

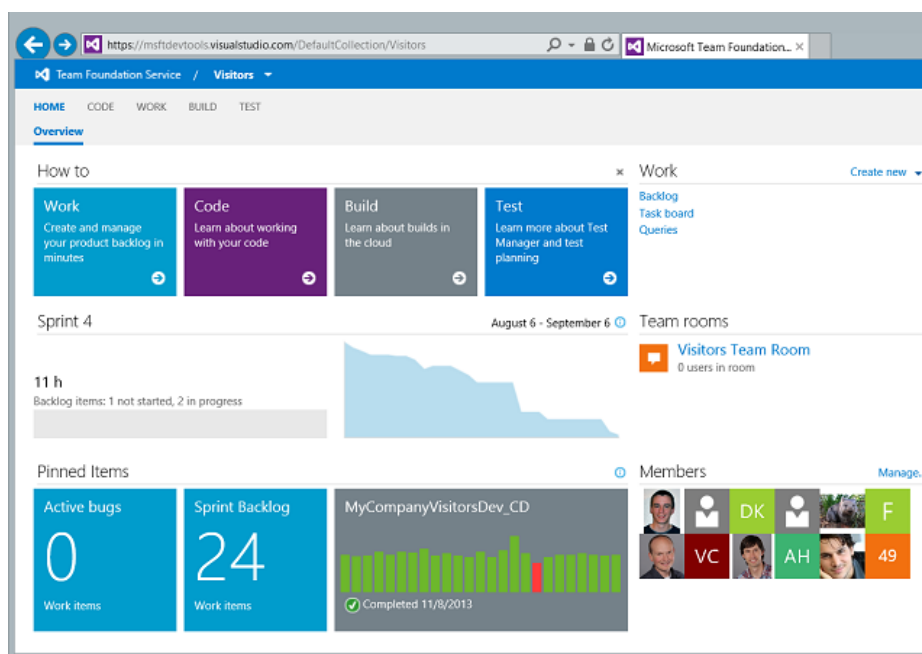
## What is Visual Studio code?

Erich Gamma หนึ่งในผู้นำทีมพัฒนา Visual Studio Code เล่าความหลัง 10 ปีว่าความสำเร็จของ VS Code ที่เราเห็นในปัจจุบัน เกิดจากความล้มเหลวของโครงการก่อนหน้านี้คือ Visual Studio Online ที่เป็น code editor บนเบราว์เซอร์ ตัวของ Gamma เองเป็นหนึ่งในผู้เขียนหนังสือ Design Patterns เคยเป็นพนักงานของ IBM ที่ดูแลโครงการ Eclipse ก่อนย้ายมาอยู่ไมโครซอฟท์ในปี 2011

เขาเล่าว่า “วิสัยทัศน์ของไมโครซอฟท์ไม่ใช่เป็นการสร้าง Visual Studio เวอร์ชันเว็บ แต่เป็นการสร้าง "กล่องเครื่องมือสำหรับนักพัฒนา" ที่เหมาะกับบางสถานการณ์เท่านั้น ตัวอย่างการใช้งาน Monaco ในตอนนี้คือใช้แก้ไขเว็บไซต์ที่โฮสต์อยู่บน Azure และใช้ร่วมกับ "Napa" ซึ่งเป็นเครื่องมือสร้างแอปสำหรับ Office 365 การที่ Monaco ทำงานผ่านเบราว์เซอร์ก็ไม่มีปัญหากับนักพัฒนามากนัก”

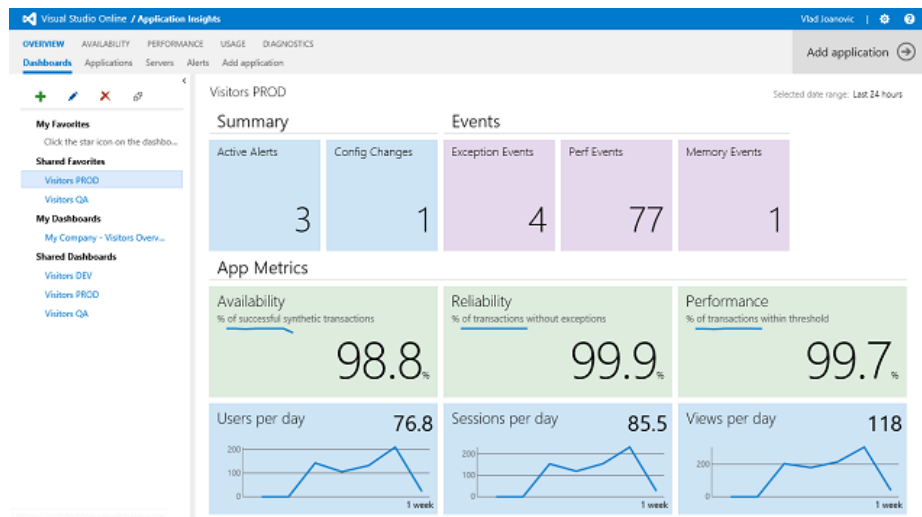
Gamma บอกว่านักพัฒนาในปัจจุบันก็แทบจะออนไลน์ตลอดเวลาอยู่แล้ว การใช้งานแบบออนไลน์จึงเป็นไปได้น้อยมาก Gamma ค่อนข้างได้อิสระจากไมโครซอฟท์ในการทำงานนี้ (เขาบอกว่าเหมือนเป็น startup อยู่ภายในบริษัท) ทีมของเขาเลือกใช้ภาษา TypeScript ที่พัฒนาโดยทีมอื่นจากไมโครซอฟท์เช่นกัน ด้วยเหตุผลว่าทีมของเขาเชี่ยวชาญ JavaScript ที่มีลักษณะคล้ายคลึงกันอยู่แล้ว

เพื่อเป็นส่วนหนึ่งของสิ่งที่เป็นของใหม่จริงๆ ในงานนี้กลับเป็นการเปิดตัว Visual Studio Online บริการกลุ่มเมฆสำหรับการพัฒนาโปรแกรม ถือเป็นส่วนขยายของ Visual Studio ที่อยู่บนอินเทอร์เน็ต (ลักษณะเดียวกับ Office.com ที่เป็นส่วนต่อขยายของ Microsoft Office มาได้สักระยะหนึ่งแล้ว)



Visual Studio Online เป็นชื่อรวมๆ ของบริการหลายอย่างที่เกี่ยวข้องกับการพัฒนาโปรแกรม ที่เปิดตัวตอนนี้ได้แก่

- **Hosted Source Control** มันคือการเอา Team Foundation Version Control ไปรันบนกลุ่มเมฆของไมโครซอฟท์แล้วคิดเงินแบบ "เช่าใช้" ไมโครซอฟท์ยังมี Git เป็น source control ให้เลือกด้วย (จะเรียกว่าเป็น GitHub เวอร์ชันองค์กรก็ได้)
- **Work Items and Agile Planning** บริการด้านการจัดการโครงการด้วยเทคนิค agile
- **Hosted Build Service** เป็น build service หรือการคอมไพล์โปรแกรมบนเซิร์ฟเวอร์ของไมโครซอฟท์ ช่วงนี้ไมโครซอฟท์จัดโปรโมชั่นทดลองใช้งานฟรี บัญชีละ 60 นาที (นับตามเวลาคอมไพล์) ต่อเดือน
- **Elastic Load Test Service** บริการด้านการทดสอบโปรแกรมโดยจำลองโหลดหนักๆ
- **Application Insights** บริการวิเคราะห์ประสิทธิภาพการทำงานของแอป ว่าทำงานราบรื่นดีไหมเมื่อเผชิญกับโหลดจริงๆ (หน้าตาตามภาพ) ตอนนี้ยังอยู่ในขั้นพรีวิว และรองรับเฉพาะแอปที่เขียนด้วย .NET/Java เท่านั้น

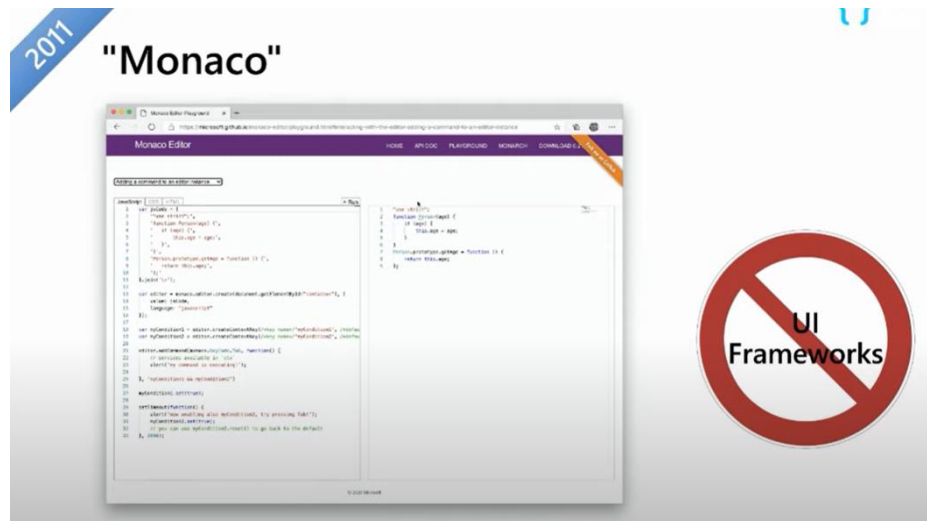


ฟีเจอร์เด็ดอีกอย่างของ Visual Studio Online คือตัวแก้ไขโค้ดเวอร์ชันเว็บแอป ทำงานผ่านเบราว์เซอร์ได้เลย (cloud IDE) ตอนนี้มันยังมีชื่อเรียกเป็นโค้ดเนมว่า "Monaco" และยังอยู่ในสถานะพรีวิวเท่านั้น ไมโครซอฟท์อธิบายว่าไม่ได้ตั้งใจพัฒนา Monaco ให้มาแข่งกับ VS2013 ตัวเต็ม แต่เน้นการแก้ไขงานง่ายๆ เล็กๆ สำหรับโค้ดที่รันอยู่บนกลุ่มเมฆเป็นหลัก เอนจินการทำงานของ Monaco นำบางส่วนมาจากบริการลักษณะเดียวกันของไมโครซอฟท์คือ Office 365 "Napa" (ตัวสร้างแอปสำหรับ Office 365) และตัวแก้ไขไฟล์ออนไลน์ของ SkyDrive เบื้องต้น Monaco เปิดให้ทดสอบแล้ว โดยยังใช้ได้กับการแก้ไขไฟล์เว็บไซต์ที่อยู่บน Azure Websites เท่านั้น

ไอเดียที่น่าสนใจและกลายมาเป็นจุดเด่นของ VS Code ในภายหลังคือ Monaco เลือกไม่ใช่ UI Framework ใดๆ เลย เขียนเองทั้งหมด ด้วยเหตุผลว่าเน้นประสิทธิภาพมาตั้งแต่ต้น จึงต้องการกำหนดชีวิตเอง ไม่พึ่งพา UI Framework เพราะควบคุมประสิทธิภาพโดยตรงไม่ได้



Visual Studio Online ประสบความสำเร็จในระดับหนึ่ง มีผู้ใช้ต่อเดือนจำนวนหลายพันคน แต่ตัวเลขแค่นี้ถือว่าน้อยมากสำหรับบริษัทขนาดไมโครซอฟท์ ที่ต้องการยอดผู้ใช้เยอะกว่านี้ระดับ 10 เท่าตัว



ในปี 2014 ทางทีมของ Gamma จึงปรับทิศทางการใหม่ (pivot) มาเป็น Project Ticino (โค้ดเนมในตอนนั้น) เป็นเครื่องมือพัฒนาแบบข้ามแพลตฟอร์ม ทำงานบน OSX และ Linux ได้ เน้นการแก้ไขโค้ด Node.js และ .NET เป็นสำคัญ VS Code ถูกวางตัวอยู่ตรงกลางระหว่าง editor แบบดั้งเดิม และ IDE เต็มรูปแบบ ช่วงนั้น

มีกระแส Electron เกิดขึ้นพอดี การที่ Monaco เขียนขึ้นเป็นเว็บอยู่แล้ว นำมารันบน Electron ได้ไม่ยาก ทีมของ Gamma ใช้เวลาประมาณหนึ่งปี พัฒนา Monaco มาเป็น Ticino และเปิดตัวในงาน Build 2015 โดยโชว์เดโมการเขียน .NET บนลินุกซ์ เรียกเสียงฮือฮาอย่างมากในตอนนั้น

2014

## Pivot to VS Code

### Project Ticino

Modern, Open, Multi-Platform Developer Tools

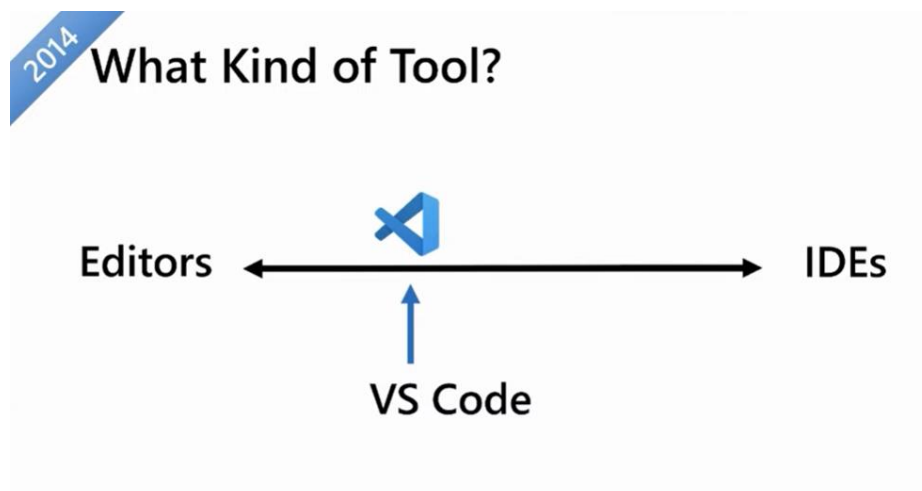
Ticino is a project to build a new primary developer tool in Microsoft's tools portfolio and incubate an associated tools platform and ecosystem. Ticino features a developer tool (the Ticino workbench) that runs on Windows, OSX, and Linux, and is built on a platform design and set of services that can be easily integrated with a more open ecosystem and tools.

In its first incarnation, we will deliver a preview of Ticino, aimed at born-on-the-web developers<sup>1</sup>, with scenarios aimed at multi-platform development, debugging, and deployment of Node.js applications, and cross-platform .NET components, at and beyond BUILD 2015.

#### Why Ticino?

Born-on-the-web developers, broadly face many of the same challenges as traditional developers. But their day-to-day primary tools workflow is very different. Few use general purpose IDEs, opting to do their

From the Ticino 2-Pager



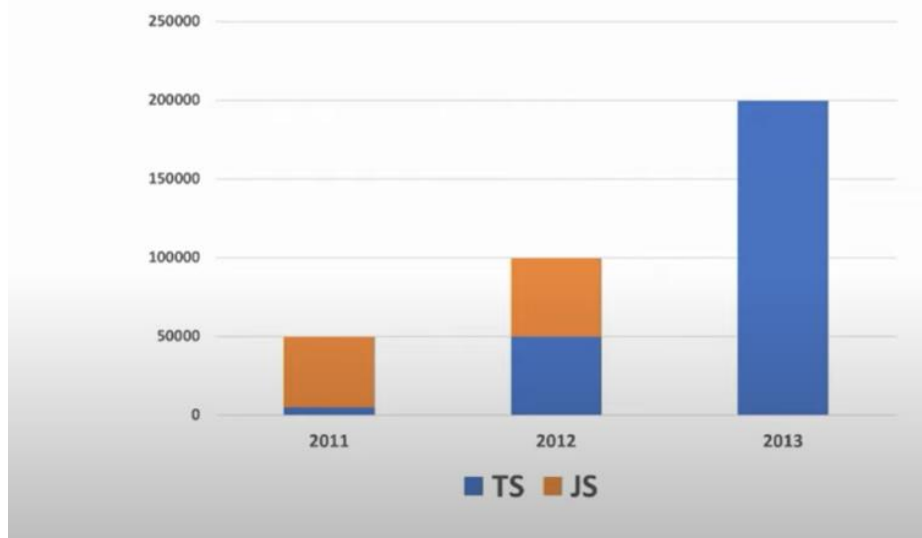
ก้าวถัดมาของ VS Code เกิดขึ้นในช่วงครึ่งหลังของปี 2015 มีตั้งแต่การโอเพนซอร์สบน GitHub และการรองรับส่วนขยาย (extension) Gamma เล่าว่าเขามีประสบการณ์จาก Eclipse ที่มีส่วนขยายจำนวนมาก แต่แนวคิดของ VS Code คือ "ต้องเซฟงานได้เสมอ" ถ้าส่วนขยายมีผลต่อโปรแกรมหลักจนแครช เสียงานที่ยังไม่ได้เซฟไป ก็ถือว่าไม่ดี ดังนั้น VS Code จึงออกแบบส่วนขยายให้รันคนละโพรเซสกับโปรแกรมหลัก และคุยกันผ่าน RPC แทน

## 2015 Connect() [November] Extension API



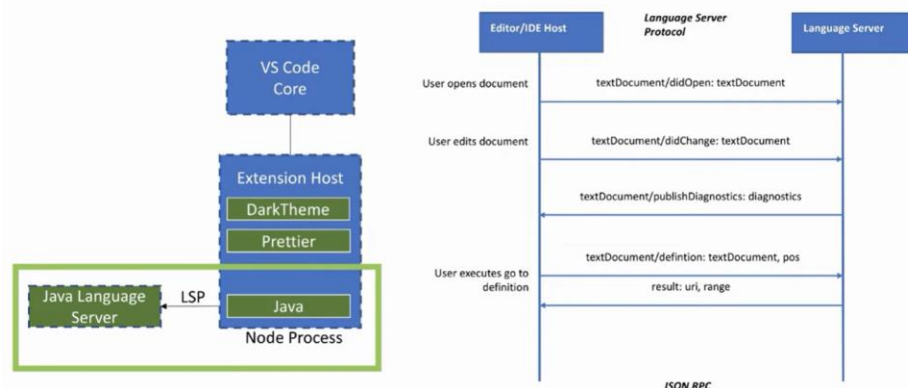
อีกปัจจัยที่ทำให้ VS Code ประสบความสำเร็จคือโครงการ TypeScript ที่ไมโครซอฟท์เริ่มพัฒนาในช่วงไล่เลี่ยกัน (เริ่มปี 2010) ทำให้การพัฒนา VS Code ง่ายขึ้นมาก ช่วงแรก VS Code ยังสร้างด้วย JavaScript เป็นหลัก แต่พอถึงปี 2013 โค้ดทั้งหมดกลายเป็น TypeScript เรียบร้อยแล้ว

## TypeScript ❤️



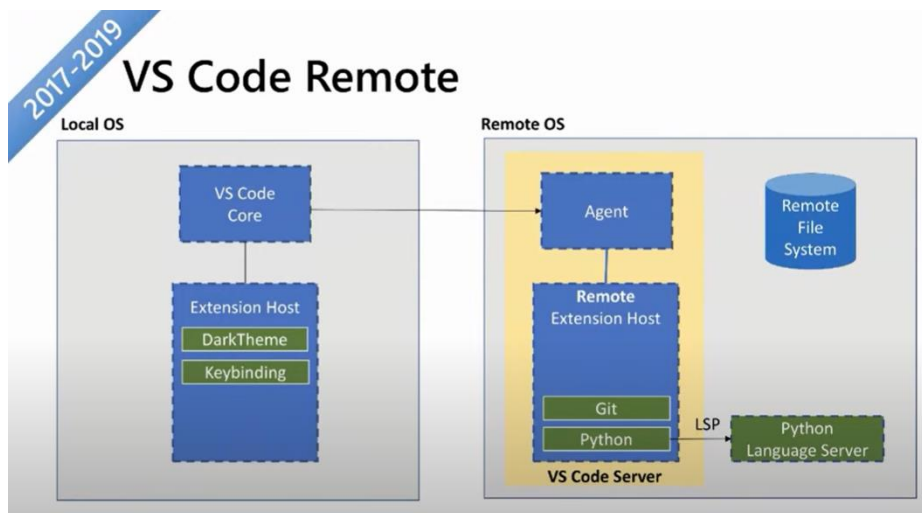
ในปี 2016 เป็นจุดเริ่มต้นของฟีเจอร์สำคัญอีกอย่างคือ Language Server Protocol (LSP) ที่ช่วยให้ VS Code รองรับภาษาโปรแกรมได้เป็นจำนวนมาก ผ่านการสร้าง Language Server โดยชุมชน ไมโครซอฟท์ไม่ต้องทำเองทั้งหมด แค่อัปเดตตัวโปรโตคอล LSP ขึ้นมาให้เป็นมาตรฐานเท่านั้น

## 2016 LSP - Language Server Protocol



ทีมของ Gamma ที่เริ่มพัฒนา VS Code อยู่ที่เมืองซูริกในสวิตเซอร์แลนด์ แต่เมื่อกระแส VS Code เริ่มจุดติดในปี 2016 ไมโครซอฟท์ก็เพิ่มอีกทีมที่สำนักงานใหญ่ใน Redmond โดยผลงานแรกของทีมนี้คือการสร้าง terminal ขึ้นมาภายในตัว VS Code เอง (xterm.js)

ช่วงปี 2017-2019 เป็นความพยายามผลักดัน VS Code ให้ทำงานได้ทุกที่ เริ่มมีแนวคิดของการทำงานรีโมทเข้ามา (เขียนบนเครื่อง รันบนอีกเครื่อง) แนวคิดนี้เริ่มได้รับความนิยมเมื่อเกิดกระแส container และไมโครซอฟท์เองมีลินุกซ์ WSL รันอยู่ในวินโดวส์

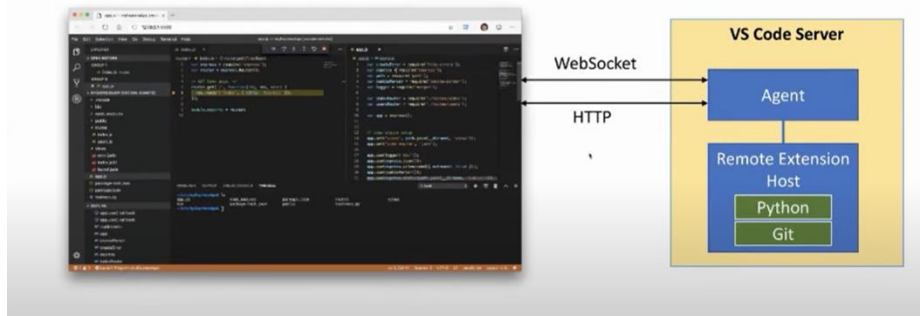


ปี 2020 เป็นการนำ VS Code กลับมาสู่รากเหง้าคือเว็บอีกครั้ง เป้าหมายคือการใช้งานกับ GitHub Codespaces ที่สามารถกดแก้โค้ดได้จากหน้าเว็บ GitHub แล้วส่งคอมไพล์ได้เลย Gamma เล่าว่าการที่ VS Code รันบน Electron อยู่ก่อน ต่างจากเบราว์เซอร์อยู่บ้าง จึงต้องปรับโค้ดข้างใต้ใหม่ให้เป็นเวอร์ชันเดียวที่รันได้ทั้งบน Electron และเบราว์เซอร์

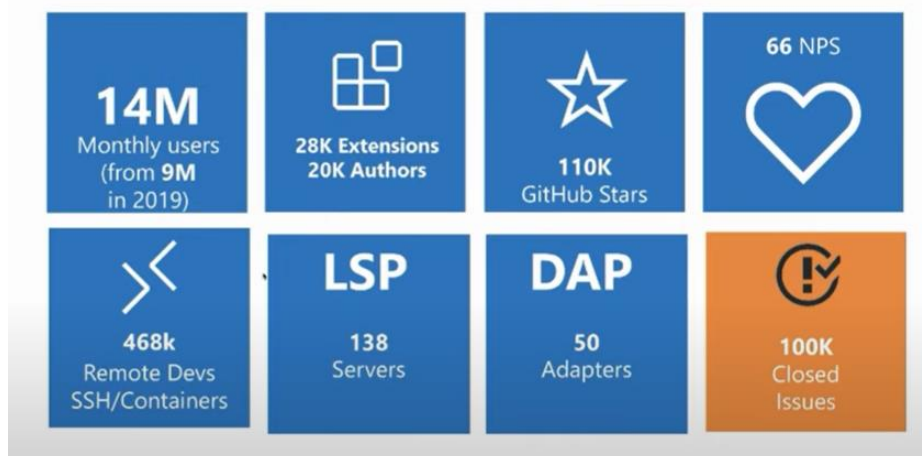


2020

## Back in the Browser - VS Code Web



ปัจจุบัน VS Code มีผู้ใช้ 14 ล้านคนต่อเดือน ส่วนขยาย 28,000 ตัว, รองรับ LSP 138 ตัว และมี Debug Adaptor Protocol (DAP) สำหรับเชื่อมต่อ debugger ลักษณะเดียวกับการเปิด LSP ให้เชื่อมต่อภาษาด้วย



แนวทางการพัฒนา VS Code คือออกรุ่นใหม่ทุกเดือน ประกาศแผนต่อสาธารณะบน GitHub การออกรุ่นใหม่ทุกเดือนทำให้ทีมงานต้องขยันปรับแก้โค้ดให้ดีขึ้น ช่วยลดหนี้ทางเทคนิค (technical debt) ลงได้ตลอดเวลา ฟังเสียงของผู้ใช้ และยังคงแนวคิดดั้งเดิม "ไฟท์ที่ประสิทธิภาพ"

## How to use Visual studio code

- วิธีการดาวน์โหลดโปรแกรม

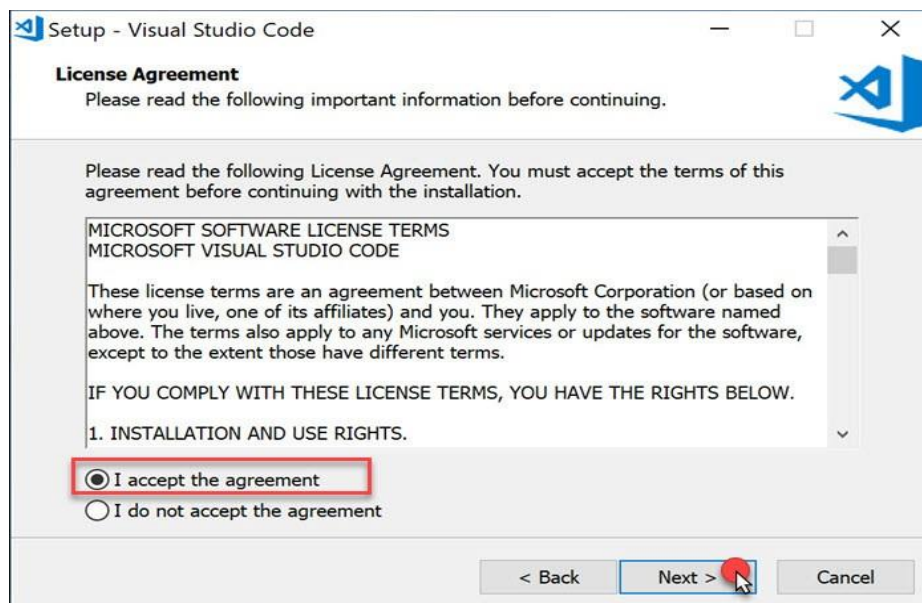
อันดับแรกเราต้องเข้าไปที่เว็บ <https://code.visualstudio.com/> เมื่อเข้าเว็บแล้วให้กดปุ่มดาวน์โหลด

- วิธีการติดตั้งโปรแกรม

- เมื่อเปิดตัวติดตั้งขึ้นมาหน้าจอแสดงว่า ( Welcome to the Visual Studio Code Setup Wizard ) แล้วเราให้กดปุ่ม Next >

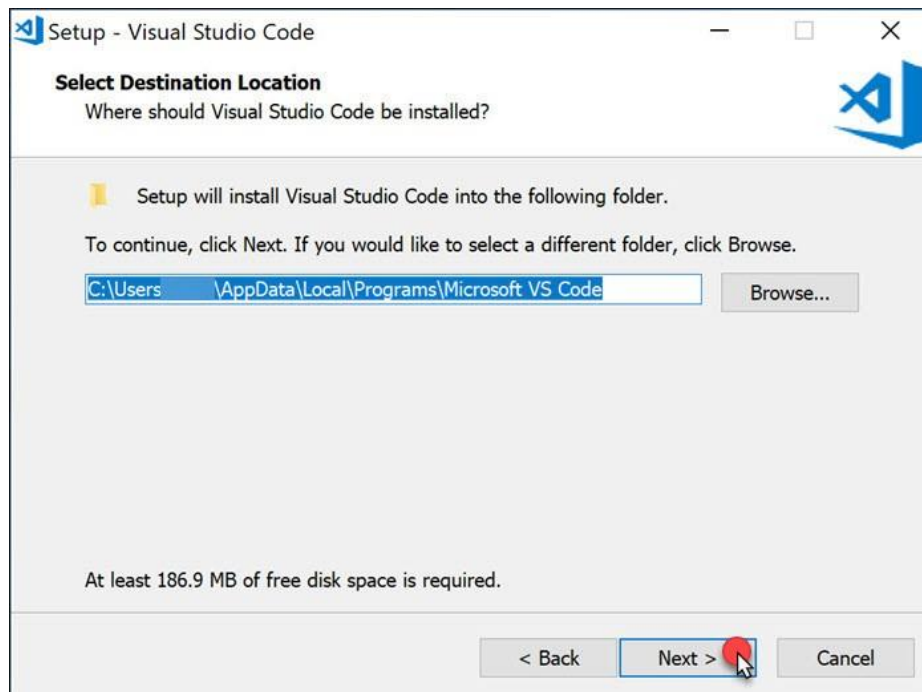


- หลังจากนั้นก็จะแสดงหน้าจอข้อตกลงของโปรแกรม ( License Agreement ) แล้วเลือก ☒ I accept the agreement หลังจากนั้นก็กด Next >

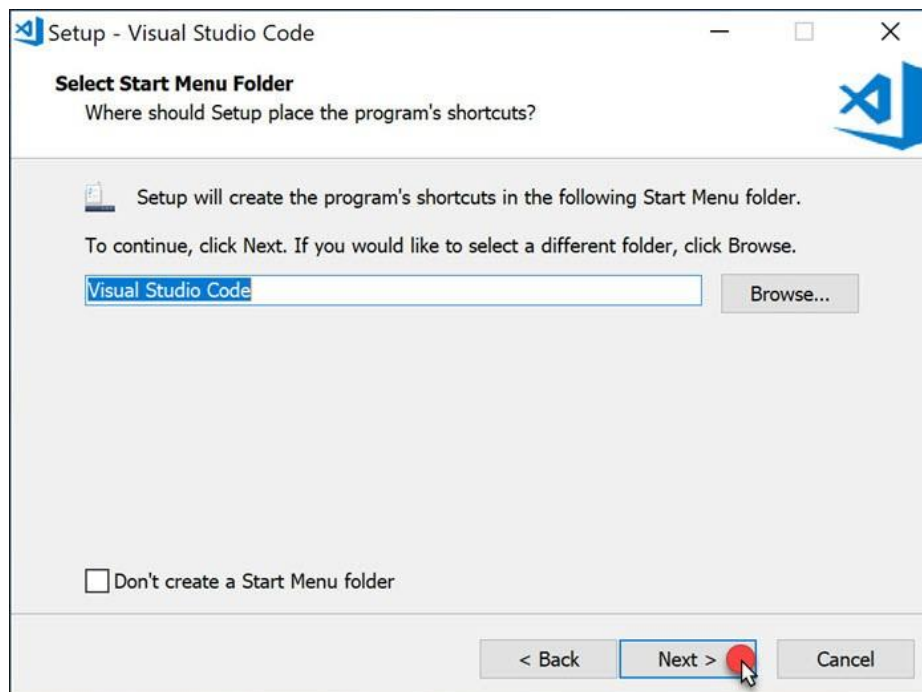




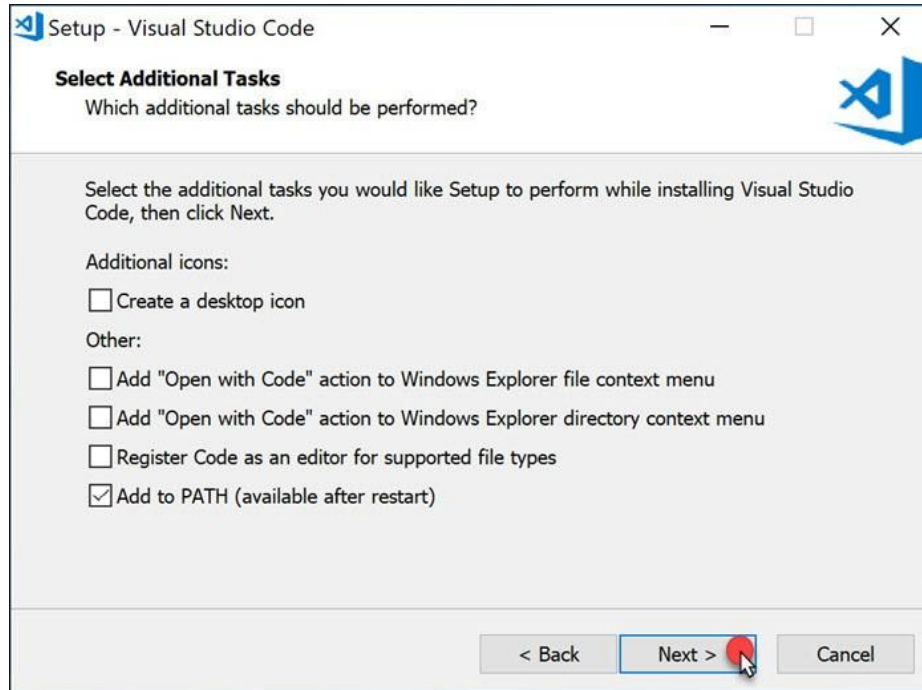
- หลังจากนั้นจะแสดงหน้าจอหน้าต่างนี้จะแสดงโฟลเดอร์ที่จะใช้เก็บไฟล์ของโปรแกรม ( Select Destination Location ) แล้วก็กดปุ่ม Next >



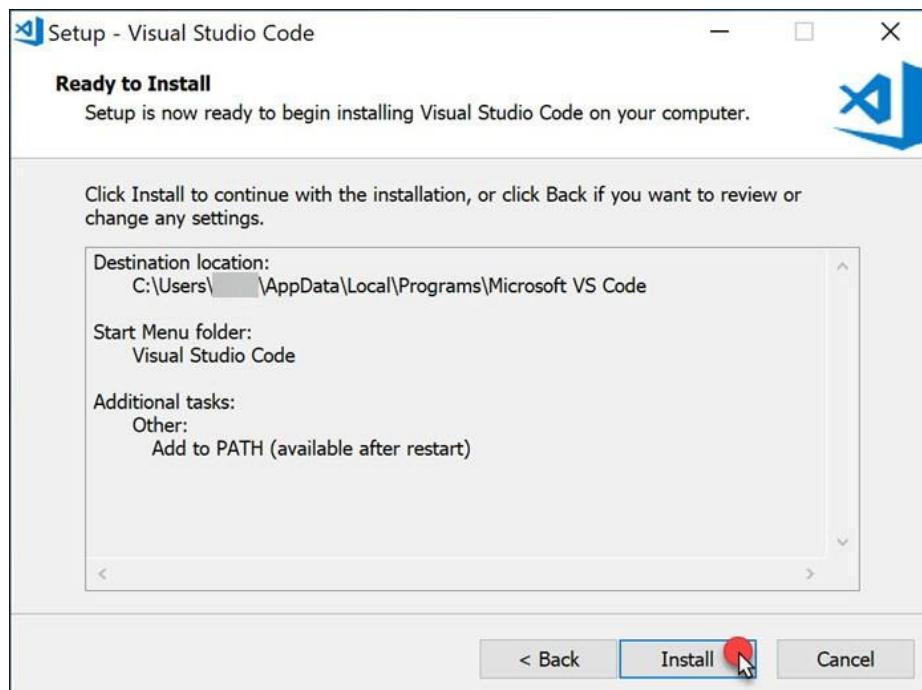
- จากนั้นจะแสดงหน้าจอเลือก ( Select Start Menu Folder ) แล้วก็กดปุ่ม Next >



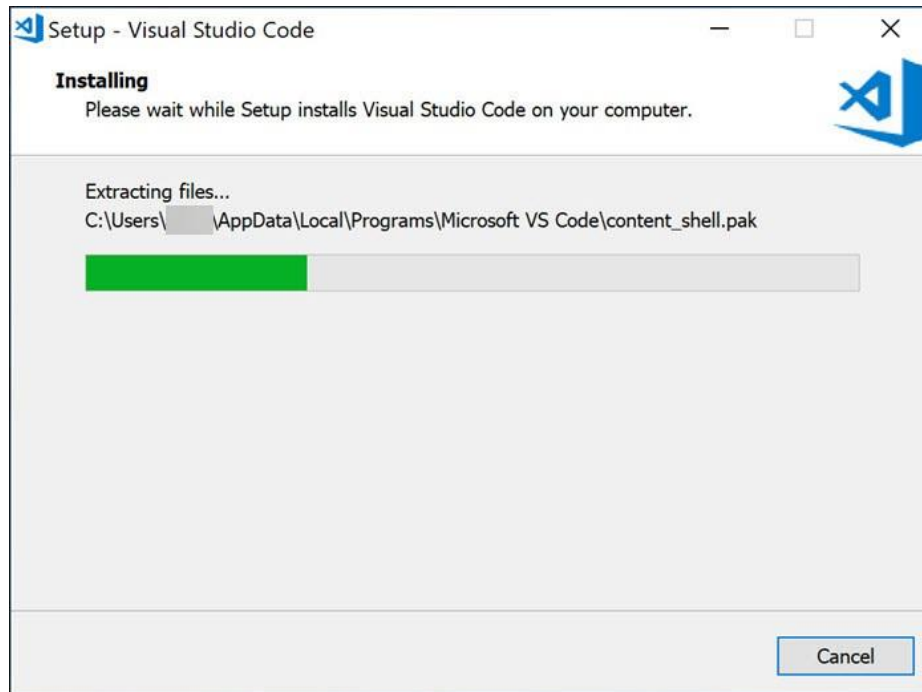
- จากนั้นก็จะแสดงหน้าจอ ( Select Additional Tasks ) แล้วก็กดปุ่ม Next >



- หลังจากนั้นจะแสดงหน้าจอ ( Ready to Install ) แล้วก็กดปุ่ม Install



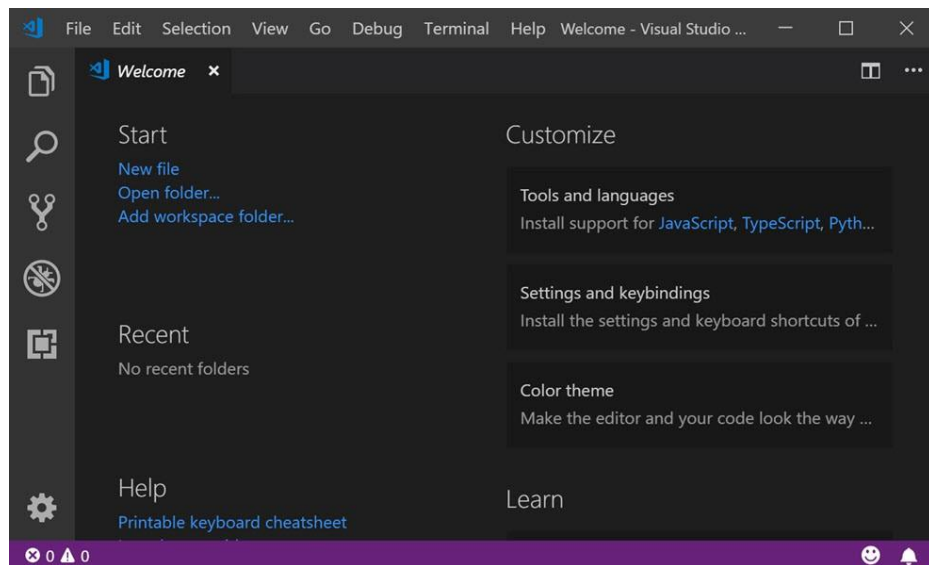
- จากนั้นรอโปรแกรมทำการติดตั้งสักครู่



- เมื่อโปรแกรมทำการติดตั้งเสร็จแล้วจะแสดงหน้าจอ ( Completing the Visual Studio Code Setup Wizard ) ให้กดปุ่ม Finish >

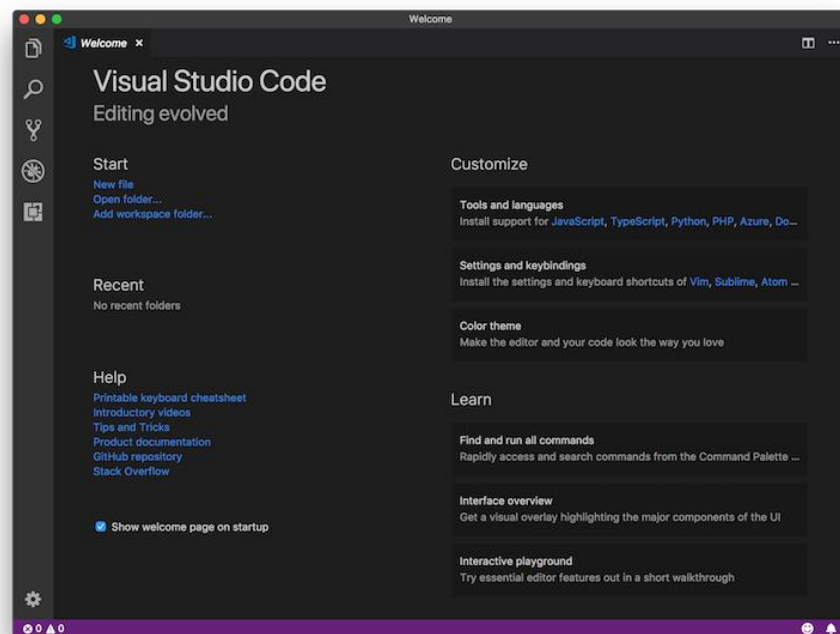


- จากนั้นโปรแกรม Visual Studio Code จะทำการเปิดขึ้นมา ก็เป็นถือว่าทำการติดตั้งเสร็จในขั้นตอนการติดตั้งโปรแกรม Visual Studio Code

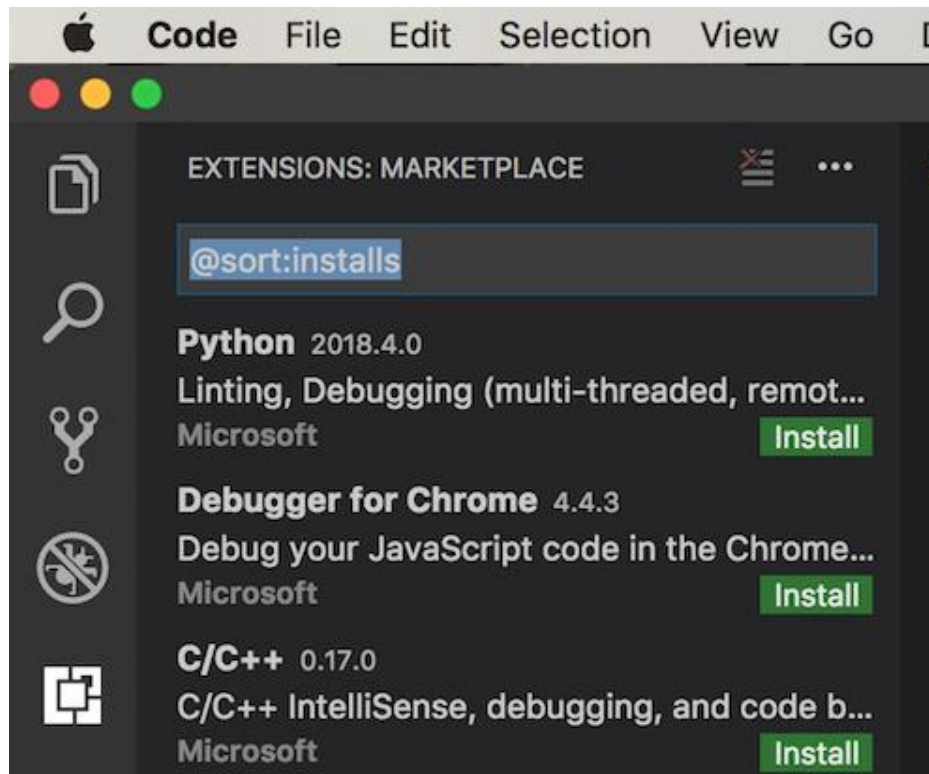


## - วิธีการใช้งาน Visual Studio Code

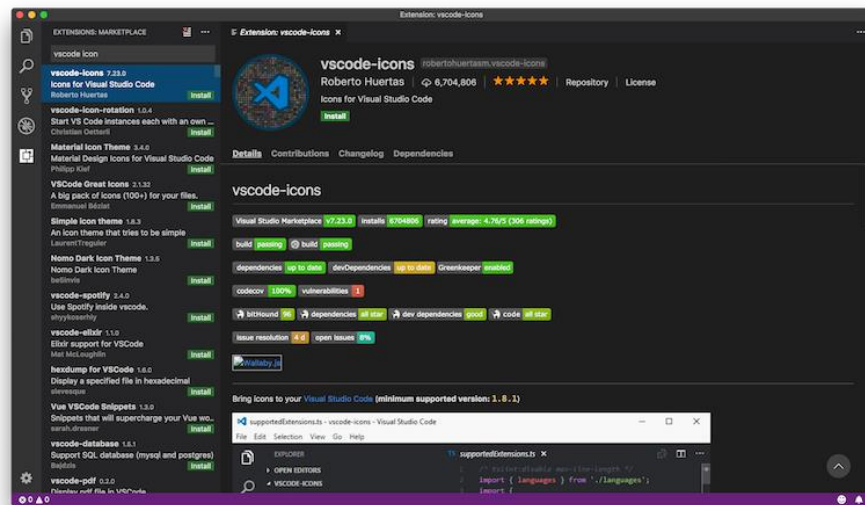
- เปิดโปรแกรม Visual Studio Code ขึ้นมา



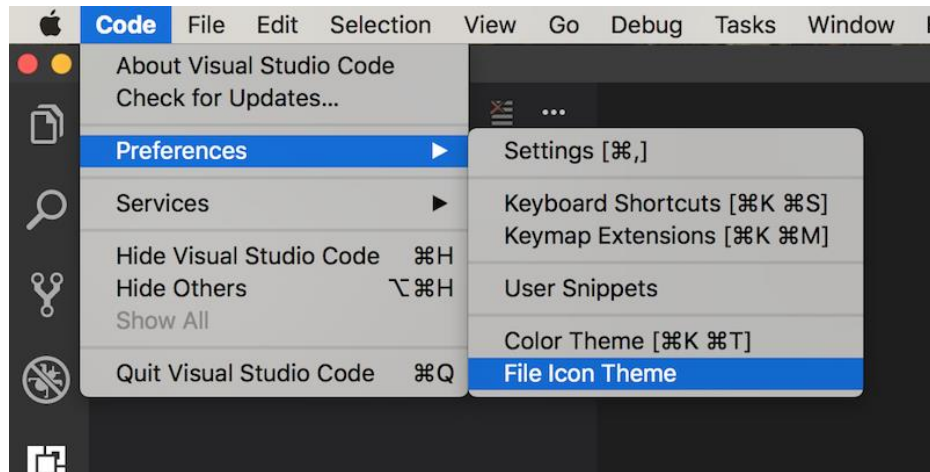
- สิ่งแรกที่ต้องก็ ทำคือ ติดตั้ง Plugin ไปที่ Search



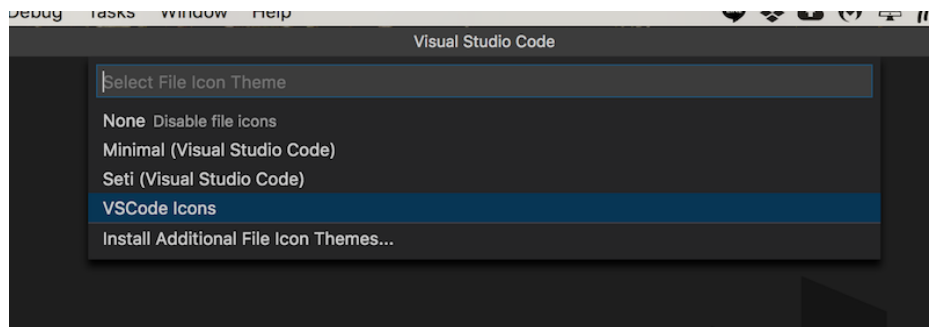
- แล้วค้นหาคำว่า vscode icon หลังจากนั้นติดตั้งได้เลย



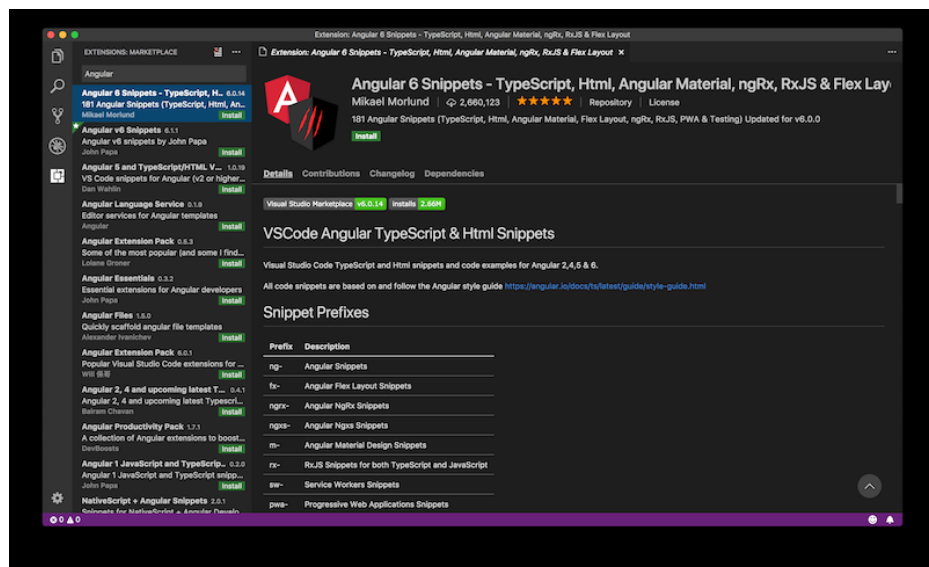
- การเปิดใช้งาน Code > Preferences . File Icon Theme



- แล้วกดเลือก VSCode Icons



- Plugin ตัวต่อไป Snippet ที่จะช่วยทำ Auto Complete



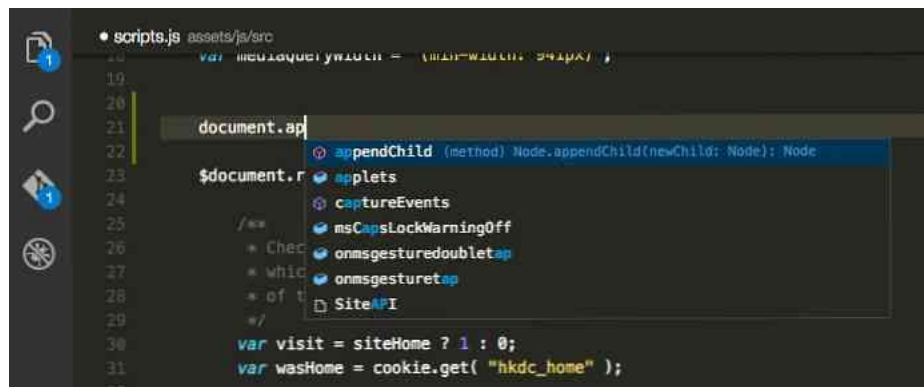


## - เทคนิคการใช้งาน Visual Studio Code (Editor) ที่น่าสนใจ

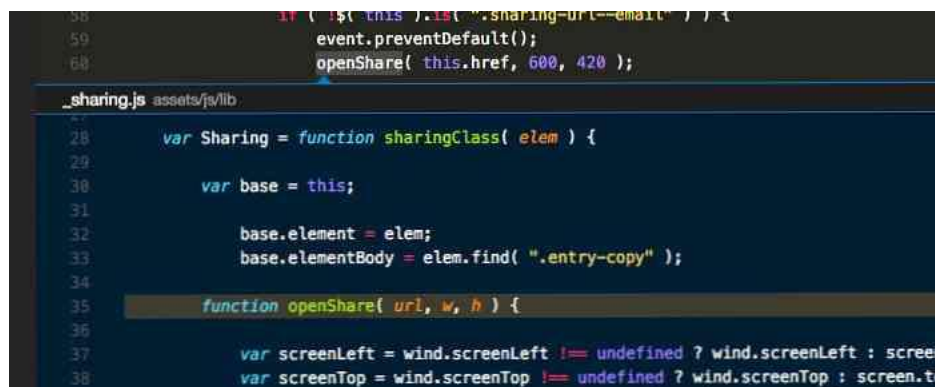
- Ctrl+P เปิดไฟล์อย่างรวดเร็ว โดยการพิมพ์ชื่อไฟล์ได้เลย
- Ctrl+Shift+F ค้นหาแบบรวดเร็วในไฟล์ต่างๆ
- Ctrl+Shift+P แสดงชุดคำสั่งต่างๆ เช่น ค้นหาการตั้งค่าของ editor ลองพิมพ์ color theme เพื่อเปลี่ยน theme ของ editor
- ลองใช้ฟอนต์ Menlo-Regular (สวยดี) ติดตั้งลง windows ก่อน แล้วตั้งค่าที่ User Settings ดังนี้ “editor.fontFamily”: “‘Menlo Regular’, Consolas, ‘Courier New’, monospace”,
- Ctrl+B เปิด-ปิด Sidebar
- ซ่อน แถบ Open Editors ใน EXPLORER ตั้งค่าที่ User Settings ดังนี้ “explorer.openEditors.visible”: 0,
- ซ่อน mini map ด้านขวาจะได้ดูสะอาดๆ ตั้งค่าที่ User Settings ดังนี้ “editor.minimap.enabled”: false,
- สำหรับคนที่เขียน PHP ให้ติดตั้ง extensions phpintelephense และ php debug
- ลองติดตั้ง extension ชื่อว่า snippet creator ไว้สร้าง snippet ใช้เองได้
- หากพิมพ์โค้ดแล้ว เวลากด tab ต้องการใช้งาน snippet ด้วย ตั้งค่าที่ User Settings ดังนี้ “editor.tabCompletion”: true,.
- หากต้องการใช้ terminal เป็น git แทน powershell หรือ cmd ก็ตั้งค่าที่ User Settings ดังนี้ “terminal.integrated.shell.windows”: “C:\Program Files\Git\bin\bash.exe”,  
“terminal.integrated.shellArgs.windows”: [  
“-l”,  
“-j”  
]
- ใช้ Ctrl+K เพื่อเคลียร์ข้อความใน terminal
- จัดรูปแบบโค้ด ใช้ Shift+Alt+F
- หากต้องการจัดรูปแบบโค้ดตอนบันทึกไฟล์ทุกครั้งก็ให้ตั้งค่า User Settings ดังนี้ “editor.formatOnSave”: true
- สำหรับคนที่เขียน Angular ติดตั้ง extension ชื่อว่า Angular Essentials ได้เลย
- สำหรับคนที่เขียน Vue.js ก็ใช้ extension ชื่อว่า Vetur
- สำหรับคนที่ใช้ Git แนะนำ extensions ชื่อว่า Git History และ GitLens
- กด Alt ค้างไว้แล้วคลิกเมาส์ที่บรรทัดนั้นๆ หากต้องการใช้ multi-cursor
- หากใครใช้ Sublime Text ก็ให้ติดตั้ง extension ชื่อว่า Sublime Text Keymap เพื่อใช้ keyboard shortcuts ที่คุ้นเคยได้

- ประโยชน์ของ Visual studio code

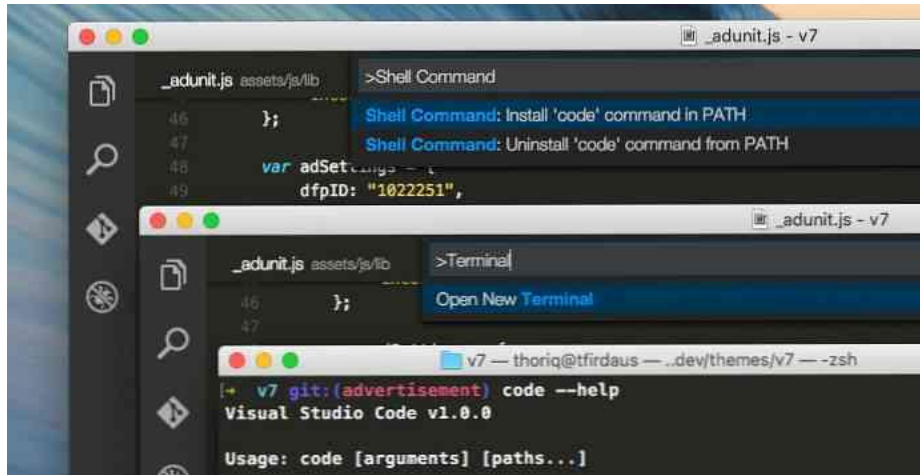
- **Intellisense** เป็นรหัส เติมข้อความอัตโนมัติตรงตามปัญญาประดิษฐ์. ยูทิลิตีนี้มีรายการคำแนะนำพร้อมกับคำใบ้หรือคำอธิบายสั้น ๆ ในขณะที่เรากำลังเขียนรหัส เหล่านี้คือ ได้มาจากปัจจัยตามบริบทหลายประการ เช่นภาษาการเขียนโปรแกรมไวยากรณ์ตัวแปรฟังก์ชัน เช่นเดียวกับรหัสทั้งหมดที่อยู่ในไฟล์ IntelliSense รองรับภาษาการเขียนโปรแกรมจำนวนมาก รวมถึง Sass, LESS, JavaScript, TypeScript และ PHP ส่วนขยายบางรายการที่จะรวม IntelliSense สำหรับภาษาการเขียนโปรแกรมอื่น ๆ ก็มีให้เช่นกัน มันเป็นคุณสมบัติที่ยอดเยี่ยมในการปรับปรุงประสิทธิภาพ



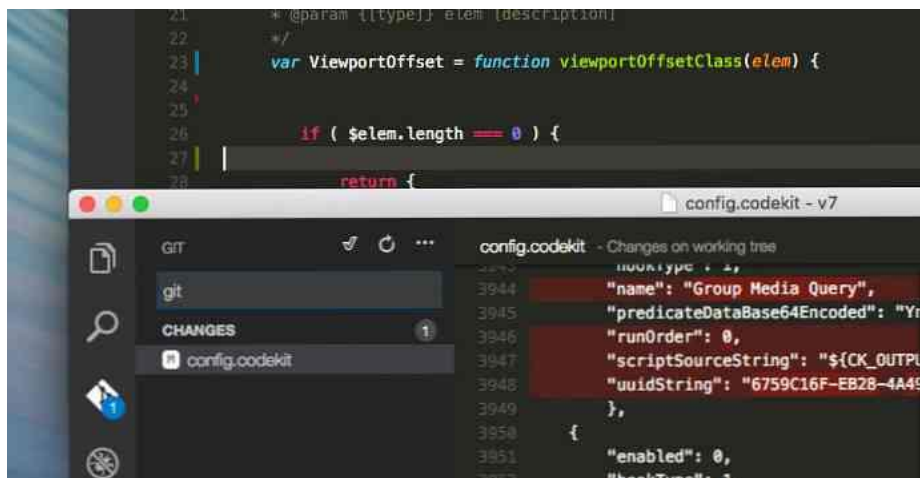
- **การมอง** ในขณะที่เขียนโค้ดคุณมักจะลืมฟังก์ชันบางอย่างโดยที่ฟังก์ชันนั้นถูกกำหนดไว้ในตอนแรกและสิ่งที่เป็นพารามิเตอร์ที่จำเป็น กับ แอบมอง, คุณสามารถเลือกฟังก์ชันจากนั้นกด Shift + F12 การคัดลอก ขยายเข้าไปในหน้าต่างออนไลน์แสดงคำจำกัดความที่สมบูรณ์ของฟังก์ชัน เช่นเดียวกับที่กำหนดฟังก์ชัน ปัจจุบันคุณลักษณะนี้ทำงานใน C, C #, JavaScript, TypeScript, .NET และภาษาการเขียนโปรแกรมอื่น ๆ



- CLI คล้ายกับ SublimeText ที่มี subl บรรทัดคำสั่งให้ Visual Studio ติดตั้ง CLI ชื่อ รหัส และติดตั้งง่าย ใน Windows CLI จะถูกติดตั้งพร้อมกับแอปแล้ว CLI สามารถเข้าถึงได้ผ่าน Command Prompt ใน OS X สามารถติดตั้งและถอนการติดตั้งผ่าน Command Palette



- Git ในตัว Visual Studio Code มี Git อยู่ในตัวแก้ไข ที่ด้านซ้ายของแถบด้านข้างคุณจะพบ ไอคอน Git ที่คุณสามารถเริ่มต้น Git ได้เช่นกัน ดำเนินการคำสั่ง Git หลายรายการ เช่น ผูกมัด, ดึง, ดัน, rebase, ประกาศ, และดูการเปลี่ยนแปลงภายในไฟล์ นอกจากนี้หากคุณกำลังทำการเปลี่ยนแปลงในพื้นที่เก็บข้อมูล Git, Visual Studio จะ แสดงตัวบ่งชี้สีใน gutter editor, ระบุตำแหน่งที่คุณทำการแก้ไข



- Task Runner Visual Studio ยังมี Task Runner ในตัวซึ่งให้ความสะดวกสบายในระดับหนึ่ง หากต้องการใช้คุณลักษณะนี้เราสามารถตั้งค่า tasks.json ไฟล์หรือผ่านการกำหนดค่างานยอตนิยมบางอย่างเช่น Grunt, Gulp หรือ MSBuild หากมีให้ เมื่อตั้งค่าการกำหนดค่าแล้วเราสามารถเรียกใช้ Task Runner ผ่าน Command Palette โดยพิมพ์ เรียกใช้งาน. การพิมพ์ งาน จะแสดงรายการคำสั่งที่เกี่ยวข้องทั้งหมด