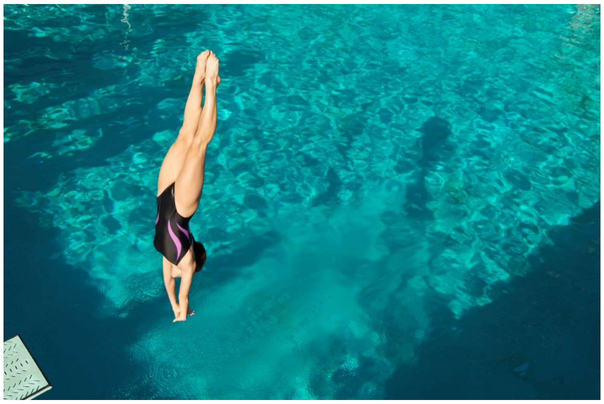
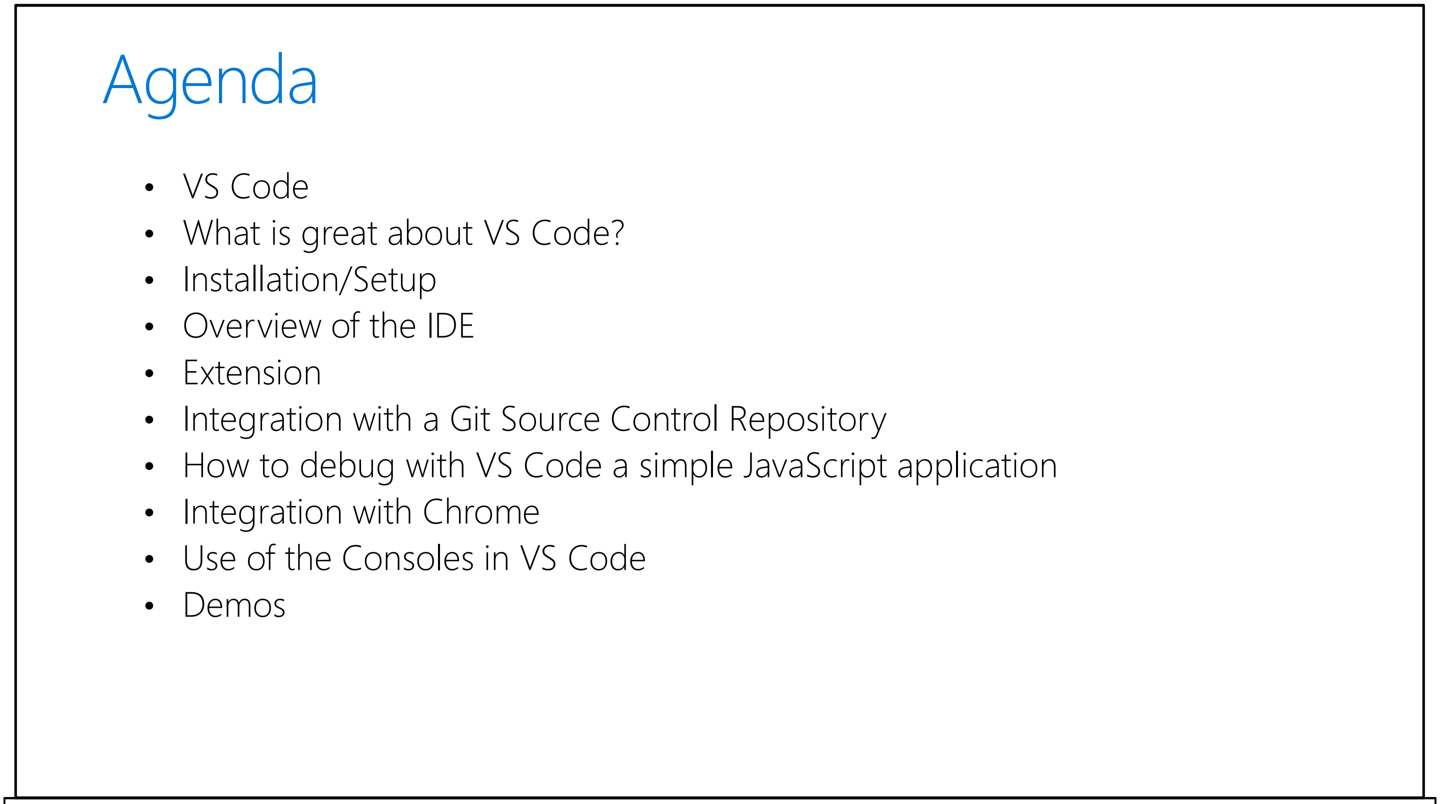
Visual Studio Code Tech Talk



Lizet Pena De Sola
Microsoft Developer Consultant. Premier Support for Developers.
lizet.pena@microsoft.com



Agenda

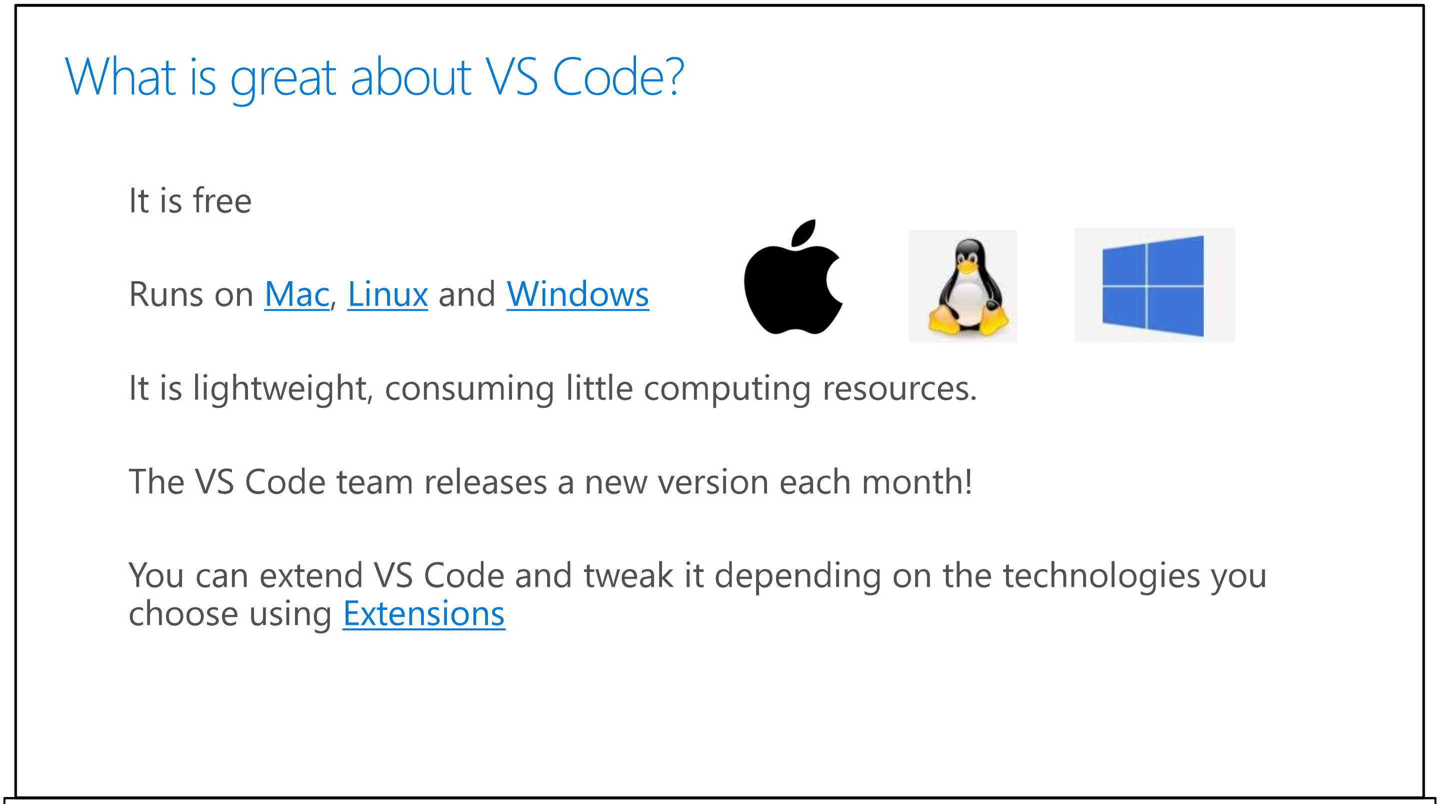
- VS Code
- What is great about VS Code?
- Installation/Setup
- Overview of the IDE
- Extension
- Integration with a Git Source Control Repository
- How to debug with VS Code a simple JavaScript application
- Integration with Chrome
- Use of the Consoles in VS Code
- Demos

VS Code

Microsoft Services

Hello and welcome to this Microsoft technical talk on Visual Studio Code also known as VS Code.

During this module, you’ll navigate the Visual Studio Code environment with me and I’ll be your guide during this learning experience.



What is great about VS Code?

It is free

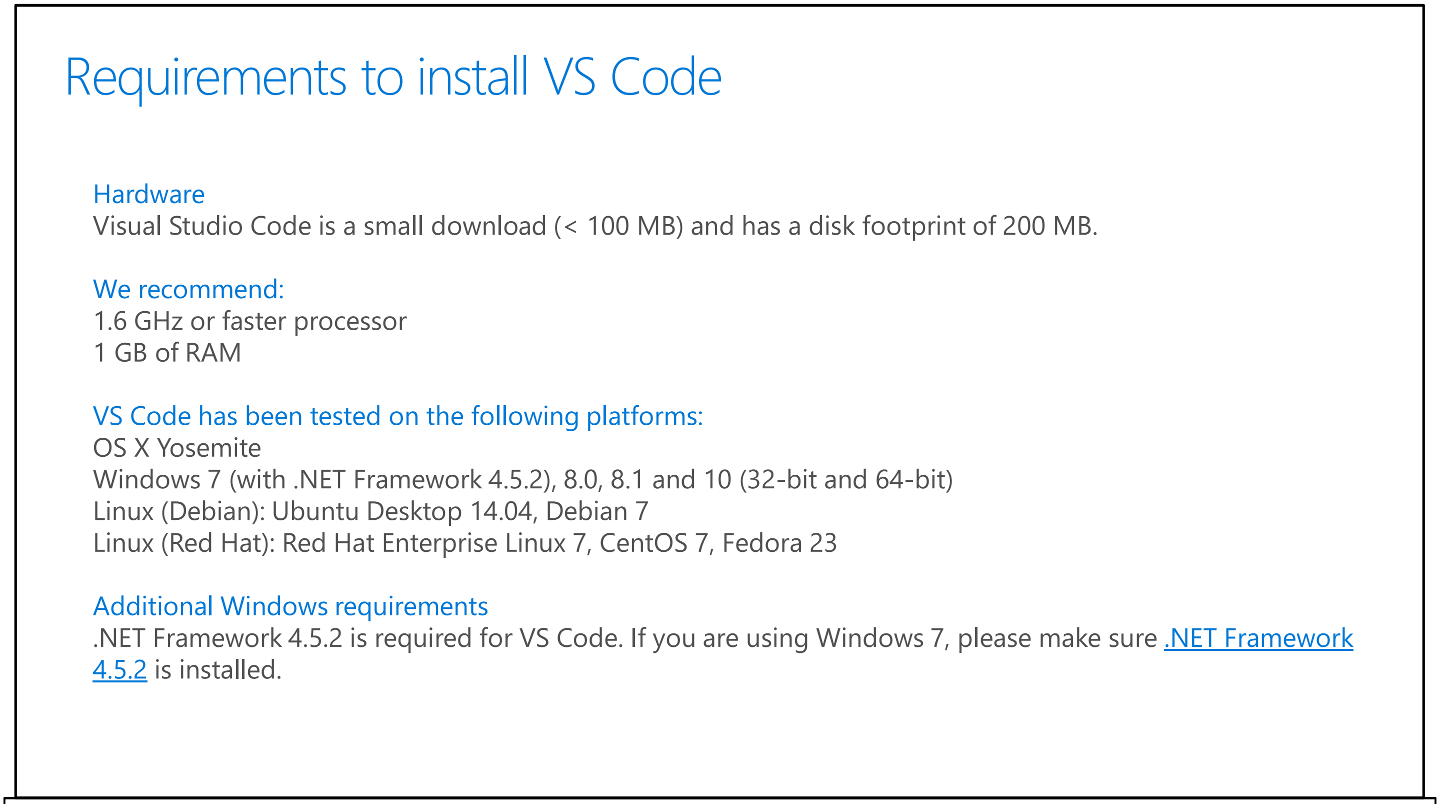
Runs on Mac, Linux and Windows

It is lightweight, consuming little computing resources.

The VS Code team releases a new version each month!

You can extend VS Code and tweak it depending on the technologies you choose using Extensions

- Free is great, we all want free, reducing the development costs and licensing costs for the development environments we choose to work with. This, in turn, allows us to solve business problems faster and better and we (developers) can give the company we work for the best advantage on the marketplace.
- And we like to code, yes, developers like to code.
- We like to have an IDE that adapts to the different technologies used in the world!
- Hint to the audience: each text on the presentation that has an underlined blue font is a hyperlink.
- You can click on it and it will navigate you through the web to documentation pages on the particular topic.



Requirements to install VS Code

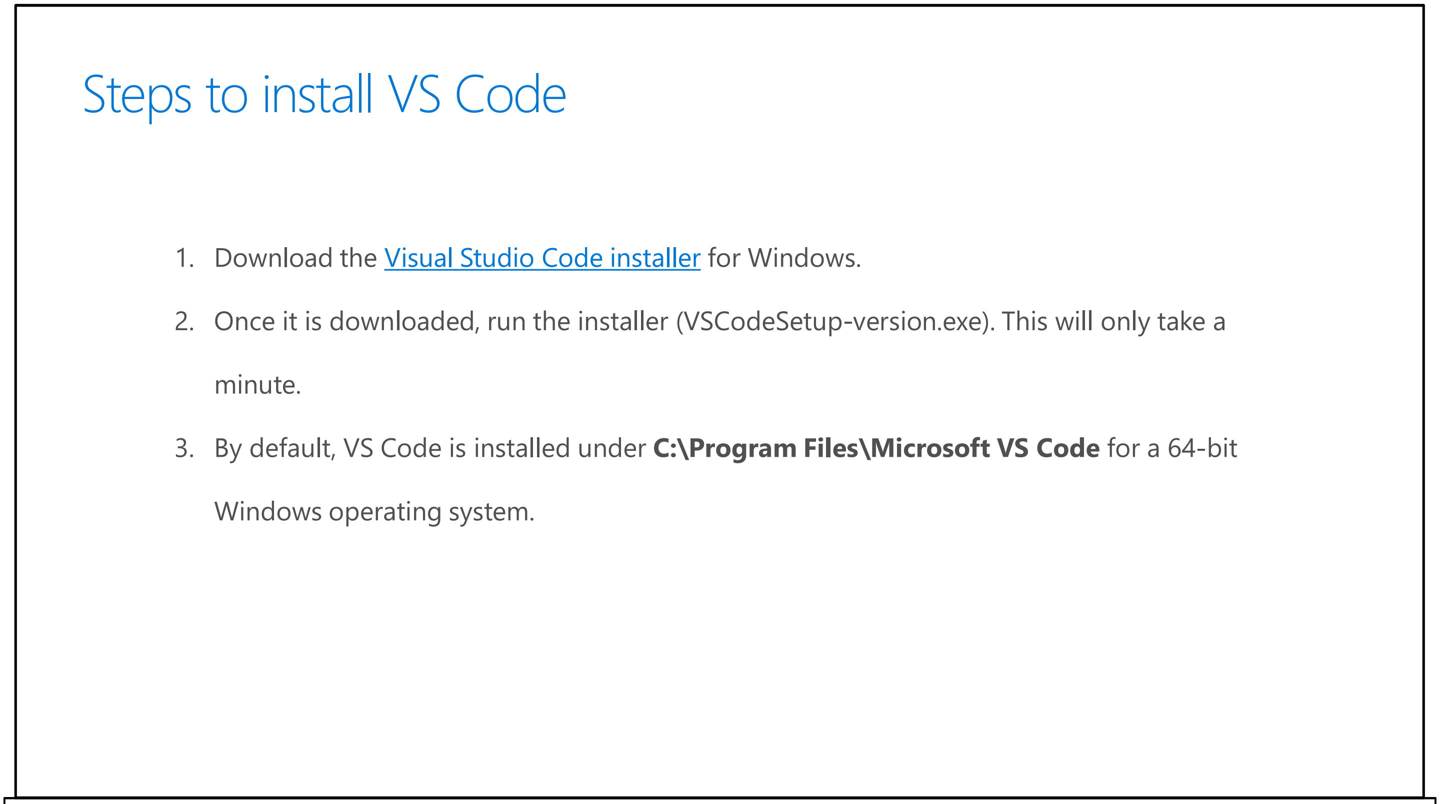
Hardware
Visual Studio Code is a small download (< 100 MB) and has a disk footprint of 200 MB.

We recommend:
1.6 GHz or faster processor
1 GB of RAM

VS Code has been tested on the following platforms:
OS X Yosemite
Windows 7 (with .NET Framework 4.5.2), 8.0, 8.1 and 10 (32-bit and 64-bit)
Linux (Debian): Ubuntu Desktop 14.04, Debian 7
Linux (Red Hat): Red Hat Enterprise Linux 7, CentOS 7, Fedora 23

Additional Windows requirements
.NET Framework 4.5.2 is required for VS Code. If you are using Windows 7, please make sure .NET Framework 4.5.2 is installed.

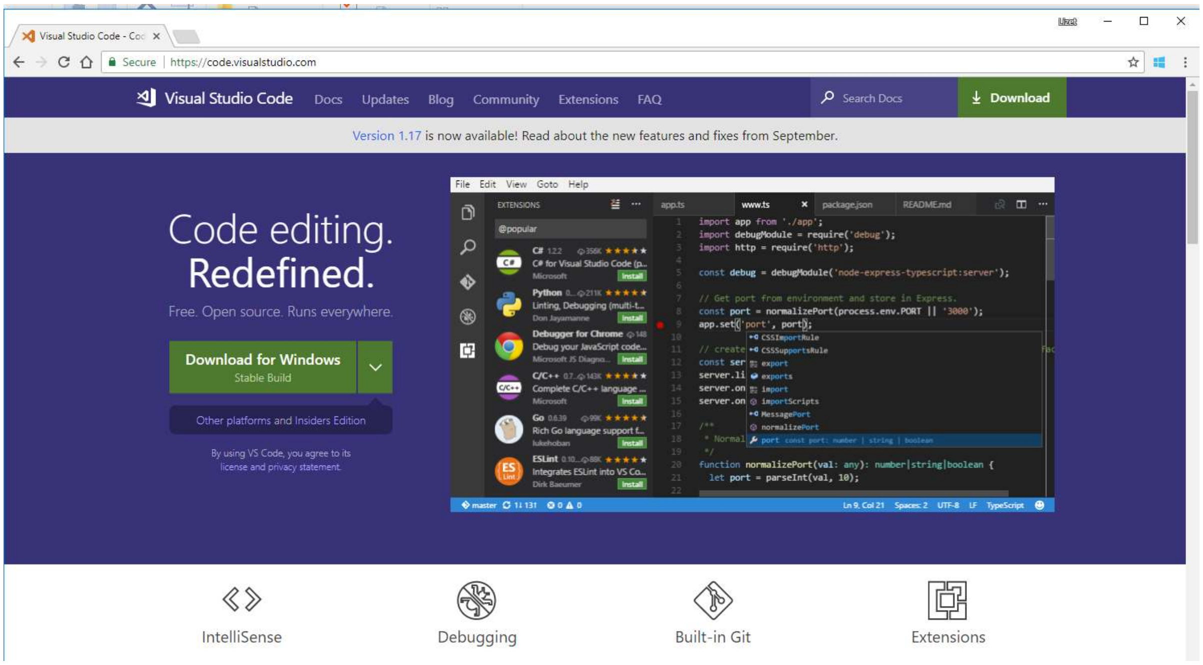
- Disk footprint is the size of the the visual studio code files on the computer hard drive.
- The .NET Framework minimum requirements is the 4.5.2 version, you can have a higher version installed as well. The .NET Framework 4.5.2 can be downloaded from the hyperlink on this slide. You are ok to just install VS Code on Windows 10 as Windows 10 already ships with the .NET Framework 4.5.2.



Steps to install VS Code

1. Download the Visual Studio Code installer for Windows.
2. Once it is downloaded, run the installer (VSCodeSetup-version.exe). This will only take a minute.
3. By default, VS Code is installed under **C:\Program Files\Microsoft VS Code** for a 64-bit Windows operating system.

- Link to download Visual Studio Code or VSCode https://go.microsoft.com/fwlink/?LinkID=534107



Version 1.17 is now available! Read about the new features and fixes from September.

Code editing. Redefined.

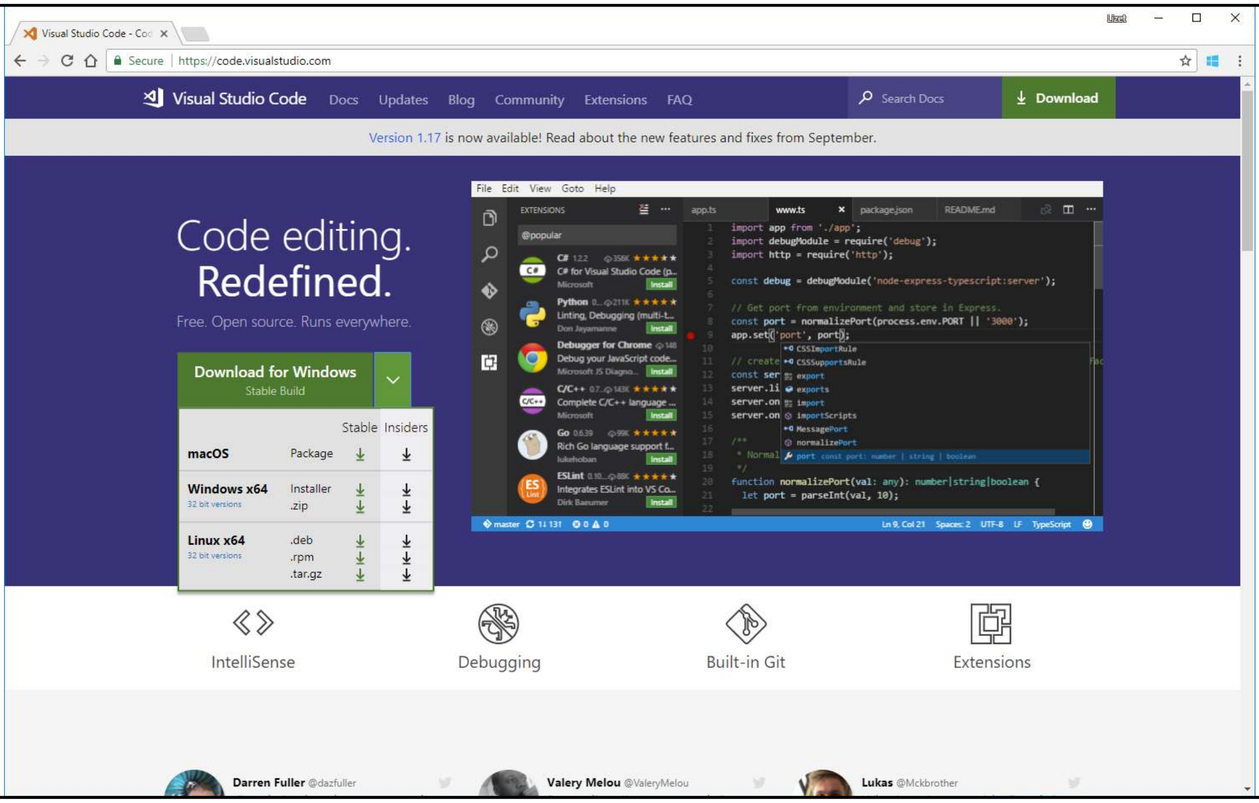
Free. Open source. Runs everywhere.

By using VS Code, you agree to its license and privacy statement.

```
import app from './app';
import debugModule = require('debug');
import http = require('http');

const debug = debugModule('node-express-typescript:server');

// Get port from environment and store in Express.
const port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

// create
const ser
server.li
server.on
server.on

/**
 * Normal
 */
function normalizePort(val: any): number|string|boolean {
  let port = parseInt(val, 10);
```

https://code.visualstudio.com/

- When you request the URL code.visualstudio.com, if your OS is a Mac OS, the page will show the proper file to be downloaded for a Mac operating system.
- You can also download files using the green drop down on the page.



Visual Studio Code - Coc
Secure | https://code.visualstudio.com
Visual Studio Code
Docs
Updates
Blog
Community
Extensions
FAQ
Search Docs
Download

Version 1.17 is now available! Read about the new features and fixes from September.

Code editing.
Redefined.

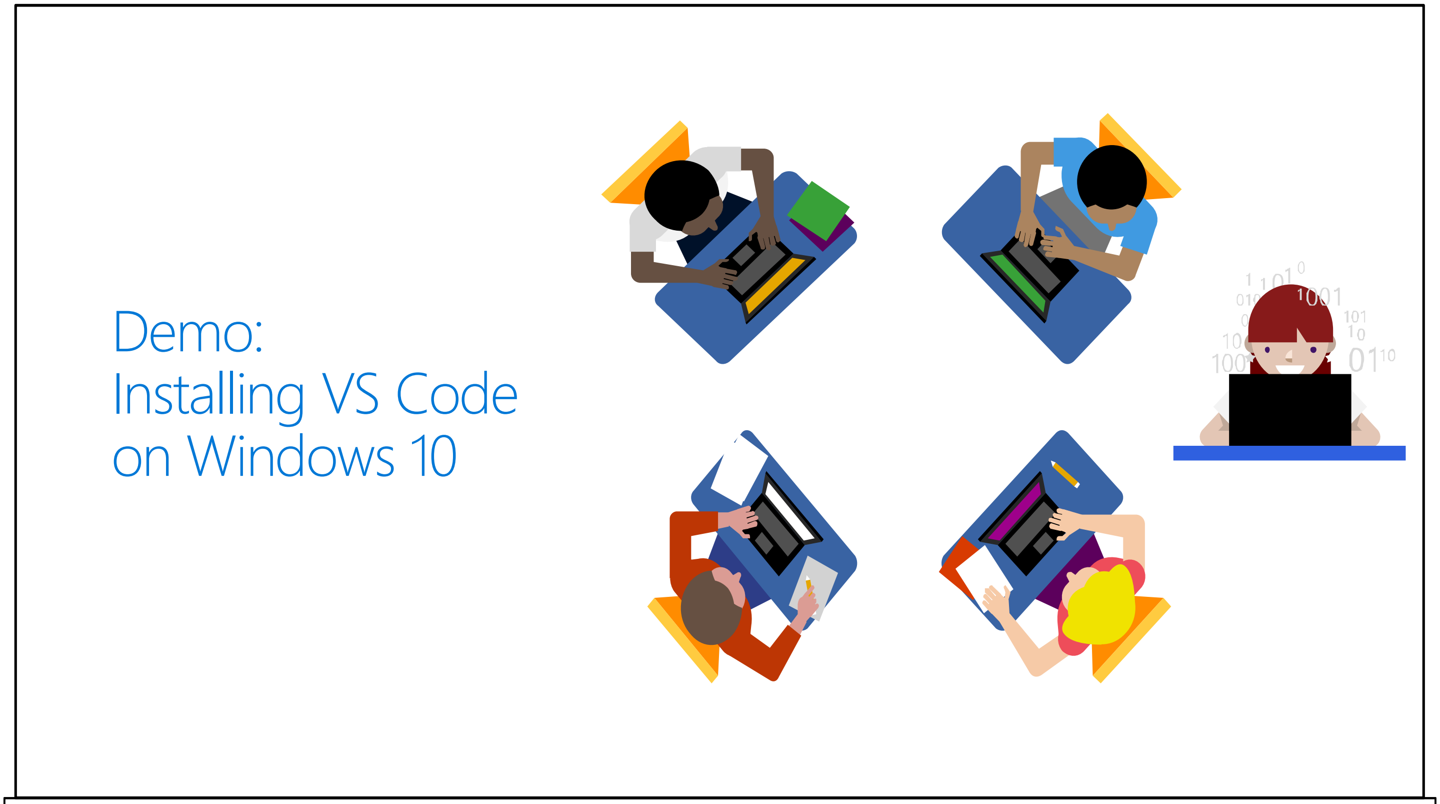
Free. Open source. Runs everywhere.

Download for Windows
Stable Build

		Stable	Insiders
macOS	Package	⤓	⤓
Windows x64 32 bit versions	Installer .zip	⤓ ⤓	⤓ ⤓
Linux x64 32 bit versions	.deb .rpm .tar.gz	⤓ ⤓ ⤓	⤓ ⤓ ⤓

File Edit View Goto Help
EXTENSIONS
@popular
C# 1.2.2 C# for Visual Studio Code (p... Microsoft Install
Python Linting, Debugging (multi-t... Don Jayamanne Install
Debugger for Chrome Debug your JavaScript code... Microsoft JS Diagno... Install
C/C++ Complete C/C++ language ... Microsoft Install
Go 0.6.39 Rich Go language support f... lukehoban Install
ESLint Integrates ESLint into VS Co... Dirk Baeumer Install
app.ts
www.ts
package.json
README.md

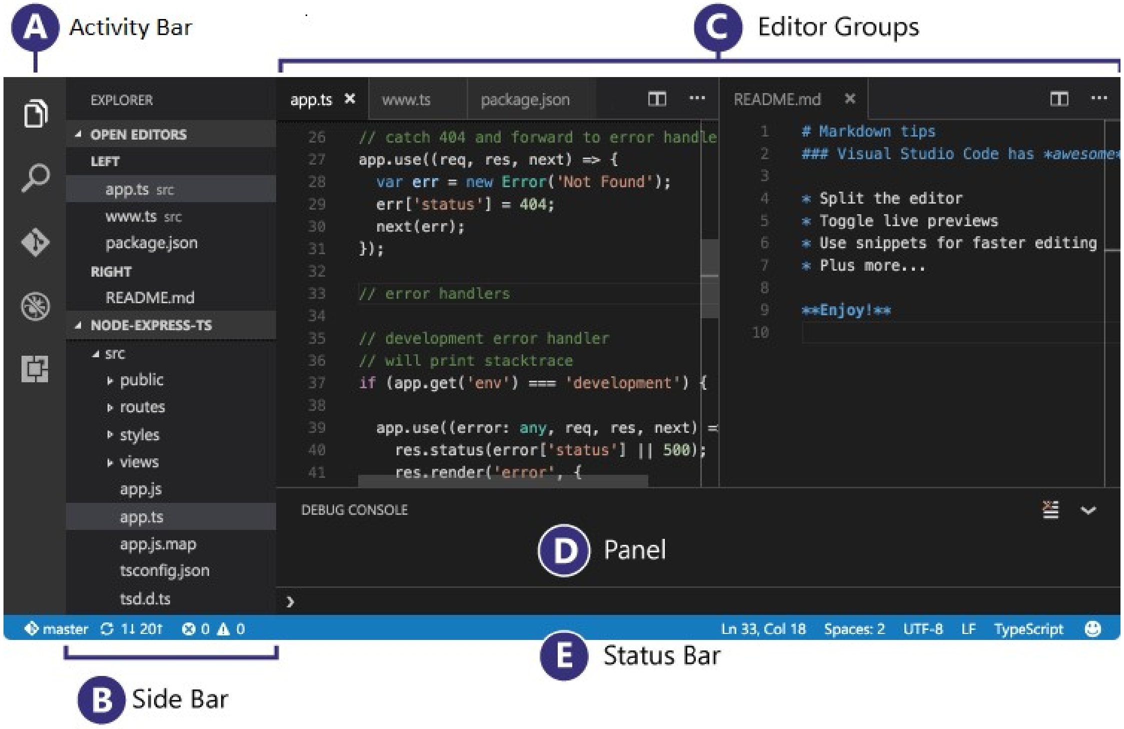
```
import app from './app';
import debugModule = require('debug');
import http = require('http');

const debug = debugModule('node-express-typescript:server');

// Get port from environment and store in Express.
const port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

// create
const ser
server.li
server.on
server.on

/**
 * Normal
 */
function normalizePort(val: any): number|string|boolean {
  let port = parseInt(val, 10);
```



Demo:
Installing VS Code on Windows 10

VSCode Overview

Microsoft Services



```
// catch 404 and forward to error handle
app.use((req, res, next) => {
  var err = new Error('Not Found');
  err['status'] = 404;
  next(err);
});

// error handlers

// development error handler
// will print stacktrace
if (app.get('env') === 'development') {

  app.use((error: any, req, res, next) =
    res.status(error['status'] || 500);
    res.render('error', {
```

```
# Markdown tips
### Visual Studio Code has *awesome*

* Split the editor
* Toggle live previews
* Use snippets for faster editing
* Plus more...

**Enjoy!**
```

- At its heart, Visual Studio Code is a code editor. Like many other code editors, VS Code adopts a common user interface and layout of an explorer on the left, showing all of the files and folders you have access to, and an editor on the right, showing the content of the files you have opened.

- **Files, Folders & Projects**
- VS Code is file and folder based - you can get started immediately by opening a file or folder in VS Code.
- On top of this, VS Code can read and take advantage of a variety of project files defined by different frameworks and platforms. For example, if the folder you opened in VS Code contains one or more package.json, project.json, tsconfig.json, or .NET Core Visual Studio solution and project files, VS Code will read these files and use them to provide additional functionality, such as rich IntelliSense in the editor.

- **The UI is divided into five areas:**
- Editor - The main area to edit your files. You can open up to three editors side by side.
- Side Bar - Contains different views like the Explorer to assist you while working on your project.
- Status Bar - Information about the opened project and the files you edit.
- Activity Bar - Located on the far left-hand side, this lets you switch between views and gives you additional context-specific indicators, like the number of outgoing changes when Git is enabled.
- Panels - You can display different panels below the editor region for output or debug information, errors and warnings, or an integrated terminal.

Demo:
Overview of the IDE

https://code.visualstudio.com/docs/getstarted/userinterface

At its heart, Visual Studio Code is a code editor. Like many other code editors, VS Code adopts a common user interface and layout of an explorer on the left, showing all of the files and folders you have access to, and an editor on the right, showing the content of the files you have opened.

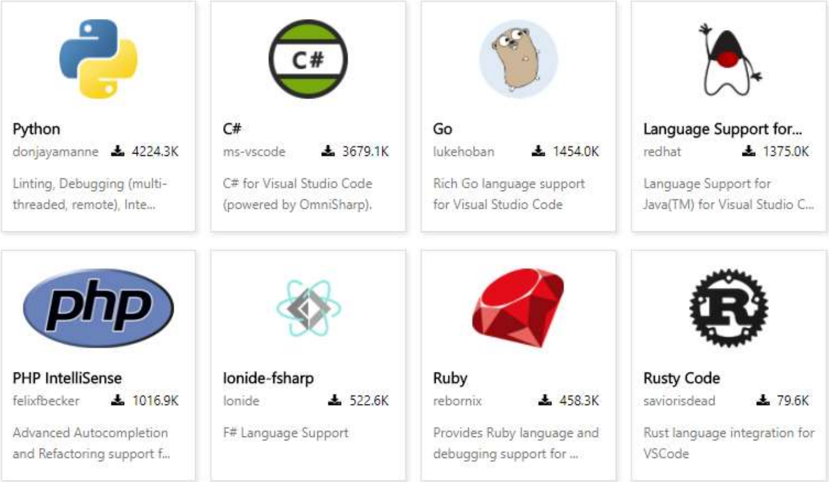
Demo steps:

- Open src folder on your desktop and show them the structure of a C# solution that has a Web API projects “TodoListService” and a Single Page Application that uses Angular JS
- Show them the different icons that identify the files
- Show them the extensions that you have installed
- Also show how to add a PowerShell console.

Extensions

Microsoft Services

Most Popular Extensions



Python
donjayamanne ⬇ 4224.3K
Linting, Debugging (multi-threaded, remote), Inte...

C#
ms-vscode ⬇ 3679.1K
C# for Visual Studio Code (powered by OmniSharp).

Go
lukehoban ⬇ 1454.0K
Rich Go language support for Visual Studio Code

Language Support for...
redhat ⬇ 1375.0K
Language Support for Java(TM) for Visual Studio C...

PHP IntelliSense
felixfbecker ⬇ 1016.9K
Advanced Autocompletion and Refactoring support f...

Ionide-fsharp
Ionide ⬇ 522.6K
F# Language Support

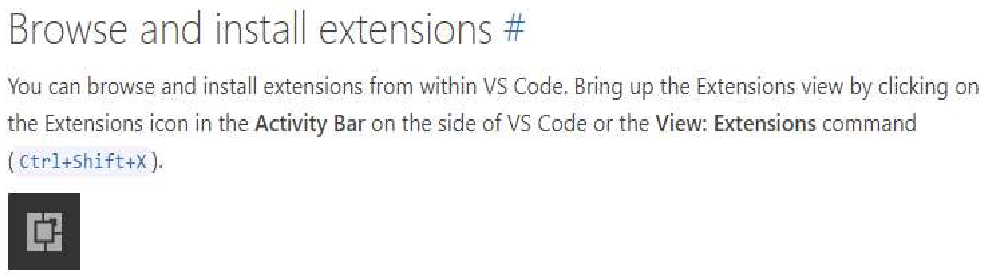
Ruby
rebornix ⬇ 458.3K
Provides Ruby language and debugging support for ...

Rusty Code
saviorisdead ⬇ 79.6K
Rust language integration for VSCode

•Outline

- Find extensions to install using the Extensions View.
- Install an extension.
- See what features are added via the Contributions tab.
- See the list of Recommended Extensions.
- Disable and uninstall extensions.

How to install Extensions

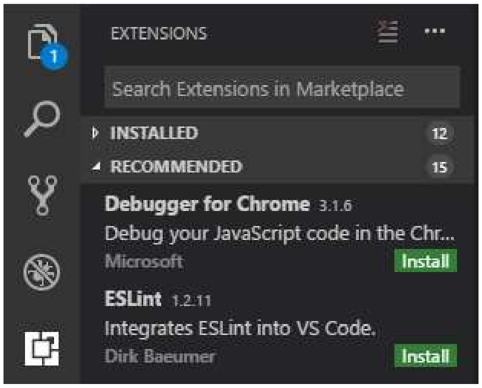


Browse and install extensions

You can browse and install extensions from within VS Code. Bring up the Extensions view by clicking on the Extensions icon in the **Activity Bar** on the side of VS Code or the **View: Extensions** command (`Ctrl+Shift+X`).

- In Visual Studio Code, we have support for almost every major programming language. Several ship in the box, for example, JavaScript, TypeScript, CSS, and HTML but more rich language extensions can be found in the VS Code Marketplace.

Demo:
Installing Extensions



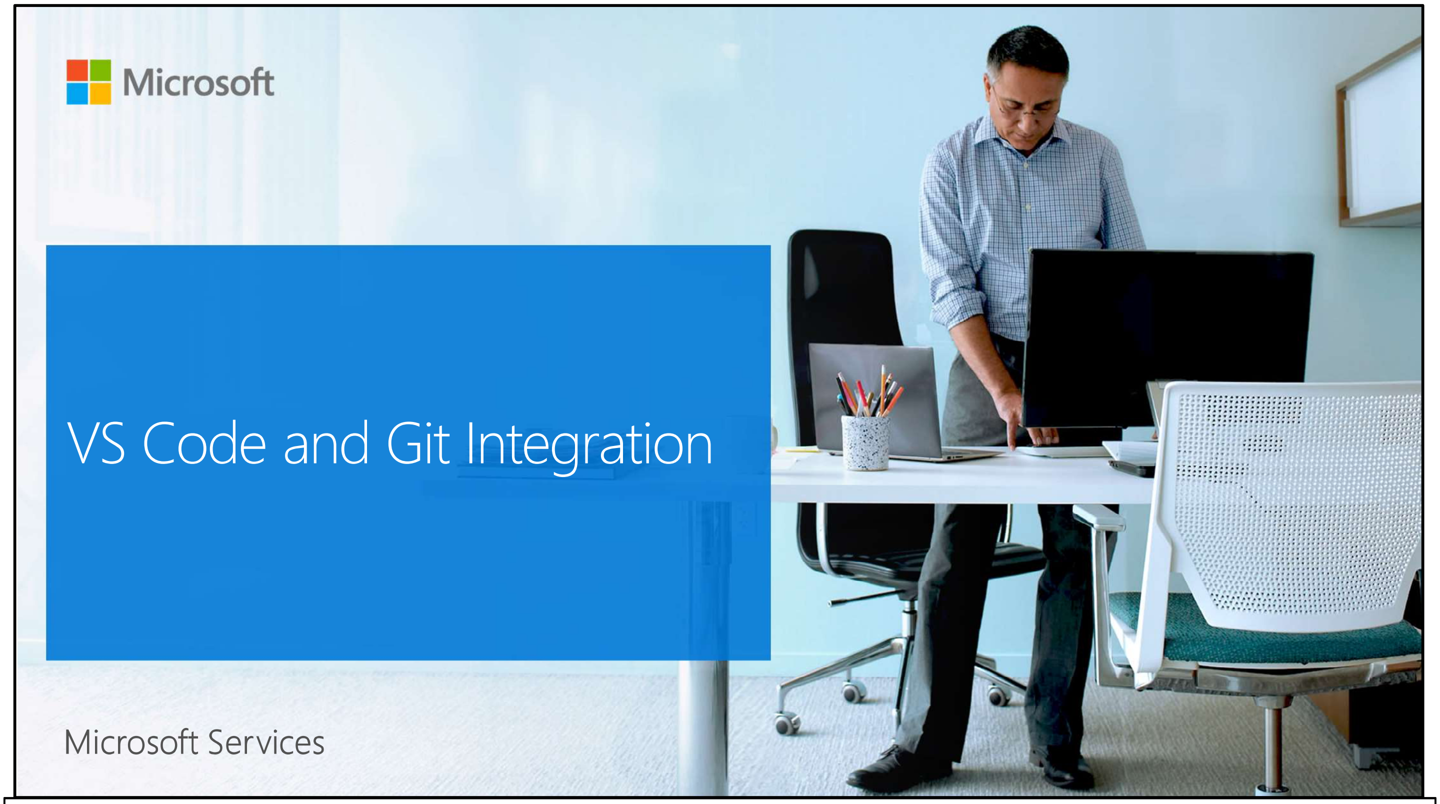
Debugger for Chrome 3.1.6
Debug your JavaScript code in the Chr...
Microsoft

ESLint 1.2.11
Integrates ESLint into VS Code.
Dirk Baeumer

https://code.visualstudio.com/docs/editor/extension-gallery

On this demo:

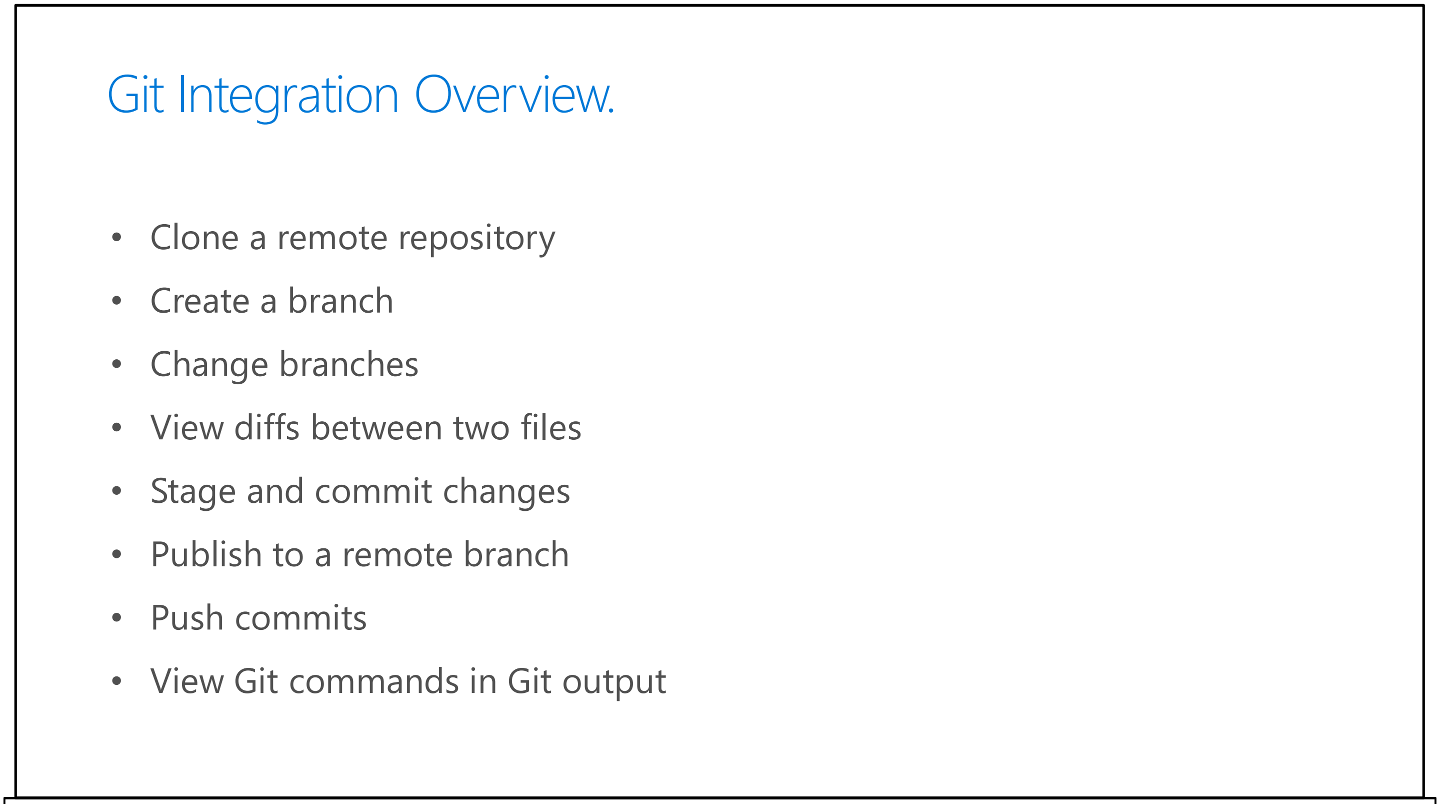
- Open VS Code,
- click the extensions icon
- download the C# extension



Microsoft

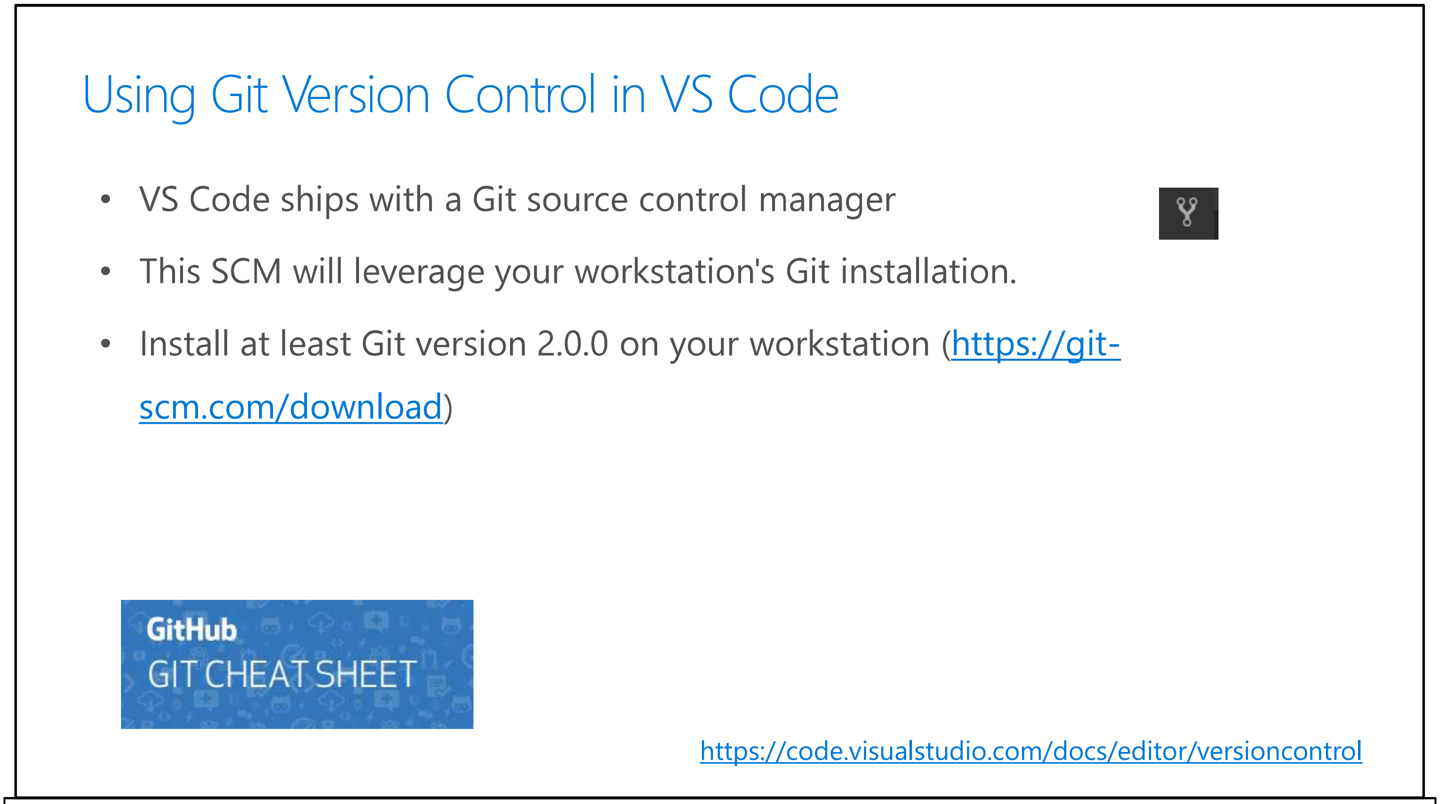
VS Code and Git Integration

Microsoft Services



Git Integration Overview.

- Clone a remote repository
- Create a branch
- Change branches
- View diffs between two files
- Stage and commit changes
- Publish to a remote branch
- Push commits
- View Git commands in Git output



Using Git Version Control in VS Code

- VS Code ships with a Git source control manager
- This SCM will leverage your workstation's Git installation.
- Install at least Git version 2.0.0 on your workstation (https://git-scm.com/download)

GitHub
GIT CHEAT SHEET

https://code.visualstudio.com/docs/editor/versioncontrol

- How to add version control to VSCode

https://code.visualstudio.com/docs/editor/versioncontrol

Git Source Control Introduction Video

https://code.visualstudio.com/docs/introvideos/versioncontrol

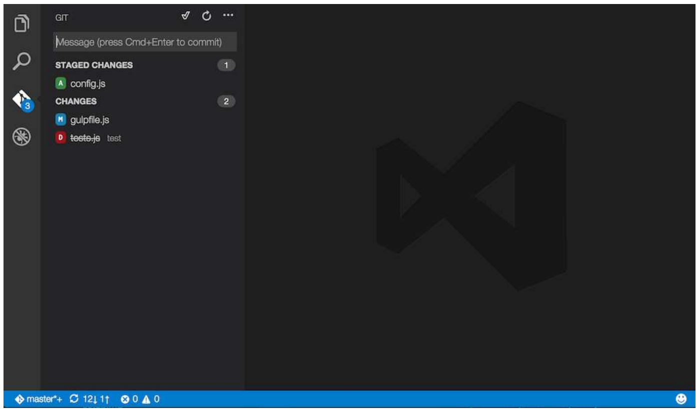
- The documentation on VSCode assumes you're already familiar with Git. If not, check out this free book on the Git version control

https://git-scm.com/book/en/v2

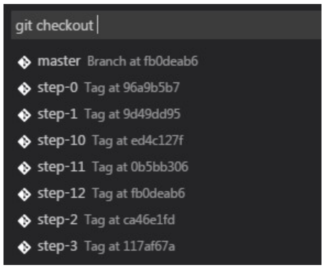
Git Hub Cheat Sheet on most common Git Commands

https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

Changed files to be committed



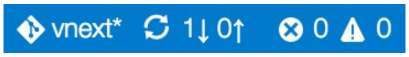
Command Palette



git checkout

master	Branch at fb0deab6
step-0	Tag at 96a9b5b7
step-1	Tag at 9d49dd95
step-10	Tag at ed4c127f
step-11	Tag at 0b5bb306
step-12	Tag at fb0deab6
step-2	Tag at ca46e1fd
step-3	Tag at 117af67a

Status Bar



VS Code offers you useful actions to push, pull and sync with the remote branch (the latter will run a pull command followed by a push command). You can find these actions in the **...** menu.

- There is a Synchronize Changes action in the Status Bar, next to the branch indicator, when the current checked out branch has an upstream branch configured. Synchronize Changes will pull remote changes down to your local repository and then push local commits to the upstream branch.
- The command palette can be reached from the View menu and allows you to invoke Git commands.
- Initialize a Repository
- If your workspace isn't under Git source control, you can easily create a Git repository with the Initialize Git Repository command.

Clone Remote Repository.

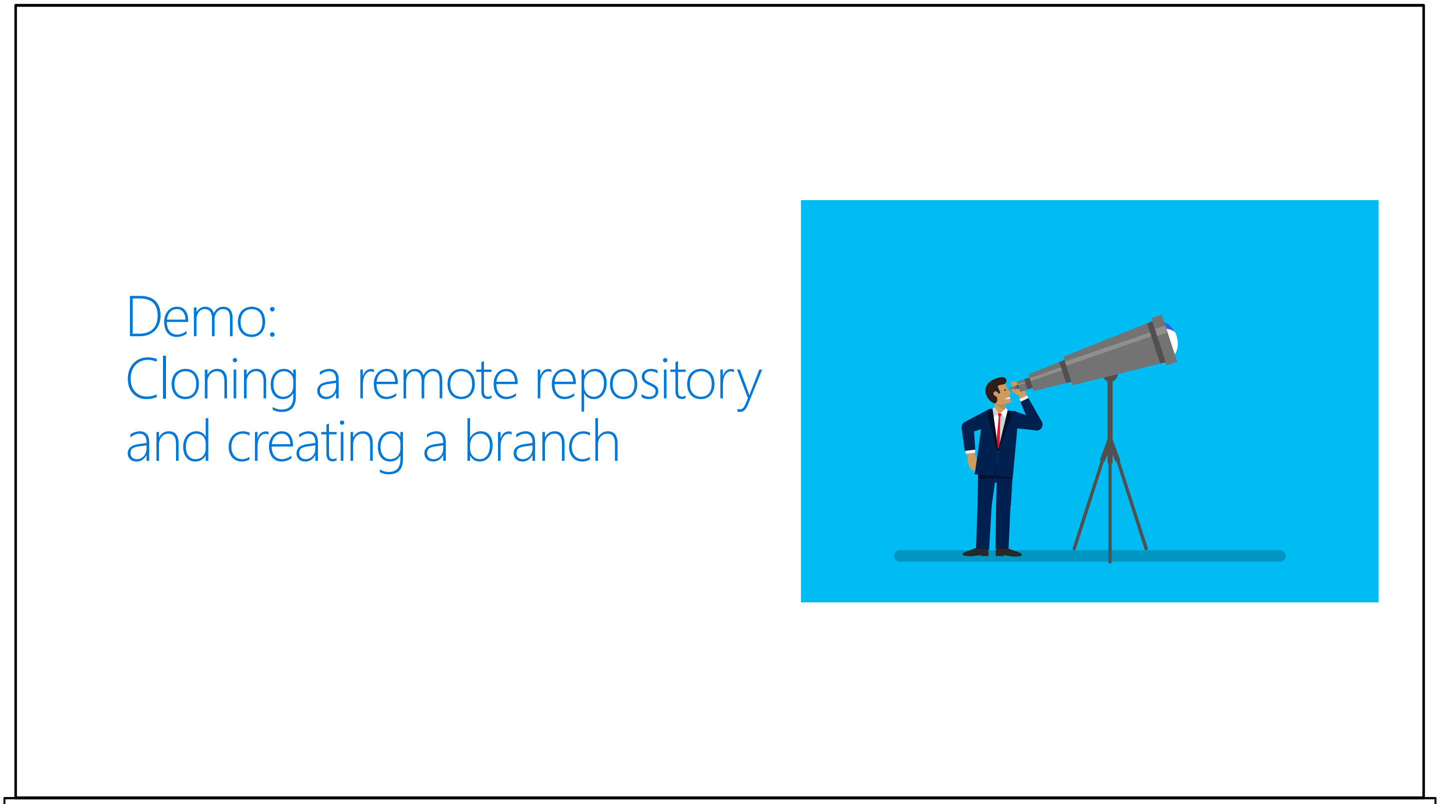
Git: Clone command in the Command Palette (Ctrl+Shift+P). You will be asked for the URL of the remote repository and the parent directory under which to put the local repository.

You can also use the Welcome Page link GitHub repository to clone an existing remote repository.

VSCode Git Branches

You can create and checkout branches directly within VS code through the **Git: Create Branch...** and **Git: Checkout to**.. commands in the Command Palette (Ctrl+Shift+P).

Git: Create Branch... command lets you quickly create a new branch.
Provide the name of your new branch and VS Code will create the branch and switch to it.



Demo:
Cloning a remote repository and creating a branch

Cloning a remote repository

Open GitHub demo repository: https://github.com/lizetpena/VSCode_AngularSPA_AAD

Click on Clone or Download button, copy the Repository URL

Open VS Code on the folder where you want to clone the repository on

Click on View menu, Open the Command Palette, type Git Clone, it should ask for the URL of the remote repository and the location on the local hard drive.

Creating a branch off the master branch

On the command palette type Git: Create Branch

Give the new branch a name

See how to switch between main and the created branch.

Changes branches on the status bar:

Click on the status bar and see how you can change between branches of the code.

Edit one file on one of the branches and see how the Git icon shows the number or modified files.

File diff demo

1) write something to the file

2) see the M icon showing next to the filename

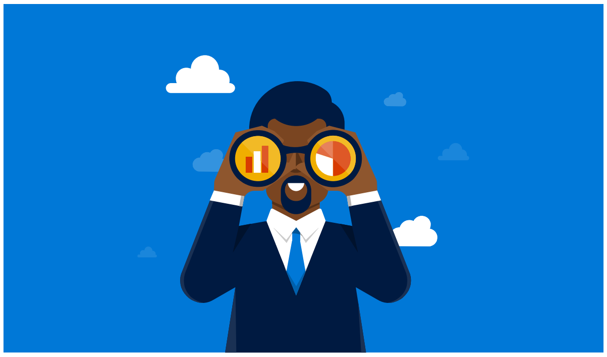
3) See the number of changes showing on the Git icon

4) click the Git icon

5) click on the file name, you'll see the differences

between the local changes and the last committed file

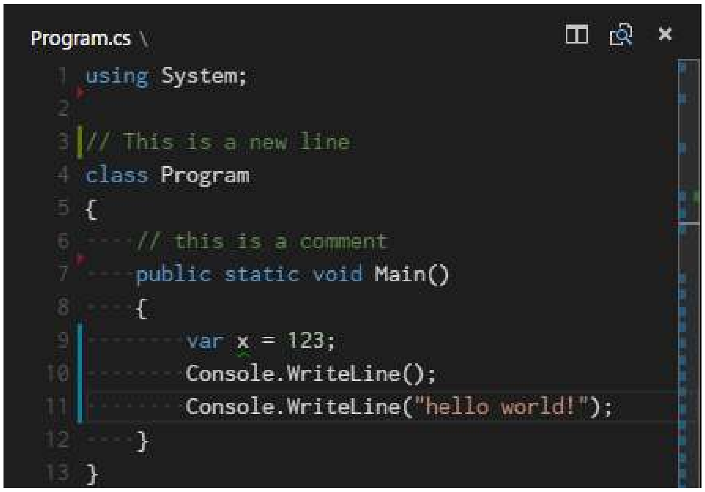
Demo:
Commit, Push changes
View Git Output



Git Integration. Gutter Indicators.

If you open a folder that is a Git repository and begin making changes, VS Code will add useful annotations to the gutter and to the overview ruler.

- A red triangle indicates where lines have been deleted
- A green bar indicates new added lines
- A blue bar indicates modified lines

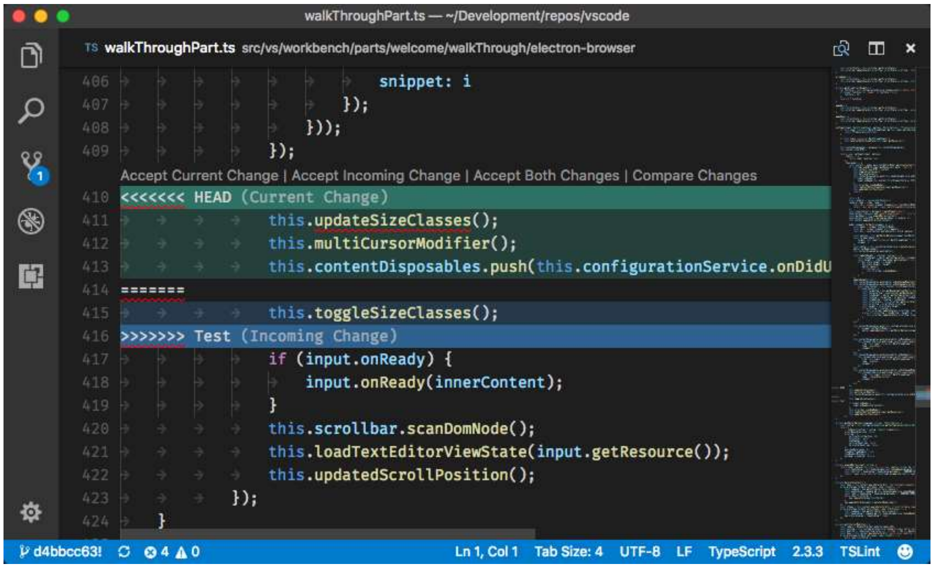


```
using System;

// This is a new line
class Program
{
    // this is a comment
    public static void Main()
    {
        var x = 123;
        Console.WriteLine();
        Console.WriteLine("hello world!");
    }
}
```

Git Integration. Merge Conflicts.

Differences are highlighted and there are inline actions to accept either or both changes.

Once the conflicts are resolved, stage the conflicting file so you can commit those changes.



walkThroughPart.ts — ~/Development/repos/vscode

TS walkThroughPart.ts src/vs/workbench/parts/welcome/walkThrough/electron-browser

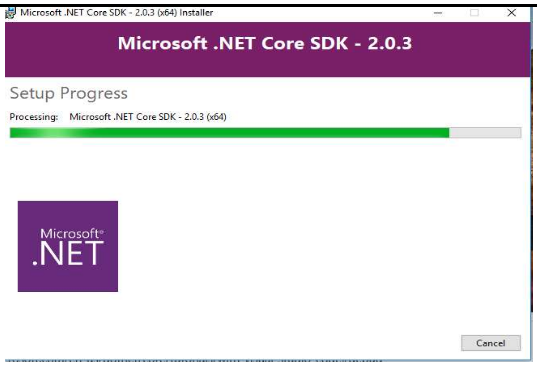
```
                            snippet: i
                        });
                    }));
                });
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<<< HEAD (Current Change)
                this.updateSizeClasses();
                this.multiCursorModifier();
                this.contentDisposables.push(this.configurationService.onDidU
=======
                this.toggleSizeClasses();
>>>>>>> Test (Incoming Change)
                if (input.onReady) {
                    input.onReady(innerContent);
                }
                this.scrollbar.scanDomNode();
                this.loadTextEditorViewState(input.getResource());
                this.updatedScrollPosition();
            });
    }
```

- Differences are highlighted and there are inline actions to accept either or both changes. Once the conflicts are resolved, stage the conflicting file so you can commit those changes.

Debugging C# Console App

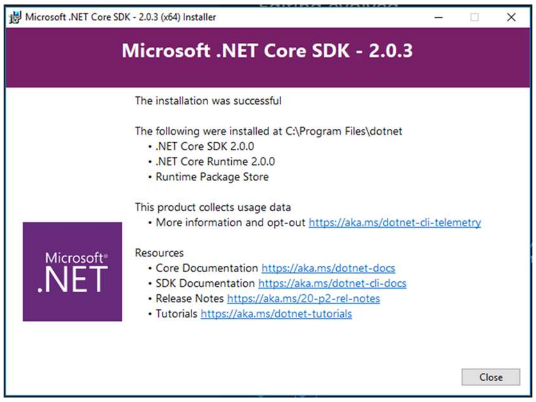
Microsoft Services

Install .NET Core SDK.


Microsoft .NET Core SDK - 2.0.3 (x64) Installer
Microsoft .NET Core SDK - 2.0.3
Microsoft® .NET
Cancel

Setup Progress

Processing: Microsoft .NET Core SDK - 2.0.3 (x64)


Microsoft .NET Core SDK - 2.0.3 (x64) Installer
Microsoft .NET Core SDK - 2.0.3
Microsoft® .NET
Close

The installation was successful

The following were installed at C:\Program Files\dotnet
- .NET Core SDK 2.0.0
- .NET Core Runtime 2.0.0
- Runtime Package Store

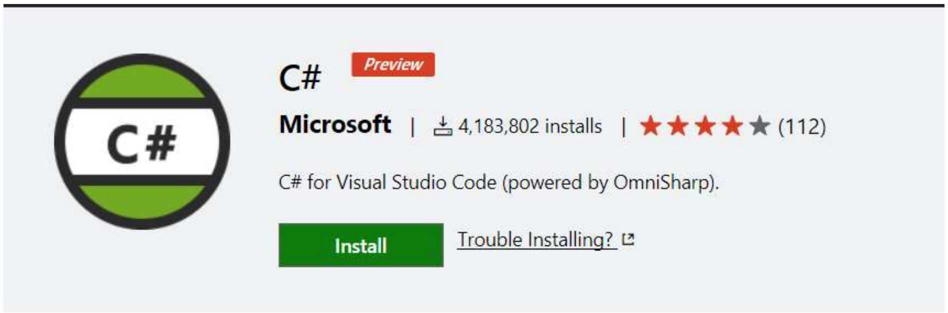
This product collects usage data
- More information and opt-out https://aka.ms/dotnet-cli-telemetry

Resources
- Core Documentation https://aka.ms/dotnet-docs
- SDK Documentation https://aka.ms/dotnet-cli-docs
- Release Notes https://aka.ms/20-p2-rel-notes
- Tutorials https://aka.ms/dotnet-tutorials

- https://code.visualstudio.com/docs/other/dotnet

- https://docs.microsoft.com/en-us/dotnet/core/tutorials/with-visual-studio-code#debug

Install the C# extension from the VS Code Marketplace.

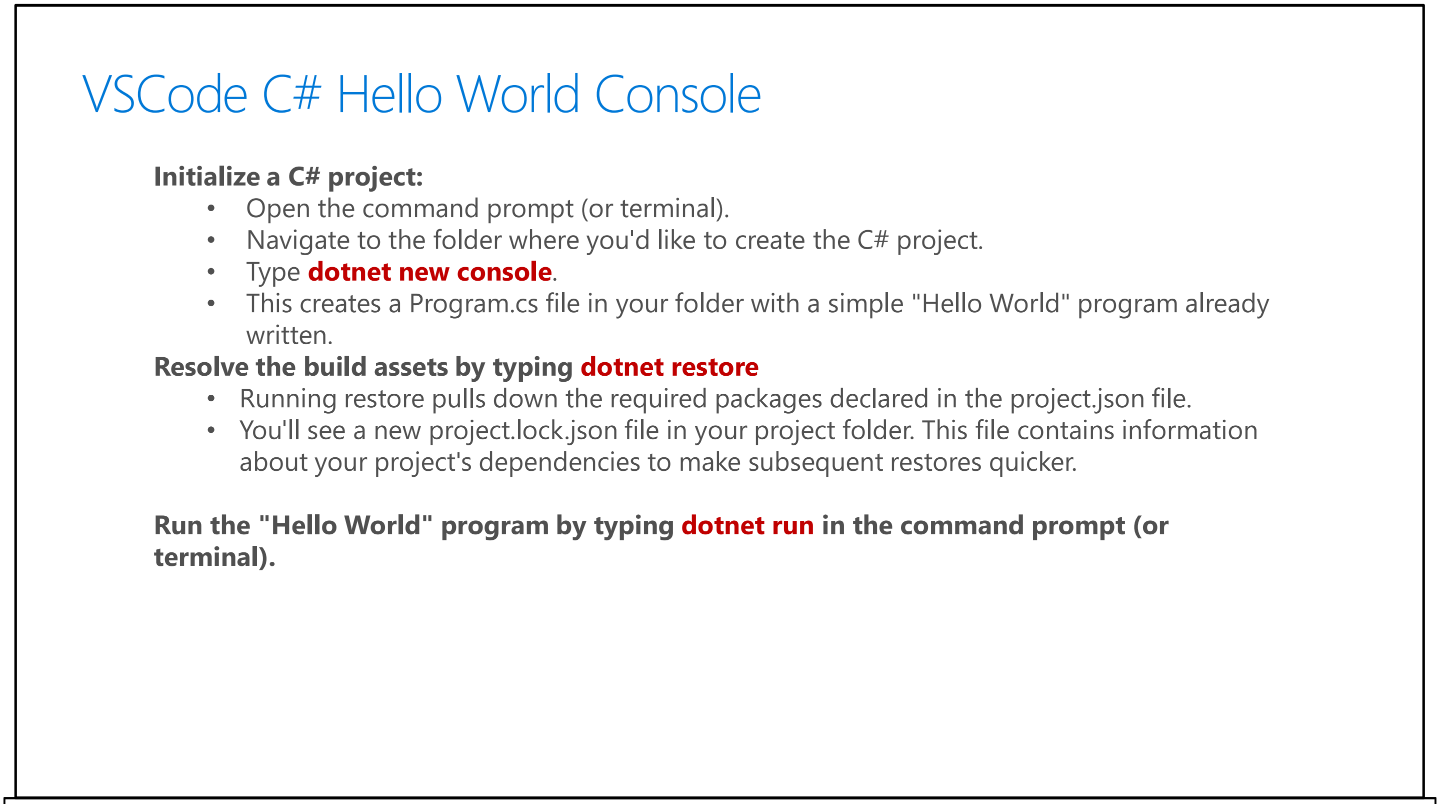


C# *Preview*

Microsoft | 4,183,802 installs | ★★★★★ (112)

C# for Visual Studio Code (powered by OmniSharp).

- https://marketplace.visualstudio.com/items?itemName=ms-vscode.csharp



VSCode C# Hello World Console

Initialize a C# project:

- Open the command prompt (or terminal).
- Navigate to the folder where you'd like to create the C# project.
- Type **dotnet new console**.
- This creates a Program.cs file in your folder with a simple "Hello World" program already written.

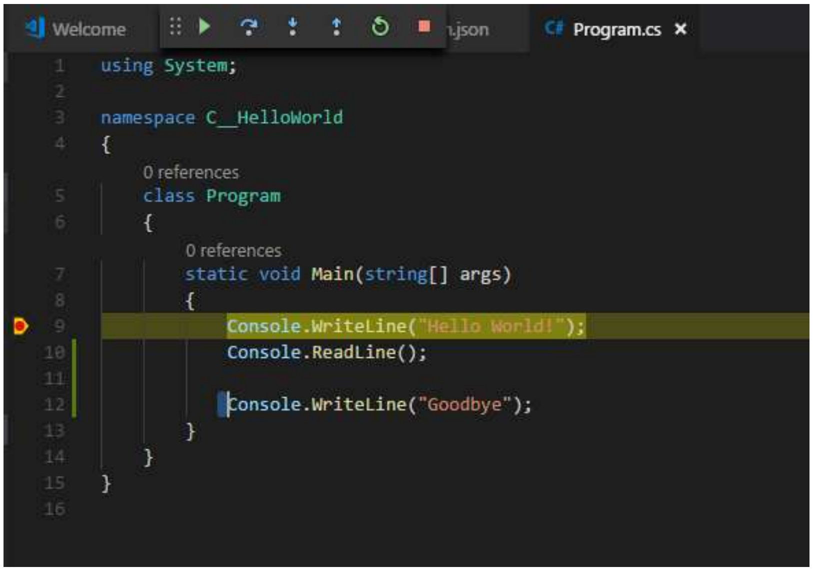
Resolve the build assets by typing dotnet restore

- Running restore pulls down the required packages declared in the project.json file.
- You'll see a new project.lock.json file in your project folder. This file contains information about your project's dependencies to make subsequent restores quicker.

Run the "Hello World" program by typing dotnet run in the command prompt (or terminal).

- Hello World
- If you'd like to get started with a simple "Hello World" program on .NET Core, follow the steps below:
- Initialize a C# project:
- Open the command prompt (or terminal).
- Navigate to the folder where you'd like to create the C# project.
- Type dotnet new console.
- This creates a Program.cs file in your folder with a simple "Hello World" program already written.
- Resolve the build assets by typing dotnet restore.
- Tip: .NET Core Tools are now MSBuild-based. This means a .csproj project file will be created instead of a project.json. Read more.
- Running restore pulls down the required packages declared in the project.json file.
- You'll see a new project.lock.json file in your project folder.
- This file contains information about your project's dependencies to make subsequent restores quicker.
- When the project folder is first opened in VS Code, a notification will appear at the top of the window asking if you'd like to add the required assets to build and debug your project. Select Yes.
- Run the "Hello World" program by typing dotnet run in the command prompt (or terminal).

VSCode C# Hello World Console



```
using System;

namespace C_HelloWorld
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            Console.ReadLine();

            Console.WriteLine("Goodbye");
        }
    }
}
```

Demo:
C# Hello World

https://docs.microsoft.com/en-us/dotnet/core/tutorials/with-visual-studio-code#debug

- git init
- git add README.md
- git commit -m "first commit"
- git remote add origin https://github.com/lizetpena/HelloWorldC-.git
- git push -u origin master

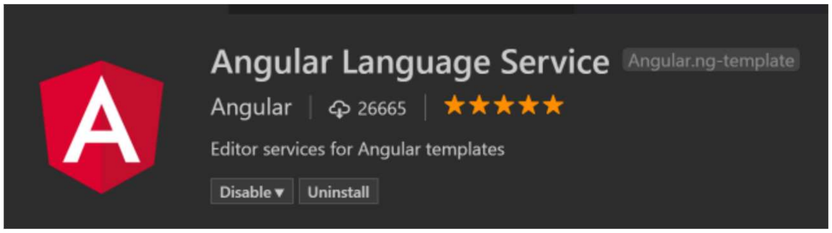
Debugging Angular sample app

Microsoft Services

Hello and welcome to this Microsoft technical talk on Visual Studio Code also known as VS Code.

During this module, you'll navigate the Visual Studio Code environment with me and I'll be your guide during this learning experience.

VSCode Angular Recommended Extension



Angular Language Service Angular.ng-template

Angular | 26665 | ★★★★★

Editor services for Angular templates

- Download the extension from the marketplace

Angular Application Start

ng new

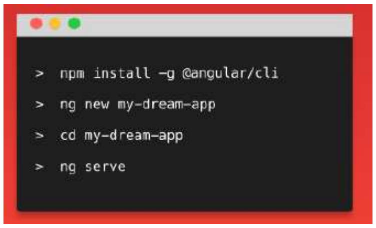
The Angular CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!

ng generate

Generate components, routes, services and pipes with a simple command. The CLI will also create simple test shells for all of these.

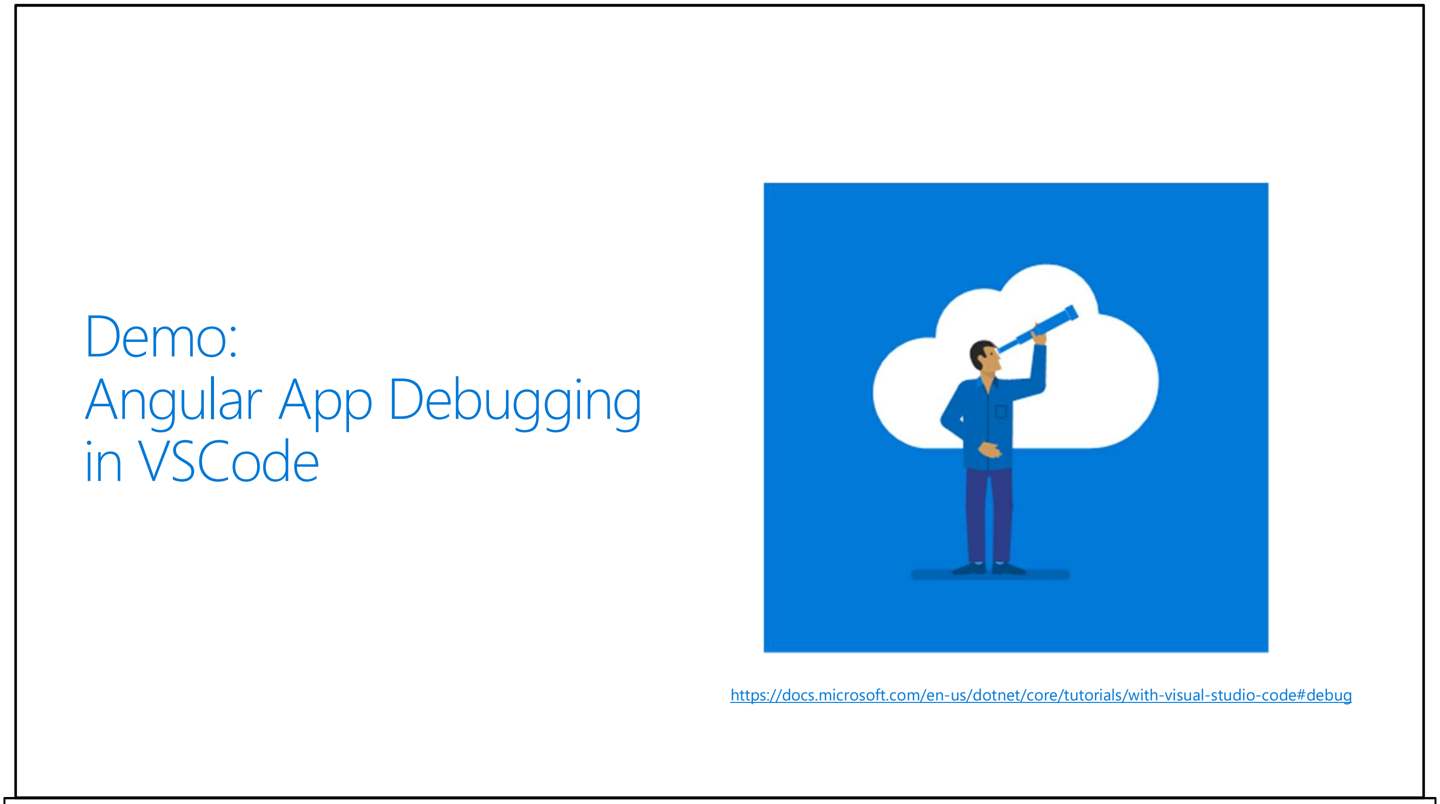
ng serve

Easily test your app locally while developing.



```
>  npm install -g @angular/cli
>  ng new my-dream-app
>  cd my-dream-app
>  ng serve
```

- The source code for this application is on https://github.com/lizetpena/AngularSample



Demo:
Angular App Debugging in VSCode

https://docs.microsoft.com/en-us/dotnet/core/tutorials/with-visual-studio-code#debug

- Open C:\src\AngularWorkshop in VSCode
- Set a few breakpoints in the TypeScript code
- Go to the terminal windows and type npm start
- Navigate with Chrome to http://localhost:4200

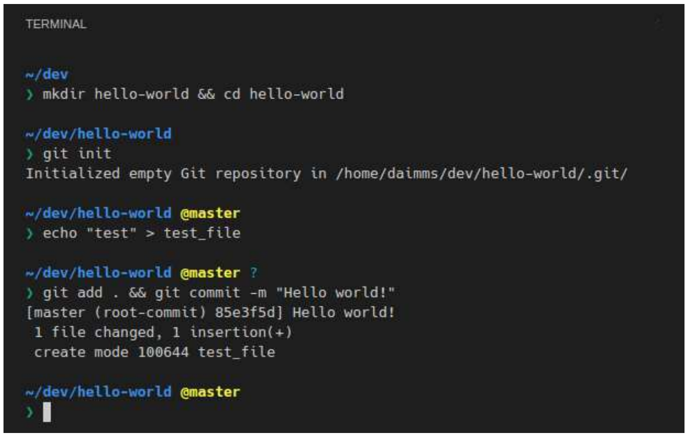
VSCode Integrated Terminal

To open the terminal:

Use the **Ctrl+`** keyboard shortcut with the backtick character.

Use the **View | Integrated Terminal** menu command.

From the Command Palette **(Ctrl+Shift+P),** use the **View:Toggle Integrated Terminal** command.



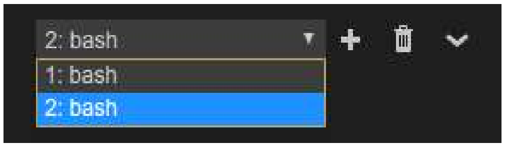
```
~/dev
› mkdir hello-world && cd hello-world

~/dev/hello-world
› git init
Initialized empty Git repository in /home/daimms/dev/hello-world/.git/

~/dev/hello-world @master
› echo "test" > test_file

~/dev/hello-world @master ?
› git add . && git commit -m "Hello world!"
[master (root-commit) 85e3f5d] Hello world!
 1 file changed, 1 insertion(+)
 create mode 100644 test_file

~/dev/hello-world @master
› ▌
```

Managing Multiple Terminals

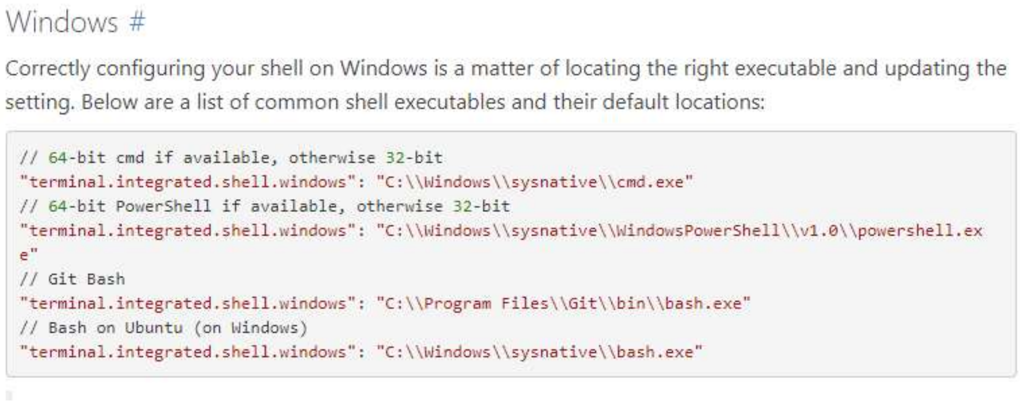
You can create multiple terminals open to different locations and easily navigate between them.

Terminal instances can be added by hitting the plus icon on the top-right of the TERMINAL panel or by triggering the **Ctrl+Shift+`** command.

This creates another entry in the dropdown list that can be used to switch between them



Managing   Multiple Terminals in VSCode



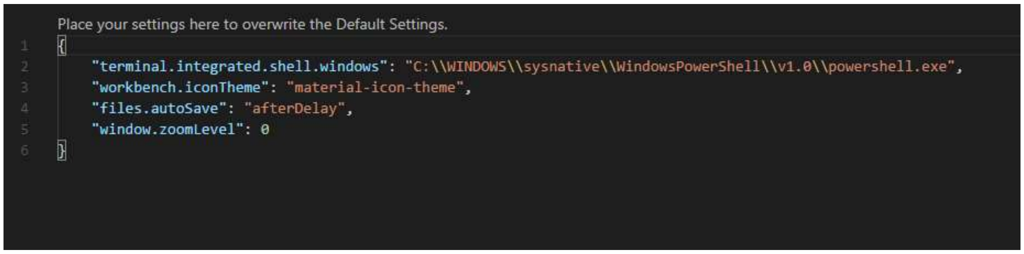
Windows

Correctly configuring your shell on Windows is a matter of locating the right executable and updating the setting. Below are a list of common shell executables and their default locations:

```
// 64-bit cmd if available, otherwise 32-bit
"terminal.integrated.shell.windows": "C:\\Windows\\sysnative\\cmd.exe"
// 64-bit PowerShell if available, otherwise 32-bit
"terminal.integrated.shell.windows": "C:\\Windows\\sysnative\\WindowsPowerShell\\v1.0\\powershell.ex
e"
// Git Bash
"terminal.integrated.shell.windows": "C:\\Program Files\\Git\\bin\\bash.exe"
// Bash on Ubuntu (on Windows)
"terminal.integrated.shell.windows": "C:\\Windows\\sysnative\\bash.exe"
```

- // 64-bit cmd if available, otherwise 32-bit "terminal.integrated.shell.windows": "C:\\Windows\\sysnative\\cmd.exe"
- // 64-bit PowerShell if available, otherwise 32-bit
- **"terminal.integrated.shell.windows": "C:\\Windows\\sysnative\\WindowsPowerShell\\v1.0\\powershell.exe“**
- // Git Bash "terminal.integrated.shell.windows": "C:\\Program Files\\Git\\bin\\bash.exe"
- // Bash on Ubuntu (on Windows) "terminal.integrated.shell.windows": "C:\\Windows\\sysnative\\bash.exe"

Override Default Settings to configure Terminal



Place your settings here to overwrite the Default Settings.

```
{
    "terminal.integrated.shell.windows": "C:\\WINDOWS\\sysnative\\WindowsPowerShell\\v1.0\\powershell.exe",
    "workbench.iconTheme": "material-icon-theme",
    "files.autoSave": "afterDelay",
    "window.zoomLevel": 0
}
```

Demo:
Managing Multiple Terminals in VSCode

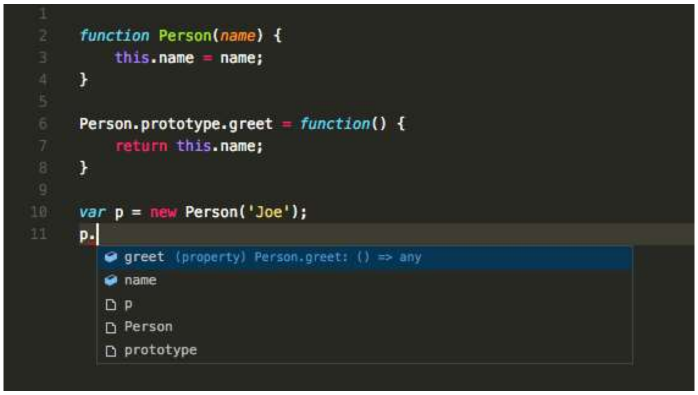
https://code.visualstudio.com/docs/editor/integrated-terminal

- Open C:\src\C# HelloWorld in VSCode
- View->Integrated Terminal
- Add a Terminal/Console
- Go to File Menu->Preferences and change the UserSettings setting "terminal.integrated.shell.windows“ to use powershell C:\\Windows\\sysnative\\WindowsPowerShell\\v1.0\\powershell.exe
- Execute cmdlet Get-Date on terminal
- Execute Get-Host cmdlet in terminal

VSCode IntelliSense

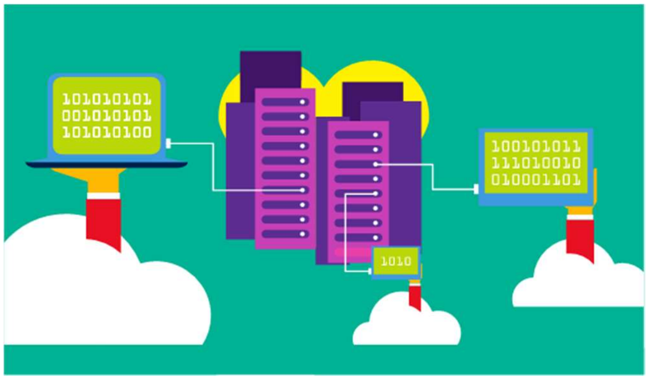
VS Code IntelliSense features are powered by a language service.
Intelligent code completions based on language semantics and an analysis of the source code

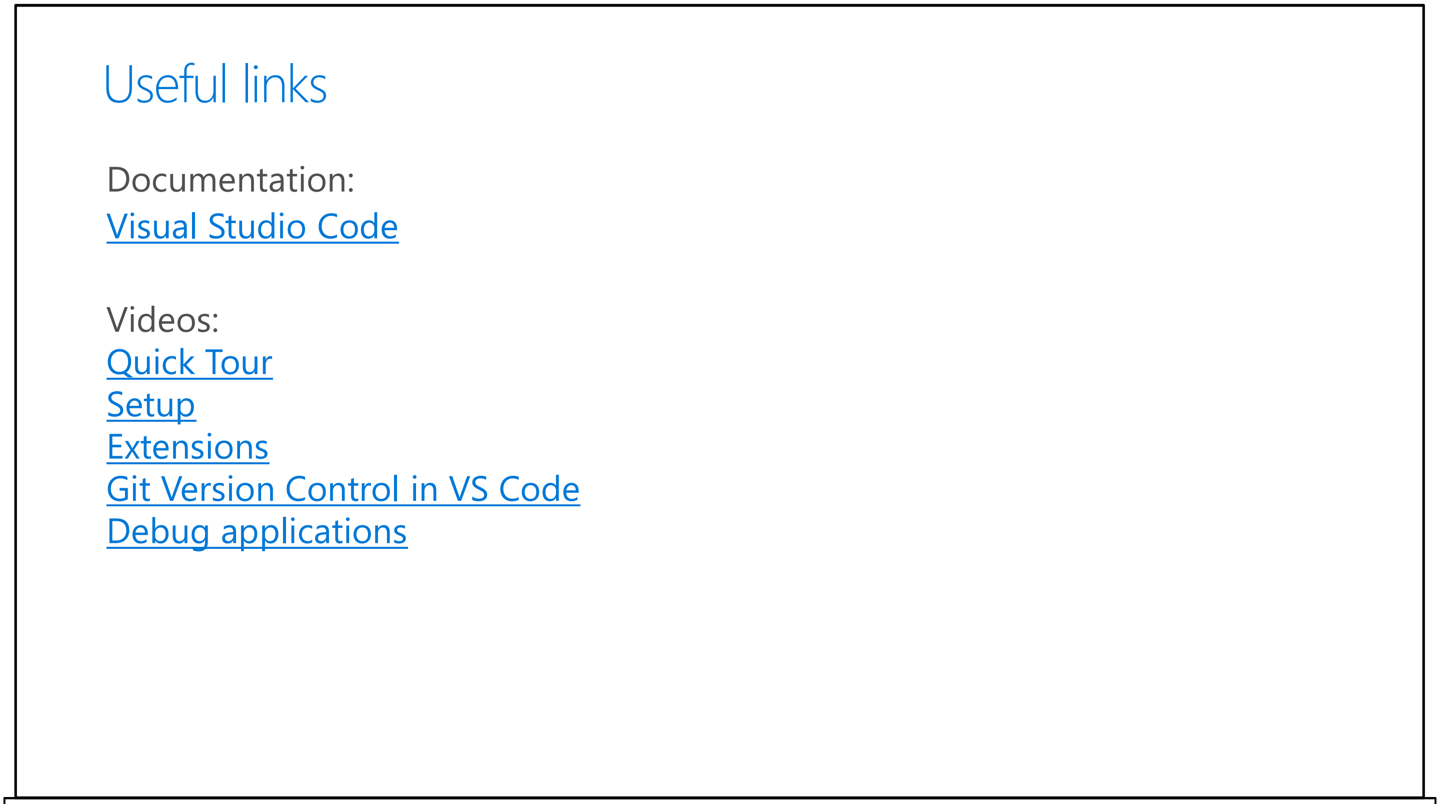
You can trigger IntelliSense in any editor window by typing **Ctrl+Space** or by typing a trigger character (such as the dot character (.) in JavaScript or C#).



```
function Person(name) {
    this.name = name;
}

Person.prototype.greet = function() {
    return this.name;
}

var p = new Person('Joe');
p.
```

Demo:
Explore the Help Menu options



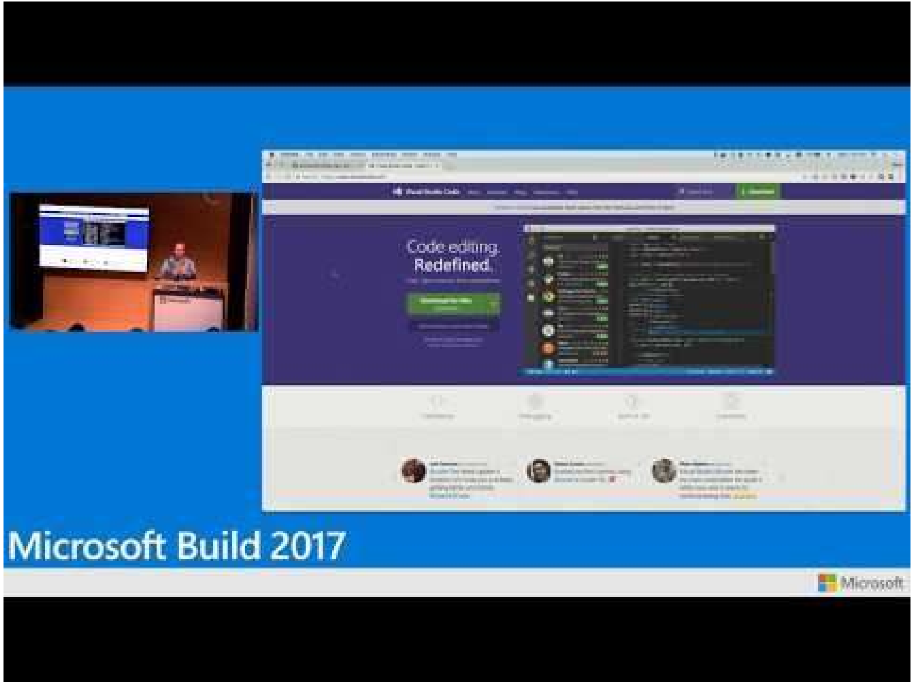


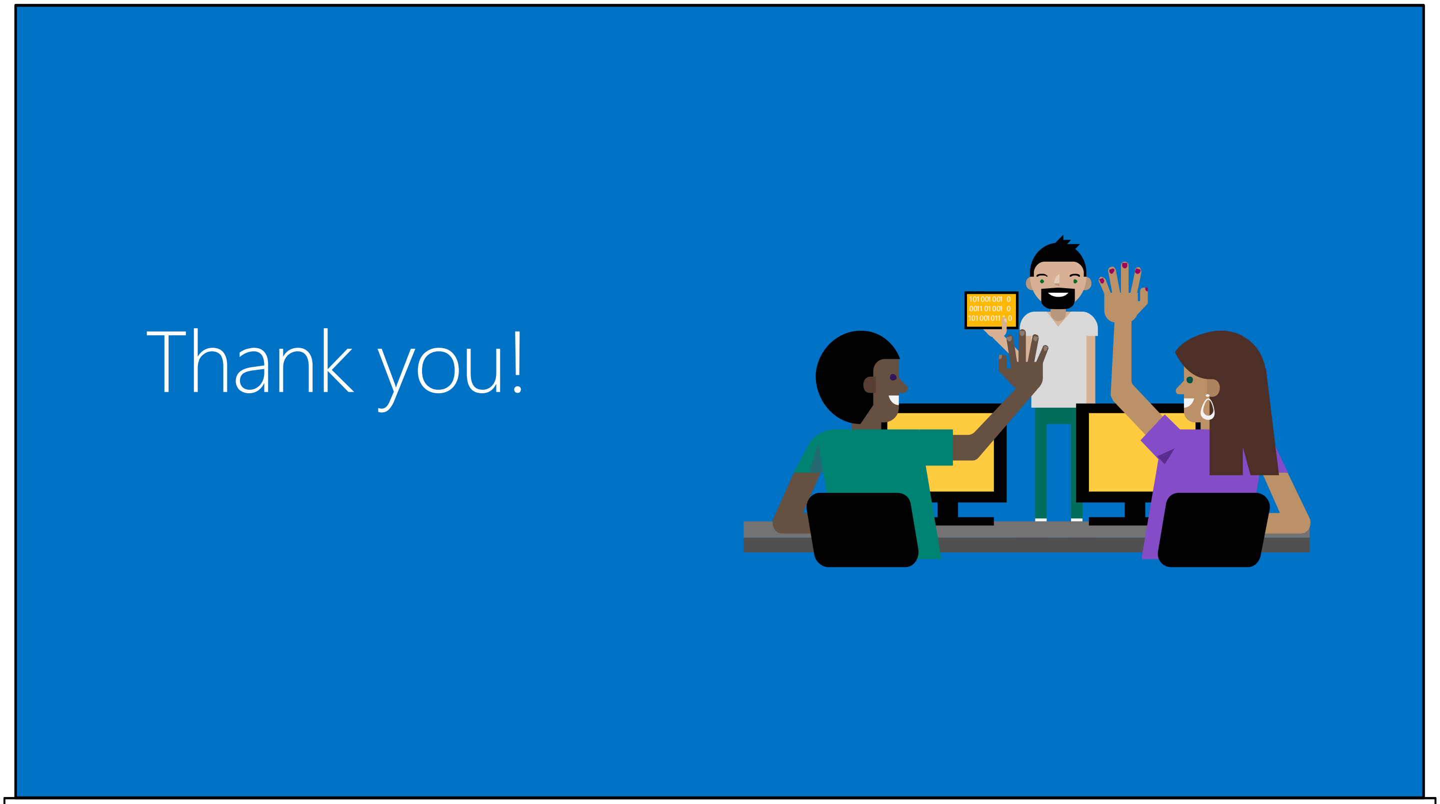
Useful links

Documentation:
Visual Studio Code

Videos:
Quick Tour
Setup
Extensions
Git Version Control in VS Code
Debug applications

VSCode Tips and Tricks from MS Build 2017





Thank you!

Thank you! And Reach out to Chris Tjoumas or me via email if you have questions or would like to set up a Hackathon or Workshop using VS Code!