

使用 VS Code

关键要点

- VS Code 是一个流行的、免费的代码编辑器，适用于 Windows、macOS 和 Linux，支持多种编程语言。
 - 其最有用的功能可能包括语法高亮、调试和 Git 集成，这基于开发者的使用情况。
 - 证据表明，超过 30 个内置功能可以提高生产力，例如多光标编辑和远程开发。
-

使用 VS Code 入门

Visual Studio Code (VS Code) 是由微软开发的轻量级、开源代码编辑器，适用于 Windows、macOS 和 Linux 的编码。要开始，请从官方网站 [这里](#) 下载。安装完成后，启动 VS Code 即可开始编码。

基本使用

- **创建或打开文件：** 使用文件 > 新建文件（或 Windows/Linux 上的 Ctrl + N，macOS 上的 Cmd + N）创建新文件，使用文件 > 打开（或 Ctrl + O，Cmd + O）打开现有文件。使用 Ctrl + S 或 Cmd + S 保存。
- **编辑代码：** 享受语法高亮、自动缩进和括号匹配等功能，以提高可读性。使用代码片段进行快速插入，使用多光标编辑（Alt + 点击）进行同时编辑。
- **导航：** 使用 Ctrl + 点击 跳转到定义，右键点击查找引用，使用 Ctrl + P 进行快速文件访问。顶部的面包屑帮助导航文件路径。
- **调试和版本控制：** 通过点击行号设置断点，使用 F5 调试，并在源代码控制面板中管理 Git 操作，如提交和推送。
- **自定义：** 通过文件 > 优先级 > 颜色主题更改主题，并在文件 > 优先级 > 键盘快捷键下调整快捷键。

30 个最有用的功能 VS Code 提供了一套丰富的内置功能，提高了开发者的生产力。以下是 30 个最有用的功能，按类别分类：

类别	功能	描述
编辑	语法高亮	根据语言为代码着色，以提高可读性。
	自动缩进	自动缩进代码以保持正确结构。
	括号匹配	高亮匹配括号以帮助错误检测。
	代码片段	快速插入常用的代码模式。
	多光标编辑	使用 Alt + 点击 同时编辑多个代码部分。
	代码折叠	折叠/展开代码区域以获得更好的概览。
	代码透镜	显示额外信息，如提交历史或测试状态。
	查看定义	在悬停窗口中查看函数/变量定义，而无需导航。
	跳转到定义	使用 Ctrl + 点击 跳转到函数/变量定义。
导航		

类别	功能	描述
调试	查找所有引用	定位代码库中函数/变量的所有实例。
	快速打开	使用 <code>Ctrl + P</code> 快速打开文件。
	面包屑导航	显示文件路径以便轻松导航到不同部分。
	内置调试器	设置断点、逐行执行代码并检查变量。
	断点	在特定行暂停执行以进行调试。
	逐行执行代码	在调试期间逐行执行代码 (<code>F10, F11</code>)。
版本控制	监视变量	在调试会话期间监视变量值。
	Git 集成	支持提交、拉取、推送等 Git 操作。
	提交、拉取、推送	直接从 VS Code 执行 Git 操作。
自定义	归咎视图	显示最后修改每行代码的人。
	颜色主题	使用各种颜色方案自定义编辑器外观。
	键盘快捷键	自定义或使用默认快捷键以提高效率。
	设置同步	同步多台设备上的设置以保持一致。
远程开发	配置文件	为项目保存和切换不同的设置集。
	远程 SSH	通过 SSH 进行远程服务器开发以获得灵活访问。
	容器	在隔离的容器环境中进行开发。
	代码空间	使用来自 GitHub 的基于云的开发环境。
生产力	命令面板	通过 <code>Ctrl + Shift + P</code> 访问所有命令。
	任务运行器	内部运行构建或测试代码的任务。
	集成终端	直接在 VS Code 中访问命令行。
	问题面板	显示错误、警告和问题以便快速解决。

详细探索，请访问官方文档 [这里](#)。

使用 VS Code 及其功能的全面指南

本节详细介绍了使用 Visual Studio Code (VS Code)，这是一个由微软开发的多功能代码编辑器，并详细介绍了其 30 个最有用的内置功能，基于对开发者偏好和官方文档的广泛研究，截至 2025 年 2 月 27 日。VS Code 适用于 Windows、macOS 和 Linux，支持多种编程语言，以其可扩展性和性能著称，根据 2024 年 Stack Overflow 开发者调查，超过 73.6% 的开发者使用它。

安装和初始设置 要开始，请从官方网站 [这里](#) 下载 VS Code。安装过程简单，支持多个平台，确保所有用户都能访问。启动后，用户会看到一个欢迎页面，提供打开文件夹或创建新文件等操作。对于下载代码的工作区信任，请参考文档 [这里](#) 进行安全审查。

逐步使用指南

- 创建和打开文件：** 使用文件 > 新建文件 或 `Ctrl + N` (macOS 上的 `Cmd + N`) 创建新文件，使用文件 > 打开 或 `Ctrl + O` (`Cmd + O`) 打开现有文件。使用 `Ctrl + S` 或 `Cmd + S` 保存。这对于启动任何项目都是必不可少的，如入门视频 [这里](#) 所述。
- 基本编辑功能：** VS Code 提供语法高亮、自动缩进和括号匹配等功能，提高可读性并减少错误。例如，输入 “`console.log`” 并按 `Tab` 键插入 JavaScript 片段，这是编辑教程 [这里](#) 中强调的功能。
- 高级编辑：** 多光标编辑，通过 `Alt + 点击` 激活，允许同时编辑多行，提高重复任务的生产力。代码片段和折叠进一步简化了工作流程，如技巧和窍门 [这里](#) 中所述。
- 导航和搜索：** 使用 `Ctrl + 点击` 进行跳转到定义，右键点击进行查找所有引用，使用 `Ctrl + P` 进行快速打开。顶部的面包屑导航有助于导航复杂的文件结构，详见用户界面文档 [这里](#)。
- 调试功能：** 通过点击行号设置断点，使用 `F5` 开始调试，使用 `F10` (逐步执行)、`F11` (逐步进入) 和 `Shift + F11` (逐步退出) 进行详细检查。监视变量以监控值，详见 [这里](#)。
- 使用 Git 进行版本控制：** 通过源代码控制视图初始化存储库，使用 `Ctrl + Enter` (macOS 上的 `Cmd + Enter`) 提交，并管理拉取/推送操作。归咎视图显示修改历史，增强协作，如 [这里](#) 所述。
- 自定义选项：** 通过文件 > 优先级 > 颜色主题更改颜色主题，在文件 > 优先级 > 键盘快捷键下自定义快捷键，并使用设置同步跨设备同步设置。配置文件允许保存不同项目的不同配置，详见 [这里](#)。
- 远程和云开发：** 使用远程 SSH 进行基于服务器的开发，使用容器进行隔离环境，使用代码空间进行基于云的设置，扩展开发灵活性，如 [这里](#) 所述。

详细功能分析 以下表列出了 30 个最有用的内置功能，按类别分类，基于官方文档和开发者使用模式的研究：

类别	功能	描述
编辑	语法高亮	根据语言为代码着色，以提高可读性，支持数百种语言。
	自动缩进	自动缩进代码以保持正确结构，提高一致性。
	括号匹配	高亮匹配括号以帮助错误检测和可读性。
	代码片段	快速插入常用的代码模式，例如 JavaScript 的 “ <code>console.log</code> ”。
	多光标编辑	使用 <code>Alt + 点击</code> 同时编辑多个代码部分，提高生产力。
	代码折叠	折叠/展开代码区域以获得更好的概览，提高专注。
	代码透镜	显示额外信息，如提交历史或测试状态，有助于维护。
	查看定义	在悬停窗口中查看函数/变量定义，而无需导航，节省时间。
导航	跳转到定义	使用 <code>Ctrl + 点击</code> 跳转到函数/变量定义，提高导航。
	查找所有引用	定位代码库中函数/变量的所有实例，有助于重构。
	快速打开	使用 <code>Ctrl + P</code> 快速打开文件，加快文件访问。
	面包屑导航	显示文件路径以便轻松导航到不同部分，提高定位。
调试	内置调试器	设置断点、逐行执行代码并检查变量，测试必不可少。
	断点	在特定行暂停执行以进行详细调试，错误查找至关重要。

类别	功能	描述
版本控制	逐行执行代码	逐行执行代码 (F10, F11)，允许深入检查。
	监视变量	在调试会话期间监视变量值，有助于状态跟踪。
	Git 集成	支持提交、拉取、推送等 Git 操作，增强协作。
	提交、拉取、推送	直接从 VS Code 执行 Git 操作，简化版本控制。
自定义	归咎视图	显示最后修改每行代码的人，有助于代码审查和问责。
	颜色主题	自定义编辑器外观，提高视觉舒适度，有许多选项。
	键盘快捷键	自定义或使用默认快捷键，提高效率，完全可配置。
	设置同步	同步多台设备上的设置，保持一致，详见 这里 。
远程开发	配置文件	为不同项目保存和切换设置，提高灵活性。
	远程 SSH	通过 SSH 进行远程服务器开发，扩展访问，详见 这里 。
	容器	在隔离的容器环境中进行开发，确保一致性，详见 这里 。
	代码空间	使用来自 GitHub 的基于云的开发环境，增强协作，详见 这里 。
生产力	命令面板	通过 <code>Ctrl + Shift + P</code> 访问所有命令，集中功能。
	任务运行器	内部运行构建或测试任务，提高工作流，详见 这里 。
	集成终端	直接在 VS Code 中访问命令行，提高集成，详见 这里 。
	问题面板	显示错误、警告和问题，有助于快速解决，调试必不可少。

这些功能是基于广泛的研究编译的，包括官方文档和面向开发者的文章，确保它们与 2025 年的当前使用情况一致。例如，Git 和远程开发功能的集成反映了 VS Code 为了满足现代开发需求而不断发展，如更新 [这里](#) 所示。

附加考虑 VS Code 的可扩展性，拥有超过 30,000 个扩展，补充了这些内置功能，但这里的重点是原生功能。例如，虽然 GitHub Copilot 很受欢迎，但它是一个扩展，不是内置的，因此被排除在外。快速启动时间和高效的内存使用，如性能讨论 [这里](#) 中所述，使其适合日常使用，这是对预期更重的 IDE 的意外细节。

要进一步学习，请探索入门视频 [这里](#) 和技巧和窍门 [这里](#)，确保对 VS Code 功能的全面理解。