



VSCode-ObjectScript-HowTo *german*

Eine Einführung in [Microsoft's Visual Studio Code](#) und wie man Visual Studio Code mit [InterSystems ObjectScript](#) nutzt.

Überblick

- [Erste Schritte mit Visual Studio Code](#)
 - [Installiere Visual Studio Code](#)
 - [Ein Projekt öffnen](#)
 - [Die Visual Studio Code Oberfläche](#)
 - [Code schreiben und editieren](#)
 - [Workspaces](#)
 - [Einstellungen](#)
 - [Farbschema](#)
 - [Debuggen und Testen](#)
 - [Quellcodekontrolle](#)
- [Visual Studio Code mit InterSystems ObjectScript nutzen](#)
 - [InterSystems Extensions installieren](#)
 - [ObjectScript Farbschema wählen](#)
 - [Zu einem Server verbinden](#)
 - [Anmeldeinformationen speichern](#)
 - [Einen Namespace öffnen](#)
 - [ObjectScript Code schreiben](#)
 - [SQLTools benutzen](#)
- [Tipps zur Verwendung von Visual Studio Code](#)
- [Tastenkombinationen für Visual Studio Code](#)
 - [Keyboard reference sheet](#)
 - [Command Palette](#)
 - [Ansicht](#)
 - [Navigation](#)
 - [Bearbeiten](#)
- [Die wichtigsten Visual Studio Code Extensions](#)

Erste Schritte mit Visual Studio Code

Visual Studio Code (VSCode) ist ein leichtgewichtiger und dennoch leistungsstarker Quellcode-Editor, der von Microsoft für Windows, Linux und macOS entwickelt wurde. Er verfügt über integrierte Unterstützung für Debugging, Quellcodekontrolle und Entwicklungsaufgaben wie Linting und Build. In dieser Dokumentation werden wir die Grundlagen der Verwendung von Visual Studio Code zum Schreiben, Debuggen und Verwalten Ihres Codes erläutern.

Installiere Visual Studio Code

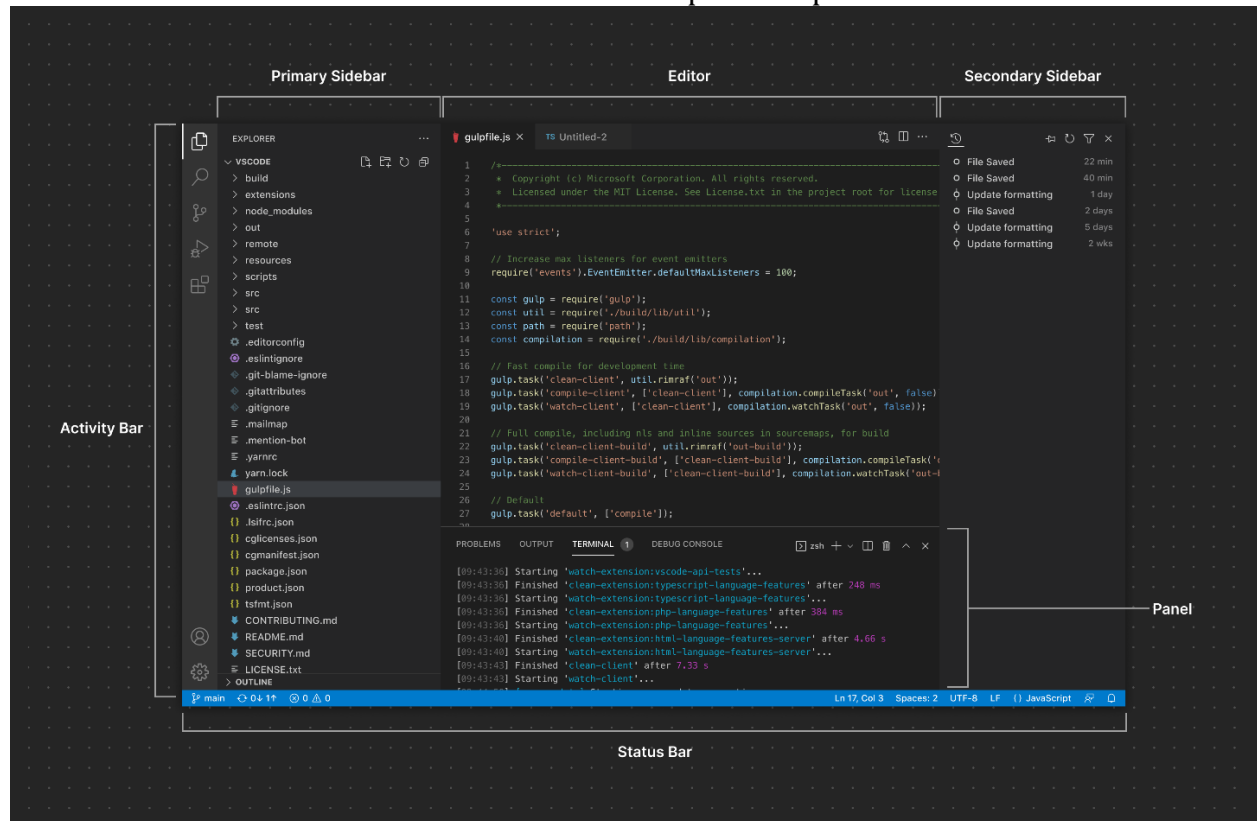
Bevor du anfangen kannst Visual Studio Code zu benutzen, musst du die passende Version für dein Betriebssystem [downloaden und installieren](#). Wenn die Installation abgeschlossen ist, kannst du Visual Studio Code starten, indem du das Symbol auf deinem Desktop anklickst oder im Start Menü (Windows) oder Spotlight (macOS) nach "Visual Studio Code" suchst.

Ein Projekt öffnen

Wenn du Visual Studio Code das erste Mal startest, wird dir ein Fenster mit *Open Folder* angezeigt. Hiermit kannst du ein bereits existierendes Projekt öffnen oder ein neues Projekt erstellen, indem du nach **File > New Folder** navigierst und dann den neuen Ordner auswählst. Wenn du einen Ordner geöffnet hast, kannst du anfangen Dateien zu deinem Projekt hinzuzufügen und bearbeiten.

Die Visual Studio Code Oberfläche

Die Visual Studio Code Oberfläche teilt sich in ein paar Hauptbereiche auf:



Activity Bar

Die Activity Bar ist eine der Hauptnavigationsflächen in Visual Studio Code, die sich auf der linken Seite des Fensters befindet und Zugriff auf verschiedene Ansichten und Funktionen, wie den Datei-Explorer und die Debug-Konsole, bietet.

Primary Sidebar

Hier werden die Inhalte der Activity Bar dargestellt.

Secondary Sidebar

Eine zweite Seitenansicht. Hier können Inhalte von anderen Bereichen hineingezogen werden, um einen besseren Überblick zu haben.

Editor

Hier kann man code schreiben und editieren.

Panel

Das Panel ist eine andere Ansicht um Inhalt anzeigen zu lassen.

Status Bar

Am unteren Rand des Fensters finden Sie Informationen und Kurzzugriffe für gängige Aktionen.

Code schreiben und editieren

Visual Studio Code verfügt über viele leistungsstarke Funktionen zum Schreiben und Bearbeiten von Code, darunter:

- **Syntax highlighting:** Visual Studio Code erkennt automatisch die Art des Codes, mit dem Sie arbeiten, und wendet eine Farbcodierung an, um das Lesen und Schreiben zu erleichtern.
 - **IntelliSense:** Visual Studio Code bietet intelligente Code-Vervollständigung und Vorschläge für viele Sprachen, die das schnelle und korrekte Schreiben von Code erleichtern.
 - **Code snippets:** Visual Studio Code enthält eine große Sammlung von Codeschnipseln, die mit wenigen Tastendrücken in Ihren Code eingefügt werden können, wodurch Sie Zeit und Mühe sparen.
-

Workspaces

Wählen Sie **File > Open File / Open Folder** um eine Datei oder Ordner als Workspace zu öffnen.

Zum aktuellen Workspace hinzufügen

Wählen Sie **File > Add Folder to Workspace...** und wählen Sie dann einen Ordner aus, den Sie zu Ihrem aktuellen Workspace hinzufügen möchten.

Workspace speichern

Damit Sie nicht jedes Mal alle Ordner in Ihrem Workspace öffnen müssen, wenn Sie Visual Studio Code öffnen, können Sie Ihren Arbeitsbereich als Datei speichern. Wählen Sie **File > Save Workspace As...** und wählen Sie ein Ziel und einen Namen für Ihre Workspacedatei. Anschließend wird eine `.code-workspace` Datei erstellt. Sie können den Workspace dann öffnen, indem Sie einfach die Datei öffnen oder über **File > Open Workspace from File...** die Datei auswählen.

Einstellungen

Öffne die Einstellungen mit **File > Preferences > Settings**.

User Einstellungen

Beim Bearbeiten von Einstellungen im *User* Tap gelten alle Einstellungen global für alle Visual Studio Code Workspaces.

Workspace Einstellungen

Beim Bearbeiten von Einstellungen im *Workspace* Tap werden alle Einstellungen nur auf den aktuell geöffneten Workspace angewendet werden.

Farbschema

Sie können das Aussehen der Visual Studio Code-Oberfläche an Ihre Bedürfnisse anpassen. Navigieren Sie zu `File > Preferences > Color Theme`, dort können Sie aus einem vorinstallierten Farbschema wählen oder mit `+ Browse Additional Color Themes...` ein neues Farbschema installieren.

Debuggen und Testen

Visual Studio Code bietet integrierte Unterstützung für das Debugging vieler Sprachen, darunter JavaScript, TypeScript und Python. Um mit dem Debuggen zu beginnen, müssen Sie eine "launch.json"-Datei konfigurieren, die Visual Studio Code mitteilt, wie die Anwendung gestartet und angehängt werden soll. Sobald Sie eine Startkonfiguration eingerichtet haben, können Sie das Debuggen starten, indem Sie in der Debug-Ansicht auf die Schaltfläche "Play" klicken oder F5 drücken.

Quellcodekontrolle

Visual Studio Code unterstützt Quellcodekontrollsysteme wie Git, SVN und TFS. Das Symbol für die Quellcodekontrolle wird in der linken Seitenleiste angezeigt, sobald Sie einen Ordner öffnen, der ein Git-Repository ist. So können Sie unter anderem die Änderungen sehen, Übertragungen vornehmen und Konflikte beim Zusammenführen behandeln.


Dies sind nur einige der vielen Funktionen, die Visual Studio Code zu bieten hat. Mit seinen leistungsstarken Bearbeitungsfunktionen, der integrierten Unterstützung für Debugging und Tests und der Integration in die Quellcodekontrolle ist Visual Studio Code eine hervorragende Wahl für eine Vielzahl von Entwicklungsaufgaben. Wenn Sie sich mit dem Tool vertraut machen, werden Sie feststellen, dass es Ihnen helfen kann, effizienter und effektiver zu arbeiten.

Visual Studio Code mit InterSystems ObjectScript nutzen

InterSystems ObjectScript ist eine Programmiersprache, die in InterSystems-Datenbanken wie InterSystems IRIS und Caché verwendet wird. Visual Studio Code (VSCode) ist ein beliebter Texteditor, der zum Schreiben und Bearbeiten von ObjectScript-Code verwendet werden kann.

InterSystems Extensions installieren

Gehen Sie zur Activity Bar auf der linken Seite und klicken Sie auf **Extensions** .

Geben Sie in die Suchleiste *InterSystems* ein und installieren Sie dann das [InterSystems ObjectScript Extension Pack](#) . Dieses Extension Pack enthält:



- [InterSystems ObjectScript](#): Fügt Unterstützung für die Sprache InterSystems ObjectScript hinzu.
 - [InterSystems Language Server](#): Fügt InterSystems ObjectScript language server hinzu.
 - [InterSystems Server Manager](#): Definieren Sie Verbindungen zu InterSystems-Servern. Durchsuchen und verwalten Sie diese Server.
-

ObjectScript Farbschema wählen

Um eine korrekte Farbhervorhebung bei der Arbeit mit ObjectScript zu gewährleisten, wählen Sie ein InterSystems Farbschema.

- Navigieren Sie zu **File > Preferences > Color Theme**
 - Wählen Sie **InterSystems Default Dark** oder **InterSystems Default Light** aus.
-

Zu einem Server verbinden

- Öffnen Sie **InterSystems Tools**  in der Activity Bar.
- Sie können die Standardkonfigurationen unter **ALL Servers** sehen.
- Wenn Sie eine neue Serververbindung erstellen möchten, klicken Sie oben auf die Schaltfläche  und geben Sie dann die Verbindungseigenschaften ein:
 - *Name of new server definition*: Geben Sie Ihrer Verbindung einen Namen.
 - *Optional Description*: Eine optionale Beschreibung für Ihren Server. Tippen Sie auf **Enter**, um es leer zu lassen.
 - *Hostname or IP address of web server*: Der Hostname oder die IP-Adresse Ihres Servers.
 - *Port of web server*: Der Port des Webservers.

- *Username*: Der Benutzername Ihres IRIS-Benutzers, von dem aus Sie sich verbinden möchten.
- *Confirm connection type*: *http* oder *https* Verbindungstyp.

Nach erfolgreichem Hinzufügen eines neuen Servers können Sie den Server unter **ALL Servers** sehen.

Tipp: Sie können Ihre Servereigenschaften bearbeiten unter: ... in der oberen rechten Ecke > Edit Settings > InterSystems: Servers > Edit in settings.json.

Anmeldeinformationen speichern

Jedes Mal, wenn Sie Visual Studio Code schließen und erneut öffnen, werden Sie aufgefordert, sich mit Ihren Anmeldeinformationen anzumelden. Um dies zu vermeiden, können Sie Ihre Anmeldeinformationen speichern:

- Benutzernamen speichern:
 - In dem **Command Palette** schreibe *Preferences: Open User Settings (JSON)*. Hier sehen Sie Ihre Serververbindung im JSON-Format.
 - Wählen Sie den Server aus, dem Sie Ihren Benutzernamen hinzufügen möchten, und fügen Sie eine Eigenschaft namens *username* hinzu.


Beispiel:

```
"default~iris": {
  "webServer": {
    ...
  },
  "username": "<your-username>",
  "description": "Connection to local InterSystems IRIS™ installed
with default settings."
}
```

- Passwort speichern
 - Wenn Sie Visual Studio Code das nächste mal öffnen, werden Sie zu einem Dialogfeld aufgefordert und nach einem Passwort gefragt
 - Geben Sie Ihr Passwort ein
 - Klicken Sie auf die *Schlüssel*-Schaltfläche in der oberen rechten Ecke des Dialogs, um das Passwort zu speichern
- Passwort löschen
 - Navigieren Sie zu *Accounts* in der Activity Bar
 - Wählen Sie Ihren Benutzer aus
 - Wählen Sie **Sign Out**
 - Wählen Sie **Delete** wenn Sie aufgefordert werden, das Passwort zu löschen

Einen Namespace öffnen

Einen Namespace mit InterSystems Tools öffnen

- Öffnen Sie **InterSystems Tools**  in der Activity Bar.
- Wählen Sie einen Server und klicken Sie darauf.
 - Wenn die Verbindung erfolgreich ist, sehen Sie ein neues Verzeichnis namens *Namespaces*.
 - Wenn die Verbindung fehlgeschlagen ist, sehen Sie *Unavailable at <current-time>*.
- Wählen Sie einen Namespace.
 - Klicken Sie auf das *Augen-Symbol* neben Ihrem Namespace, um den Namespace in Ihrem Workspace anzuzeigen.
 - Klicken Sie auf das *Stift-Symbol* neben Ihrem Namespace, um den Namespace in Ihrem Workspace zu bearbeiten.

Einen Namespace mit dem Explorer öffnen

- Öffnen Sie den **Explorer** in der Activity Bar.
 - Machen Sie einen Rechtsklick und wählen Sie **Add Server Namespace to Workspace**.
 - Wählen Sie nun Ihren Server aus oder erstellen Sie einen neuen mit dem **+** Knopf in der oberen rechten Ecke des Dialogs.
 - Wähle Sie einen Namespace.
-

ObjectScript Code schreiben


Nachdem Sie eine Verbindung zu einem Namespace hergestellt haben, können Sie mit dem Schreiben von ObjectScript-Code beginnen. Hier sind die grundlegenden Schritte dafür:

- Erstellen Sie eine neue Datei mit der Dateiendung **.cls**. (z.B. **MyPackage.MyClass.cls**)
- Schreiben Sie Ihren ObjectScript-Code in diese Datei.
- Die Klasse wird kompiliert, nachdem Sie die Datei gespeichert haben.

Sie können auch andere Dateien mit der Endung **.mac**, **.inc** usw. erstellen.

SQLTools benutzen

Wenn Sie einen Blick in Ihre Datenbank werfen möchten, ohne Visual Studio Code zu verlassen, können Sie die SQLTools-Extension verwenden.

- Installieren Sie die **SQLTools** Extension
- Installieren Sie die **SQLTools InterSystems IRIS** Extension
- Navigieren Sie zum Tap **SQLTools**  in der Activity Bar und wählen Sie **Add New Connection**
- Wählen Sie **InterSystems IRIS** und füllen Sie die Verbindungseigenschaften aus

- Wählen Sie **Connect Now** und Sie sehen eine neue Verbindung unter *Connections*
- Jetzt können Sie eine neue SQL-Datei erstellen und die Abfrage auf Ihrer Verbindung mit **Run on active connection** ausführen

ODER

- Durchsuchen Sie Ihre Tabellen und Views, indem Sie die Verbindung aufklappen, und zeigen Sie dann die Tabelle oder Ansicht an, indem Sie auf das Lupensymbol klicken.

Tipps zur Verwendung von Visual Studio Code

Hier sind ein paar Tipps, die Ihnen dabei helfen sollen, Visual Studio Code optimal zu nutzen.

Editor Playground

Wenn Sie gerade erst mit Visual Studio Code beginnen, können Sie den *Editor Playground* unter **Help > Editor Playground** ausprobieren. Hier erfahren Sie, wie Sie einige der leistungsstarken [Codebearbeitungsfunktionen](#) verwenden.

Command Palette

Die Command Palette ist ein leistungsstarkes Tool, mit dem Sie schnell und einfach auf fast alle Funktionen oder Befehle in Visual Studio Code zugreifen können. Hier können Sie auch schnell die Tastenbelegungen nachschlagen, falls Sie eine vergessen haben. Um die Command Palette zu öffnen, drücken Sie **Ctrl + Shift + P** (Windows/Linux) oder **Cmd + Shift + P** (macOS).

Verwenden Sie das integrierte Terminal

Visual Studio Code enthält ein integriertes Terminal, mit dem Sie Befehlszeilentools wie git und npm ausführen können. Um das Terminal zu öffnen, drücken Sie **Ctrl + Shift + Ö** (Windows/Linux) oder **Cmd + Shift + Ö** (macOS).

Verwenden Sie Codeschnipsel

Visual Studio Code enthält eine Vielzahl integrierter Codeausschnitte, mit denen Sie beim Schreiben von Code Zeit sparen können. Sie können zum Beispiel "for" eingeben und dann **Ctrl + Space** (Windows/Linux) oder **Cmd + Space** (macOS) drücken, um schnell eine for-Schleife zu erstellen. Sie können auch Ihre eigenen benutzerdefinierten Codeausschnitte erstellen, indem Sie zu **File > Preferences > Configure User Snippets** gehen.

Passen Sie das Farbschema an

Visual Studio Code enthält mehrere integrierte Farbschema, mit denen Sie das Erscheinungsbild des Editors anpassen können. Sie können das Farbschema ändern, indem Sie zu **File > Preferences > Color Theme** gehen. Um weitere Farbschemen zu finden, können Sie den Marktplatz verwenden.

Extensions benutzen

Visual Studio Code verfügt über ein riesiges Ökosystem von [Extensions](#), die dem Editor neue Features und Funktionen hinzufügen. Sie können Erweiterungen suchen und installieren, indem Sie zu **File > Extensions** gehen. Sehen Sie sich auch [Die wichtigsten Visual Studio Code Extensions](#) an.

Lernen Sie die Tastenkombinationen

Visual Studio Code verfügt über eine große Anzahl von Tastenkombinationen, mit denen Sie schneller und effizienter arbeiten können. Sie können die vollständige Liste der Tastenkombinationen anzeigen, indem Sie zu `File > Preferences > Keyboard Shortcuts` gehen, wo Sie auch die Tastaturkürzel anpassen können. Sehen Sie sich auch [Tastenkombinationen für Visual Studio Code](#) an.

Beim Speichern formatieren

Es ist nicht immer einfach, Ihren Code manuell in einem gut lesbaren Zustand zu halten. Glücklicherweise bietet Visual Studio Code einen Formatierer für fast jede Sprache. Um Ihren Code gleichzeitig mit dem Speichern zu formatieren, können Sie die Funktion `Format On Save` aktivieren:

- Öffnen Sie die Einstellungen unter `File > Preferences > Settings` oder über `Strg + ,`
- Tippen Sie `Format On Save`
- Aktivieren Sie das Kästchen für `Editor: Format On Save`

Sticky Scroll

Wenn Sie Code mit langen Klassen und Methoden arbeiten, die über die vertikale Größe Ihres Bildschirms hinausgehen, kann es schwierig sein, den Überblick zu behalten, in welchem Bereich Sie arbeiten. Visual Studio Code bietet jetzt eine Funktion, um einen besseren Überblick über Ihren Code zu haben: **Sticky Scroll**

- Öffnen Sie die Einstellungen unter `File > Preferences > Settings` oder über `Strg + ,`
- Tippen Sie `Sticky Scroll`
- Aktivieren Sie das Kästchen für `Editor > Sticky Scroll: Enabled`

Passen Sie die Einstellungen an

Mit Visual Studio Code können Sie verschiedene Einstellungen anpassen, z. B. die Schriftgröße, die Tabgröße und die Anzahl der für Einzüge verwendeten Leerzeichen. Sie können auf diese Einstellungen zugreifen, indem Sie zu `File > Preferences > Settings` gehen. Hier können Sie die Datei `settings.json` bearbeiten, die alle Ihre Visual Studio Code-Einstellungen enthält.

Hier finden Sie nützliche Einstellungen sowie

- `editor.renderWhitespace`: Rendern Sie Leerzeichen in Ihrem Code für eine bessere Sichtbarkeit.
- `editor.tabSize`: Passen Sie die Größe des Tabs an
- `editor.fontSize`: Passen Sie die Größe der Schriftart an

Settings Sync

Sie können Ihre Einstellungen, Tastenkombinationen und Extensions auf allen Geräten synchronisieren mit [Settings Sync](#). Melden Sie sich einfach mit Ihrem GitHub- oder Microsoft-Benutzer an.

Verwenden Sie Code-Navigationsfunktionen

Visual Studio Code enthält Features wie Go to Definition, Find All References und Peek Definition, die beim Navigieren durch Ihre Codebasis nützlich sein können.

FiraCode Schriftart

Installieren Sie die [FiraCode Schriftart](#) um Symbole, die mit mehreren Zeichen kodiert sind, wie `>=` oder `!=`, zu einem Zeichen anzeigen zu lassen.

Erstellen Sie Ihre eigenen Extensions

Visual Studio Code bietet eine Vielzahl von Extensions und es gibt fast für alles eine Extension. Aber manchmal stoßen Sie auf ein Problem, bei dem keine Extensions vorhanden ist, um dieses Problem zu lösen. Das könnte der Punkt sein, an dem Sie beginnen [Ihre eigenen Erweiterungen erstellen](#).


Tastenkombinationen für Visual Studio Code

Ein Schlüssel, um mit Visual Studio Code produktiver zu sein, ist die Verwendung der integrierten Tastenkombinationen. Sie können die vollständige Liste der Tastenkombinationen sehen, indem Sie zu **File > Preferences > Keyboard Shortcuts** gehen.

Hinweis: Einige dieser Tastenkombinationen können je nach verwendetem Betriebssystem oder Tastaturlayout (hier für deutsches Layout) abweichen.

Keyboard reference sheet

Hier sehen Sie fast alle Tastenkombinationen in einer Übersicht.



Visual Studio Code

Keyboard shortcuts for Windows

General

Ctrl+Shift+P, F1	Show Command Palette
Ctrl+P	Quick Open, Go to File...
Ctrl+Shift+N	New window/instance
Ctrl+Shift+W	Close window/instance
Ctrl+,	User Settings
Ctrl+K Ctrl+S	Keyboard Shortcuts

Basic editing

Ctrl+X	Cut line (empty selection)
Ctrl+C	Copy line (empty selection)
Alt+ ↑ / ↓	Move line up/down
Shift+Alt+ ↑ / ↓	Copy line up/down
Ctrl+Shift+K	Delete line
Ctrl+Enter	Insert line below
Ctrl+Shift+Enter	Insert line above
Ctrl+Shift+\	Jump to matching bracket
Ctrl+] / [Indent/outdent line
Home / End	Go to beginning/end of line
Ctrl+Home	Go to beginning of file
Ctrl+End	Go to end of file
Ctrl+↑ / ↓	Scroll line up/down
Alt+PgUp / PgDn	Scroll page up/down
Ctrl+Shift+[Fold (collapse) region
Ctrl+Shift+]	Unfold (uncollapse) region
Ctrl+K Ctrl+[Fold (collapse) all subregions
Ctrl+K Ctrl+]	Unfold (uncollapse) all subregions
Ctrl+K Ctrl+0	Fold (collapse) all regions
Ctrl+K Ctrl+J	Unfold (uncollapse) all regions
Ctrl+K Ctrl+C	Add line comment
Ctrl+K Ctrl+U	Remove line comment
Ctrl+/	Toggle line comment
Shift+Alt+A	Toggle block comment
Alt+Z	Toggle word wrap

Navigation

Ctrl+T	Show all Symbols
Ctrl+G	Go to Line...
Ctrl+P	Go to File...
Ctrl+Shift+O	Go to Symbol...
Ctrl+Shift+M	Show Problems panel
F8	Go to next error or warning
Shift+F8	Go to previous error or warning
Ctrl+Shift+Tab	Navigate editor group history
Alt+ ← / →	Go back / forward

Search and replace

Ctrl+F	Find
Ctrl+H	Replace
F3 / Shift+F3	Find next/previous
Alt+Enter	Select all occurrences of Find match
Ctrl+D	Add selection to next Find match
Ctrl+K Ctrl+D	Move last selection to next Find match
Alt+C / R / W	Toggle case-sensitive / regex / whole word

Multi-cursor and selection

Alt+Click	Insert cursor
Ctrl+Alt+ ↑ / ↓	Insert cursor above / below
Ctrl+U	Undo last cursor operation
Shift+Alt+I	Insert cursor at end of each line selected
Ctrl+L	Select current line
Ctrl+Shift+L	Select all occurrences of current selection
Ctrl+F2	Select all occurrences of current word
Shift+Alt+→	Expand selection
Shift+Alt+←	Shrink selection
Shift+Alt+ (drag mouse)	Column (box) selection
Ctrl+Shift+Alt+ (arrow key)	Column (box) selection
Ctrl+Shift+Alt+PgUp/PgDn	Column (box) selection page up/down

Rich languages editing

Ctrl+Space	Trigger suggestion
Ctrl+Shift+Space	Trigger parameter hints
Shift+Alt+F	Format document
Ctrl+K Ctrl+F	Format selection
F12	Go to Definition
Alt+F12	Peek Definition
Ctrl+K F12	Open Definition to the side
Ctrl+.	Quick Fix
Shift+F12	Show References
F2	Rename Symbol
Ctrl+K Ctrl+X	Trim trailing whitespace
Ctrl+K M	Change file language

Editor management

Ctrl+F4, Ctrl+W	Close editor
Ctrl+K F	Close folder
Ctrl+\	Split editor
Ctrl+ 1 / 2 / 3	Focus into 1 st , 2 nd or 3 rd editor group
Ctrl+K Ctrl+ - / -	Focus into previous/next editor group
Ctrl+Shift+PgUp / PgDn	Move editor left/right
Ctrl+K - / -	Move active editor group

File management

Ctrl+N	New File
Ctrl+O	Open File...
Ctrl+S	Save
Ctrl+Shift+S	Save As...
Ctrl+K S	Save All
Ctrl+F4	Close
Ctrl+K Ctrl+W	Close All
Ctrl+Shift+T	Reopen closed editor
Ctrl+K Enter	Keep preview mode editor open
Ctrl+Tab	Open next
Ctrl+Shift+Tab	Open previous
Ctrl+K P	Copy path of active file
Ctrl+K R	Reveal active file in Explorer
Ctrl+K O	Show active file in new window/instance

Display

F11	Toggle full screen
Shift+Alt+O	Toggle editor layout (horizontal/vertical)
Ctrl+ = / -	Zoom in/out
Ctrl+B	Toggle Sidebar visibility
Ctrl+Shift+E	Show Explorer / Toggle focus
Ctrl+Shift+F	Show Search
Ctrl+Shift+G	Show Source Control
Ctrl+Shift+D	Show Debug
Ctrl+Shift+X	Show Extensions
Ctrl+Shift+H	Replace in files
Ctrl+Shift+J	Toggle Search details
Ctrl+Shift+U	Show Output panel
Ctrl+Shift+V	Open Markdown preview
Ctrl+K V	Open Markdown preview to the side
Ctrl+K Z	Zen Mode (Esc Esc to exit)

Debug

F9	Toggle breakpoint
F5	Start/Continue
Shift+F5	Stop
F11 / Shift+F11	Step into/out
F10	Step over
Ctrl+K Ctrl+I	Show hover

Integrated terminal

Ctrl+`	Show integrated terminal
Ctrl+Shift+`	Create new terminal
Ctrl+C	Copy selection
Ctrl+V	Paste into active terminal
Ctrl+↑ / ↓	Scroll up/down
Shift+PgUp / PgDn	Scroll page up/down
Ctrl+Home / End	Scroll to top/bottom

Other operating systems' keyboard shortcuts and additional unassigned shortcuts available at aka.ms/vscodekeybindings

Für macOS und Linux.

Command Palette

Hier können Sie alle Befehle suchen und ausführen. Sie können hier auch einen Befehl einer Tastenkombination zuordnen.

- **Command Palette anzeigen:**
Windows/Linux: Strg + Shift + P
MacOS: Cmd + Shift + P
-

Ansicht

- **Neues Fenster/Instanz:**
Windows/Linux: Strg + Shift + N
MacOS: Cmd + Shift + N
- **Editor schließen:**
Windows/Linux: Strg + W
MacOS: Cmd + W
- **Fenster/Instanz schließen:**
Windows/Linux: Strg + Shift + W
MacOS: Cmd + Shift + W
- **Sidebar verstecken/anzeigen:**
Windows/Linux: Strg + B
MacOS: Cmd + B
- **Alle geöffneten Tabs schließen:**
Windows/Linux: Strg + K + W
MacOS: Cmd + K + W
- **Falten {}:**
Windows/Linux: Strg + Shift + ⌘
MacOS: Cmd + Shift + ⌘
- **Entfalten {}:**
Windows/Linux: Strg + Shift + `
MacOS: Cmd + Shift + `

- **Geteilte Editoransicht:**

Windows/Linux: Strg + ^

MacOS: Cmd + ^

- **Zen Mode:**

Windows/Linux: Strg + K + Z

MacOS: Cmd + K + Z

- **Markdown Vorschau:**

Windows/Linux: Strg + K + V

MacOS: Cmd + K + V

Navigation

- **Schnell eine Datei zum Workspace öffnen:**

Windows/Linux: Strg + P

MacOS: Cmd + P

- **Nächster Editor:**

Windows/Linux: Strg + Tab

MacOS: Cmd + Tab

- **Vorheriger Editor:**

Windows/Linux: Strg + Shift + Tab

MacOS: Cmd + Shift + Tab

- **Geh zurück:**

Windows/Linux: Alt + Linke Pfeiltaste

MacOS: Strg + Cmd + Linke Pfeiltaste

- **Vorwärts gehen:**

Windows/Linux: Alt + Rechte Pfeiltaste

MacOS: Strg + Cmd + Rechte Pfeiltaste

- **Globale Suche:**

Windows/Linux: Strg + Shift + F

MacOS: Cmd + Shift + F

- **Struktur der Datei anzeigen:**

Windows/Linux: Strg + Shift + .

MacOS: Cmd + Shift + .

- **Gehe zu einer bestimmten Zeile:**

Windows/Linux: Strg + G

MacOS: Cmd + G

Bearbeiten

- **Zeile entfernen und kopieren:**

Windows/Linux: Strg + X

MacOS: Cmd + X

- **Zeile kopieren:**

Windows/Linux: Strg + C

MacOS: Cmd + C

- **Hervorheben:**

Windows/Linux: Shift + Linke/Rechte Pfeiltaste

MacOS: Shift + Linke/Rechte Pfeiltaste

- **Wort hervorheben:**

Windows/Linux: Strg + Shift + Linke/Rechte Pfeiltaste

MacOS: Cmd + Shift + Linke/Rechte Pfeiltaste

- **Wort hervorheben und mehrere Cursor auf die nächsten gleichen Wörter setzen:**

Windows/Linux: Strg + D

MacOS: Cmd + D

- **Alle Vorkommen der aktuellen Auswahl auswählen:**

Windows/Linux: Strg + Shift + L

MacOS: Cmd + Shift + L

- **Ausschnitt einfügen:**

Windows/Linux: Shift + Alt + J

MacOS: Shift + Option + J

- **Zeile kopieren:**

Windows/Linux: Shift + Alt + Hoch/Runter Pfeiltaste

MacOS: Shift + Option + Hoch/Runter Pfeiltaste

- **Cursor oben/unten hinzufügen:**

Windows/Linux: Strg + Alt + Hoch/Runter Pfeiltaste

MacOS: Strg + Option + Hoch/Runter Pfeiltaste

- **Letzten Cursorvorgang rückgängig machen:**

Windows/Linux: Strg + U

MacOS: Cmd + U

- **Zeile auskommentieren:**

Windows/Linux: Strg + #

MacOS: Cmd + #

- **Hervorgehobenen Bereich auskommentieren:**

Windows/Linux: Shift + Alt + A

MacOS: Shift + Option + A

- **Mehrzeiliger Cursor:**

Windows/Linux: Strg + Alt + Hoch/Runter Pfeiltaste oder Alt gedrückt halten und KLICKEN

MacOS: Cmd + Option + Hoch/Runter Pfeiltaste oder Option gedrückt halten und KLICKEN

- **Aktuelle Zeile auswählen:**

Windows/Linux: Strg + L

MacOS: Cmd + L

- **Zeile nach oben/unten verschieben:**

Windows/Linux: Alt + Hoch/Runter Pfeiltaste

MacOS: Option + Hoch/Runter Pfeiltaste

- **Zeile löschen:**

Windows/Linux: Strg + Shift + K

MacOS: Cmd + Shift + K

- **Zeile unten einfügen:**

Windows/Linux: Strg + Enter

MacOS: Cmd + Enter

- **Zeile oben einfügen:**

Windows/Linux: Strg + Shift + Enter

MacOS: Cmd + Shift + Enter

- **Text formatieren:**

Windows/Linux: Shift + Alt + F

MacOS: Shift + Option + F

Die wichtigsten Visual Studio Code Extensions

Visual Studio Code ist ein leistungsstarker Texteditor, der jedoch mit den richtigen Extensions noch leistungsfähiger werden kann. Hier ist eine Liste einiger wichtiger Extensions, die Ihnen helfen können, Ihre Produktivität zu steigern und Ihren Arbeitsablauf zu optimieren.

- **Auto Complete Tag** - Extension Pakete zum automatischen Hinzufügen von Schließen-Tags und Umbenennen von gepaarten Tags für HTML/XML.
- **Auto Import** - Eine Extension, die fehlende Module während der Eingabe automatisch importiert, wodurch Sie Zeit sparen und die Wahrscheinlichkeit von Fehlern verringern.
- **Better Comments** - Verbessern Sie Ihre Codekommentierung, indem Sie mit Warnungen, Informationen, TODOs und mehr kommentieren!
- **Bookmarks** - Linien markieren und zu ihnen springen.
- **ChatGPT: write and improve code using AI** - Verwenden Sie ChatGPT direkt in der IDE, um Ihre Codierung mit KI-gestützter Unterstützung zu verbessern und zu automatisieren (inoffiziell).
- **Code Runner** - Eine Extension, mit der Sie Ihren Code direkt aus VSCode ausführen können, mit Unterstützung für mehrere Sprachen und automatischer Ausgabeformatierung.
- **Color Picker** - Eine Extension, die es einfach macht, Farben in Ihrem Code auszuwählen und zu verwenden, mit einer Farbauswahl und einer Palette gängiger Farben.
- **Debugger for Chrome** - Eine Extension, mit der Sie Ihren JavaScript-Code direkt in Chrome debuggen können, mit vollständiger Unterstützung für Haltepunkte, Aufruflisten und mehr.
- **Docker** - Vereinfacht das Erstellen, Verwalten und Debuggen von containerisierten Anwendungen.
- **ESLint** - Eine Extension, die Ihnen hilft, konsistenteren und fehlerfreien Code zu schreiben, indem sie Ihren Code auf potenzielle Probleme durchsucht.
- **GitLens** - Eine Extension, die VSCode leistungsstarke Git-Funktionen hinzufügt, einschließlich Visualisierungen von Änderungen, Schuldanmerkungen und mehr.
- **IntelliCode** - Die Visual Studio IntelliCode-Extension bietet KI-unterstützte Entwicklungsfunktionen für Python-, TypeScript/JavaScript- und Java-Entwickler in Visual Studio Code mit Erkenntnissen, die auf dem Verständnis Ihres Codekontexts in Kombination mit maschinellem Lernen basieren.

- **IntelliCode API Usage Examples** - Sehen Sie sich relevante Codebeispiele von GitHub für über 100.000 verschiedene APIs direkt in Ihrem Editor an.
- **Jupyter** - Jupyter-Notebook Unterstützung, interaktives Programmieren und Computing, das Intellisense, Debugging und mehr unterstützt.
- **Live Server** - Eine Extension, mit der Sie einen lokalen Webserver ausführen und eine Vorschau Ihrer HTML-, CSS- und JavaScript-Änderungen in Echtzeit anzeigen können.
- **Markdown Preview Enhanced** - Markdown Preview Enhanced portiert nach vscode.
- **Material Icon Theme** - Eine Extension, die die Standardsymbole im Explorer in ein moderneres und minimalistischeres Design umwandelt.
- **Multiple cursor case preserve** - Behält die Groß-/Kleinschreibung bei der Bearbeitung mit mehreren Cursors bei.
- **open in browser** - Dadurch können Sie die aktuelle Datei in Ihrem Standardbrowser oder Ihrer Standardanwendung öffnen.
- **Prettier** - Eine Extension, die Ihren Code automatisch so formatiert, dass er einem einheitlichen Stil entspricht.
- **SonarLint** - SonarLint ist eine IDE-Erweiterung, die Ihnen hilft, Qualitätsprobleme zu erkennen und zu beheben, während Sie Code in C, C, Java, JavaScript, PHP, Python, HTML und TypeScript schreiben.
- **SQLTools** - Verbinden von Benutzern mit vielen der am häufigsten verwendeten Datenbanken. Willkommen bei der richtigen Datenbankverwaltung.
- **Thunder Client** - Leichtgewichtiger-Rest-API-Client für VSCode.
- **Turbo Console Log** - Automatisches Schreiben aussagekräftiger Logmeldungen.
- **XML Tools** - XML-Formatierung, XQuery- und XPath-Tools für Visual Studio Code.

Dies sind nur einige Beispiele für die vielen nützlichen Extensions, die für Visual Studio Code verfügbar sind. Experimentieren Sie mit verschiedenen Extensions, um diejenigen zu finden, die für Sie und Ihren Workflow am besten geeignet sind.

Schauen Sie sich auch gerne meine Visual Studio Code Extensions an:

- **OwnVscodeExtension** - Bereitstellung von Werkzeugen für Visual Studio Code.
- **OwnGitExtension** - Eine Visual Studio Code Extension, die Werkzeuge für die Arbeit mit GitHub-Projekten bietet.
- **OwnObjectScriptExtension** - Eine Visual Studio Code Extension, die Werkzeuge für InterSystems ObjectScript bereitstellt.

[GitHub](#)

Von [Philipp B.](#)

powered by [InterSystems.](#)

2023