

# Help for Visual Studio Code

Jim Fawcett

CSE686 – Internet Programming

Spring 2019

# Visual Studio Code (VS Code)

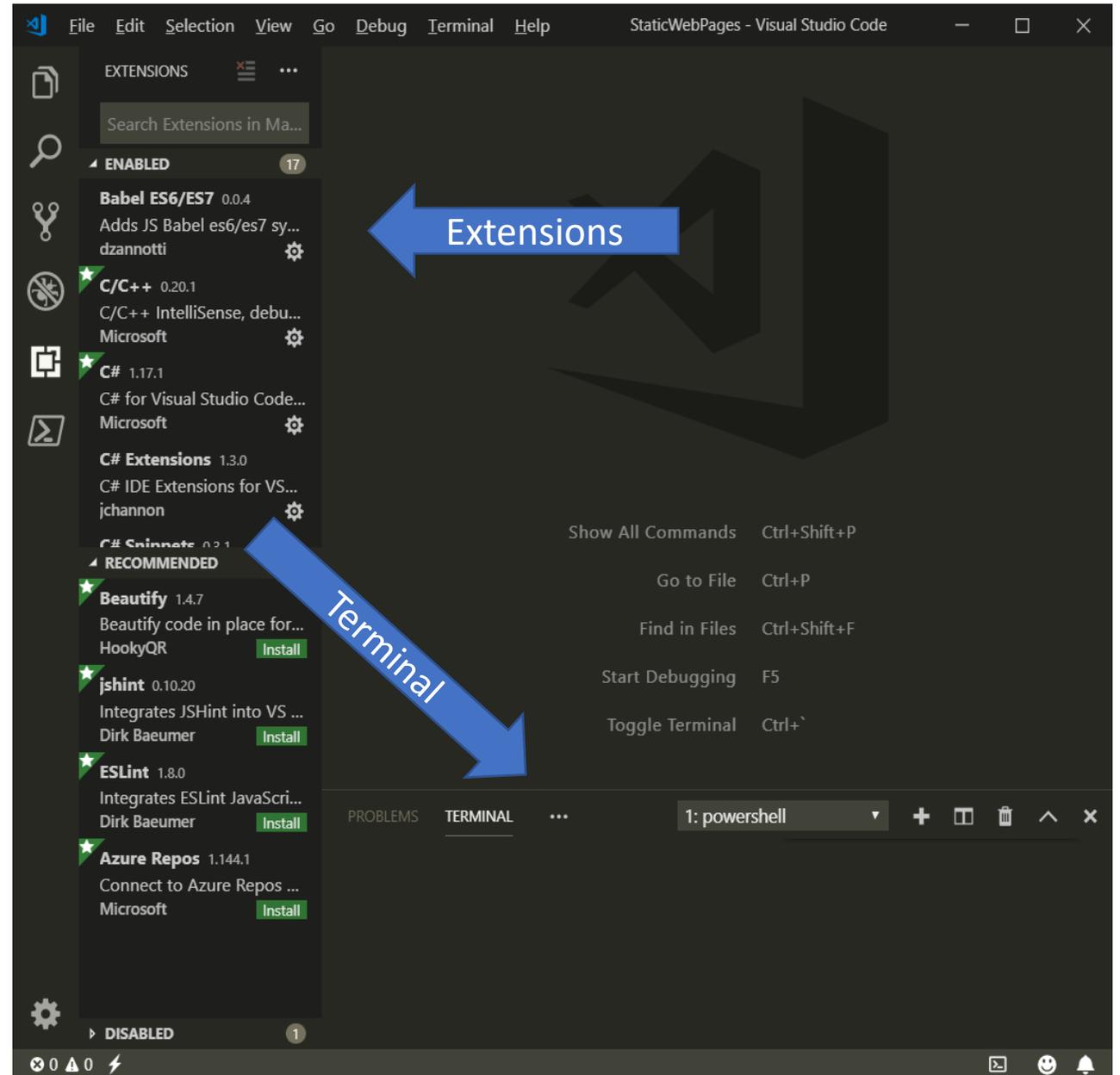
- **VS Code** is a text editor build with the Electron.js JavaScript framework.
- It is easy to use for building static web pages.
- In conjunction with the **dotnet** command, and a default C# plugin, it is a capable Integrated Development Environment (IDE) for working with Asp.Net Core applications.
  - Visual Studio has more facilities, and, once you learn how it works, it is a very effective tool for building static pages and working with Asp.Net Core projects.
  - VS Code will do everything we need for the first half of the course.

# Using VS Code

- The first thing to do is download the latest version – it's free. Just ask Dr. Google to take you to VS Code download.
- That will download an installer in your Download directory. Run it – that takes just a minute or so - then start up VS Code.

# VS Code Window

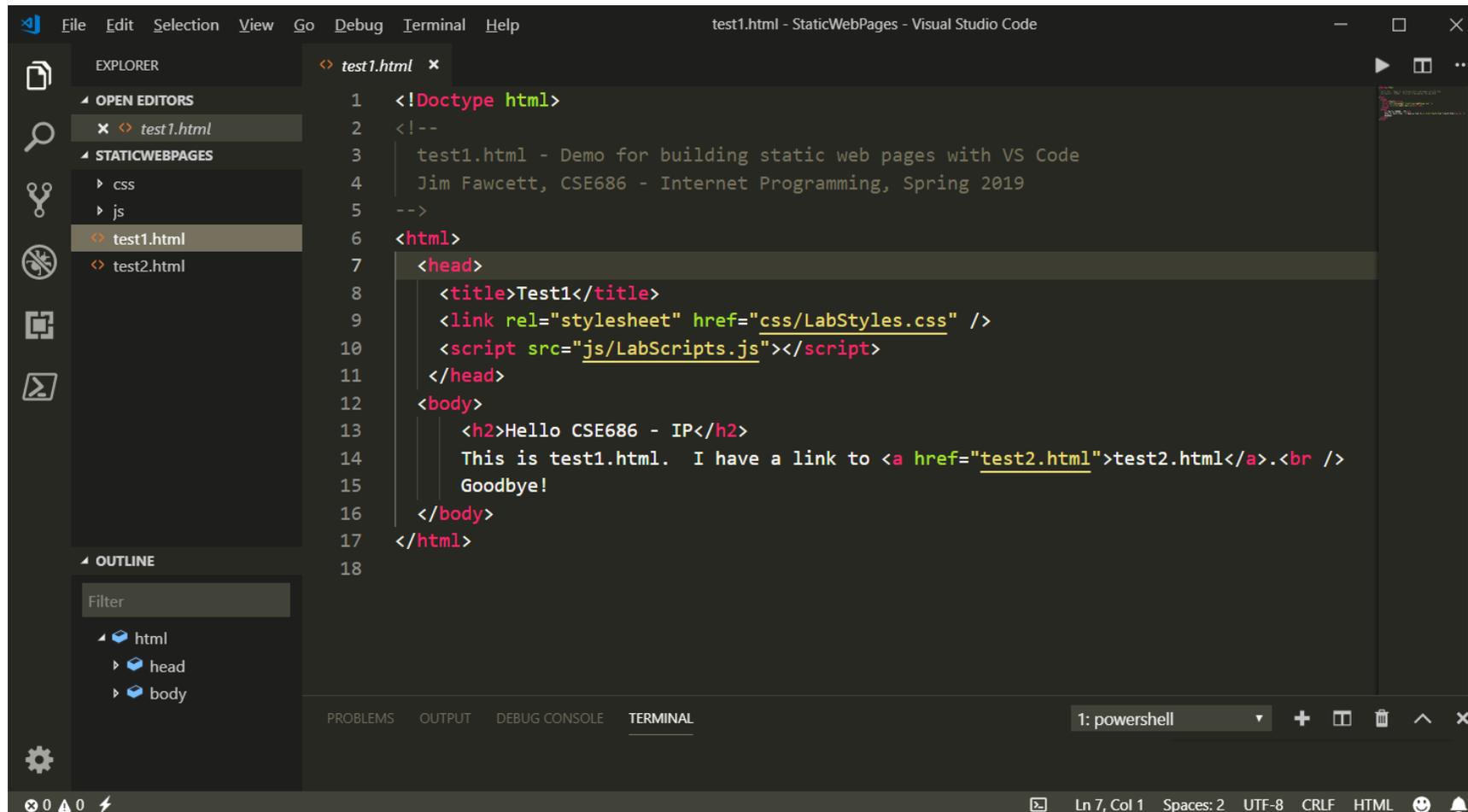
- In the left panel VS Code shows folder, search, Source Control, Debugging, Extensions, and a PowerShell command views.
- To start, you open the Folder view.
- VS Code doesn't have projects, but it does have settings for build and launch.



# Building Static Web Pages

- File > Open Folder > select work folder in dialog (or create)
- File > New File > Enter opens editor on an un-named text file
- Enter some small amount of text
- File > Save As > Enter File name (ext html, css, js) > Save
- Made sub folders css and js in work folder
  - File > Save As > right-click on dialog > new > folder > CSS
  - File > Save As > right-click on dialog > new > folder > JS
- Now you fill in the html, css, and js details.

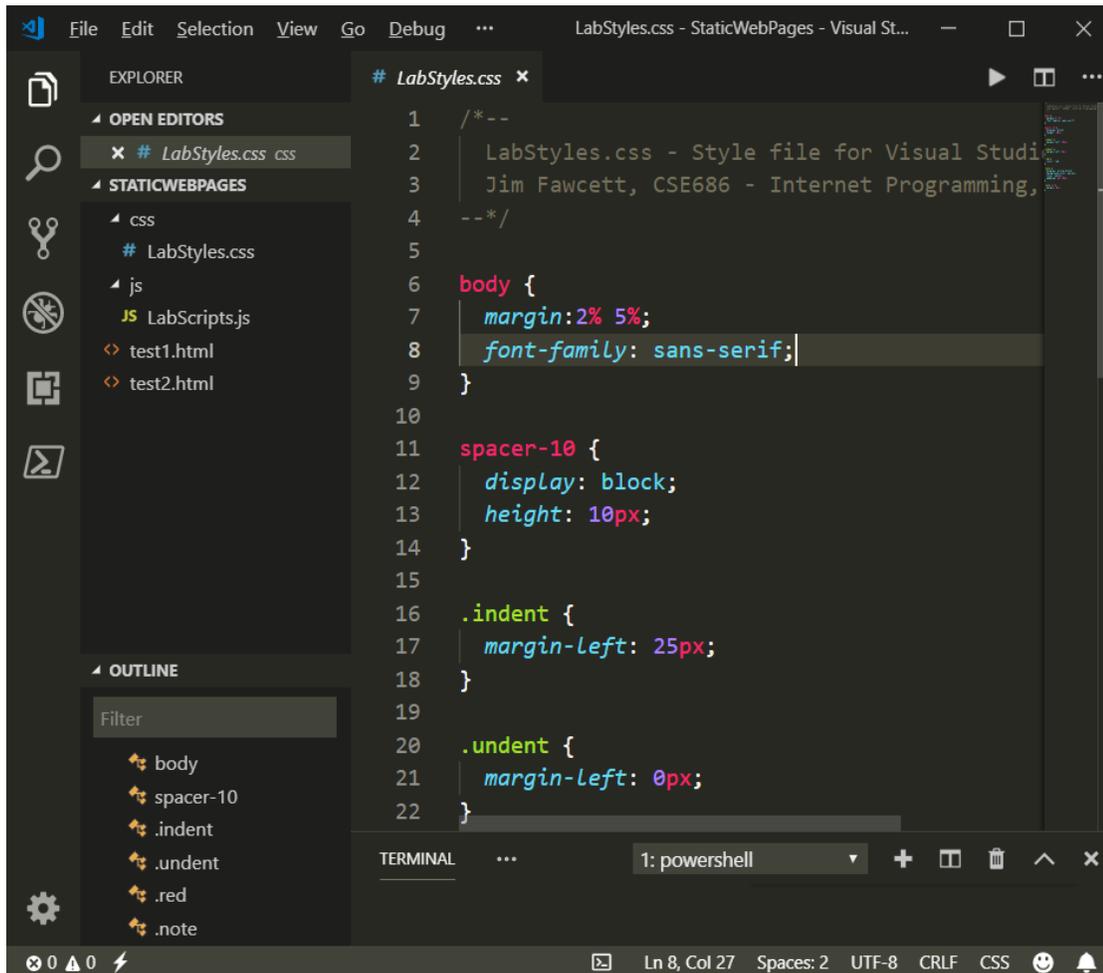
# Building Static Web Pages with VS Code



```
1 <!DOCTYPE html>
2 <!--
3   test1.html - Demo for building static web pages with VS Code
4   Jim Fawcett, CSE686 - Internet Programming, Spring 2019
5 -->
6 <html>
7   <head>
8     <title>Test1</title>
9     <link rel="stylesheet" href="css/LabStyles.css" />
10    <script src="js/LabScripts.js"></script>
11  </head>
12  <body>
13    <h2>Hello CSE686 - IP</h2>
14    This is test1.html. I have a link to <a href="test2.html">test2.html</a>.<br />
15    Goodbye!
16  </body>
17 </html>
18
```

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the project structure: 'STATICWEBPAGES' containing 'css', 'js', 'test1.html', and 'test2.html'. The Outline sidebar shows the document structure: 'html' containing 'head' and 'body'. The main editor area displays the HTML code for 'test1.html' with line numbers 1 through 18. The code includes a DOCTYPE declaration, a comment, a title, a link to a CSS file, a script tag, and a body with an h2 and a link to test2.html. The status bar at the bottom indicates 'Ln 7, Col 1', 'Spaces: 2', 'UTF-8', 'CRLF', and 'HTML'.

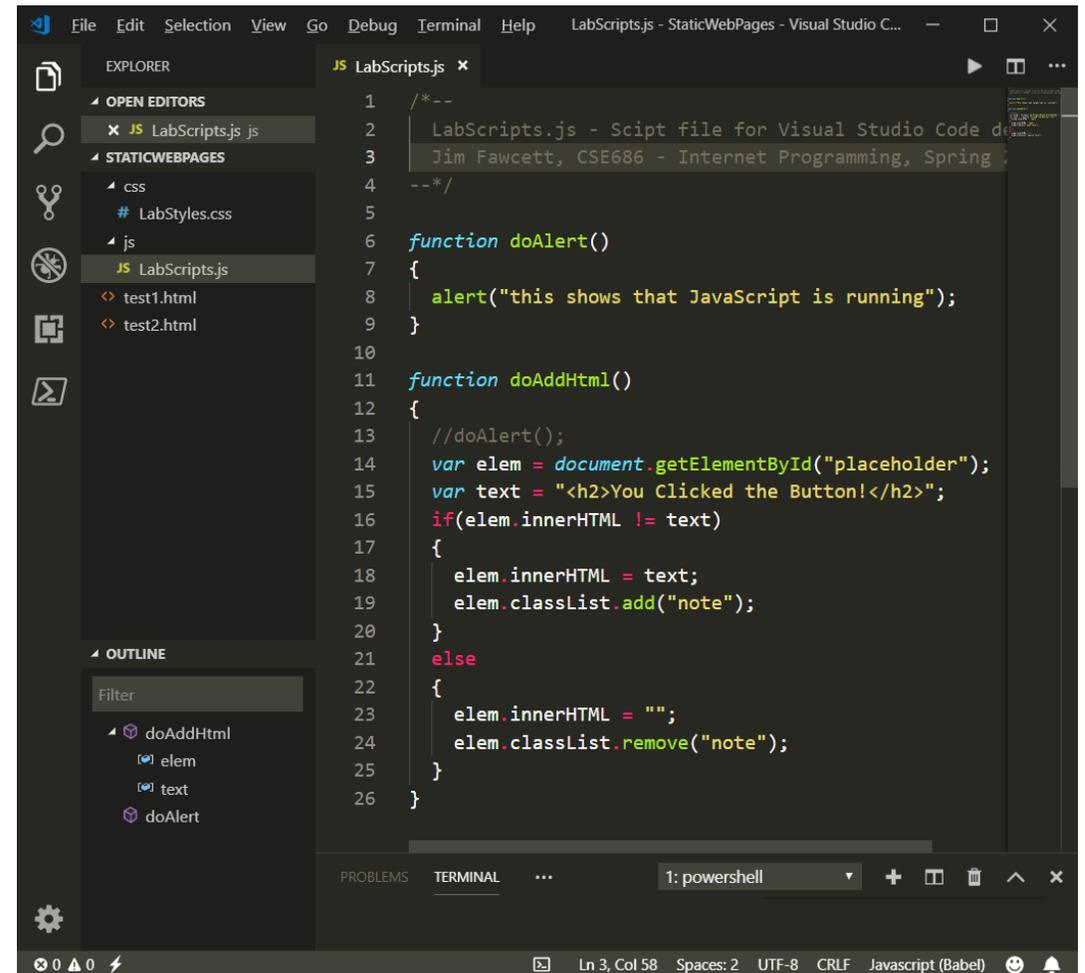
# Building Static Web Pages with VS Code



This screenshot shows the Visual Studio Code editor with the `LabStyles.css` file open. The Explorer sidebar on the left shows the project structure under `STATICWEBPAGES`, including `css` and `js` folders. The Outline sidebar shows the CSS selectors: `body`, `spacer-10`, `.indent`, `.indent`, `.red`, and `.note`. The main editor area displays the following CSS code:

```
1 /*--
2 LabStyles.css - Style file for Visual Studi
3 Jim Fawcett, CSE686 - Internet Programming,
4 --*/
5
6 body {
7   margin:2% 5%;
8   font-family: sans-serif;
9 }
10
11 spacer-10 {
12   display: block;
13   height: 10px;
14 }
15
16 .indent {
17   margin-left: 25px;
18 }
19
20 .indent {
21   margin-left: 0px;
22 }
```

The status bar at the bottom indicates the cursor is at line 8, column 27, with 2 spaces, UTF-8 encoding, CRLF line endings, and CSS mode.



This screenshot shows the Visual Studio Code editor with the `LabScripts.js` file open. The Explorer sidebar on the left shows the project structure under `STATICWEBPAGES`, including `css` and `js` folders. The Outline sidebar shows the JavaScript functions: `doAddHtml`, `elem`, `text`, and `doAlert`. The main editor area displays the following JavaScript code:

```
1 /*--
2 LabScripts.js - Script file for Visual Studio Code de
3 Jim Fawcett, CSE686 - Internet Programming, Spring
4 --*/
5
6 function doAlert()
7 {
8   alert("this shows that JavaScript is running");
9 }
10
11 function doAddHtml()
12 {
13   //doAlert();
14   var elem = document.getElementById("placeholder");
15   var text = "<h2>You Clicked the Button!</h2>";
16   if(elem.innerHTML != text)
17   {
18     elem.innerHTML = text;
19     elem.classList.add("note");
20   }
21   else
22   {
23     elem.innerHTML = "";
24     elem.classList.remove("note");
25   }
26 }
```

The status bar at the bottom indicates the cursor is at line 3, column 58, with 2 spaces, UTF-8 encoding, CRLF line endings, and Javascript (Babel) mode.

# Building Console Applications

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS: C:\su\CSE686\code\Asp.Net.Core
>cd c:\su\temp
PS: C:\su\temp
>mkdir demoVSCodeConsoleApp

Directory: C:\su\temp

Mode                LastWriteTime         Length Name
----                -
d-----          1/13/2019   4:51 PM                demoVSCodeConsoleApp

PS: C:\su\temp
>cd demo*
PS: C:\su\temp\demoVSCodeConsoleApp
>dotnet new console
The template "Console Application" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on C:\su\temp\demoVSCodeConsoleApp\demoVSCodeConsoleApp.csproj...
Restoring packages for C:\su\temp\demoVSCodeConsoleApp\demoVSCodeConsoleApp.csproj...
Generating MSBuild file C:\su\temp\demoVSCodeConsoleApp\obj\demoVSCodeConsoleApp.csproj.nuget.g.props.
Generating MSBuild file C:\su\temp\demoVSCodeConsoleApp\obj\demoVSCodeConsoleApp.csproj.nuget.g.targets.
Restore completed in 122.07 ms for C:\su\temp\demoVSCodeConsoleApp\demoVSCodeConsoleApp.csproj.

Restore succeeded.

PS: C:\su\temp\demoVSCodeConsoleApp
>ls

Directory: C:\su\temp\demoVSCodeConsoleApp

Mode                LastWriteTime         Length Name
----                -
d-----          1/13/2019   4:51 PM                obj
-a----          1/13/2019   4:51 PM             178 demoVSCodeConsoleApp.csproj
-a----          1/13/2019   4:51 PM             202 Program.cs

PS: C:\su\temp\demoVSCodeConsoleApp
>
```

```
launch.json - demoVSCodeConsoleApp - Visual Studio Code

EXPLORER
├── OPEN EDITORS
│   └── Program.cs
├── demoVSCodeConsoleApp
│   ├── .vscode
│   │   ├── launch.json
│   │   └── tasks.json
│   └── bin
│       └── Debug
│           ├── netcoreapp2.1
│           ├── demoVSCodeConsoleApp.deps.json
│           ├── demoVSCodeConsoleApp.dll
│           ├── demoVSCodeConsoleApp.pdb
│           ├── demoVSCodeConsoleApp.runtimeconf...
│           └── demoVSCodeConsoleApp.runtimeconf...
└── obj
    └── demoVSCodeConsoleApp.csproj
        └── Program.cs

1 {
2 // Use IntelliSense to learn about possible attributes.
3 // Hover to view descriptions of existing attributes.
4 // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830386
5 "version": "0.2.0",
6 "configurations": [
7
8     {
9         "name": ".NET Core Launch (console)",
10        "type": "coreclr",
11        "request": "launch",
12        "preLaunchTask": "build",
13        "program": "${workspaceFolder}/bin/Debug/netcoreapp2.1/demoVSCodeConsoleApp.dll",
14        "args": [],
15        "cwd": "${workspaceFolder}",
16        "stopAtEntry": false,
17        "console": "internalConsole"
18    },
19
20    {
21        "name": ".NET Core Attach",
22        "type": "coreclr",
23        "request": "attach",
24        "processId": "${command:pickProcess}"
25    }
26 ]
27 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

You may only use the Microsoft .NET Core Debugger (vsdbg) with Visual Studio Code, Visual Studio or Visual Studio for Mac software to help you develop and test your applications.

Hello World!

# So What?

- If we use Visual Studio Code:
  - VS Code can't create .net core projects
  - So we create a project folder, call dotnet new console (or whatever)
  - Open Visual Studio Code
  - Open the newly created folder
    - When you do this VS Code will ask if you want to create resources to work with that folder.
    - Select yes – that builds a json settings file to manage builds and debugging
  - Start editing files and running
- Note that it is dotnet runtime that does the launching and debugging, not VS Code.

That's all Folks!