



VSCode-ObjectScript-HowTo

An introduction to [Microsoft's Visual Studio Code](#) and how to use Visual Studio Code with [InterSystems ObjectScript](#).

Overview

- [Getting Started with Visual Studio Code](#)
 - [Install Visual Studio Code](#)
 - [Opening a project](#)
 - [The Visual Studio Code interface](#)
 - [Writing and editing code](#)
 - [Workspaces](#)
 - [Settings](#)
 - [Color Theme](#)
 - [Debugging and testing](#)
 - [Source Control](#)
- [Using Visual Studio Code with InterSystems ObjectScript](#)
 - [Install InterSystems Extensions](#)
 - [Set ObjectScript Color Theme](#)
 - [Connect to a Server](#)
 - [Store Credentials](#)
 - [Open a Namespace](#)
 - [Writing ObjectScript Code](#)
 - [Use SQLTools](#)
- [Tips for Using Visual Studio Code](#)
- [Visual Studio Code Keyboard Shortcuts](#)
 - [Keyboard reference sheet](#)
 - [Command Palette](#)
 - [View](#)
 - [Navigation](#)
 - [Edit](#)
- [Essential Visual Studio Code Extensions](#)



Getting Started with Visual Studio Code

Visual Studio Code (VSCode) is a lightweight yet powerful source code editor developed by Microsoft for Windows, Linux, and macOS. It comes with built-in support for debugging, source control, and development tasks such as linting and building. In this guide, we will walk through the basics of using Visual Studio Code to write, debug, and manage your code.

Install Visual Studio Code

Before you can start using Visual Studio Code, you need to [download and install](#) the appropriate version for your operating system. Once the installation is complete, you can launch Visual Studio Code by double-clicking the icon on your desktop or by searching for “Visual Studio Code” in the start menu (on Windows) or spotlight (on macOS).

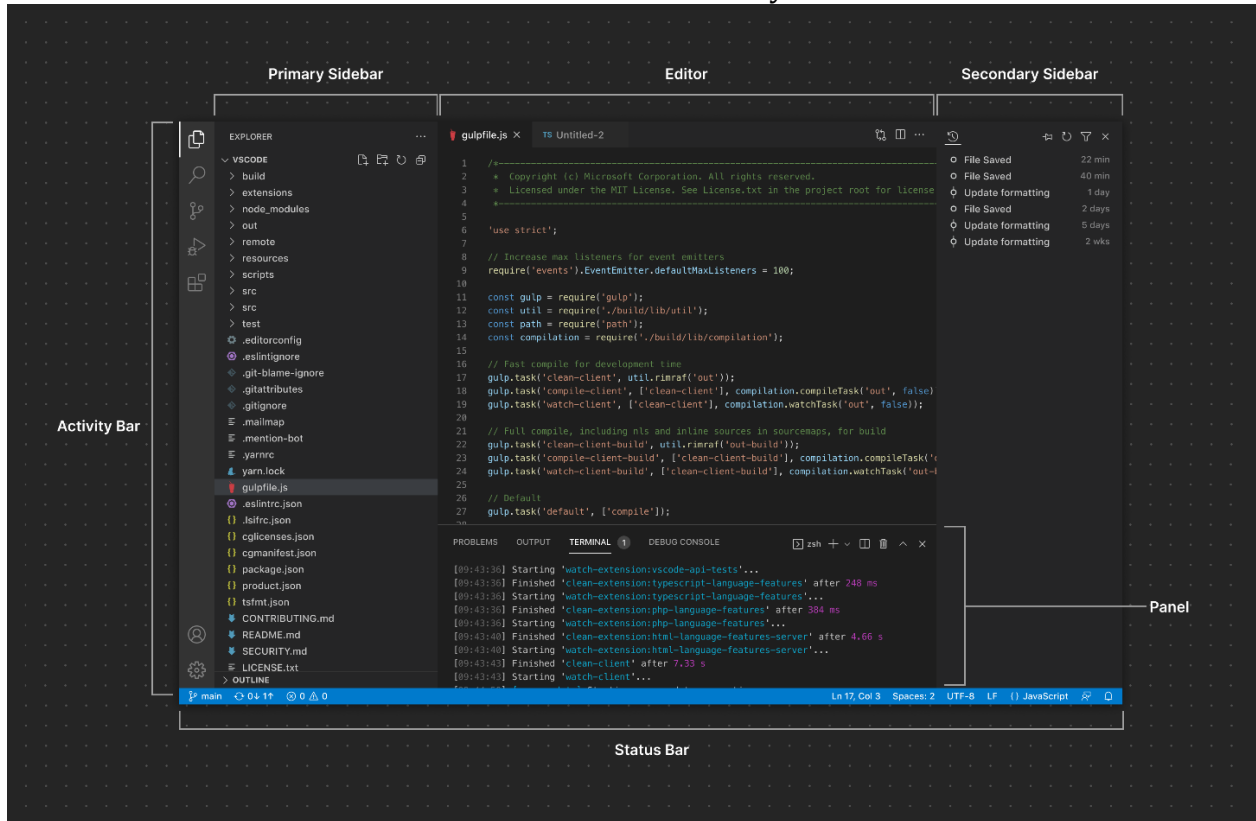
Opening a project

When you first launch Visual Studio Code, you will be presented with an *Open Folder* dialog. You can use this to open an existing project, or you can create a new one by clicking **File > New Folder** and then selecting the new folder. Once you have a folder open, you can start adding and editing files within that folder.



The Visual Studio Code interface

The Visual Studio Code interface is divided into a few key areas:



Activity Bar

The Activity Bar is a core navigation surface in Visual Studio Code located on the left side of the window, provides access to various views and features such as the file explorer and debug console.

Primary Sidebar

Will open the content of a Activity Bar Item.

Secondary Sidebar

A secondary sidebar. You can drag items from other containers to here, to see more at a time.

Editor

Here you will spend most of your time, writing and editing code.

Panel

The Panel is an other area to view containers.



Status Bar

Located at the bottom of the window, provides information and shortcuts for common actions.

Writing and editing code

Visual Studio Code has many powerful features for writing and editing code, including:

- **Syntax highlighting:** Visual Studio Code automatically recognizes the type of code you're working with and applies color coding to make it easier to read and write.
 - **IntelliSense:** Visual Studio Code provides intelligent code completion and suggestions for many languages, making it easier to write code quickly and correctly.
 - **Code snippets:** Visual Studio Code includes a large collection of code snippets that can be inserted into your code with a few keystrokes, saving you time and effort.
-

Workspaces

Select `File > Open File / Open Folder` to open a file or a folder as a workspace.

Add to current Workspace

Select `File > Add Folder to Workspace...` then select a folder to add to your current Workspace.

Save Workspace

So you don't have to open all your folders in your workspace everytime you open Visual Studio Code, you can save your Workspace as a file. Select `File > Save Workspace As...` and choose a destination and name for your workspace file. Afterwards a `.code-workspace` file was created. You can then open the workspace by just open the file or via `File > Open Workspace from File...`

Settings

Open Settings via `File > Preferences > Settings`.

User Settings

When editing settings in the *User* tab all settings apply globally across all Visual Studio Code Workspaces.



Workspace Settings

When editing settings in the *Workspace* tap all settings applied only effects the current opened workspace.

Color Theme

You can customize the appearance of the Visual Studio Code interface to your needs. Navigate to *File > Preferences > Color Theme* there you can choose from a preinstalled Color Theme or install a new Color Theme with *+ Browse Additional Color Themes...*

Debugging and testing

Visual Studio Code includes built-in support for debugging many languages, including JavaScript, TypeScript, and Python. To start debugging, you need to configure a “launch.json” file, which tells Visual Studio Code how to start and attach to your application. Once you have a launch configuration set up, you can start debugging by clicking the play button in the debug view or by pressing F5.

Source Control

Visual Studio Code support source control systems like Git, SVN, and TFS. The source control icon will appear on the left sidebar once you open a folder that is a git repository, allowing you to see the changes, make commits and handle merge conflicts, among other functionalities.


These are just a few of the many features that Visual Studio Code has to offer. With its powerful editing capabilities, built-in support for debugging and testing, and integration with source control, Visual Studio Code is a great choice for a wide range of development tasks. As you become more familiar with the tool, you’ll find that it can help you work more efficiently and effectively.




Using Visual Studio Code with InterSystems ObjectScript

InterSystems ObjectScript is a programming language that is used in InterSystems databases such as InterSystems IRIS and Caché. Visual Studio Code (VSCode) is a popular text editor that can be used to write and edit ObjectScript code.

Install InterSystems Extensions

Go to the Activity Bar on the far left-hand side and click on the **Extensions**  tap.

Type in the search bar *InterSystems* and then install the [InterSystems ObjectScript Extension Pack](#) . This Extension pack includes:


- [InterSystems ObjectScript](#): Adds InterSystems ObjectScript language support.
 - [InterSystems Language Server](#): Adds InterSystems ObjectScript language server.
 - [InterSystems Server Manager](#): Define connections to InterSystems servers. Browse and manage those servers.
-

Set ObjectScript Color Theme

To provide correct color highlighting when working with Objectscript choose a InterSystems Color Theme.

- Navigate to *File > Preferences > Color Theme*
 - Choose *InterSystems Default Dark* or *InterSystems Default Light*.
-

Connect to a Server

- Open the **InterSystems Tools**  tap in the Activity Bar.
- You can see default configurations under **ALL Servers**
- If you want to create a new Server connection click on the **+** button in the top and then fill in the connection properties:
 - *Name of new server definition*: Give your Connection a name.
 - *Optional Description*: A optional description for your server. Tap **Enter** to leave empty.
 - *Hostname or IP address of web server*: The Hostname or IP Address of your server.
 - *Port of web server*: The Port of the web server.
 - *Username*: The Username of your Iris user you want to connect from.



- Confirm connection type: *http* or *https* connection type.

After successfully adding a new Server you can see the Server under **All Servers**.

Tip: You can edit your Server Properties under: ... in the top right corner > Edit Settings > InterSystems: Servers > Edit in settings.json.

Store Credentials

Everytime you close Visual Studio Code and reopen it, it will prompt you a message to log in with your credentials. To avoid this you can store your credentials:

- Store Username:
 - In the **Command Palette** type *Preferences: Open User Settings (JSON)*. Here you can see your server connection in a JSON format.
 - Choose the server you want to add your username and add a property called *username*

Example:


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

- Store password
 - Next time you open Visual Studio code, it will prompt you a dialog and ask for a password
 - Type in your password
 - Click on the *key* button in the top right corner of the dialog to store the password
 - Delete password
 - Navigate to *Accounts* in the Activity Bar
 - Select your user
 - Click on **Sign Out**
 - select **Delete** when asked to delete password
-




Open a Namespace

Open a Namespace via InterSystems Tools

- Open the `InterSystems Tools`  tap in the Activity Bar.
- Choose a Server and click on it.
 - If the connection is successfully you will see a new directory named *Namespaces*.
 - If the connection failed you will see *Unavailable at <current-time>*.
- Choose a namespace.
 - Click on the *eye-icon* next to your namespace to view the namespace in your workspace.
 - Click on the *pen-icon* next to your namespace to edit the namespace in your workspace.

Open a Namespace via the Explorer

- Open the `Explorer` tap in the Activity Bar.
 - Do a right-click and choose `Add Server Namespace to Workspace`.
 - Now choose your Server or create a new one with the  in the top right corner of the dialog.
 - Choose a Namespace.
-

Writing ObjectScript Code


Once you have connected to a Namespace, you can begin writing ObjectScript code. Here are the basic steps for doing so:

- Create a new file with the `.cls` file extension (for example, `MyPackage.MyClass.cls`)
- Write your ObjectScript code in this file.
- The class will be compiled after you saved the file.

You can also create other files with the extension `.mac`, `.inc` etc.

Use SQLTools

When you want to have a look inside your database without leaving Visual Studio Code, you can use the SQLTools extension.

- Install the [SQLTools](#) extension
- Install the [SQLTools InterSystems IRIS](#)
- Navigate to the SQLTools  tap in the Activity Bar and choose `Add New Connection`
- Choose *InterSystems IRIS* and fill out the connection properties
- Choose `Connect Now` and you will see a new connection under *Connections*



- Now you can create a new SQL file and run the Query on your connection with **Run on active connection**

OR

- Browse through your tables and views, by unfolding the connection, and then view the table or view by clicking on the magnifyingglasses icon.



Tips for Using Visual Studio Code

Here are a few tips to help you get the most out of Visual Studio Code.

Editor Playground

If you just getting started with Visual Studio Code you can try the *Editor Playground* under `Help > Editor Playground`. Here you learn how to use some of the powerful [code editing features](#).

Use the Command Palette

The Command Palette is a powerful tool that allows you to access almost any feature or command in Visual Studio Code quickly and easily. Here you also can look up quickly the keybindings in case you forgot one. To open the Command Palette, press `Ctrl + Shift + P` (Windows/Linux) or `Cmd + Shift + P` (macOS).

Use the Built-in Terminal

Visual Studio Code includes a built-in terminal that you can use to run command-line tools, such as git and npm. To open the terminal, press `Ctrl + Shift + `` (Windows/Linux) or `Cmd + Shift + `` (macOS).

Use Code Snippets

Visual Studio Code includes a wide variety of built-in code snippets that can save you time when writing code. For example, you can type “for” and then press `Ctrl + Space` (Windows/Linux) or `Cmd + Space` (macOS) to quickly create a for loop. You can also create your own custom code snippets by going to `File > Preferences > Configure User Snippets`.

Customize the Theme

Visual Studio Code includes several built-in themes that you can use to customize the look and feel of the editor. You can change the theme by going to `File > Preferences > Color Theme`. To find more themes you can use the marketplace.

Use Extensions

Visual Studio Code has a vast ecosystem of [extensions](#) that add new features and functionality to the editor. You can browse and install extensions by going to `File > Extensions`. View [Essential visual Studio Code Extensions](#).



Learn the Keyboard Shortcuts

Visual Studio Code has a large number of keyboard shortcuts that can help you work faster and more efficiently. You can view the full list of keyboard shortcuts by going to `File > Preferences > Keyboard Shortcuts`, where you also can customize the keyboard shortcuts. View [Visual Studio Code Keyboard Shortcuts](#).

Format on Save

Its not always easy to keep your code in a good readable condition manually. Thankfully Visual Studio Code provides a formatter for almost every language. To format your file at the same time when you save your file, you can enable the `Format On Save` feature:

- Open the Settings under `File > Preferences > Settings` or via `Ctrl + ,`
- Type `Format On Save`
- Check the checkbox for `Editor: Format On Save`

Sticky Scroll

When working in code with long classes and methods that stretch beyond the vertical size of your screen, it can be difficult to keep track of which scope you're working in. You may be editing a long method or exploring an unfamiliar codebase. Visual Studio Code now offers a feature to have a better overview of your code: **Sticky Scroll**

- Open the Settings under `File > Preferences > Settings` or via `Ctrl + ,`
- Type `Sticky Scroll`
- Check the checkbox for `Editor > Sticky Scroll: Enabled`

Customize the settings

Visual Studio Code allows you to customize various settings, such as the font size, tab size, and the number of spaces used for indentation. You can access these settings by going to `File > Preferences > Settings` (or `Code > Preferences > Settings` on a Mac). Here, you can edit the `settings.json` file, which contains all of your Visual Studio Code settings.

Here you can find useful settings as well as

- `editor.renderWhitespace`: render whitespaces in your code for better visibility.
- `editor.tabSize`: customize the size of the tab
- `editor.fontSize`: customise the size of the font

Settings Sync

You can sync your settings, keybindings and extensions across all devices with [Settings Sync](#). Just log in with your GitHub or Microsoft user.



Use code navigation features

Visual Studio Code includes features such as Go to Definition, Find All References, and Peek Definition, which can be useful for navigating through your codebase.

FiraCode Font

Install the [FiraCode Font](#) to replace symbols that are encoded with several characters, such as `>=` or `!=`, to one symbol.

Create your own Extensions

Visual Studio Code offers a wide variety of extensions and there is almost a extension for everthing. But sometimes you run into a problem where no extension exists to solve this problem. That could be the point to start [creating your own extensions](#).




Visual Studio Code Keyboard Shortcuts

A key to be more productive with Visual Studio Code is to use the builtin keyboard shortcuts. You can view the full list of keyboard shortcuts by going to **File > Preferences > Keyboard Shortcuts**.

Note: Some of these shortcuts may differ based on the specific operating system or keyboard layout you're using (here for german layout).

Keyboard reference sheet

Here you can view almost all keyboard shortcuts in one overview.



Visual Studio Code

Keyboard shortcuts for Windows

General	
Ctrl+Shift+P, FI	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
FB	Go to next error or warning
Shift+FB	Go to previous error or warning
Ctrl+Shift+Tab	Navigate editor group history
Alt+ - / -	Go back/ forward
Ctrl+M	Toggle Tab moves focus
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+V	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

For macOS and Linux.



Command Palette

Here you can search and run all Commands. You can also map a command to a Keyboard shortcut here.

- **Show Command Palette:**
Windows/Linux: Ctrl + Shift + P
MacOS: Cmd + Shift + P
-

View

- **New window/instance:**
Windows/Linux: Ctrl + Shift + N
MacOS: Cmd + Shift + N
- **Close editor:**
Windows/Linux: Ctrl + W
MacOS: Cmd + W
- **Close window/instance:**
Windows/Linux: Ctrl + Shift + W
MacOS: Cmd + Shift + W
- **Toggle Sidebar:**
Windows/Linux: Ctrl + B
MacOS: Cmd + B
- **Close all open tabs:**
Windows/Linux: Ctrl + K + W
MacOS: Cmd + K + W
- **Fold {}:**
Windows/Linux: Ctrl + Shift + F
MacOS: Cmd + Shift + F



- **Unfold {}:**
Windows/Linux: Ctrl + Shift + `
MacOS: Cmd + Shift + `
 - **Split editor view:**
Windows/Linux: Ctrl + ^
MacOS: Cmd + ^
 - **Zen Mode:**
Windows/Linux: Ctrl + K + Z
MacOS: Cmd + K + Z
 - **Markdown Preview:**
Windows/Linux: Ctrl + K + V
MacOS: Cmd + K + V
-

Navigation

- **Quick Open file from Workspace:**
Windows/Linux: Ctrl + P
MacOS: Cmd + P
- **Next editor:**
Windows/Linux: Ctrl + Tab
MacOS: Cmd + Tab
- **Previous editor:**
Windows/Linux: Ctrl + Shift + Tab
MacOS: Cmd + Shift + Tab
- **Go back:**
Windows/Linux: Alt + Left Arrow
MacOS: Ctrl + Cmd + Left Arrow



- **Go forward:**
Windows/Linux: Alt + Right Arrow
MacOS: Ctrl + Cmd + Right Arrow
 - **Global Find:**
Windows/Linux: Ctrl + Shift + F
MacOS: Cmd + Shift + F
 - **Show outline of file:**
Windows/Linux: Ctrl + Shift + .
MacOS: Cmd + Shift + .
 - **Go to specific Line:**
Windows/Linux: Ctrl + G
MacOS: Cmd + G
-

Edit

- **Cut Line:**
Windows/Linux: Ctrl + X
MacOS: Cmd + X
- **Copy Line:**
Windows/Linux: Ctrl + C
MacOS: Cmd + C
- **Highlight:**
Windows/Linux: Shift + Left/Right Arrow
MacOS: Shift + Left/Right Arrow
- **Highlight word:**
Windows/Linux: Ctrl + Shift + Left/Right Arrow
MacOS: Cmd + Shift + Left/Right Arrow



- **Highlight word and put multiple cursor to next same words:**

Windows/Linux: Ctrl + D

MacOS: Cmd + D

- **Select all occurrences of current selection:**

Windows/Linux: Ctrl + Shift + L

MacOS: Cmd + Shift + L

- **Insert Snippet:**

Windows/Linux: Shift + Alt + J

MacOS: Shift + Option + J

- **Copy Line:**

Windows/Linux: Shift + Alt + Up/Down Arrow

MacOS: Shift + Option + Up/Down Arrow

- **Add cursor above/below:**

Windows/Linux: Ctrl + Alt + Up/Down

MacOS: Ctrl + Option + Up/Down

- **Undo last cursor operation:**

Windows/Linux: Ctrl + U

MacOS: Cmd + U

- **Comment line:**

Windows/Linux: Ctrl + #

MacOS: Cmd + #

- **Comment highlighted array out:**

Windows/Linux: Shift + Alt + A

MacOS: Shift + Option + A

- **Multiple Line cursor:**

Windows/Linux: Ctrl + Alt + Up/Down Arrow or hold Alt and CLICK

MacOS: Cmd + Option + Up/Down Arrow or hold Option and CLICK



- **Select current Line:**

Windows/Linux: Ctrl + L

MacOS: Cmd + L

- **Move line up/down:**

Windows/Linux: Alt + Up/Down Arrow

MacOS: Option + Up/Down Arrow

- **Delete line:**

Windows/Linux: Ctrl + Shift + K

MacOS: Cmd + Shift + K

- **Insert line below:**

Windows/Linux: Ctrl + Enter

MacOS: Cmd + Enter

- **Insert line above:**

Windows/Linux: Ctrl + Shift + Enter

MacOS: Cmd + Shift + Enter

- **Format Text:**

Windows/Linux: Shift + Alt + F

MacOS: Shift + Option + F



Essential Visual Studio Code Extensions

Visual Studio Code is a powerful text editor, but it can be even more powerful with the right extensions. Here's a list of some essential extensions that can help you boost your productivity and streamline your workflow.

- **Auto Complete Tag** - Extension Packs to add close tag and rename paired tag automatically for HTML/XML.
- **Auto Import** - An extension that automatically imports missing modules as you type, saving you time and reducing the chance of errors.
- **Better Comments** - Improve your code commenting by annotating with alert, informational, TODOs, and more!
- **Bookmarks** - Mark lines and jump to them.
- **ChatGPT: write and improve code using AI** - Use ChatGPT right inside the IDE to enhance and automate your coding with AI-powered assistance (unofficial).
- **Code Runner** - An extension that allows you to run your code directly from VSCode, with support for multiple languages and automatic output formatting.
- **Color Picker** - An extension that makes it easy to pick and use colors in your code, with a color picker and palette of common colors.
- **Debugger for Chrome** - An extension that allows you to debug your JavaScript code directly in Chrome, with full support for breakpoints, call stacks, and more.
- **Docker** - Makes it easy to create, manage, and debug containerized applications.
- **ESLint** - An extension that helps you write more consistent and error-free code by linting your code for potential issues.
- **GitLens** - An extension that adds powerful Git functionality to VSCode, including visualizations of changes, blame annotations, and more.
- **IntelliCode** - The Visual Studio IntelliCode extension provides AI-assisted development features for Python, TypeScript/JavaScript and Java developers in Visual Studio Code, with insights based on understanding your code context combined with machine learning.
- **IntelliCode API Usage Examples** - See relevant code examples from GitHub for over 100K different APIs right in your editor.
- **Jupyter** - Jupyter notebook support, interactive programming and computing that supports Intellisense, debugging and more.



- **Live Server** - An extension that allows you to run a local web server and preview your HTML, CSS, and JavaScript changes in real-time.
- **Markdown Preview Enhanced** - Markdown Preview Enhanced ported to vscode.
- **Material Icon Theme** - An extension that changes the default icons in the explorer to a more modern and minimalistic design.
- **Multiple cursor case preserve** - Preserves case when editing with multiple cursors.
- **open in browser** - This allows you to open the current file in your default browser or application.
- **Prettier** - An extension that automatically formats your code to conform to a consistent style.
- **SonarLint** - SonarLint is an IDE extension that helps you detect and fix quality issues as you write code in C, C++, Java, JavaScript, PHP, Python, HTML and TypeScript.
- **SQLTools** - Connecting users to many of the most commonly used databases. Welcome to database management done right.
- **Thunder Client** - Lightweight Rest API Client for VS Code.
- **Turbo Console Log** - Automating the process of writing meaningful log messages.
- **XML Tools** - XML Formatting, XQuery, and XPath Tools for Visual Studio Code.

These are just a few examples of the many useful extensions available for Visual Studio Code. Experiment with different extensions to find the ones that work best for you and your workflow.

Check out my Visual Studio Code Extensions:

- **OwnVscodeExtension** - Providing tools for Visual Studio Code.
- **OwnGitExtension** - A Visual Studio Code Extension that offers tools for working with GitHub projects.
- **OwnObjectScriptExtension** - A Visual Studio code extension that supplies tools for InterSystems ObjectScript.



GitHub

by Philipp B.

powered by InterSystems.

2023

