VS CODE

Các chức năng built-in & Extension cần thiết

Open Settings

Shift+Ctrl+P 🡪 gõ ‘setting’ 🡪 chọn:

* + Open Settings (UI): change settings with UI
  + Open Settings (JSON): change settings with file settings.json

Built-in

Open màn hình Settings: Ctrl+,

* Emmet in VS Code <https://code.visualstudio.com/docs/editor/emmet>

Emmet is a plugin for many popular text editors which greatly improves HTML & CSS workflow:

**Emmet commands: F1 – gõ emmet**

* + **Bao ngoài vị trí con trỏ với Element nào đó: Wrap With Abbreviation**
  + **Remove tag đang chọn (ko xóa nội dung bên trong): Remove Tag**
  + **Chọn cặp đóng-mở tag: Balance (outward)**
  + **Chỉ Chọn nội dung bên trong tag: Balance (inward)**
  + **Di chuyển tới tag mở/đóng của tag đang chọn: Go to Matching Pair**
  + **Gom về 1 dòng những dòng đang chọn: Merge Lines**

**Syntax editor**

* + Lorem: tạo văn bản demo. Lorem100 – tao 100 từ
  + Child: >

You can use > operator to nest elements inside each other:

div>ul>liHTML to CSS autocompletionax

* + Sibling: +

Use + operator to place elements near each other, on the same level:

div+p+bq

* + Climb-up: ^

With ^ operator, you can climb one level up the tree and change context where following elements should appear:

div+div>p>span+em^bq

You can use as many ^ operators as you like, each operator will move one level up:

div+div>p>span+em^^bq

* + Multiplication: \*

With \* operator you can define how many times element should be outputted:

ul>li\*5

* + Grouping: ()

Parenthesises are used by Emmets’ power users for grouping subtrees in complex abbreviations:

div>(header>ul>li\*2>a)+footer>p

You can nest groups inside each other and combine them with multiplication \* operator:

(div>dl>(dt+dd)\*3)+footer>p

* + ID and CLASS

In CSS, you use elem#id and elem.class notation to reach the elements with specified id or class attributes. In Emmet, you can use the very same syntax to *add* these attributes to specified element:

div#header+div.page+div#footer.class1.class2.class3

* + Custom attributes

You can use [attr] notation (as in CSS) to add custom attributes to your element:

td[title="Hello world!" colspan=3]

* + Item numbering: $

With multiplication \* operator you can repeat elements, but with $ you can *number* them. Place $ operator inside element’s name, attribute’s name or attribute’s value to output current number of repeated element:

ul>li.item$\*5

* + Text: {}

You can use curly braces to add text to element:

a{Click me}

a>{click}+b{here}

p>{Click }+a{here}+{ to continue}

* Settings nhắc code:
  + Inlay Hints: nhập “Inlay Hints” để định vị setting

Enable/check option tại giao diện hoặc insert trong file settings.json

* "javascript.inlayHints.parameterTypes.enabled": true
* "javascript.inlayHints.parameterNames.enabled": "all"
* User Snippets

Tự định nghĩa snippets cá nhân

* Bấm F1 (or menu File\Preferences\User Snippets)🡪 'Preferences: Configure User Snippets' 🡪 chọn option tạo file (.code-snippets) or edit file đã có
* Viết code snippet theo mẫu trong file hoặc vào trang <https://snippet-generator.app/> tạo code snippet, rồi copy – past vào

Các input trong trang <https://snippet-generator.app/>

* + - * + Description: môt tả ý nghĩa snippet
        + Tab trigger: định nghĩa các phím bấm call snippet
        + You snippet: tự viết hoặc copy code paste vào

Một số Ext extension cần thiết:

**Open settings của 1 extension cụ thể:**

* Open slidebar Extension: Ctrl+Shift+X
* Chọn extention, trong khung nội dung extention ở giữa: click icon setting hình bánh răng (cạnh icon Install)
* [GitHub Copilot extension](https://marketplace.visualstudio.com/items?itemName=GitHub.copilot) to power your AI suggestions in VS Code.
* Auto Rename Tag: auto rename tag khi edit 1 trong 2 tag đóng hoặc mở
* Bracket Pair Colorizer: định dạng, định vị từng cặp dấu ngoặc rõ ràng
* JavaScript Snippet Pack: https://marketplace.visualstudio.com/items?itemName=akamud.vscode-javascript-snippet-pack
* Turbo Console Log: hỗ trợ tạo lệnh console.log nhanh
  + - Insert meaningful log message automatically

Two steps:

* + - * Selecting the variable which is the subject of the debugging
      * Pressing ctrl + alt + L

Cho phép chọn nhiều chỗ cùng lúc

* + - Comment all log messages, inserted by the extension, from the current document: Alt+Shift+C.
    - UnComment all log messager inserted by the extension: Alt+Shift+U
    - Delete all log messager inserted by the extension: Alt+Shift+C
* Flutter Print - Help insert and remove print(\*) statement  
  Mặc định xuất ra lệnh log nên sau khi install xong, cần cấu hình như sau để xuất ra lệnh print:
  + - Tại màn hình thông tin extention click vào icon ❊ để vào setting, chọn Extension setting
    - Tại tab User trong phần Flutter print chọn hàm print, qua tab Workspace làm y như vậy

Cách dùng:

* + - Print object: quét chọn object muốn đưa vào lệnh print, bấm tổ hợp Ctrl + Shift + L
    - Delete all print command được tạo bởi extension: Shift + Win + D
* HTML to CSS autocompletion
* Code runner: cho quét chọn run test từng line/block code or nguyên file
  + - Setup: file settings.json
      * Set language: "code-runner.defaultLanguage": "javascript"
    - Run: quét chọn code, bấm Ctrl+Alt+N.

Kết quả xuất ra cửa sổ Output

* + - Clear output mỗi lần run (làm sạch output): insert "code-runner.clearPreviousOutput": true vào file C:\Users\nmson\AppData\Roaming\Code\User\settings.json (tùy biến path)
* Live Server Extention: extention auto update page on browser khi edit code
  + - Mở browser trước khi open live server
    - Canh đều ½ màn hình cửa sổ code và browser: Win + mũi tên trái/phải
    - Open/Close server: 1 trong các cách sau
      * Tại góc dưới phải cửa sổ VS Code, click icon Go Live để open server (sẽ biến thành icon Port... khi server đang open), click lần nữa để close server
      * Righ click trên list file (hoặc trong cửa sổ edit code), chọn Open with Live Server.
      * Shortcut: open: Alt + L, Alt + O; close: Alt + L, Alt + C
    - Settings: File->preference->Settings->Extensions->LiveServerConfig (Ctrl + ,),

Chọn các option…. hoặc click Edit in Settings.json.

* + - Một số settings thường dùng:
      * + Chọn browser chạy live server: "liveServer.settings.CustomBrowser": "chrome"
        + liveServer.settings.port: Customize Port Number of your Live Server. If you want random port number, set it as 0
        + liveServer.settings.root: To change root of server in between workspace folder structure, use / and absolute path from workspace.

Example: /sub\_folder1/sub\_folder2. Now sub\_folder2 will be root of the server.

Default value is "/".(The Workspace Root).

* MySQL

Database Client for Visual Studio Code. It supports databases MySQL/MariaDB, Microsoft SQL Server, PostgreSQL, SQLite, MongoDB, Redis, and ElasticSearch.

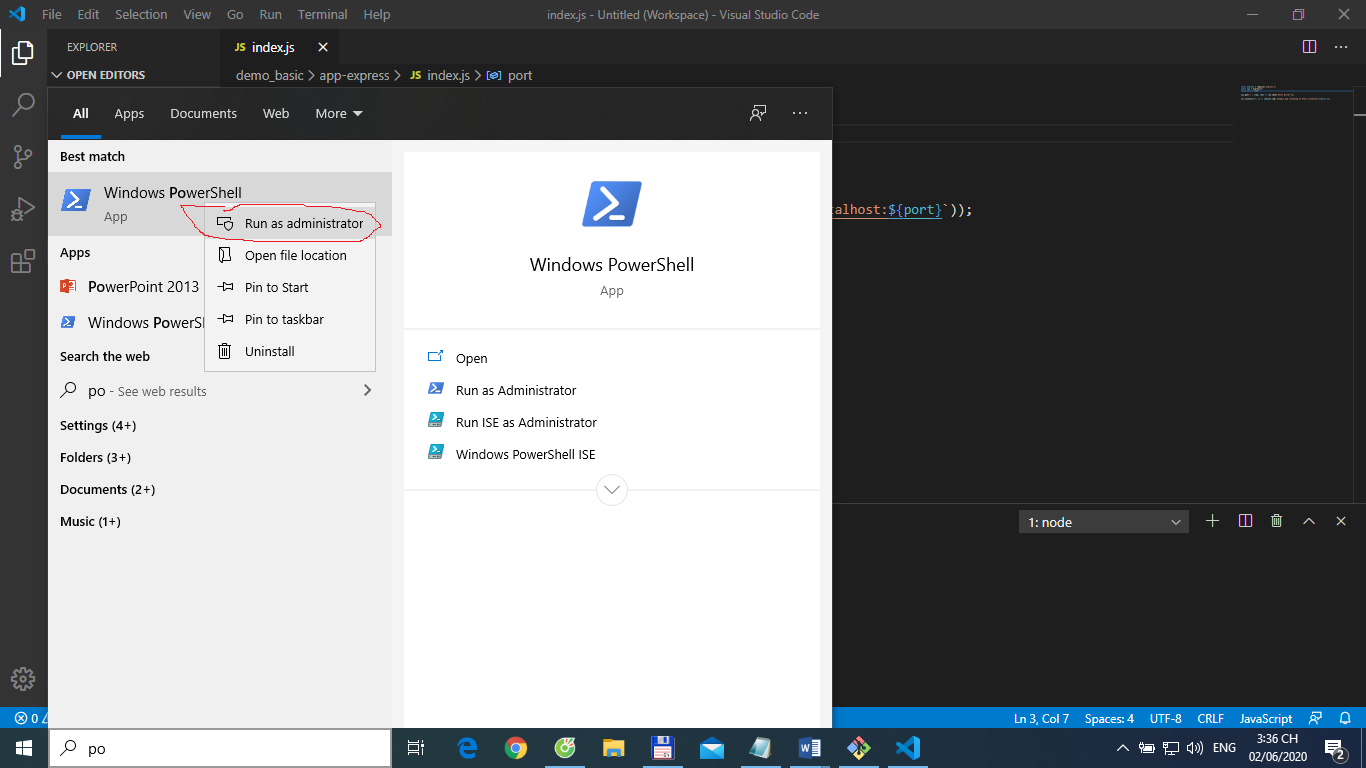
https://marketplace.visualstudio.com/items?itemName=cweijan.vscode-mysql-client2

Xử lý một số vấn đề phát sinh

Run nodemon:

default do bị giới hạn quyền truy cập nên để run nodemon in Visual Studio Code ta phải thực hiện gỡ giới hạn như sau

* Chạy Power Shell với quyền admin

****

* Tại cmd gõ lệnh:

Set-ExecutionPolicy -Scope LocalMachine Unrestricted

Copy code từ Word vào dính stt dòng

=> Vào menu Selection \ Column Selection Mode, rồi quét chọn theo cột để delete stt

# Basic Editing

[(tham](https://code.visualstudio.com/docs/editor/codebasics(tham) khảo https://code.visualstudio.com/docs/editor/codebasics  
or Help > Editor Playground.)

* + Bật terminal tại 1 đường dẫn nào đó: định vị folder trong khung File Explorer (Ctrl+Shift+E), Rigth click folder – Open in Integrated Tẻminal
  + Set path working cho terminal: dò tới setting Terminal › Integrated: Cwd, nhập path vào textbox or add dòng sau vào file setting.json

"terminal.integrated.cwd": "D:\\MY\\TEST\\CNWeb\_UD\\TEST\_API",

Tổ chức folder, file của project vào Workspace

Một Workspace có thể chứa 1 or nhiều folder chứa data của 1 or nhiều project

Khi save project vào workspace VSCode sẽ save thông tin vào file ten-workspace.code-workspace. Double click vào file .code-workspace này sẽ start VSCode thẳng vào workspace luôn

Một số lệnh liên quan tới workspace: từ menu File

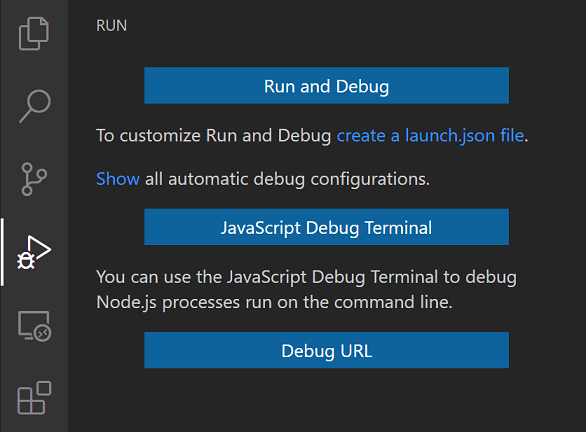
* Open workspace from file: mở workspace từ fie .code-workspace đã lưu
* Add folder to Workspace: add folder vào workspace đang làm việc
* Save Workspace as…: save workspace với tên mới
* Duplicate Workspace: copy workspace đang làm việc và mở 1 cửa sổ làm việc mới

Run and Debug – Ctrl+Shift+D: bật màn hình Run

VS Code has built-in debugging support for the [Node.js](https://nodejs.org/) runtime and can debug JavaScript, TypeScript, or any other language that gets transpiled to JavaScript.

For debugging other languages and runtimes (including [PHP](https://marketplace.visualstudio.com/items?itemName=felixfbecker.php-debug), [Ruby](https://marketplace.visualstudio.com/items?itemName=rebornix.Ruby), [Go](https://marketplace.visualstudio.com/items?itemName=golang.go), [C#](https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.csharp), [Python](https://marketplace.visualstudio.com/items?itemName=ms-python.python), [C++](https://marketplace.visualstudio.com/items?itemName=ms-vscode.cpptools), [PowerShell](https://marketplace.visualstudio.com/items?itemName=ms-vscode.PowerShell) and [many others](https://marketplace.visualstudio.com/search?term=debug&target=VSCode&category=Debuggers&sortBy=Relevance)), look for Debuggers [extensions](https://code.visualstudio.com/docs/editor/extension-marketplace) in the VS Code [Marketplace](https://marketplace.visualstudio.com/vscode/Debuggers) or select **Install Additional Debuggers** in the top-level Run menu.

If running and debugging is not yet configured (no launch.json has been created), VS Code shows the Run start view.



* Link create a launch.json fie tạo file launch.json tại folder .vscode trong root folder project
* Nếu mất file launch.json, có thể tạo manual tại folder .vscode đã nói ở trên. Dùng Ctrl+Spacebar để kích hoạt snippet nhập cấu hình debug.
* Nếu ko có file launch.json thì VSCode lưu cấu hình debug trong file .code-wordspace, mục '”launch”:{….}

Các bước setup debug:

* Start live server
* Mở sidebar Sources and control: Ctrl + Shift + G
* Tại combobox Launch … Click chọn Add Config theo 1 trong các option
  + - * Add Config theo folder: cấu hình lưu trong file launch.json
      * Add Config theo workspace: cấu hình lưu trong file .code-workspace
* Trong nội dung cấu hình edit 2 field sau:
* “name”: đặt tên dễ nhớ
* “url”: chèn port của live server, path chính xác tới file cần debug.

Nếu ko cần run live server thì set “url”: "file:///D:/MY/TEST/CNWeb\_UD/dong-ho.html" (path static)

 {

            "name": "Launch DeQui",

            "request": "launch",

            "type": "pwa-chrome",

            "url": "http://127.0.0.1:5500/TEST\_API/index2.html",

            "webRoot": "${workspaceFolder}"

        },

        {

            "name": "Launch Chrome JSON-server",

            "request": "launch",

            "type": "pwa-chrome",

            "url": "http://localhost:3000",

            "webRoot": "${workspaceFolder}"

        }

Fix lỗi ko debug được

* Với file run trực tiếp từ path local: fix url auto thành file path (double click để open file .html trực tiếp với browser, copy url trong address bar dán vào cấu hình url)

Vd: “url”: ": <http://localhost:8080> fix thành "url": "file:///D:/MY/TEST/CNWeb\_UD/dong-ho.html"

* Với live server local: fix port trong url trùng với port của live server.

Vd: "url": <http://localhost:8080> fix thành "url": <http://localhost:5500> với file index tại root folder

Vd: fix thành "url": <http://localhost:5500/dong-ho.html> - chỉ định file cần run

* Xuất thông tin bằng console.log: bật cửa sổ Debug Console bằng Ctrl+Shift+Y
* Debug actions

Debug Actions

* Continue / Pause F5
* Step Over F10
* Step Into F11
* Step Out Shift+F11
* Restart Ctrl+Shift+F5
* Stop Shift+F5
* Conditional breakpoints: Right click trên icon breckpoint

A powerful VS Code debugging feature is the ability to set conditions based on expressions, hit counts, or a combination of both.

* Expression condition: The breakpoint will be hit whenever the expression evaluates to true.

Vd: i==5; valueNumber > 10; a==b

* Hit count: The 'hit count' controls how many times a breakpoint needs to be hit before it will 'break'.

Vd: =5; <10; >20

* Log message: set message output in Debug console, chèn biến trong cặp {}

Vd: day la breakpoint: {i}

Lưu ý:

* + Hit count chỉ chạy chính xác lần Start Debugging(F5) đầu tiên, Restart (Ctrl+Shift+F5) sẽ ko còn chính xác (do giá trị biến count ko được reset lại) 🡪 phải Stop (Shift+F5) và Start (F5) lại
  + Mỗi breakpoint chỉ áp dụng được 1 trong 3 option trên tại cùng thời điểm. Thứ tự ưu tiên sẽ là 1)Expression condition 2)Log message 3)Hit count

Python debugging in VS Code

<https://code.visualstudio.com/docs/python/debugging>

* Cách 1: Run debug ko cần  debug configuration

Bấm F5 or menu Run 🡪 Start Debugging

* Cách 2: Create/Add  debug configuration vào file launch.json

Có 1 số cách khác nhau để tạo file launch.json (trong folder .vscode của project) sau:

Ctrl+Shift+D để vào Run and Debug hoặc menu Run 🡪 Add Configuration, tại listbox Debug Configuration chọn Python File….

Một số lỗi với vòng lặp

* + Dùng for.. of với object bị lỗi: Uncaught TypeError: … is not iterable

=> dùng for…in

* => dùng method Object.keys để get list keys:

 let text = "";

    for (let key of Object.keys(person)) {

        text += person[key] + ' ';

    }

    console.log(text);

* => dùng method Object.values để get list value:

for (let value of Object.values(person)) //values có s

        console.log(value);

Luu y loai du lieu duoc duyet trên từng vòng lặp:

* For…in duyệt trên index với Array, trên key với Object
* For…of duyệt trên value
* forEach duyệt trên value

Đếm số properties của object

* Get so keys trong object: let n = Object.keys(ten\_object).length;
* Hoặcdùng method getOwnPropertyNames

 const propOwn = Object.getOwnPropertyNames(person).length;

Shortcut thường dùng

* + Tra cứu shortcut: Ctrl+K Ctrl+R (vào link https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf)
  + Format code: Shift + Alt + F
  + Rename variable: đặt con trỏ tại vị trí variable, F2 🡪 all variable cùng tên sẽ change theo
  + Select current line: Ctrl+L
  + Delete line: Ctrl+Shift+K
  + Toggle Line Comment: Ctrl+/
  + Toggle Block Comment: Shift+Alt+A (quét chọn khối code, ko lấy /\* và \*/)
  + Toggle Word Wrap: Alt+Z (mất tác dụng khi close VSCode)
  + Set Word Wrap default: Vào màn hình Setting (Ctrl+,), nhập vào khung search “word wrap”, định vị option Editor: Word Wrap, set giá trị = on
  + Go to Line...: Ctrl+G
  + Go Back/Forward: di chuyển con trỏ ngược về vị trí trước, tới vị trí sau Alt+Left/Right
  + Jump to matching bracket: chuyển con trỏ về ngoặc đóng/mở chung cặp. Ctrl+Shift+\
  + Move Line: Down: Alt+Down; Up: Alt+Up
  + Copy Line: Down: Shift+Alt+Down; Up: Shift+Alt+Up
  + Fold and unfold code: Ctrl+Shift+[; Ctrl+Shift+]
  + Tìm kiếm:
    - Search trong file đang edit: Ctrl+F; replace Ctrl+H
    - Seach trong all files đang open: Ctrl+Shift+F; replace Ctrl\_Shift+H
    - Show command palette: khung nhập lệnh tìm kiếm phía trên vùng editor: F1
      * Khi bấm F1 mặc định tiền tố là dấu >: tìm lệnh
      * Tiền tố ? cũng là tìm lệnh
      * Ctrl+P : tìm file trong project
      * Thêm tiền tố : để goto dòng code thứ x (Ctrl+G)
      * Thêm tiền tố @ để tìm object trong code (Tìm biến, hàm, object…trong file: Ctrl+Shift+O)
  + Multi Cursor Editing: nếu tổ hợp phím Ctrl+Alt xoay màn hình(xung đột) phải disable xoay màn hình trước: Right click desktop – Graphics Options – Hot keys
    - `Chọn các phần tử giống nhau trên nhiều dòng thẳng hàng nhau: : chọn vị trí hoặc phần tử đầu tiên, Ctrl + Alt + Up/Down để chọn các dòng khác, tùy chỉnh phần tử được chọn(Shift + mũi tên)
    - Chọn các phần tử hoặc vị trí bất kỳ: chọn vị trí đầu tiên, Ctrl + click chọn các vị trí tiếp theo, tùy chỉnh phần tử được chọn(Shift + mũi tên)
    - Chọn all phần tử giống nhau cùng lúc (vd thuộc tính ‘color’): chọn 1 phần tử bất kỳ, bấm tổ hợp Ctrl+Shift+L
  + Zoom code: Ctrl + phím +/-
  + Change language mode: Ctrl+K, M
  + Chuyển tab editor giữa các file đang mở: Ctrl + PgUp/PgDn hoặc Alt+ số thứ tự tab
  + Open cửa sổ Setting: Ctrl+,
  + Open file settings.json: các cách sau
    - C1: Ctrl+P 🡪 chọn file settings.json
    - C2: F1, xóa dấu > 🡪 chuyển qua C1
    - C3: Ctrl+, 🡪 click chọn icon Open settings.json ở góc trên phải

Tùy biến giao diện:

* + Show minimap (bản đồ show full nội dụng code ở khung cạnh phải): menu View – Show minimap
  + Toggle vùng Command line (Terminal, Ouput,…): Ctrl+J
  + Show/Hide Slidebar (khung cây folder/file bên trái): Ctrl+B
  + Chuyển đổi giữa các loại Slidebar
    - File Explorer: Ctrl + Shift + E
    - Extension: Ctrl + Shift + X
    - Search: Ctrl + Shift + F
    - Sources and control: Ctrl + Shift + G
    - Run and Debugger: Ctrl + Shift + D
  + Tách thêm 2, 3 cửa sổ code:Click icon Split Editor góc trên phải (Ctrl+\). Bấm Alt khi click sẽ tách trên dưới
    - Switch between editors: Ctrl+1, Ctrl+2, Ctrl+3 (Chuyển qua lại giữa các vùng Editor, 1 editor có thể mở nhiều tab cho nhiều file)
    - Chuyển qua lại giữa các tab (trên all editor): Ctrl+PgUp/PgDn
    - Chuyển qua lại giữa các tab trong cùng Editor: Alt+số

Working with Markdown files

* + Open Preview to the Side (mở cửa sổ preview và split editor song song): Ctrl+K, V (Ctrl+K, nhả hết 2 phím rồi mới bấm V)
  + Reference:
    - <https://markdown-it.github.io>
    - [File demo](file:///D:\MY\word\mark_down.md)
  + Một số thủ thuật:
    - Paragraphs: To create paragraphs, use a blank line to separate one or more lines of text. – cách 1 dòng trắng là tách paragraph mới. Để xuống dòng mà ko tách pagraraph thì gõ nối tiếp Tab và Enter

Xác định phạm vi paragraph: click chọn text trong cửa sổ Preview (file markdown) sẽ xuất hiện vạch kẻ đứng ở cạnh trái paragraph

Mỗi heading auto tách paragraph