

# Mit VS Code

## Wichtige Punkte

- VS Code ist ein beliebter, kostenloser Code-Editor für Windows, macOS und Linux, der viele Programmiersprachen unterstützt.
  - Es scheint wahrscheinlich, dass seine nützlichsten Funktionen Syntax-Hervorhebung, Debugging und Git-Integration umfassen, basierend auf der Nutzung durch Entwickler.
  - Die Beweise deuten darauf hin, dass über 30 eingebaute Funktionen die Produktivität erhöhen, wie z.B. Multi-Cursor-Bearbeitung und Remote-Entwicklung.
- 

## Erste Schritte mit VS Code

Visual Studio Code (VS Code) ist ein leichtgewichtiger, quelloffener Code-Editor, der von Microsoft entwickelt wurde und ideal für das Codieren auf Windows, macOS und Linux ist. Um zu beginnen, laden Sie ihn von der offiziellen Website hier herunter. Nach der Installation starten Sie VS Code, um mit dem Codieren zu beginnen.

## Grundlegende Nutzung

- **Erstellen oder Öffnen von Dateien:** Verwenden Sie `Datei > Neue Datei` (oder `Strg + N` auf Windows/Linux, `Cmd + N` auf macOS) für neue Dateien und `Datei > Öffnen` (oder `Strg + O`, `Cmd + O`) für bestehende Dateien. Speichern Sie mit `Strg + S` oder `Cmd + S`.
- **Code bearbeiten:** Nutzen Sie Funktionen wie Syntax-Hervorhebung, automatische Einrückung und Klammerabgleich für bessere Lesbarkeit. Verwenden Sie Code-Snippets für schnelle Einfügungen und Multi-Cursor-Bearbeitung (`Alt + Klick`) für gleichzeitige Bearbeitungen.
- **Navigation:** Springen Sie zu Definitionen mit `Strg + Klick`, finden Sie Verweise über Rechtsklick und verwenden Sie `Strg + P` für schnellen Dateizugang. Brotkrumen oben helfen bei der Navigation durch Dateipfade.
- **Debugging und Versionskontrolle:** Setzen Sie Haltepunkte durch Klicken auf den Rand, debuggen Sie mit `F5` und verwalten Sie Git-Operationen wie Commit und Push im Quellsteuerungs-Panel.
- **Anpassung:** Ändern Sie Themen über `Datei > Einstellungen > Farbschema` und passen Sie Tastenkombinationen unter `Datei > Einstellungen > Tastenkombinationen an`.

**30 Nützlichste Funktionen** VS Code bietet einen reichen Satz eingebauter Funktionen, die die Produktivität von Entwicklern erhöhen. Hier sind 30 der nützlichsten, kategorisiert für Klarheit:

<b>Kategorie</b>	<b>Funktion</b>	<b>Beschreibung</b>
<b>Bearbeiten</b>	Syntax-Hervorhebung	Färbt den Code basierend auf der Sprache für bessere Lesbarkeit.
	Automatische Einrückung	Rückt den Code automatisch ein, um eine ordentliche Struktur zu gewährleisten.
	Klammerabgleich	Hebt passende Klammern hervor, um Fehler zu erkennen.
	Code-Snippets	Fügt häufig verwendete Code-Muster schnell ein.
	Multi-Cursor-Bearbeitung	Bearbeitet mehrere Code-Teile gleichzeitig mit <code>Alt + Klick</code> .
	Code-Falten	Klappt/entfaltet Code-Bereiche für einen besseren Überblick.
	Code Lens	Zeigt zusätzliche Informationen wie Commit-Historie oder Teststatus.
<b>Navigation</b>	Definition anzeigen	Zeigt Definitionen von Funktionen/Variablen in einem Hover-Fenster ohne Navigation.
	Zu Definition gehen	Springt zu Definitionen von Funktionen/Variablen mit <code>Strg + Klick</code> .
	Alle Verweise finden	Findet alle Vorkommen einer Funktion/Variablen im Code.
	Schnell öffnen	Öffnet Dateien schnell mit <code>Strg + P</code> .
<b>Debugging</b>	Brotkrumen-Navigation	Zeigt den Dateipfad für einfache Navigation zu verschiedenen Teilen.
	Eingebauter Debugger	Setzt Haltepunkte, führt durch den Code und untersucht Variablen.
	Haltepunkte	Pausiert die Ausführung an bestimmten Zeilen für das Debugging.
	Code Schritt für Schritt ausführen	Führt den Code Zeile für Zeile während des Debuggens aus ( <code>F10</code> , <code>F11</code> ).
<b>Versionskontrolle</b>	Variablen überwachen	Überwacht Variablenwerte während Debugging-Sitzungen.
	Git-Integration	Unterstützt Git-Operationen wie Commit, Pull, Push aus der Box.
	Commit, Pull, Push	Führt Git-Aktionen direkt in VS Code aus.
	Blame-Ansicht	Zeigt, wer die letzte Änderung an jeder Codezeile vorgenommen hat.
<b>Anpassung</b>	Farbschemata	Passt das Erscheinungsbild des Editors mit verschiedenen Farbschemata an.
	Tastenkombinationen	Passt Tastenkombinationen an oder verwendet Standardkombinationen für Effizienz.
	Einstellungen synchronisieren	Synchronisiert Einstellungen auf mehreren Geräten für Konsistenz.

Kategorie	Funktion	Beschreibung
	Profile	Speichert und wechselt zwischen verschiedenen Einstellungssätzen für Projekte.
<b>Remote-Entwicklung</b>	Remote SSH	Entwickelt auf Remote-Servern über SSH für flexible Zugriff.
	Container	Entwickelt in isolierten Container-Umgebungen.
	Codespaces	Nutzt cloudbasierte Entwicklungsumgebungen von GitHub.
<b>Produktivität</b>	Befehlspalette	Greift auf alle Befehle über <code>Strg + Umschalt + P</code> zu.
	Task Runner	Führt Aufgaben wie das Bauen oder Testen von Code intern aus.
	Integriertes Terminal	Greift direkt auf die Kommandozeile innerhalb von VS Code zu.
	Probleme-Panel	Zeigt Fehler, Warnungen und Probleme für eine schnelle Behebung.

Für eine detaillierte Erforschung besuchen Sie die offizielle Dokumentation hier.

## Umfassender Leitfaden zur Verwendung von VS Code und seinen Funktionen

Dieser Abschnitt bietet einen tiefgehenden Einblick in die Verwendung von Visual Studio Code (VS Code), einem vielseitigen Code-Editor von Microsoft, und beschreibt seine 30 nützlichsten eingebauten Funktionen, basierend auf umfangreicher Forschung zu Entwicklerpräferenzen und offizieller Dokumentation bis zum 27. Februar 2025. VS Code, verfügbar für Windows, macOS und Linux, unterstützt eine breite Palette von Programmiersprachen und ist für seine Erweiterbarkeit und Leistung bekannt, mit über 73,6 % der Entwickler, die es gemäß der Umfrage der Entwickler von Stack Overflow 2024 verwenden.

**Installation und erste Einrichtung** Um zu beginnen, laden Sie VS Code von der offiziellen Website hier herunter. Die Installation ist einfach und unterstützt mehrere Plattformen, was die Zugänglichkeit für alle Benutzer gewährleistet. Beim Starten werden Benutzer mit einer Willkommenseite begrüßt, die Aktionen wie das Öffnen eines Ordners oder das Erstellen einer neuen Datei anbietet. Für den Workspace-Trust, insbesondere bei heruntergeladenem Code, überprüfen Sie ihn auf Sicherheit, wie in der Dokumentation hier beschrieben.

## Schritt-für-Schritt-Anleitung zur Nutzung

- 1. Erstellen und Öffnen von Dateien:** Verwenden Sie `Datei > Neue Datei` oder `Strg + N` (`Cmd + N` auf macOS) für neue Dateien und `Datei > Öffnen` oder `Strg + O` (`Cmd + O`) für bestehende Dateien. Spe-

ichern Sie mit `Strg + S` oder `Cmd + S`. Dies ist für den Start jedes Projekts wesentlich, wie in Einführungsvideos hier beschrieben.

2. **Grundlegende Bearbeitungsfunktionen:** VS Code bietet Syntax-Hervorhebung, automatische Einrückung und Klammerabgleich standardmäßig, was die Lesbarkeit verbessert und Fehler reduziert. Zum Beispiel fügt das Tippen von `console.log` und das Drücken der Tab-Taste ein JavaScript-Snippet ein, eine Funktion, die in Bearbeitungstutorials hier hervorgehoben wird.
3. **Erweiterte Bearbeitung:** Die Multi-Cursor-Bearbeitung, aktiviert durch `Alt + Klick`, ermöglicht gleichzeitige Bearbeitungen über mehrere Zeilen hinweg, was die Produktivität bei wiederholten Aufgaben steigert. Code-Snippets und Faltung vereinfachen den Arbeitsablauf weiter, wie in Tipps und Tricks hier beschrieben.
4. **Navigation und Suche:** Verwenden Sie `Strg + Klick` für "Zu Definition gehen", Rechtsklick für "Alle Verweise finden" und `Strg + P` für "Schnell öffnen". Brotkrumen-Navigation oben hilft bei der Navigation durch komplexe Dateistrukturen, wie in der Benutzeroberflächendokumentation hier beschrieben.
5. **Debugging-Funktionen:** Setzen Sie Haltepunkte durch Klicken auf den Rand, starten Sie das Debugging mit `F5` und verwenden Sie `F10` (Schritt über), `F11` (Schritt in) und `Umschalt + F11` (Schritt aus) für detaillierte Inspektionen. Überwachen Sie Variablen, um Werte zu überwachen, eine Funktion, die ausführlich hier beschrieben wird.
6. **Versionskontrolle mit Git:** Initialisieren Sie ein Repository über die Quellsteuerungsansicht, committen Sie mit `Strg + Enter` (macOS: `Cmd + Enter`) und verwalten Sie Pull/Push-Operationen. Die Blame-Ansicht zeigt die Änderungsgeschichte, was die Zusammenarbeit verbessert, wie hier beschrieben.
7. **Anpassungsoptionen:** Ändern Sie Farbschemata über `Datei > Einstellungen > Farbschema`, passen Sie Tastenkombinationen unter `Datei > Einstellungen > Tastenkombinationen` an und synchronisieren Sie Einstellungen auf verschiedenen Geräten mit `Einstellungen synchronisieren`. Profile ermöglichen das Speichern verschiedener Konfigurationen, wie hier beschrieben.
8. **Remote- und Cloud-Entwicklung:** Verwenden Sie Remote SSH für serverbasierte Entwicklung, Container für isolierte Umgebungen und Codespaces für cloudbasierte Setups, um die Entwicklungsflexibilität zu erweitern, wie hier beschrieben.

**Detaillierte Funktionsanalyse** Die folgende Tabelle listet die 30 nützlichsten eingebauten Funktionen auf, kategorisiert für Klarheit, basierend auf Forschung aus offizieller Dokumentation und Entwicklungsmustern:

Kategorie	Funktion	Beschreibung
<b>Bearbeiten</b>	Syntax-Hervorhebung	Färbt den Code basierend auf der Sprache für bessere Lesbarkeit, unterstützt Hunderte von Sprachen.

Kategorie	Funktion	Beschreibung
	Automatische Einrückung	Rückt den Code automatisch ein, um eine ordentliche Struktur zu gewährleisten, was die Konsistenz verbessert.
	Klammerabgleich	Hebt passende Klammern hervor, um Fehler zu erkennen und die Lesbarkeit zu verbessern.
	Code-Snippets	Fügt häufig verwendete Code-Muster schnell ein, z.B. "console.log" für JavaScript.
	Multi-Cursor-Bearbeitung	Bearbeitet mehrere Code-Teile gleichzeitig mit <code>Alt + Klick</code> , was die Produktivität steigert.
	Code-Falten	Klappt/entfaltet Code-Bereiche für einen besseren Überblick, was die Konzentration verbessert.
	Code Lens	Zeigt zusätzliche Informationen wie Commit-Historie oder Teststatus, was die Wartung erleichtert.
	Definition anzeigen	Zeigt Definitionen von Funktionen/Variablen in einem Hover-Fenster ohne Navigation, was Zeit spart.
<b>Navigation</b>	Zu Definition gehen	Springt zu Definitionen von Funktionen/Variablen mit <code>Strg + Klick</code> , was die Navigation verbessert.
	Alle Verweise finden	Findet alle Vorkommen einer Funktion/Variablen, nützlich für das Refactoring.
	Schnell öffnen	Öffnet Dateien schnell mit <code>Strg + P</code> , was den Dateizugang beschleunigt.
	Brotkrumen-Navigation	Zeigt den Dateipfad für einfache Navigation zu verschiedenen Teilen, was die Orientierung verbessert.
<b>Debugging</b>	Eingebauter Debugger	Setzt Haltepunkte, führt durch den Code und untersucht Variablen, wesentlich für das Testen.
	Haltepunkte	Pausiert die Ausführung an bestimmten Zeilen für detailliertes Debugging, kritisch für Fehlerbehebung.
	Code Schritt für Schritt ausführen	Führt den Code Zeile für Zeile aus ( <code>F10</code> , <code>F11</code> ), was eine tiefe Inspektion ermöglicht.
	Variablen überwachen	Überwacht Variablenwerte während Debugging-Sitzungen, was das Verfolgen des Zustands erleichtert.
<b>Versionskontrolle</b>	Git-Integration	Unterstützt Git-Operationen wie Commit, Pull, Push, was die Zusammenarbeit verbessert.
	Commit, Pull, Push	Führt Git-Aktionen direkt in VS Code aus, was die Versionskontrolle vereinfacht.
	Blame-Ansicht	Zeigt, wer die letzte Änderung an jeder Zeile vorgenommen hat, nützlich für Code-Reviews und Verantwortlichkeit.

Kategorie	Funktion	Beschreibung
<b>Anpassung</b>	Farbschemata	Passt das Erscheinungsbild des Editors an, was den visuellen Komfort verbessert, mit vielen Optionen.
	Tastenkombinationen	Passt Tastenkombinationen an oder verwendet Standardkombinationen, was die Effizienz erhöht, vollständig konfigurierbar.
	Einstellungen synchronisieren Profile	Synchronisiert Einstellungen auf mehreren Geräten, was die Konsistenz gewährleistet, detailliert hier. Speichert und wechselt zwischen Einstellungen für verschiedene Projekte, was die Flexibilität erhöht.
<b>Remote-Entwicklung</b>	Remote SSH	Entwickelt auf Remote-Servern über SSH, was den Zugriff erweitert, detailliert hier.
	Container	Entwickelt in isolierten Container-Umgebungen, was die Konsistenz gewährleistet, beschrieben hier.
	Codespaces	Nutzt cloudbasierte Entwicklungsumgebungen von GitHub, was die Zusammenarbeit verbessert, detailliert hier.
<b>Produktivität</b>	Befehlspalette	Greift auf alle Befehle über <code>Strg + Umschalt + P</code> zu, zentralisiert die Funktionalität.
	Task Runner	Führt Aufgaben wie das Bauen oder Testen intern aus, was den Arbeitsablauf verbessert, detailliert hier.
	Integriertes Terminal	Greift auf die Kommandozeile innerhalb von VS Code zu, was die Integration verbessert, beschrieben hier.
	Probleme-Panel	Zeigt Fehler, Warnungen und Probleme, was die schnelle Behebung erleichtert, wesentlich für das Debugging.

Diese Funktionen wurden aus umfangreicher Forschung zusammengestellt, einschließlich offizieller Dokumentation und entwicklerorientierter Artikel, um sicherzustellen, dass sie mit der aktuellen Nutzung bis 2025 übereinstimmen. Zum Beispiel spiegelt die Integration von Git- und Remote-Entwicklungsfunktionen die Entwicklung von VS Code wider, um modernen Entwicklungsanforderungen gerecht zu werden, wie in Updates hier zu sehen.

**Zusätzliche Überlegungen** Die Erweiterbarkeit von VS Code, mit über 30.000 Erweiterungen, ergänzt diese eingebauten Funktionen, aber der Fokus liegt hier auf den nativen Fähigkeiten. Zum Beispiel ist GitHub Copilot beliebt, aber es handelt sich um eine Erweiterung, keine eingebaute Funktion, daher wird es ausgeschlossen. Die schnelle Startzeit und der effiziente Speicherverbrauch, wie in Leistungsdiskussionen hier beschrieben, machen es für den täglichen Gebrauch geeignet, ein unerwartetes Detail für Benutzer, die schwerere IDEs erwarten.

Für weiteres Lernen erkunden Sie Einführungsvideos hier und Tipps und Tricks hier, um ein umfassendes Verständnis der Fähigkeiten von VS Code zu gewährleisten.

---

## Wichtige Zitate

- [Offizielle VS Code-Dokumentation](#)
- [VS Code-Funktionen-Tipps und Tricks](#)
- [VS Code-Debugging-Leitfaden](#)
- [VS Code-Git-Integration-Übersicht](#)
- [VS Code-Einstellungen-Synchronisationsdokumentation](#)
- [VS Code-Remote-SSH-Entwicklung](#)
- [VS Code-Container-Entwicklung](#)
- [VS Code-Codespaces-Entwicklung](#)
- [VS Code-Task-Runner-Dokumentation](#)
- [VS Code-Integriertes Terminal-Leitfaden](#)
- [Warum wir den VS Code-Editor gebaut haben](#)
- [VS Code-Einführungsvideos](#)
- [VS Code-Januar-2025-Update](#)