关于封面

本书封面插图中的动物是斜带吻棘鲀

(*Rhinecanthus rectangulus*)，它在夏威夷群岛享有盛誉，人们亲切地称之为humuhumunukunukuapua'a，简称humuhumu。经过民众投票和夏威夷州立法机构批准，斜带吻棘鲀于1984年被选为夏威夷州州鱼。

斜带吻棘鲀主要栖息于印度洋—太平洋及夏威夷的浅礁区域。它们的食物主要包括藻类及珊瑚礁上的无脊椎动物，如海胆和蜗牛。

尽管斜带吻棘鲀倾向于与观察者保持距离，但它们独特的冲锋习性、鲜明的领地行为及引人注目的外观——丰满的嘴巴、蓝色的上唇及长达25 cm的体型——使得它们尤为引人注目，容易被观察者发现。

斜带吻棘鲀种类繁多，大约有40个品种。O'Reilly图书封面插图中的许多动物正面临灭绝的威胁，它们对地球很重要。

本书封面插图由Karen Montgomery根据  
*Oeuvres du Comte De Lacépède*中的一幅古董线刻版画绘制。

配套资源验证码 240957

\*文中代码字体版权说明<sup>注</sup>

**[13]Copyright Copyright 2010, 2012 Adobe Systems Incorporated (<http://www.adobe.com/>), with Reserved Font Name 'Source'. License This Font Software is licensed under the SIL Open Font License, Version 1.1. This license is copied below, and is also available with a FAQ at: <http://scripts.sil.org/OFL> SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007 PREAMBLE The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others. The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives. DEFINITIONS "Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source**

**files, build scripts and documentation.** "Reserved Font Name" refers to any names specified as such after the copyright statement(s). "Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s). "Modified Version" refers to any derivative made by adding to, deleting, or substituting â€" in part or in whole â€" any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment. "Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

**PERMISSION & CONDITIONS** Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

- 1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.
- 2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.
- 3) No Modified Version of the Font Software may use

**the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.**

**4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.**

**5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.**

**TERMINATION** This license becomes null and void if any of the above conditions are not met.

**DISCLAIMER** THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,

**ARISING FROM, OUT OF THE USE OR  
INABILITY TO USE THE FONT SOFTWARE OR  
FROM OTHER DEALINGS IN THE FONT  
SOFTWARE.**