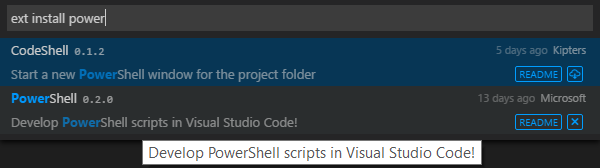
# PowerShell Extension

In the next two posts I am going to take some time to discuss the existing Visual Studio Code extensions in details. In this one we will discuss PowerShell Extension and in the next one – integration with Chrome Debugging service.

PowerShell extension is a very good example of how to use powerful Code editor not just for Web development but for many different tasks including IT Professional stuff. In order to install PowerShell extension, you can use Command Palette window and execute **Install Extension** command there:

C:\temp\@2.tif

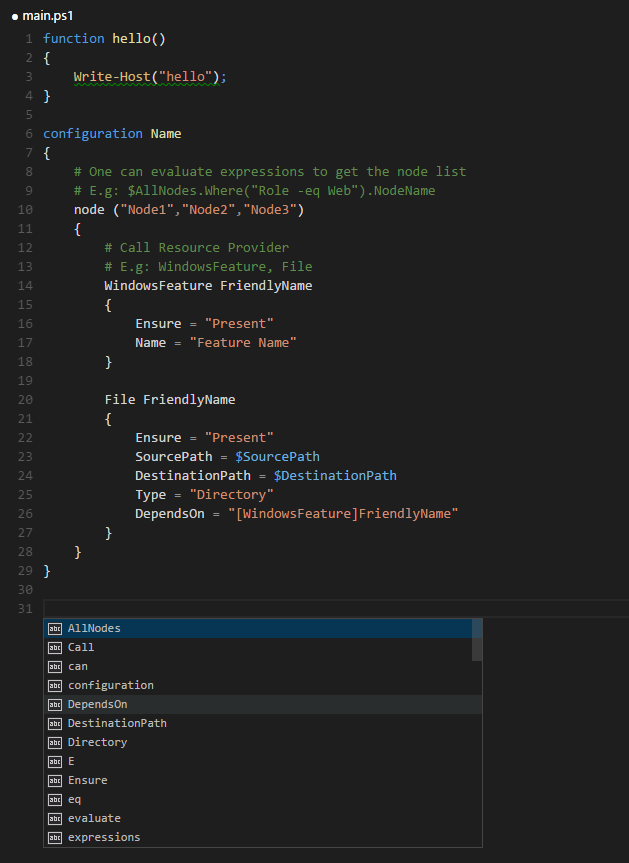
Right after that, Code will download a list of all extensions’ names and you can start typing PowerShell in order to make the list shorter and select the right one:



Once you install PowerShell extension, Visual Studio will propose to be restarted and right after the restart you can start working with PowerShell.

Of course, PowerShell is a platform for Windows but if you decide to develop some scripts on Mac you can use this PowerShell extension on Mac as well. You will not have access to some features there such as debugging but you can use at least the following things:

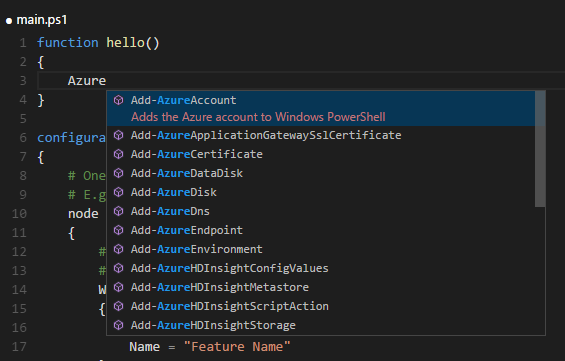
* **Syntax highlighting and code snippets** – Code will recognize all files with **.ps1** extension automatically and highlight different syntax constructions there. If you have non-standard extensions you can simply use the Status Bar to select the right file type (PowerShell, in our case). Editing the code you can activate the Code Snippets window and select a snippet from the list. I have found about 25 snippets now but the extension is open source and I think that some people will contribute new snippets soon;



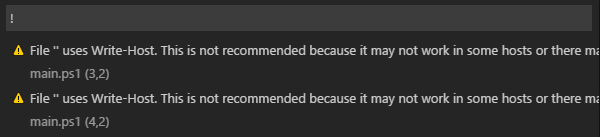
* **Basic IntelliSense support** – in case of Mac you don’t have great IntelliSense support because there is no way to get information about installed cmdlets etc. There is simply no PowerShell on Mac. But you can see that Code has basic IntelliSense support for some common constructions and for user-defined constructions in your project. It’s much better than nothing for Mac developers;
* **Go to Definition and Find References features**;
* **Integration with online help** – clicking Ctrl+F1 you can execute a browser that will use a word under your cursor in order to locate documentation on Microsoft site;

Of course, in case of Windows, PowerShell is native and you have access to much more features in Code. Thanks to open-source projects like [PowerShell Editor Services](https://github.com/PowerShell/PowerShellEditorServices) and [PowerShell Scrip Analyzer](https://github.com/PowerShell/PSScriptAnalyzer), extension’s developers got a chance to implement debugging features and code analyzer features. Here is the list of additional features that successfully work on Windows platform:

* **Full IntelliSense support** – Code provides information about all installed modules. So, you can easy get access to all cmdlets;



* **Rule-based analysis** – based on PowerShell Analyzer, Code provides information about mistakes and not very good practices in your code. Looking at the Status Bar you can find information about errors and warnings in real time and clicking this information you can read more about the problems;



* **Debugging and running scripts directly from Code** – if you are going just execute several lines of your code you can simply select them and click **F8** key. But it’s also possible to execute cmdlet using debugger. In order to do it, you need to create **launch.json** file clicking start button in Debug view. In my Code edition, I got just basic **launch.json** without any needed info inside. So, if you have a similar problem you can create **launch.json** manually. The file looks like this:

{

"version": "0.2.0",

"configurations": [

{

"name": "PowerShell",

"type": "PowerShell",

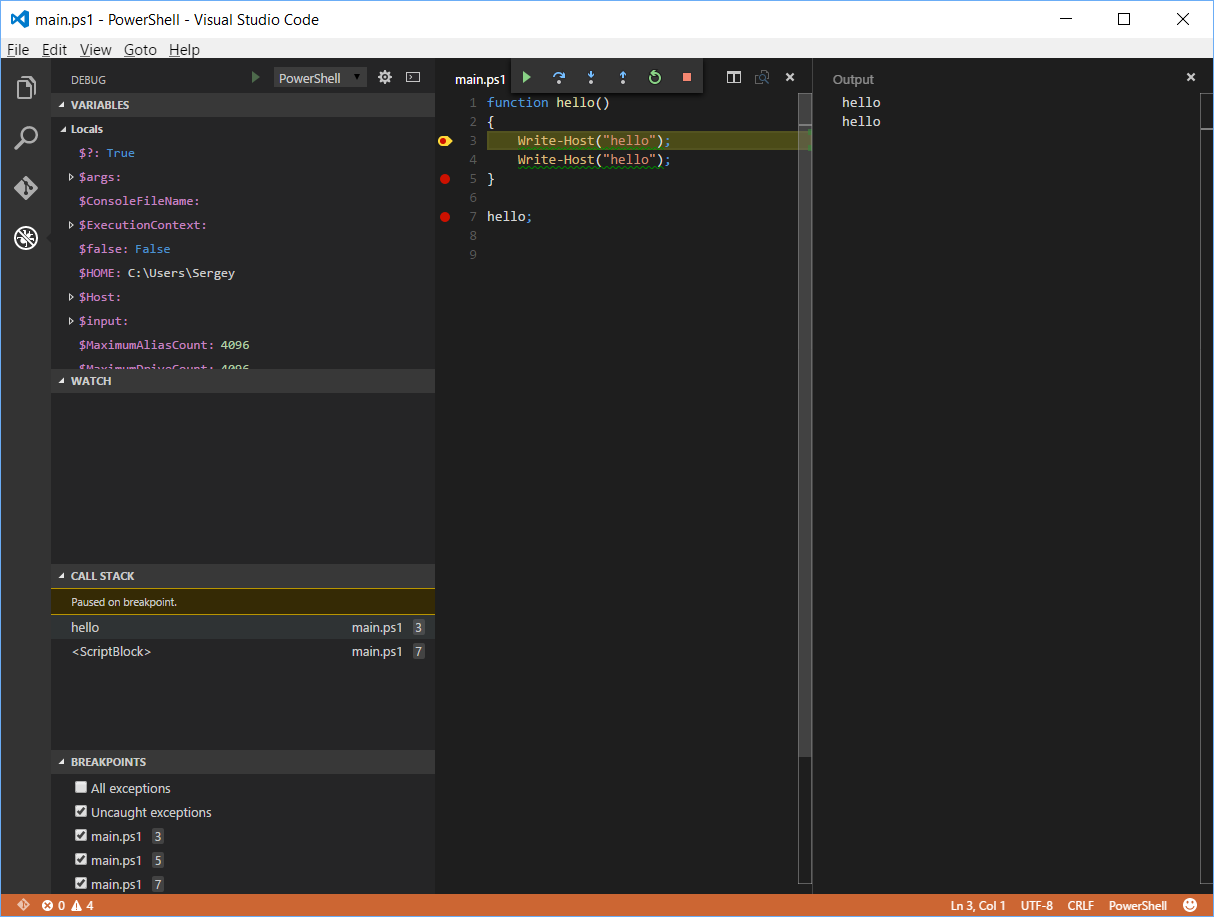
"program": "DebugTest.ps1"

}

]

}

Once you provide a file name with your cmdlets you can run debugger:



Therefore, you can easily use Code as a PowerShell editor with lots of features on Windows and even some features on Mac and Linux platform.

Finally, if you want to contribute or get access to code bits you can visit [GitHub site](https://github.com/PowerShell/vscode-powershell).