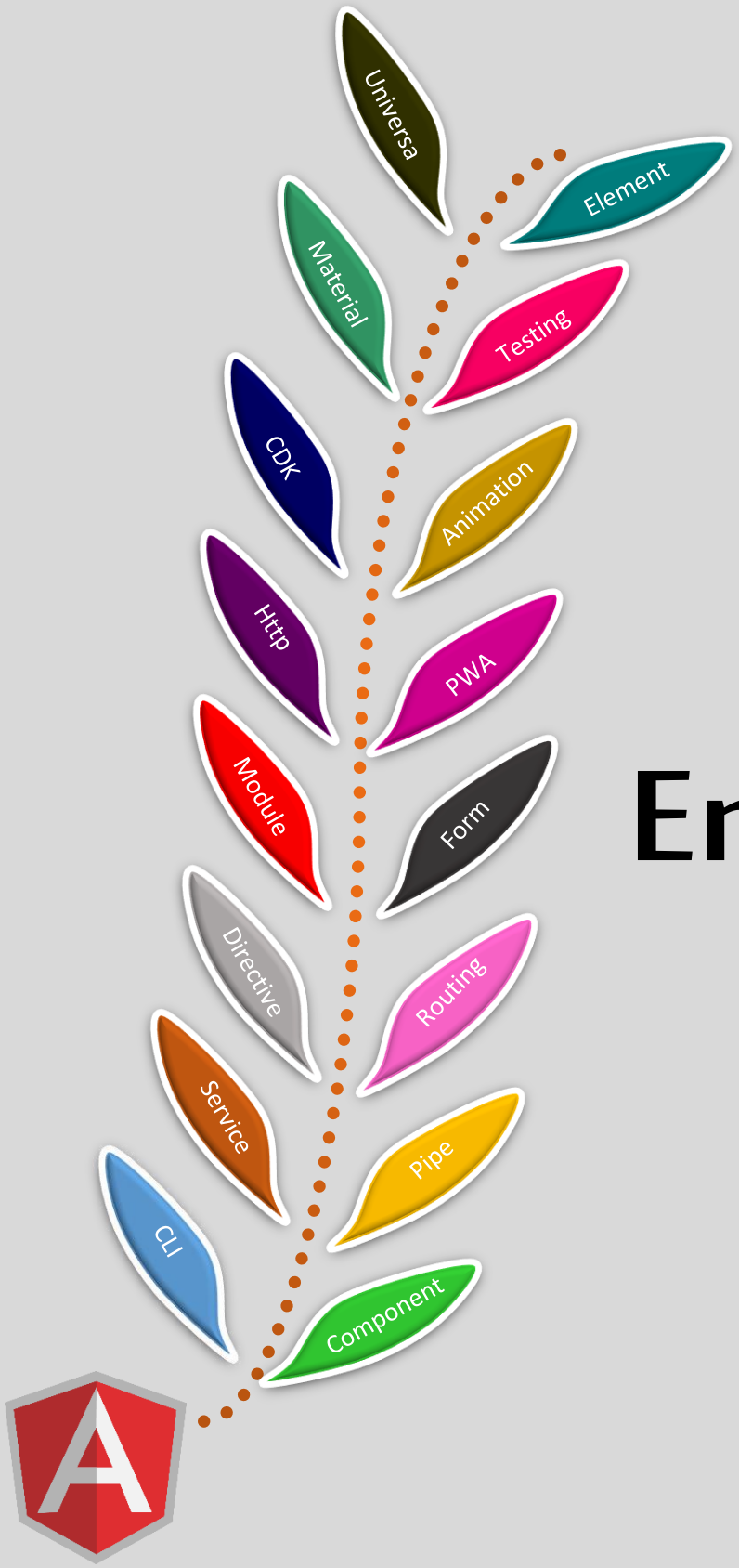
سلسلة تعلم Angular بالعربي



1

Angular Environment Setup

2020

فيصل الفهد

Angular Environment Setup

الكتاب الأول من

سلسلة تعلم Angular بالعربي

تم انتاج هذا العمل في عام

٢٠٢٠

المؤلف

فيصل الفهد

<u>إهداء</u>

إلى أطهر قلبين في حياتي... والديَّ العزيزين.

إلى من شاركني السرَّاء والضراء، ولم أرها عابسة يومًا..... زوجتي المخلصة.

إلى من أتشوَّق لأن أرى مستقبلهما المشرق بإذن الله..... ابنائي وبناتي.

إلى جميع متلهف للعلم ويتوق للمعرفة

أهديكم هذا الكتاب المتواضع، وأدعو الله أن يحوز إعجابكم.

المؤلف

مقدمة السلسلة

اللهم علمنا ما ينفعنا وانفعنا بما علمتنا إنك انت العليم الحكيم..

أضع بين إياديكم أحبتي وأخوتي مطوري الويب وبالتحديد مطوري Front End أول سلسلة عربية تتكلم عن إطار عمل Angular، حيث تقوم فكرة هذه السلسلة على حصر جميع الميزات التي يُقدمها Angular ومن ثم في كل كتاب يتم أخذ ميزة او ميزتين والإبحار فيها بشكل مستقل محاولاً بذلك تغطية أغلب وأهم جوانبها مع الشرح المفصل والأمثلة المتعددة.

ومن هذا المنطبق يجب التنويه أن هذه السلسلة ليست للمبتدئين في البرمجة، وانما لابد ان تمتلك عزيزي المتعلم على الأقل أساسيات أركان تطوير الويب الثلاثة HTML-CSS-JavaScript ومن ثم لابد ان تمتلك اساسيات Typescript لأن إطار عمل Angular يعتمد على هذه التقنية، ولا يخفى على كل مطور واجهات أمامية أهمية Typescript والتي من وجهة نظري أرى انه يجب على كل مطور ويب تعلمها واتقانها، ونكتفي ان نقول ان Deno.js وهي اللغة الجديدة من Node.js أصبحت تدعم هذه التقنية بشكل كامل، لذلك انصح كل مطور عربي بتعلم هذه التقنية ومحاولة الإلمام بجوانبها وكيفية الاستفادة منها في إطار عمل Angular بشكل خاص ولتطوير الويب بشكل عام.

ولا يخفى عليك عزيزي المتعلم أن الكمال لله ولا يخلوا عمل من الأخطاء فالخطأ وارد والتقصير موجود، لذلك في حال وجدت أي تقصير او أي خطأ في هذه السلسلة او أردت تقديم أي اقتراح لتطوير هذه السلسلة فلا تتردد بمراسلتي على البريد الإلكتروني الذي سوف اضعه بأسفل هذه الصفحة.

وأخيراً أسأل الله العلي العظيم أن ينفع به وان يتقبله عملاً خالصاً لوجهه سبحانه،

ولا تنسوني أخوتي من صالح دعائكم.

المؤلف

فيصل الفهد

Faisal.alfahd.ksa@gmail.com

جدول المحتويات

مقدمة الكتاب

في هذا الكتاب نبدأ معك عزيزي المتعلم في تعلم كيفية تهيئة إطار عمل Angular حيث نحتاج إلى تعلم بعض الأدوات المساعدة مثل npm ومعرفة كيفية التعامل مع محرر الأكواد وأدوات المطور في المتصفحات الحديثة، ومن ثم نتعلم كيف نقوم بإنشاء مشروع جديد مع شرح لأغلب وأهم ملفات المشروع التي يتم تنزيلها بشكل افتراضي مع بداية أي مشروع Angular جديد ومن ثم ننتقل الى جزء مهم وهو Angular CLI ونتعلم كيف نقوم بالتعامل مع هذه الميزة Feature مع شرح أغلب وأهم أوامرها وشرح الخيارات Options لكل أمر من هذه الأوامر أن وجد، وأخيراً نشرح بشكل سريع كيف يتم إضافة مكتبات خارجية إلى المشروع الخاص بنا.

لذلك جعلته أول كتاب من هذه السلسلة لأن جميع هذه المعارف السابقة تعتبر الأساس لتعامل مع Angular حيث تأتي بعد العلوم الأساسية التي أشرت لها سابقاً في مقدمة السلسلة وهي HTML-CSS-JavaScript-Typescript.

المؤلف

فيصل الفهد

Faisal.alfahd.ksa@gmail.com

الفصل الأول

Node Package Manager

(npm)

1.1.مقدمة:

يعتبر NPM بمثابة المدير لمخزن او مستودع يحوي اغلب المكتبات واطر العمل والأدوات التي يحتاجها أي مطور (على الأقل مطوري الويب)، بجميع توجهاتهم سواء كانوا يستخدمون Vanilla JavaScript او Angular او أي تقنية أخرى، وما يهمنا هنا بالتحديد هو الاستفادة من هذه NPM لتنزيل أطار عمل Angular.

وهذه NPM يتم تنزيله بشكل تلقائي عند تنزيل Node.js وهو عبارة عن بيئة عمل لي JavaScript تمكننا من كتابة أكواد JavaScript من ناحية Server والتعامل مع قواعد البيانات مثل القدرة على إضافة وحذف وتعديل وقراءة البيانات والتحكم بإدارة المستخدمين.

وبطبيعة الحال ليس هنا المقام لشرح Node.js وانما سوف نشرح كيف يتم تنزيل Node.js على جهاز الحاسب لكي نستطيع الاستفادة من NPM.

2.1. تنزيل وتثبيت Node.js:

لتنزيل وتثبيت Node.js في نظام تشغيل Windows، الرجاء منك عزيزي المتعلم تطبيق الخطوات التالية:

١) الذهاب إلى هذا الموقع:

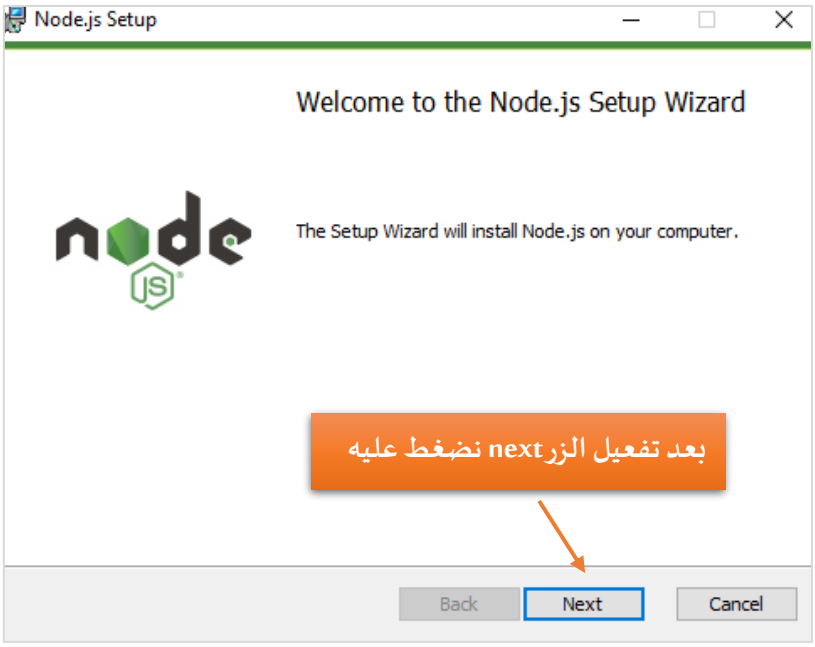
https://nodejs.org/en/

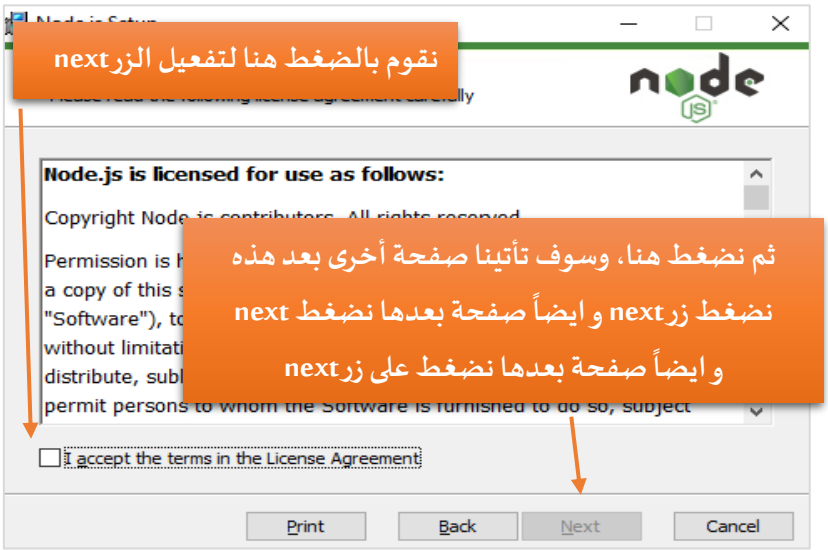
٢) وعندما تُفتح صفحة الموقع، سوف تكون بالشكل التالي:

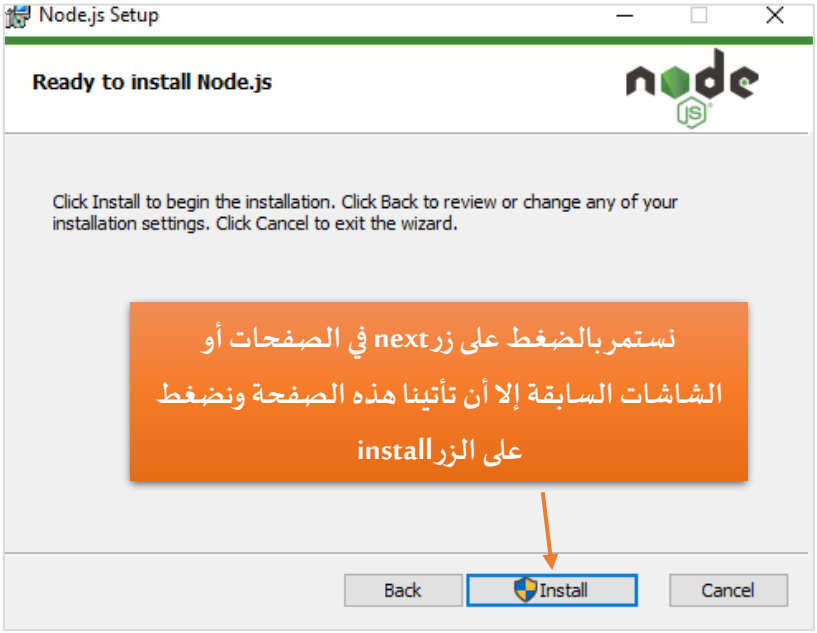
بطبيعة الحال عزيزي المتعلم لابد ان تنتبه انه في لحظة تأليف هذا الكتاب كانت الإصدارات كما هو موضح في الشكل السابق، ولكن قد تختلف الإصدارات في لحظة قراءتك لهذا الكتاب.

٣) عند الضغط على أحد الأزرار وتحميل أحد الإصدارين السابقين، قم بتثبيت node.js بالطريقة المعروفة التي يتم تثبيت بها أغلب البرامج وهي الضغط على زر التالي ثم التالي...الخ، إلا أن يتم تثبيت node.js في جهاز الحاسب لديك.

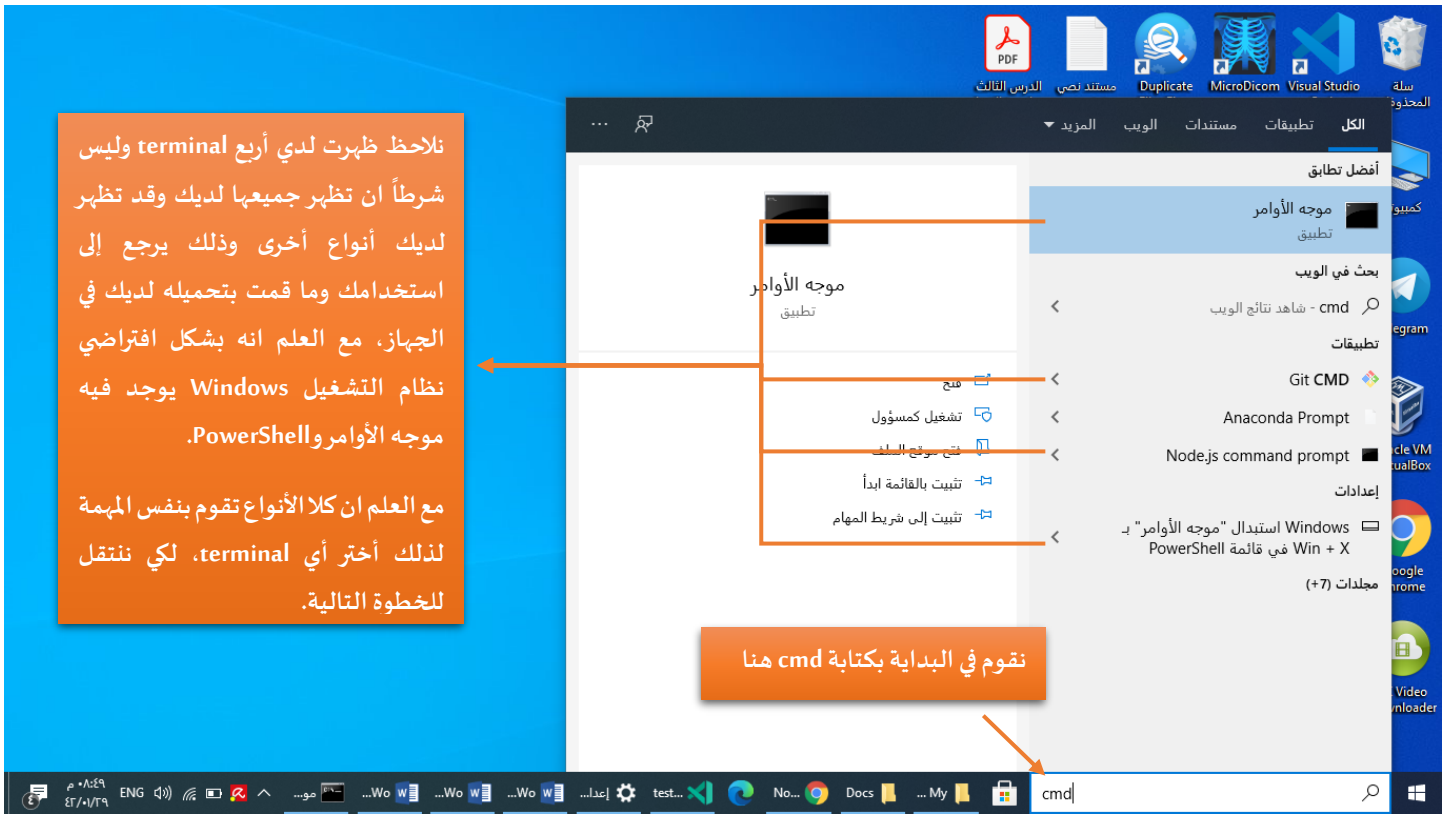

Node.js Setup
Welcome to the Node.js Setup Wizard
Please wait while the Setup Wizard prepares to guide you through the installation.
Computing space requirements
الانتظار قليلاً إلا أن يتم تفعيل الزر
Back
Next
Cancel


Node.js Setup
Welcome to the Node.js Setup Wizard
The Setup Wizard will install Node.js on your computer.
بعد تفعيل الزر next نضغط عليه
Back
Next
Cancel

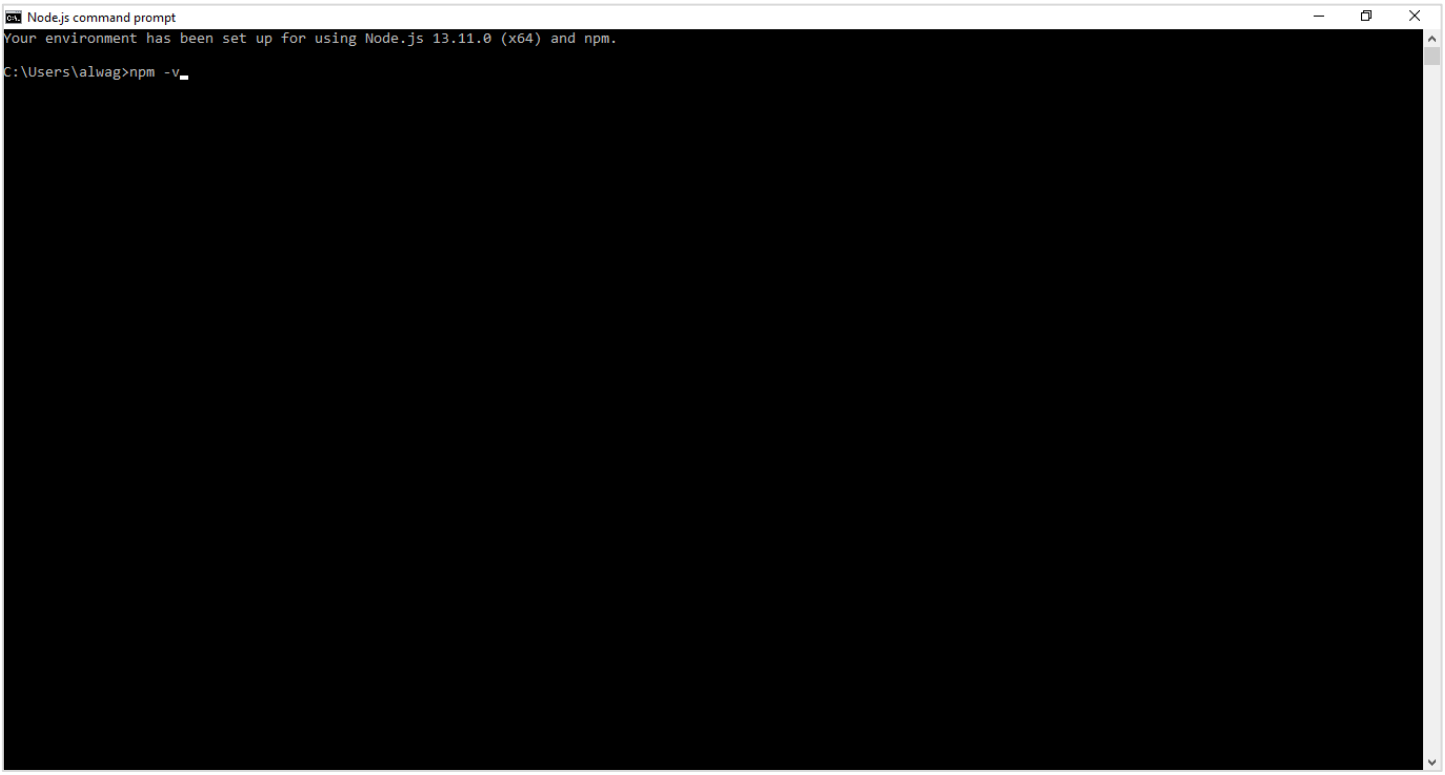

نقوم بالضغط هنا لتفعيل الزر next
Node.js is licensed for use as follows:
ثم نضغط هنا، وسوف تأتينا صفحة أخرى بعد هذه
نضغط زر next و ايضاً صفحة بعدها نضغط next
و ايضاً صفحة بعدها نضغط على زر next
I accept the terms in the License Agreement
Print
Back
Next
Cancel



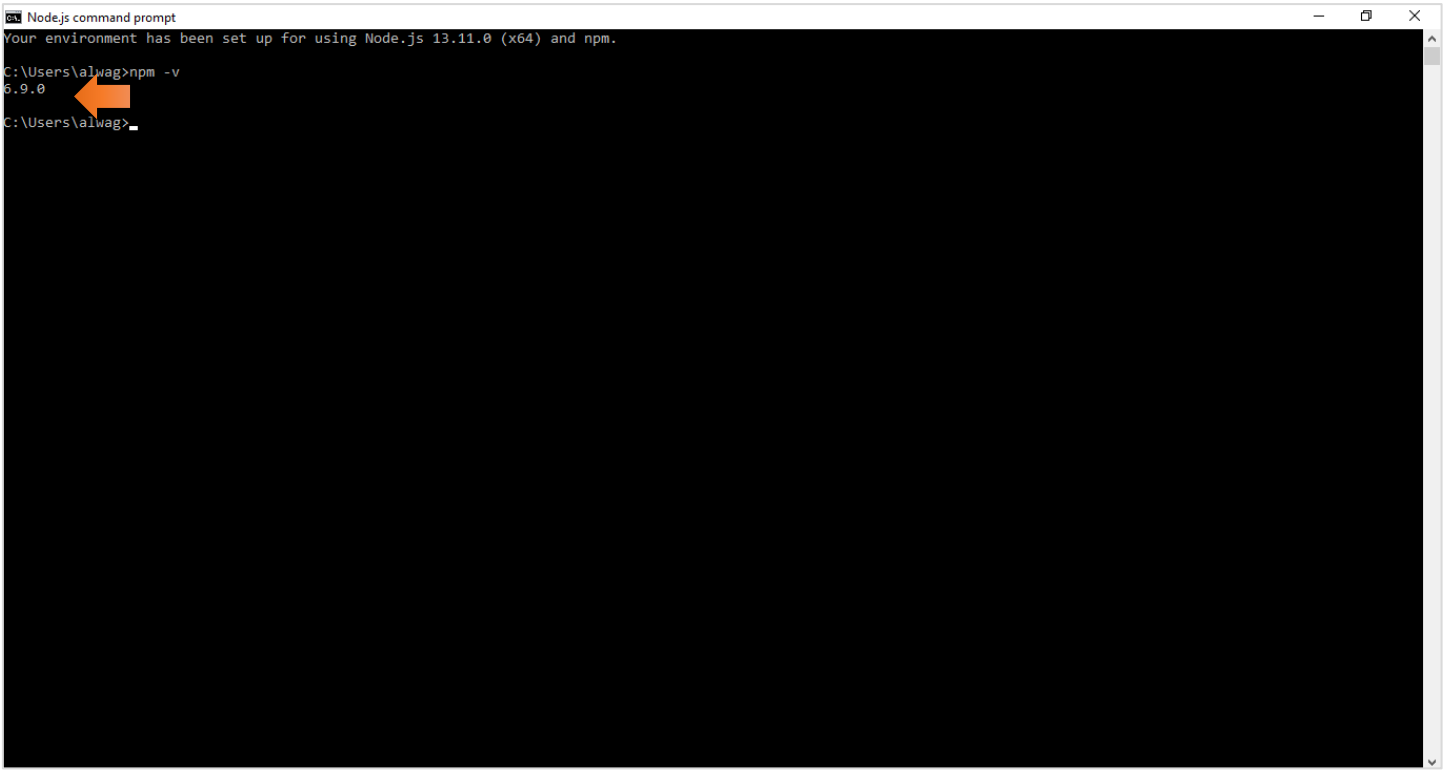
وبعدها سوف تأتي شاشة التثبيت وعند الانتهاء سوف يتم تثبيت Node.js بالإضافة إلى NPM، ولتأكد من أن NPM تم تثبيته بشكل صحيح في الجهاز لديك تستطيع الذهاب إلى terminal في نظام التشغيل لديك، وبما اننا هنا نشرح على نظام تشغيل windows فتستطيع الذهاب إلى شريط البحث ومن ثم كتابة cmd، كالتالي:



بعد اختيارنا لأحد الأنواع السابقة سوف تظهر لنا شاشة terminal، ونكتب فيها الأمر التالي npm -v حيث v اختصار لكلمة version، كالتالي:



ومن ثم نضغط زر Enter من لوحة المفاتيح وننتظر قليلاً، إلى ان تظهر لنا رقم الاصدارة الخاصة في NPM التي قمت بتحميلها على جهازك، كالتالي:



فإذا ظهرت لك رقم الاصدار، فنستطيع ان نقول مبروك عزيزي المتعلم فقد تم تحميل NPM في جهازك بدون أي مشاكل وتستطيع الانتقال إلى الخطوة التالية، وإن لم تظهر فعليك الرجوع ومحاولة معرفة السبب.

الفصل الثاني

Code Editor

(VS Code)

1.2.مقدمة:

بطبيعة الحال نحتاج إلى محرر أكواد Code Editor لكي يسهل علينا كتابة الكود البرمجي، وهنالك محررات أكواد كثيرة ومن أشهرها Visual Studio Code واختصاراً VS Code والسبب لاختيار هذا المحرر بالذات هو انه مجاني ومشهور جداً لدى المطورين وخصوصاً مطوري الويب ومفتوح المصدر وهذا يعطيه دعم هائل من مجتمع المطورين حول العالم واخيراً يكفي ان نقول انه من شركة مايكروسوفت الرائدة في عالم البرمجيات.

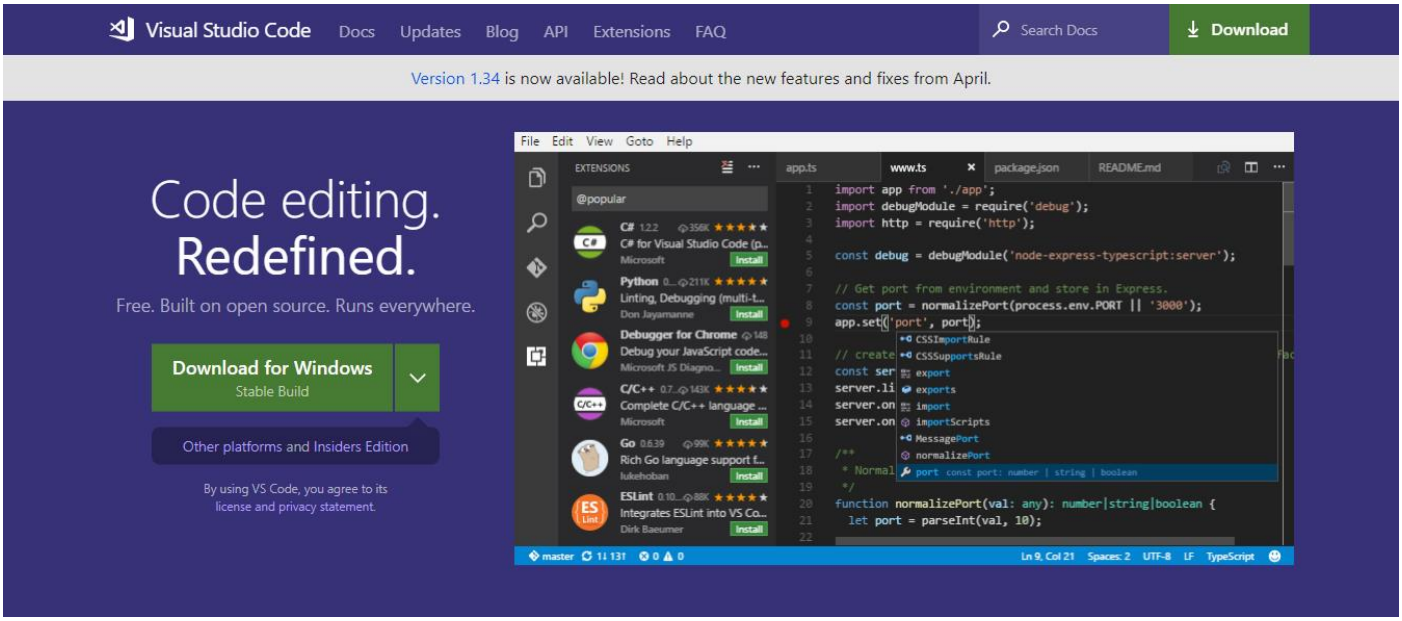
ويجب ان أُشير إلى اننا هنا لن ندخل بجميع تفاصيل هذا المحرر وانما سوف نُشير إلى بعض الأوامر والشاشات الإضافات التي نستخدمها بشكل متكرر وتسهل علينا كتابة الكود بشكل عام والتعامل مع إطار عمل Angular بشكل خاص.

وأخيراً وجب التنويه انه لا يلزمك عزيزي المتعلم ان تستخدم هذا المحرر لكي تستطيع تعلم هذه السلسلة وانما لك حرية استخدام أي محرر أكواد تجده مناسب لك.

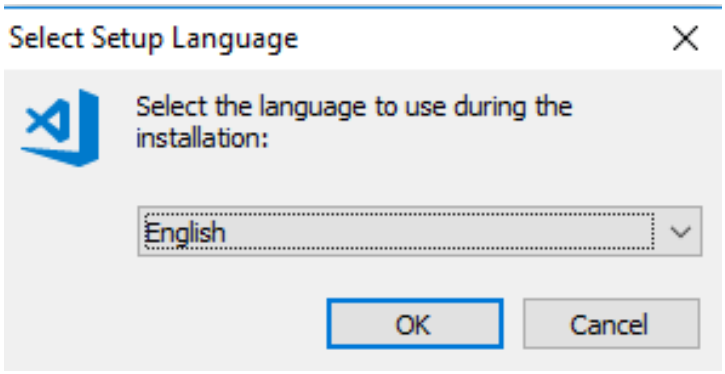
2.2.تحميل وتثبيت VS Code:

١) الانتقال إلى الموقع الموجود على هذا الرابط: https://code.visualstudio.com/

٢) ومن ثم اختيار النسخة التي تناسب نظام التشغيل لديك والضغط على الزر، مع العلم أن الموقع ذكي سوف يُظهر النسخة التي تناسب نظام التشغيل لديك بشكل تلقائي.



٣) بعد الضغط على الزر وتحميل البرنامج إلى جهاز الحاسب لديك، نبدأ الآن تثبيت البرنامج وعند الضغط على ملف التثبيت تظهر لنا هذه الشاشة التي نحدد منها اللغة، نختار اللغة الإنجليزية ثم Ok.

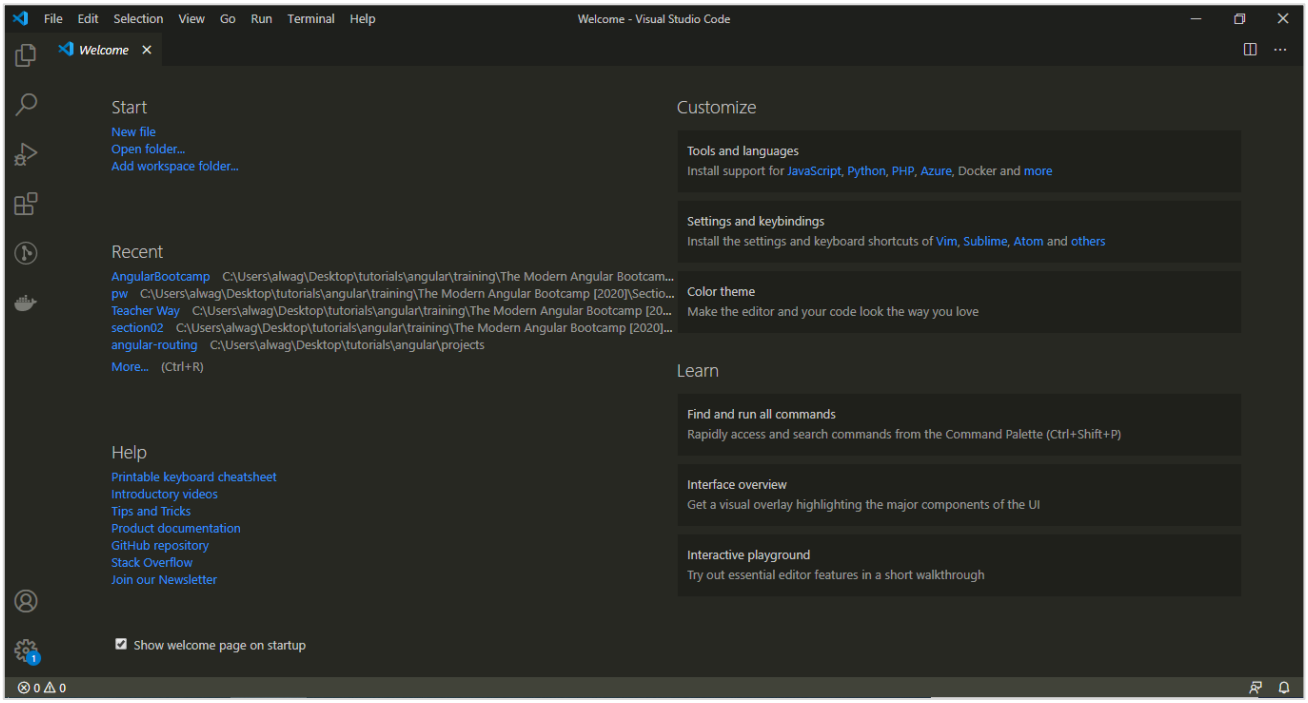


٤) ولكيلا نُعيد تكرار الخطوات فهو مشابهه لما قمنا به عندما ثبتنا node.js نستمر في ضغط التالي next إلى أن تأتينا شاشة التثبيت ومن ثم نقوم بالضغط على زر تثبيت أو install.

3.2. فتح VS Code وشرح أهم أجزائه:

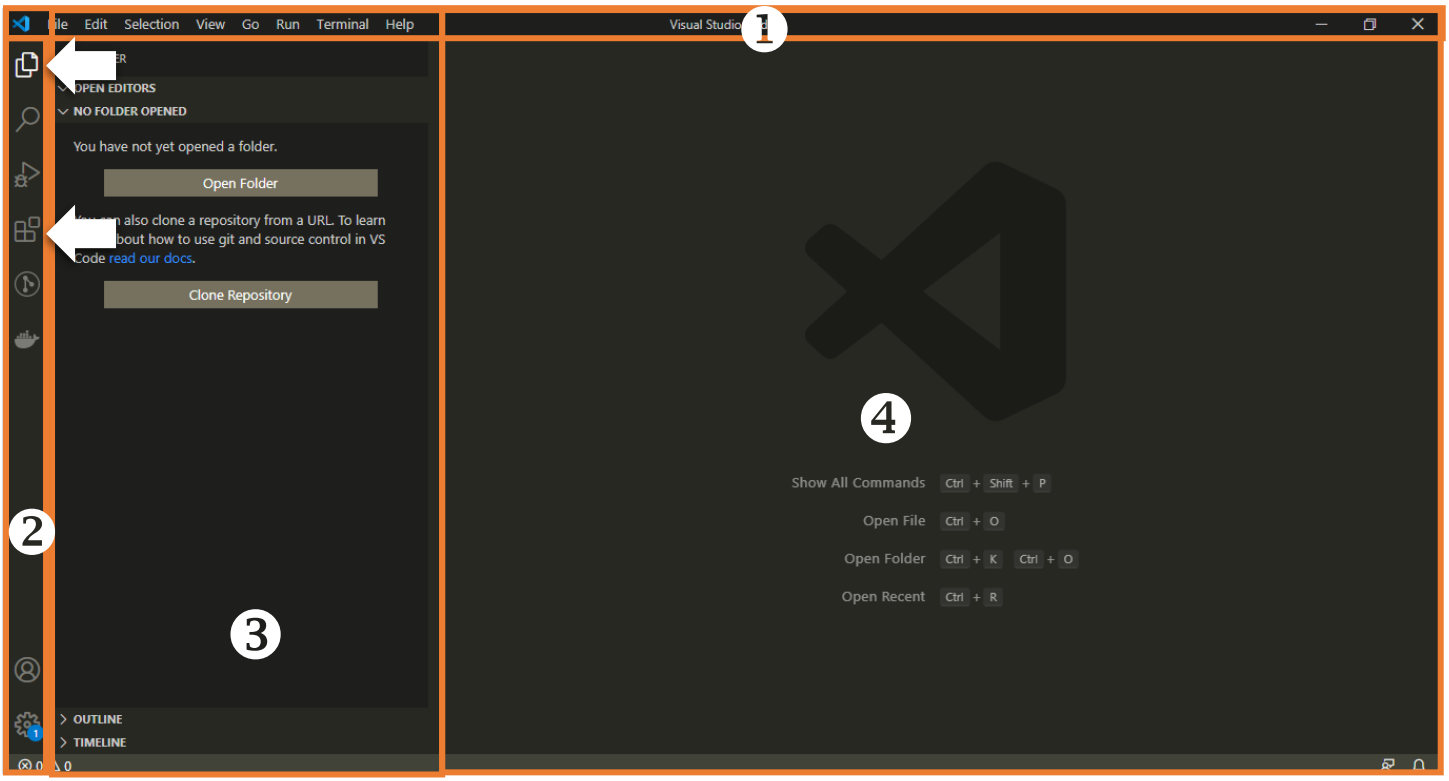
بعدما يتم تثبيت التطبيق في الجهاز لديك سوف تجد في الغالب اختصار له على سطح المكتب في نظام تشغيل Windows بهذا الشكل:

قم بالنقر عليه لفتحه وننتظر قليلاً إلى ان يتم فتحه بشكل كامل بحيث تظهر لنا الشاشة التالية:



عند بداية التشغيل سوف تظهر لنا شاشة الترحيب Welcome نستطيع اغلاقها.

ونلاحظ أن هذه الشاشة مقسمة إلى مجموعة أقسام، كالتالي:



١) شريط القوائم، ويحتوي على جميع الأوامر الخاصة بهذا التطبيق مقسمة على شكل قوائم منسدلة.

٢) شريط الأنشطة او الفعاليات، ويحتوي على مجموعة من القوائم وعند الضغط على كل قائم تظهر لنا محتوياتها في الشاشة رقم (3)، وحقيقة على الرغم من أهميتها لكن سوف أقوم بالتركيز هنا على قائمتين وهما اللذان تم التأشير عليهما بالسهمين، الأول هو Explorer والثاني Extensions، حيث الأول نستطيع عن طريقه استعراض ملفات ومجلدات المشروع وإجراء جميع التعديلات عليها من إضافة او حذف او تعديل اسم المجلد او الملف، اما الثاني فنستطيع عن طريقه إضافة كم هائل من الإضافات التي تُساعدنا في التعامل مع الكود وهذه الإضافات في الغالب تم بناءها من قبل مجتمع المطورين، فمثلاً عند الرغبة في التعامل مع إطار عمل Angular فنستطيع مثلاً تحميل إضافة تقوم بإكمال الأمر بمجرد كتابة بعض الأحرف الأولى منه، او نستطيع تحميل إضافة أخرى تستطيع التعرف على المتغيرات والخصائص التي تم تعريفها في ملف class عند استدعائها في ملف template بمجرد كتابة بعض الأحرف الأولى، وايضاً في حال اردنا التعامل مع C++ او C# او PHP، وغيرها من اللغات والمكتبات واطر العمل الأخرى، فنستطيع تحميل جميع الإضافات الخاصة بها.

وهنالك ايضاً الإضافات العامة والتي لا تختص بلغة محددة فمثلاً ترتيب وتنسيق أكواد HTML او تنسيق أكواد CSS بطريقة معينة او تلوين الأقواس المتداخلة، وهكذا من هذا النوع من الإضافات، وايضاً هنالك إضافات خاصة بتغيير Theme الخاص بتطبيق VS Code نفسه، بل هنالك بعض الإضافات التي تقوم بالتدقيق النحوي والإملائي في حال اردت إضافة بعض الملاحظات على بعض الاسطر البرمجية، وفي الحقيقة ليس هنا المقام لشرحها جميعا، ولكن سوف نتعلم كيف إضافة بعض هذه الإضافات وخصوصاً التي تساعدنا في تسهيل اعمالنا في مشاريع Angular.

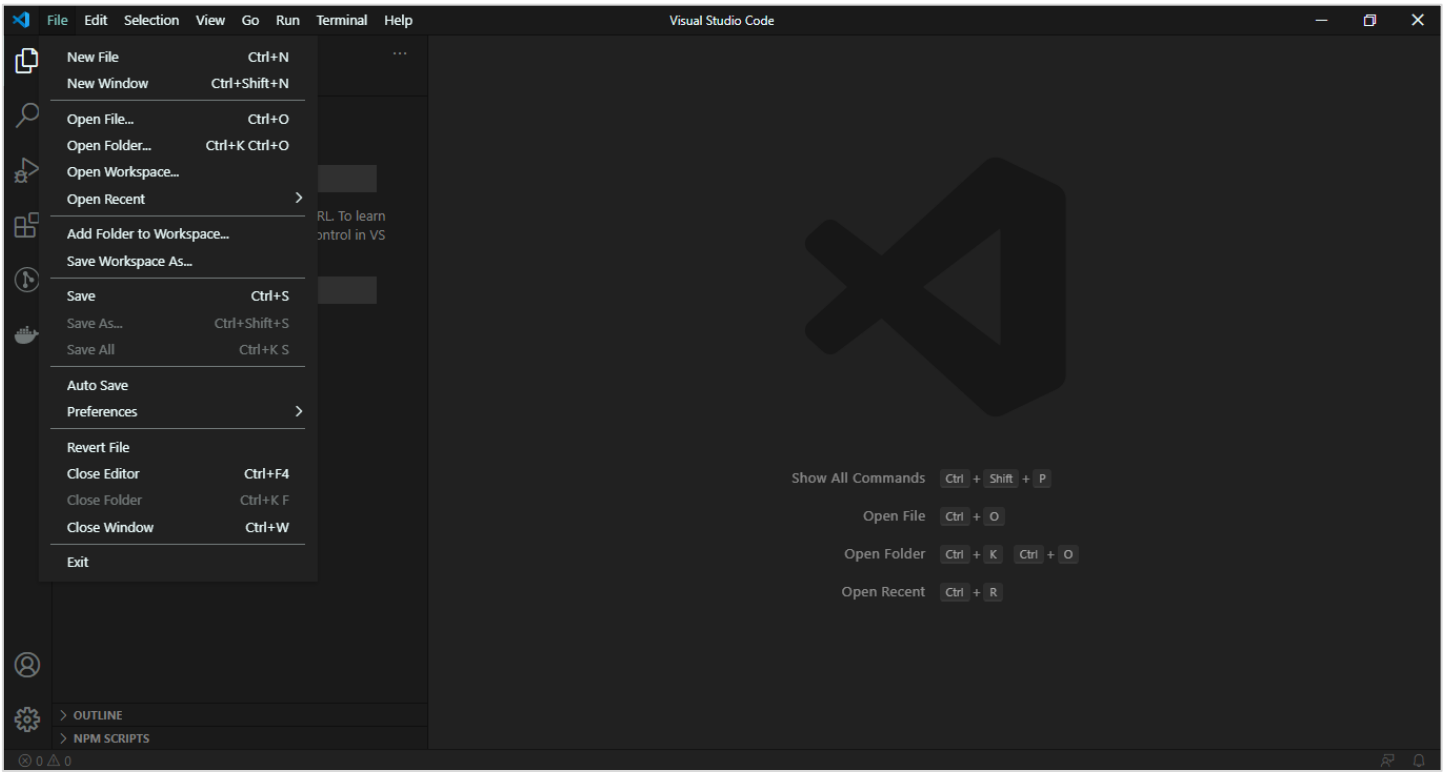
٣) هذه الشاشة أشرنا لها سابقاً وهي التي يتم فيها عرض محتويات القوائم التي يتم اختياراها في النقطة (2).

٤) في هذه الشاشة نستعرض محتويات أي ملف يتم اختياره بحيث نقوم بكتابة الأكواد وتحريرها وإجراء جميع الإجراءات اللازمة.

1.3.2.شريط القوائم:

كما أشرنا سابقاً شريط القوائم يحتوي على جميع الأوامر الخاصة بالبرنامج مجمعة على شكل قوائم، وايضاً كما أشرنا سابقاً انه ليس المقام هنا لشرح جميع هذه الأوامر ولكن سوف نتطرق إلى أشهرها وما يهمنا في تطوير الويب بشكل عام وAngular بشكل خاص.

1.1.3.2. قائمة File:



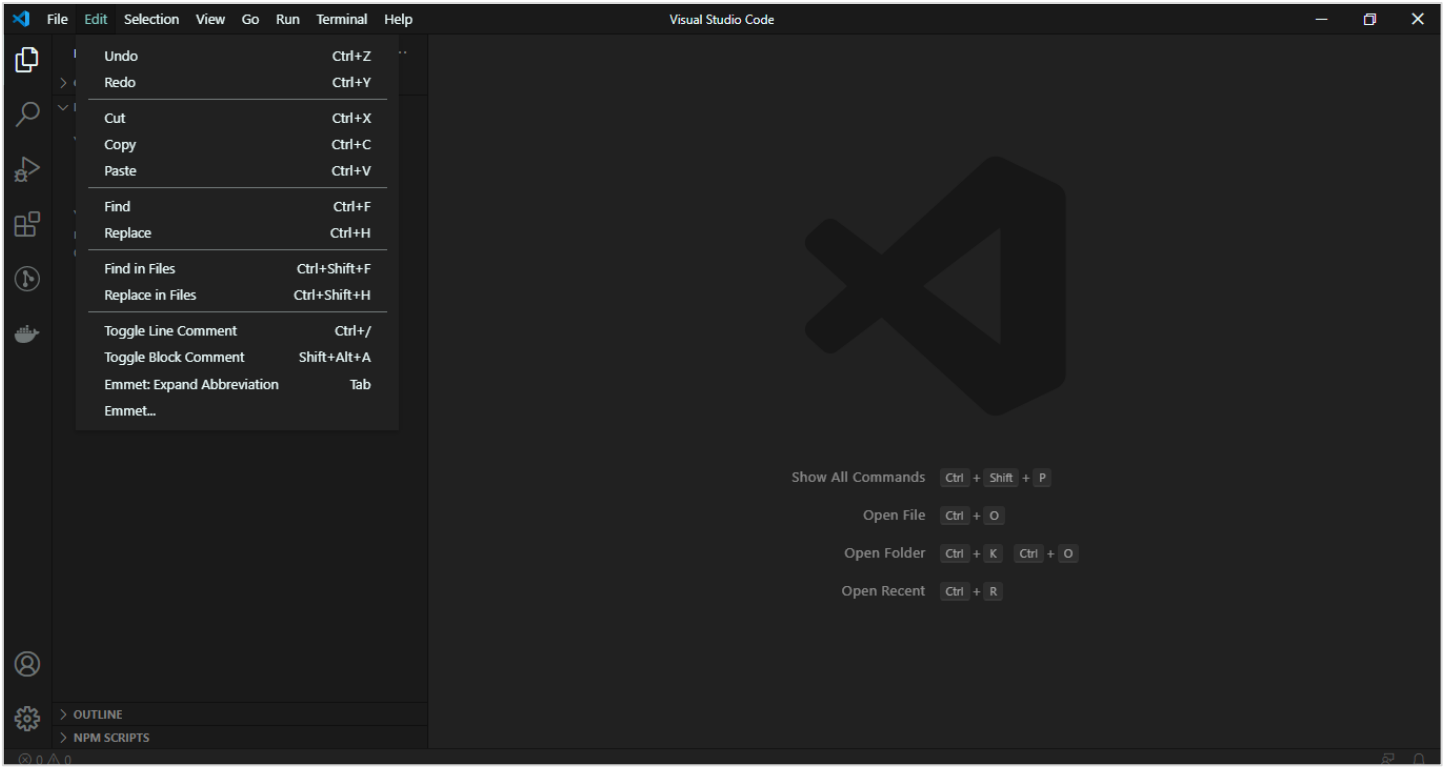
وهذه القائمة تحتوي على أغلب الأوامر الخاصة بالتعامل مع الملفات من ناحية إضافة ملف جديد، وعند الضغط على هذه الأمر سوف يتم إضافة ملف وبشكل افتراضي سيتم تسميته Untitled.txt، لذلك نحتاج إلى ان نعمل له حفظ عند طريق نفس هذه القائمة File ونختار Save ومن ثم نحدد نوع هذا الملف سواء html او js او ts او ...الخ.

وايضاً تحتوي هذه القائمة على الأمر New Window ونستفيد منه في حال كان لدينا أكثر من مشروع بحيث كل مشروع يكون في نافذة خاصة به.

ومن الأوامر المهمة وكثيرة الاستخدام هي فتح مجلد Open Folder ونستفيد من هذا الأمر لفتح مجلد مشروع سابق، فكثير من مشاريع web يتم انشاءها عند طريق terminal وينتج عنها مجلد يحتوي على جميع ملفات المشروع لذلك عن طريق هذا الامر نقوم بفتح وإدراج هذا المجلد داخل التطبيق.

وبالطبع هنالك الأوامر Save وSave As وSave All، واتوقع عزيز المتعلم بما انه وصلت إلى مرحلة تطوير تطبيقات الويب فإن هذه الأوامر تكون معتاد عليها وتعرف الفروق بينها لذلك لن نقوم بشرحها.

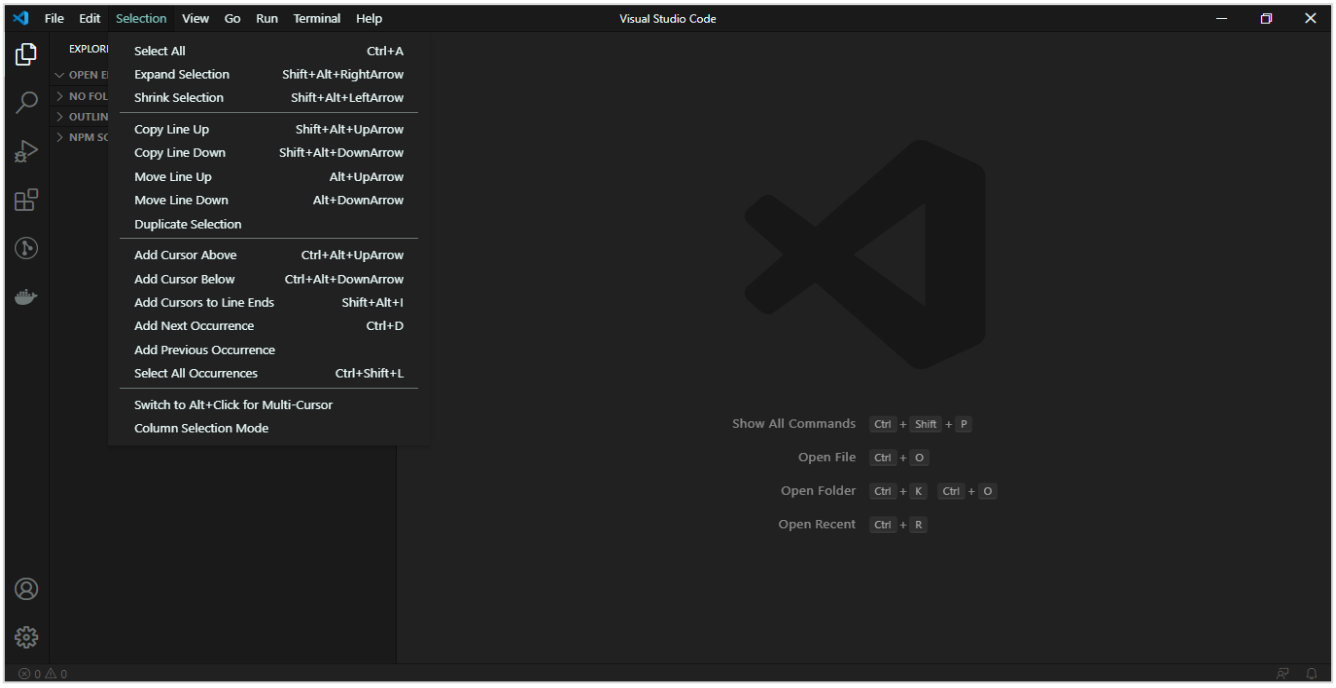
2.1.3.2.قائمة Edit:



ولعل من هذه القائمة ايضاً وما تحتويه من أوامر هي معروفة لدي أي مبتدأ في تعلم الحاسب الآلي فما بالك بشخص وصل إلى مرحلة تطوير وبرمجة Web، ولعل الجديد هنا هو الامر Toggle Line Comment ونستفيد منها لعمل comment لسطر برمجي معين او الغاء هذا comment، ونستطيع كتابة الاختصار / + Ctrl في Windows لعمل comment او الغائه.

3.1.3.2.قائمة Selection:

ولعل هذه من القوائم الخاصة بتطبيق VS Code، وتحتوي على مجموعة من الأوامر الخاصة بالتحديد وتحريك ونسخ الاسطر البرمجية، وسوف نتطرق إلى بعض هذه الأوامر كالتالي:



الأمر الأول **Select All** وخاص بتحديد كافة الأسطر البرمجية.

الأوامر الأربعة والمهمة جداً هي:

Copy Line Up: واختصارها Shift + Alt + UpArrow والمقصود UpArrow زر السهم للأعلى في لوحة المفاتيح، ويتم تنفيذ هذا الأمر عن طريق وضع مؤشر الكتابة على السطر البرمجي المستهدف ومن ثم نضغط على ازرار الاختصار معاً او اختيار الأمر **Copy Line Up:** من القائمة Selection وسوف يتم أخذ نسخة من هذا السطر في أعلى السطر البرمجي.

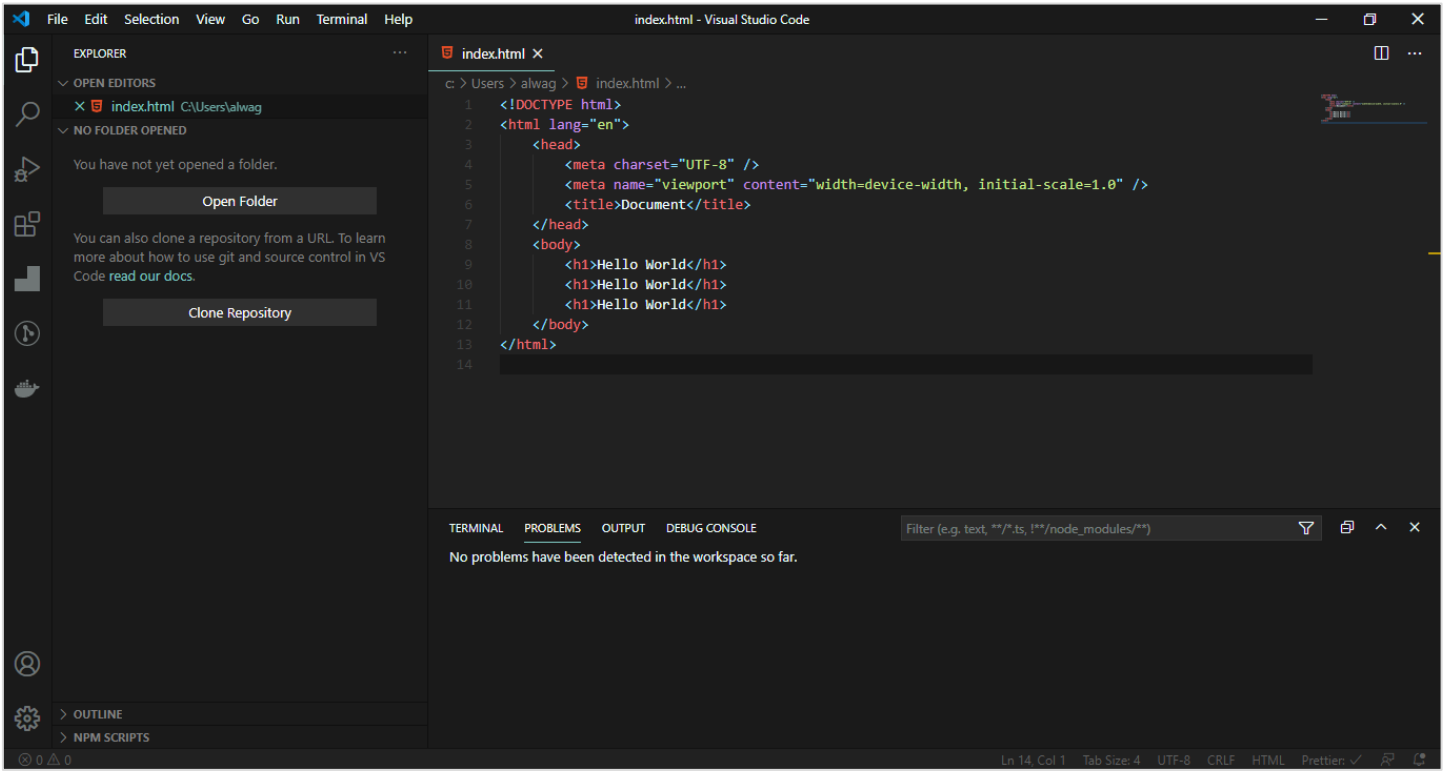
Copy Line Down: وهو مشابهه للأمر السابق والفرق هو في الاختصار حيث نضغط في هذه المرة الز UpArrow او زر السهم للأسفل او اختيار الامر Copy Line Down وسوف يتم عمل نسخة من هذا السطر في أسفل السطر المحدد.

Move Line Up: ويقوم بنقل السطر البرمجي (الأوامر او الامر البرمجي) بكامله إلى سطر أعلى، واختصاره Alt + UpArrow او اختيار الامر Move Line Up من القائمة.

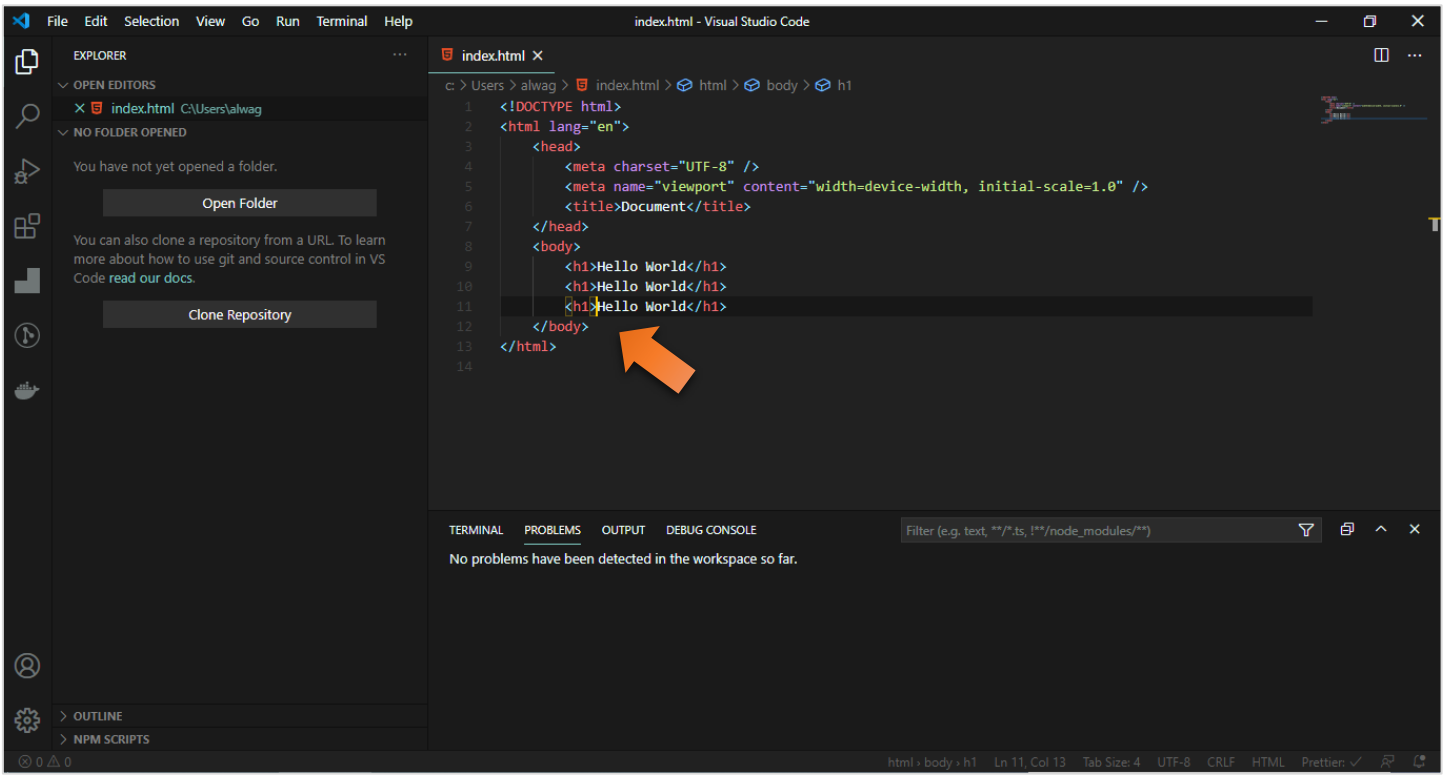
Move Line Down: وهو مشابهه لما سبقه ولكن هنا ينقله لسطر في الأسفل، واختصاره Alt + UpArrow او اختيار الامر Move Line Down من القائمة.

كما الأوامر الأربعة الأخرى ايضاً لا تقل أهمية وتساعد المطور في تسريع كتابة الكود وتعديل الأخطاء وهي خاصة بعمل بالتحديد، ويُفضل استخدام الاختصارات بدلاً من اختيار الأوامر من القائمة للحصول على النتيجة المُرادة، وهذه الأوامر هي:

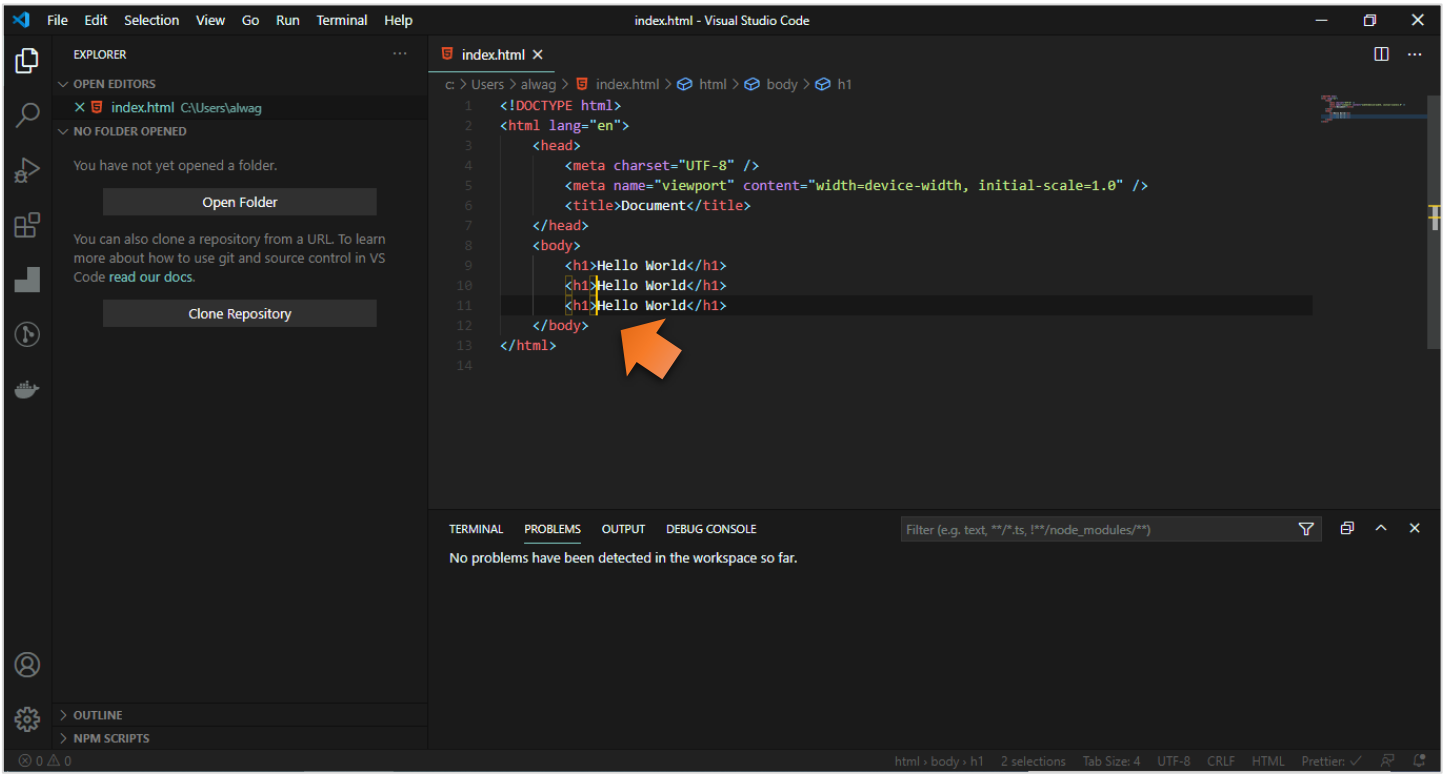
Add Cursor Above ويقوم هذا الامر بتمديد مؤشر الكتابة للأعلى مع كل ضغطة على الاختصار ومن ثم على سبيل المثال نستطيع تحديد مجموعة من الخانات بالضغط على زر shift + RightArrow او الزر Shift + LeftArrow وعند الكتابة فإن الكتابة سوف يتم تطبيقها على جميع الأسطر التي تم تحديدها، ولتوضيح لنفرض انه لدينا ملف HTML وفيه Markup التالي:



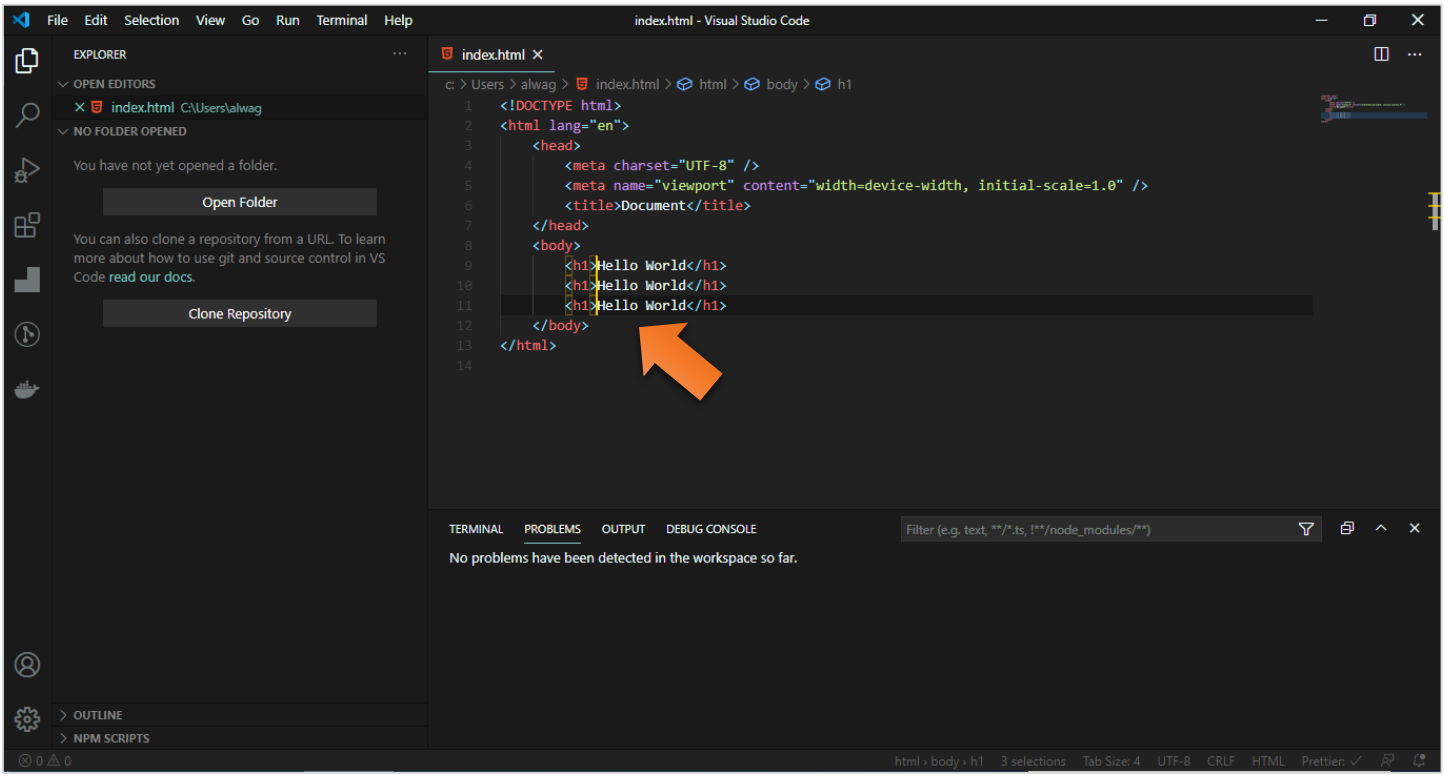
ولنضع مؤشر الفأرة في أحد الأسطر البرمجية، كالتالي:



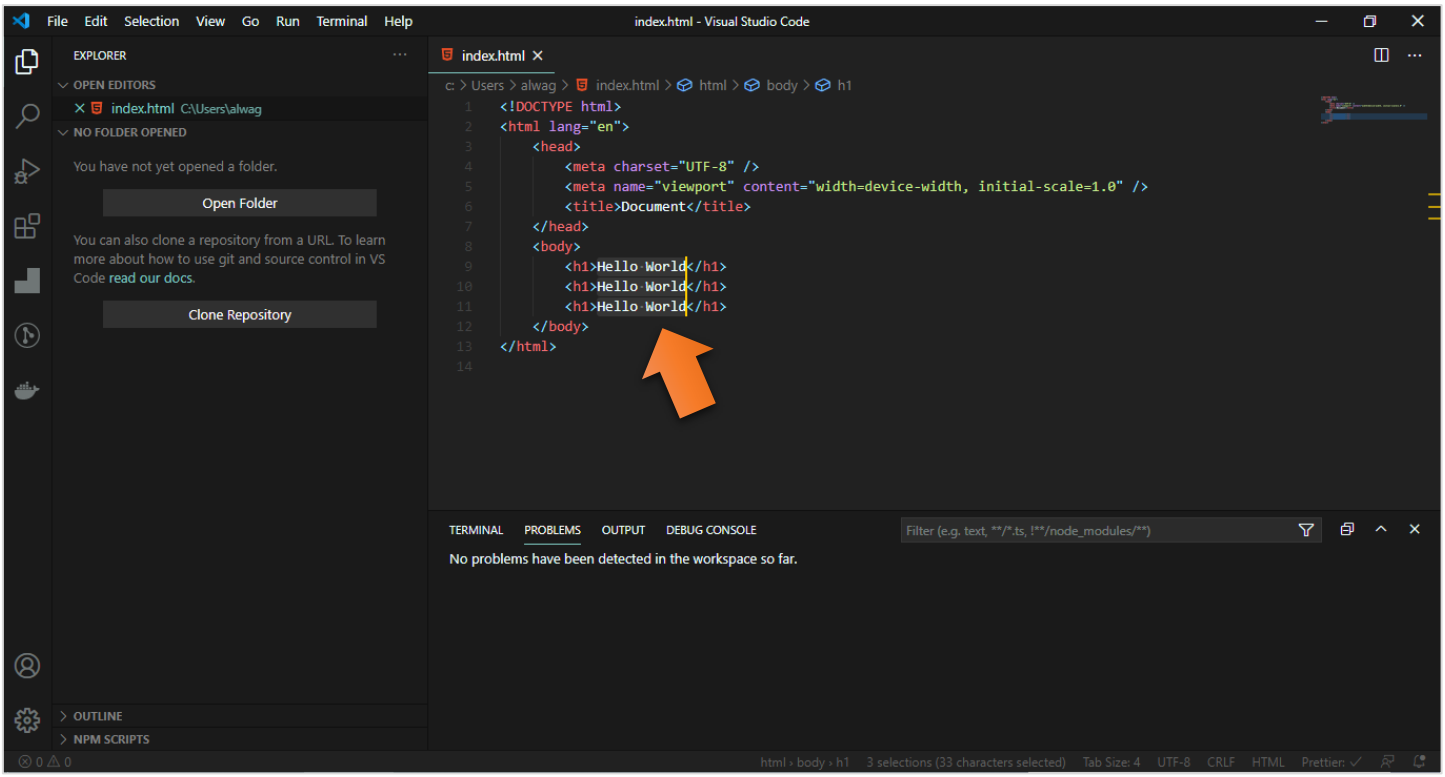
الآن نستطيع الاستفادة من الأمر Add Cursor Above ولنستخدم الاختصار Ctrl + Alt + UpArrow مرة واحدة، كالتالي:



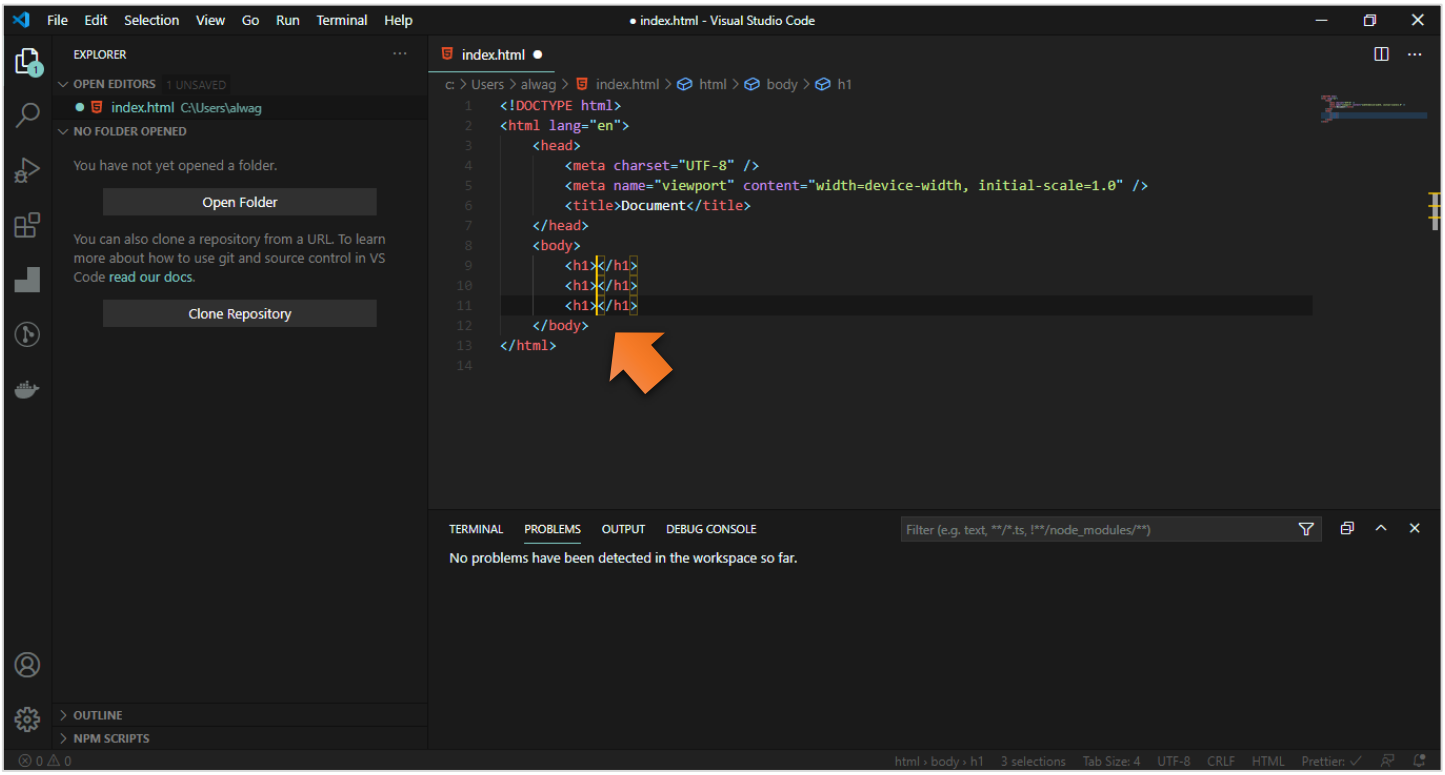
ولنقوم بالضغط عليه مرة أخرى، ولنرى النتيجة:



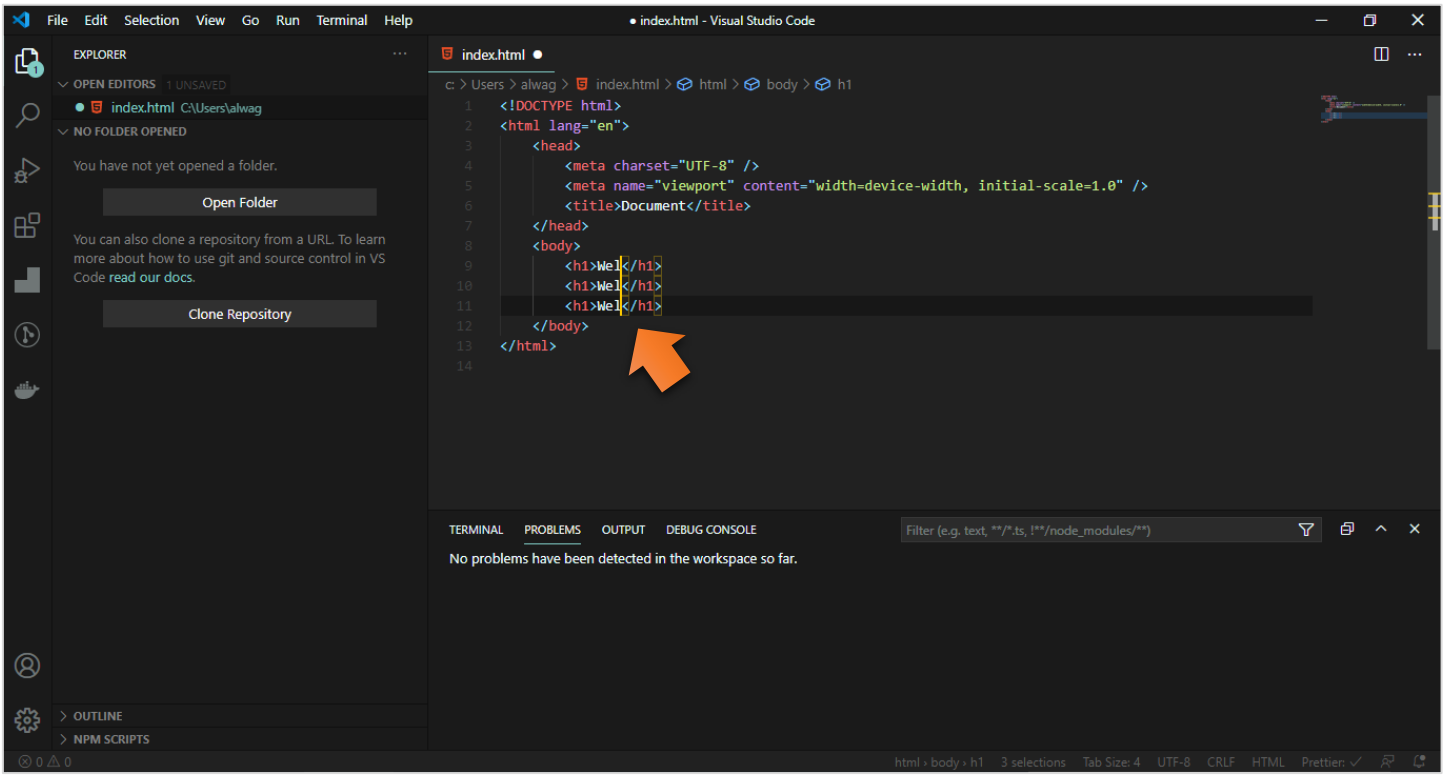
نلاحظ انه مع كل ضغطة على الاختصار فسوف يتمدد مؤشر الكتابة إلى الأعلى، والآن لنقوم بالضغط على الاختصار Shift + RightArrow لنحدد جميع الاسطر الثالثة، كالتالي:

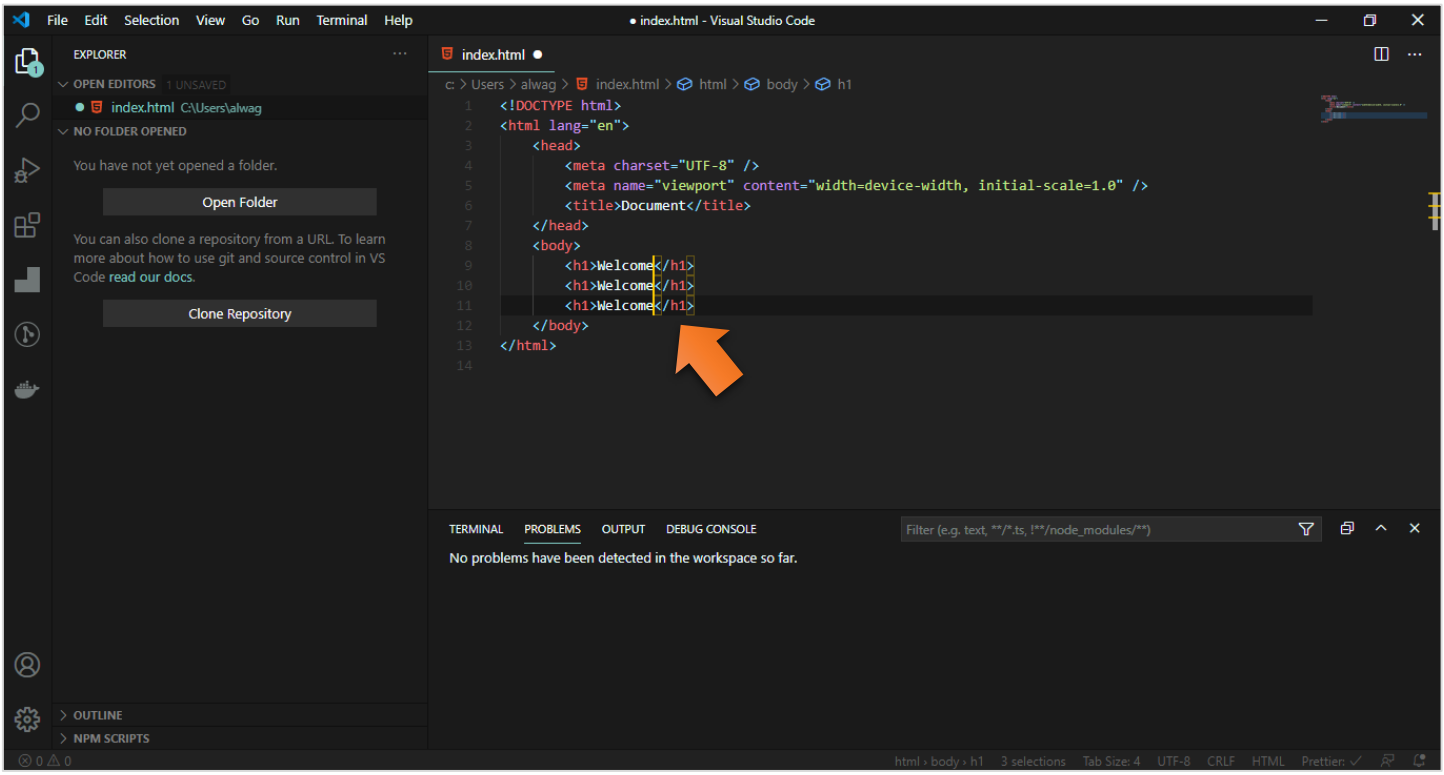


نلاحظ تم تحديد كلا الاسطر معاً وهذه من فوائد الامر السابق، وليس هذا فقط وايضاً لو اردنا الحذف فسوف يتم حذف كلا الاسطر، كالتالي:



وايضاً عند الكتابة فسوف يتم الكتابة في جميع الاسطر، كالتالي:

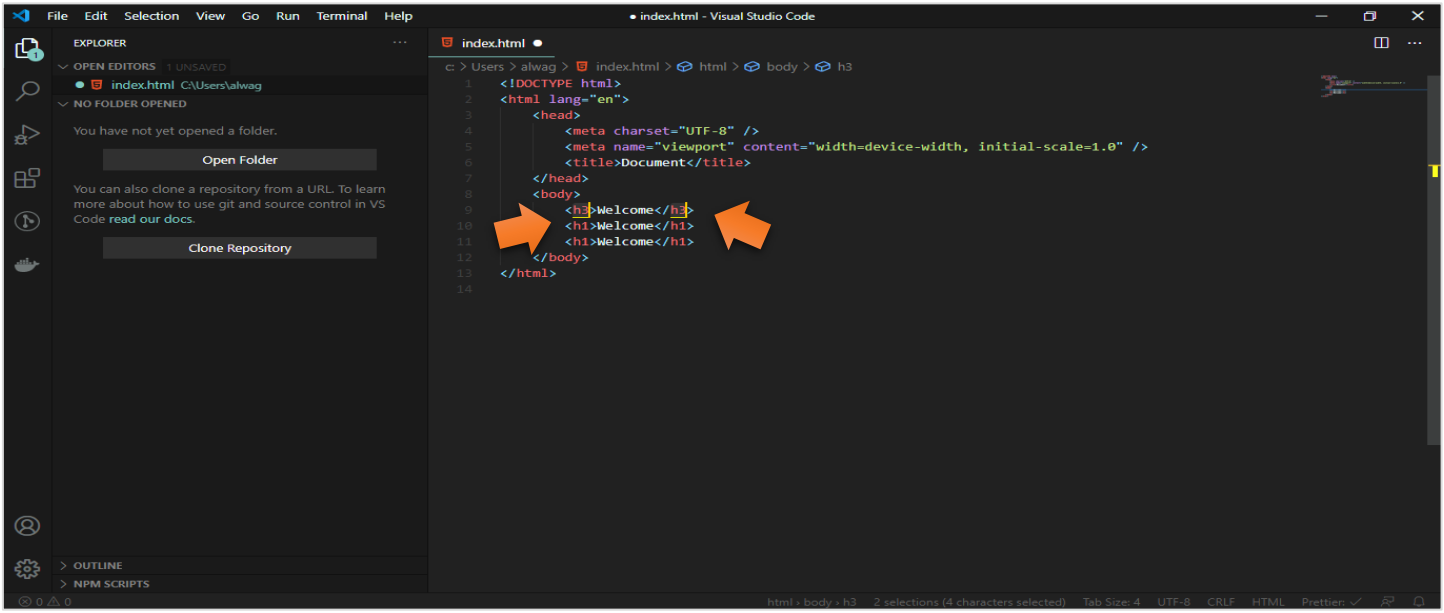




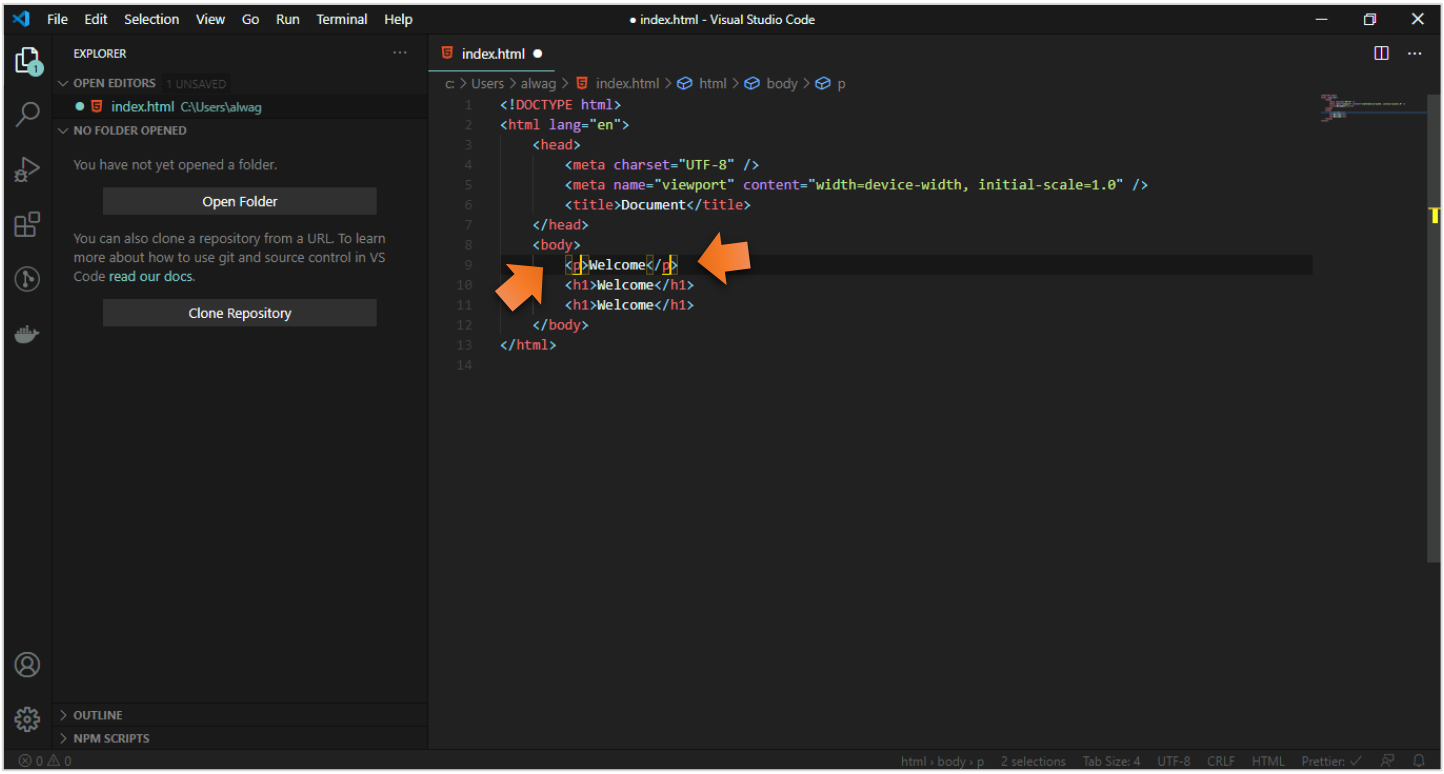
وفي حالة أردنا الخروج من هذا الوضع فقط نقوم بالضغط في أي مكان في الملف.

Add Cursor Below: وهو مشابهه للأمر السابق ولكن بدلاً من ان يقوم بالتمدد للأعلى هنا سوف يتمدد للأسفل.

Add Next Occurrence: وهذا الأمر مهم جداً ويُستخدم لتحديد أكثر من تاغ (تاغات HTML مثل او <h1></h1> وهكذا بقية التاغات الأخرى) في حال اردنا ان نقوم بتغيير التاغ، ففي حال كان لدينا Markup معقد ومتداخل واردنا تغيير احد التاغات فلابد ان نقوم بتغيير فتحة التاغ واغلاق التاغ، لذلك قد نجد صعوبة في تغيير فتحة التاغ ومن ثم البحث عن غلق التاغ لكي نقوم بتغييره هو الآخر، لذلك اتى هذا الأمر حيث نقوم بوضع مؤشر الفأرة عند بداية التاغ المُراد تغييره ومن ثم نضغط على الاختصار Ctrl + D وسوف يقوم بتحديد هذا التاغ ومن ثم نقوم بالضغط عليه مرة أخرى وسوف يقوم بتحديد قفلة هذا التاغ، كالتالي:



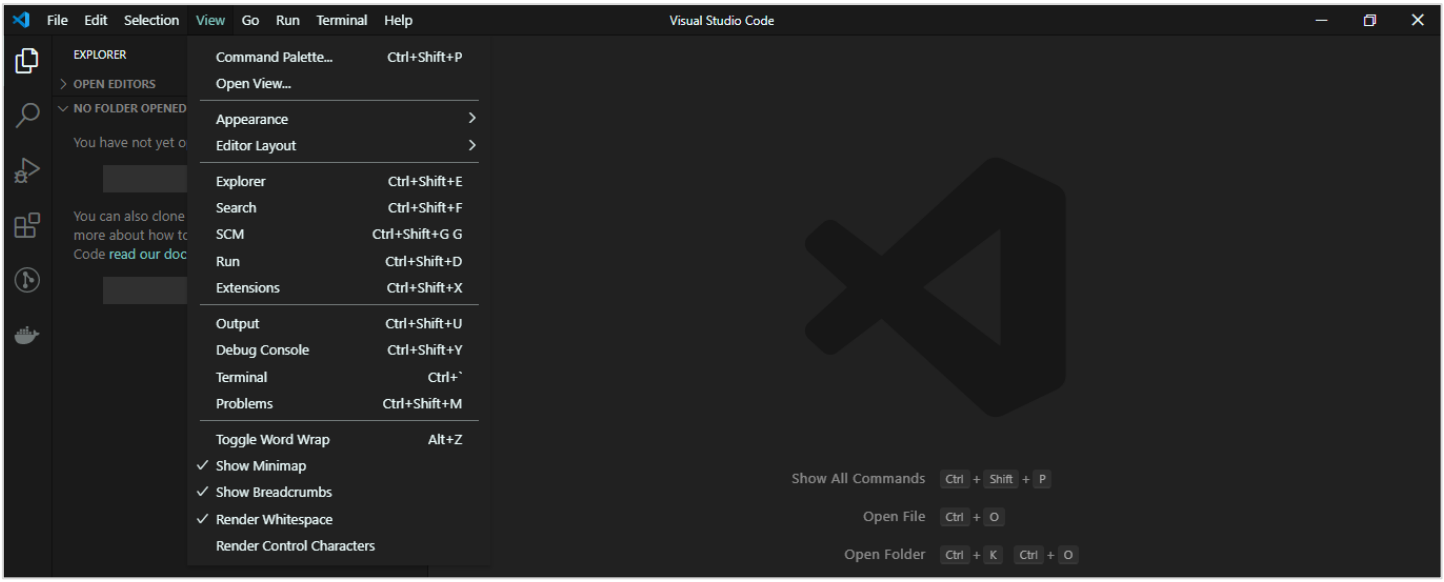
نلاحظ قمنا في البداية بوضع مؤشر الكتابة في بداية التاغ <h1> ومن ثم ضغطنا على الاختصار CTRL + D فقام بتحديد هذا التاغ ومن ثم قمنا بالضغط عليه مرة أخرى على الاختصار وقام بتحديد نهاية (قفلة) التاغ </h1>، الآن نستطيع حذف هذا التاغ واستبداله مثلاً بالتاغ <p>، كالتالي:



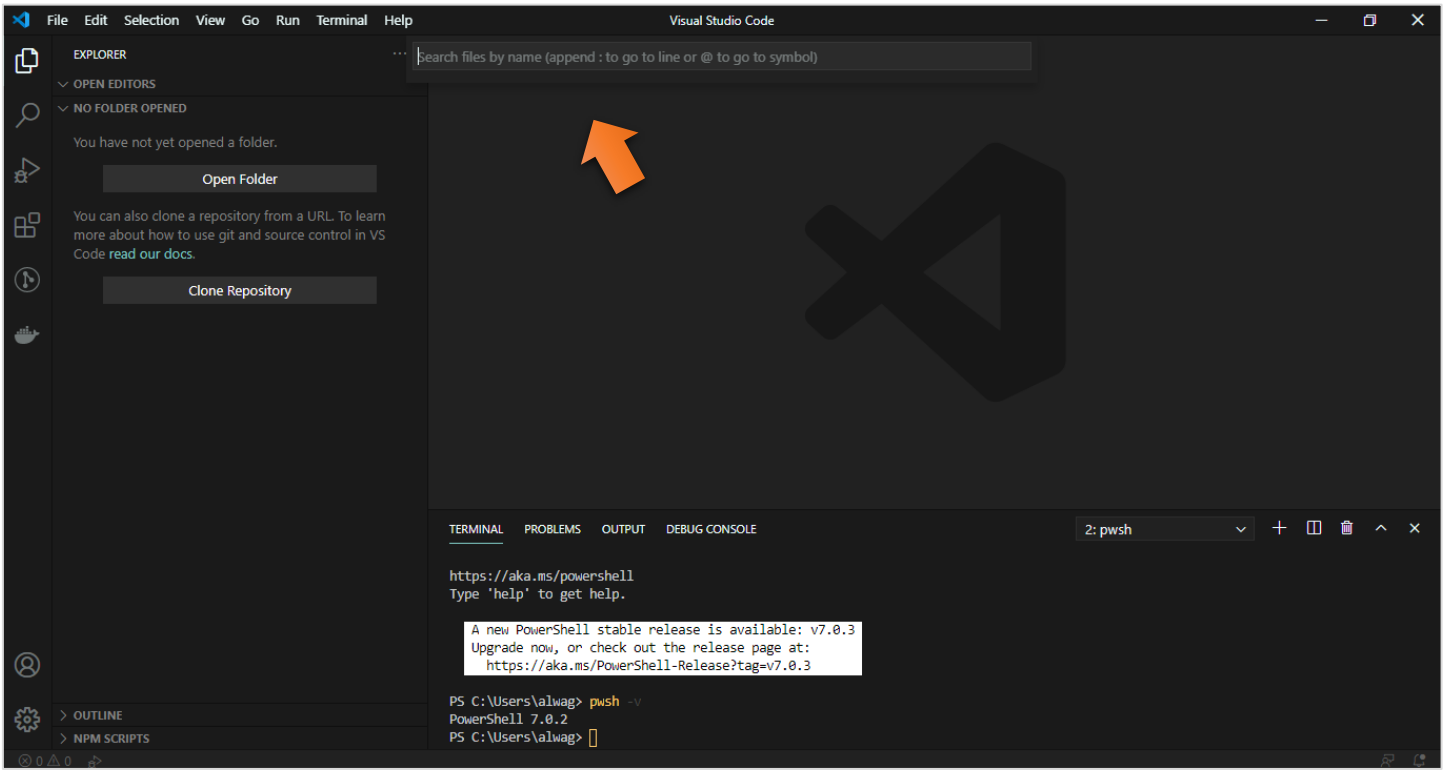
Select All Occurrences: وهذا الامر يقوم بتحديد كافة التاغات المحددة، فمثلاً في المثال السابق لدينا اكثر من <h1> تاغ فلذلك نستطيع وضع المؤشر على أحدى هذه التاغات ومن ثم اختيار الأمر Select All Occurrences او الاختصار Ctrl + Shift + L.

4.1.3.2.قائمة View:

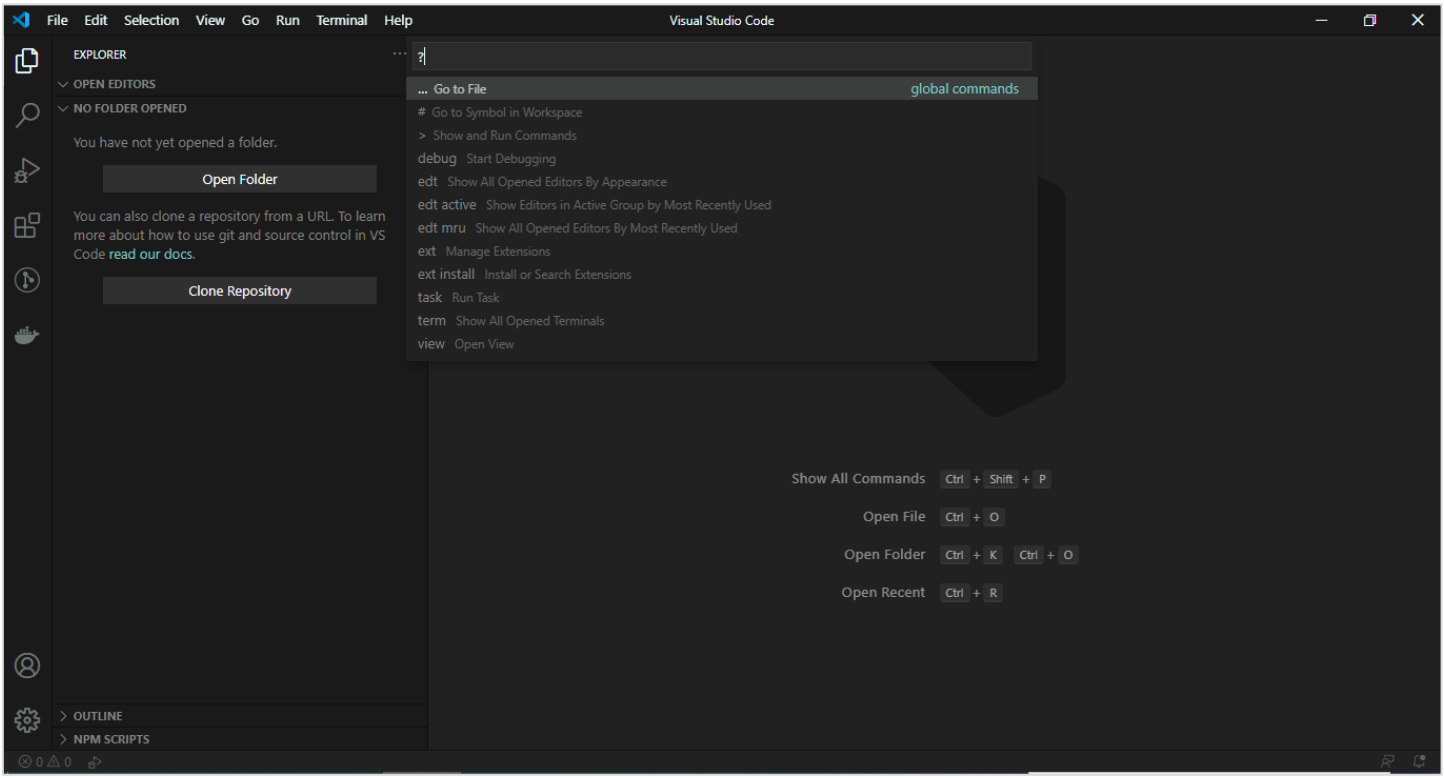
وهذه القائمة تحتوي على الأوامر الخاصة بالتحكم بإظهار واخفاء بعض أجزاء التطبيق بالإضافة إلى الأوامر الخاصة بطريقة عرض أجزاء التطبيق وتقسيمه إلى أجزاء، كالتالي:



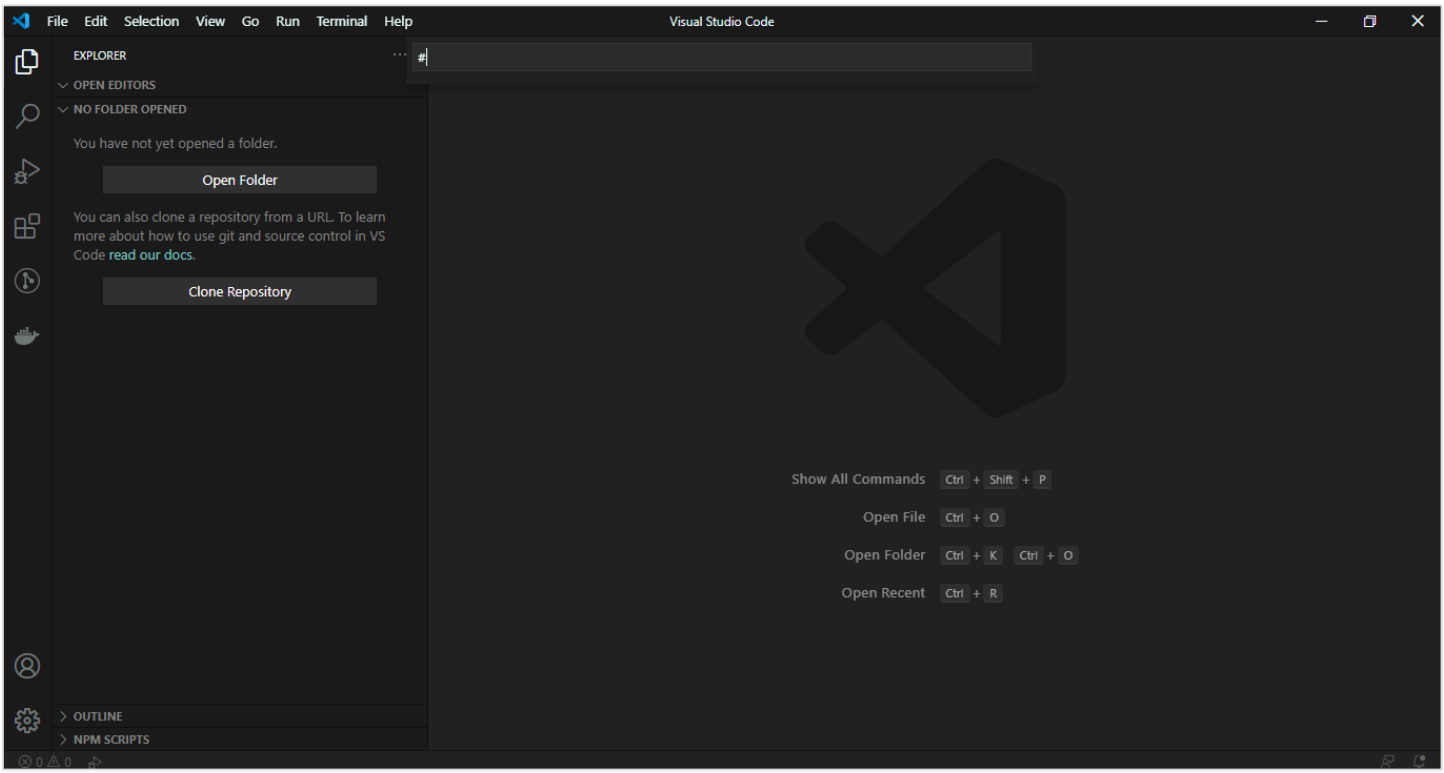
اول أمر سوف يمر علينا وهو أهمها Command Palette: وعن طريق هذا الأمر نستطيع القيام بكثير من الأمور ولمعرفة هذه الأوامر، التي نستطيع كتابتها في Command Palette نقوم بكتابة الاختصار Ctrl + P، كالتالي:



ونستفيد من هذا المربع في التنقل بين ملفات المشروع، حيث نقوم بكتابة اسم الملف او جزء منه وسوف يظهر لنا قائمة بكافة الملفات المشابهة ومن ثم عن طريق الأسهم في لوحة المفاتيح نختار الملف المستهدف، وايضاً لتبديل بين آخر ملفين قمنا بفتحهما نقوم بالضغط على الاختصار Ctrl + P + P، ولمعرفة الأوامر الأخرى نكتب علامة الاستفهام، كالتالي:

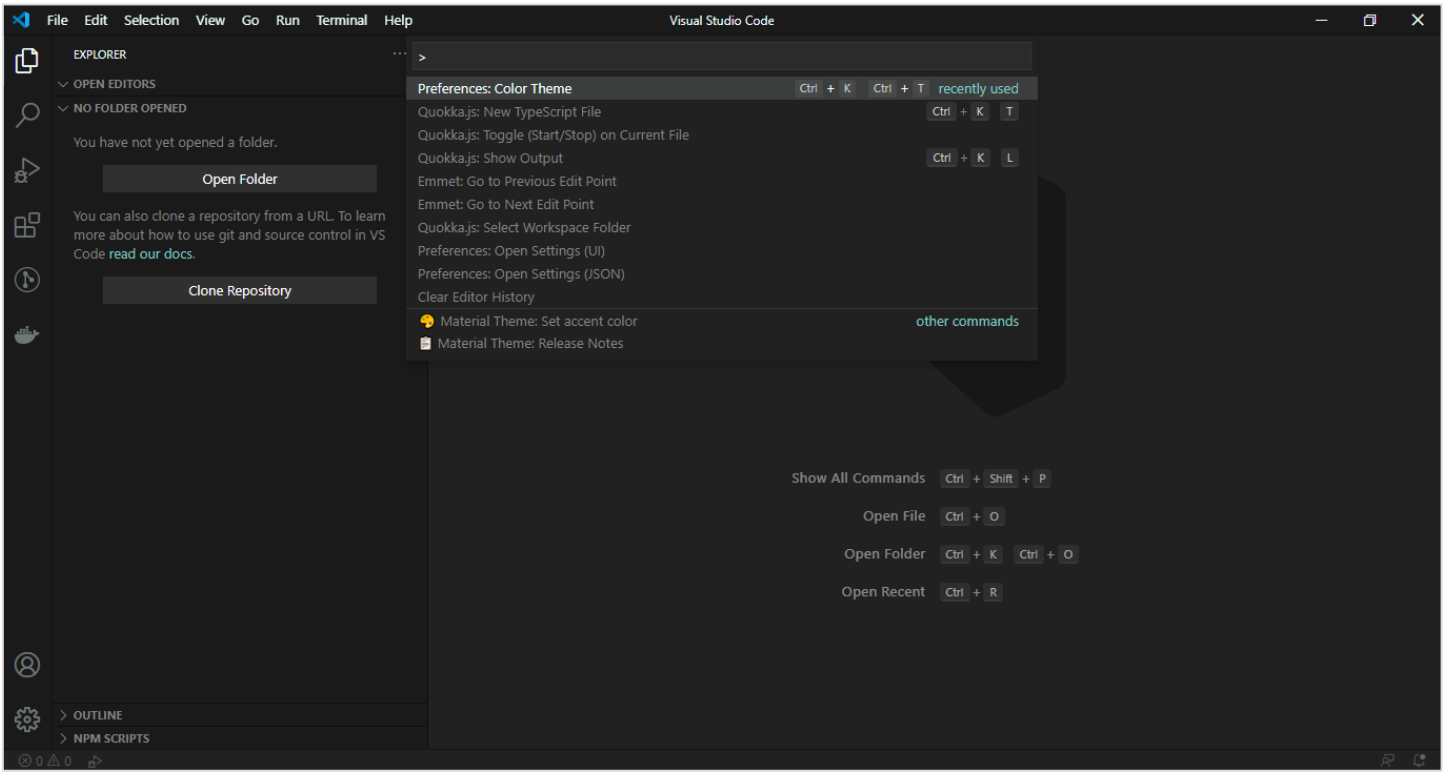


الأول go to file هو الذي أشرنا له سابقاً وهو التنقل بين ملفات المشروع، اما الثاني وهو # فنقوم بكتابته، كالتالي:

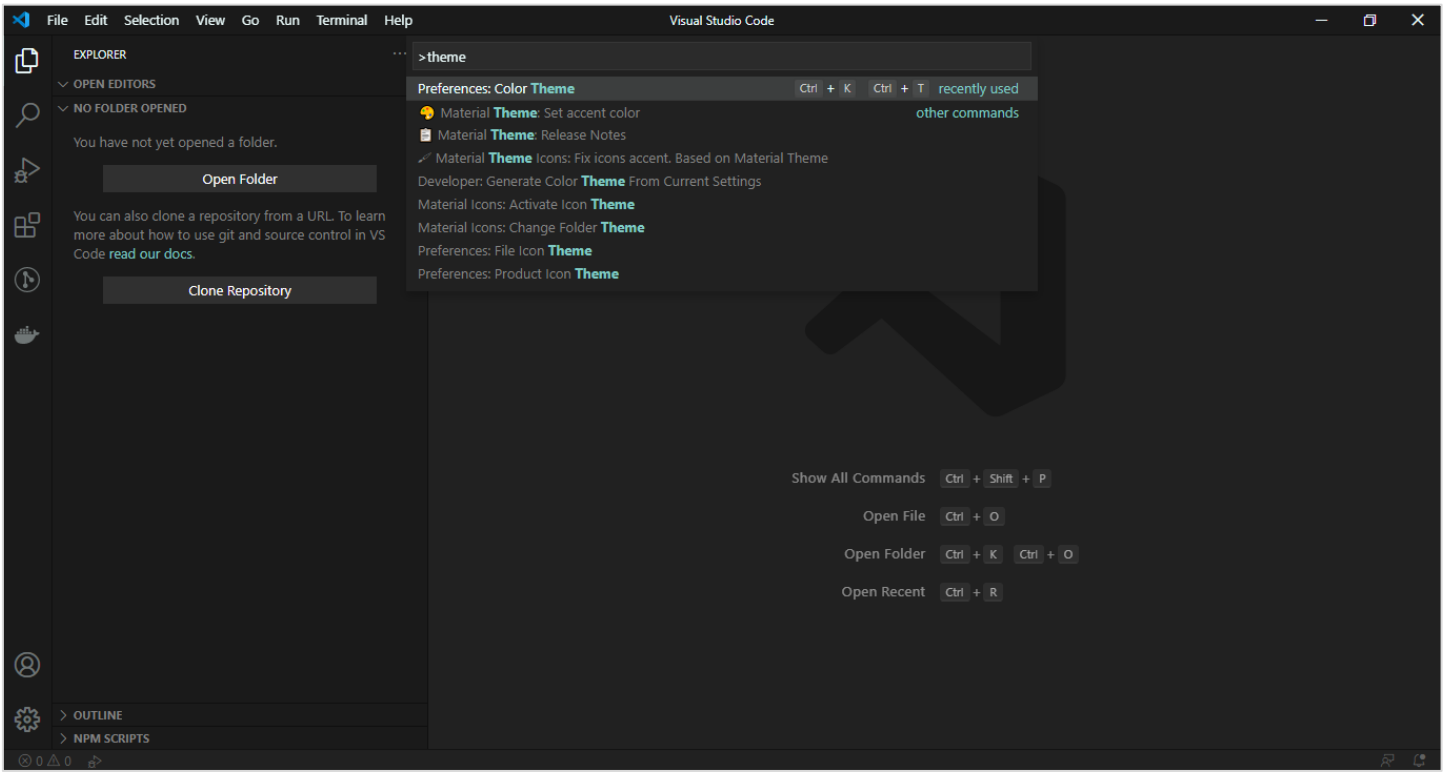


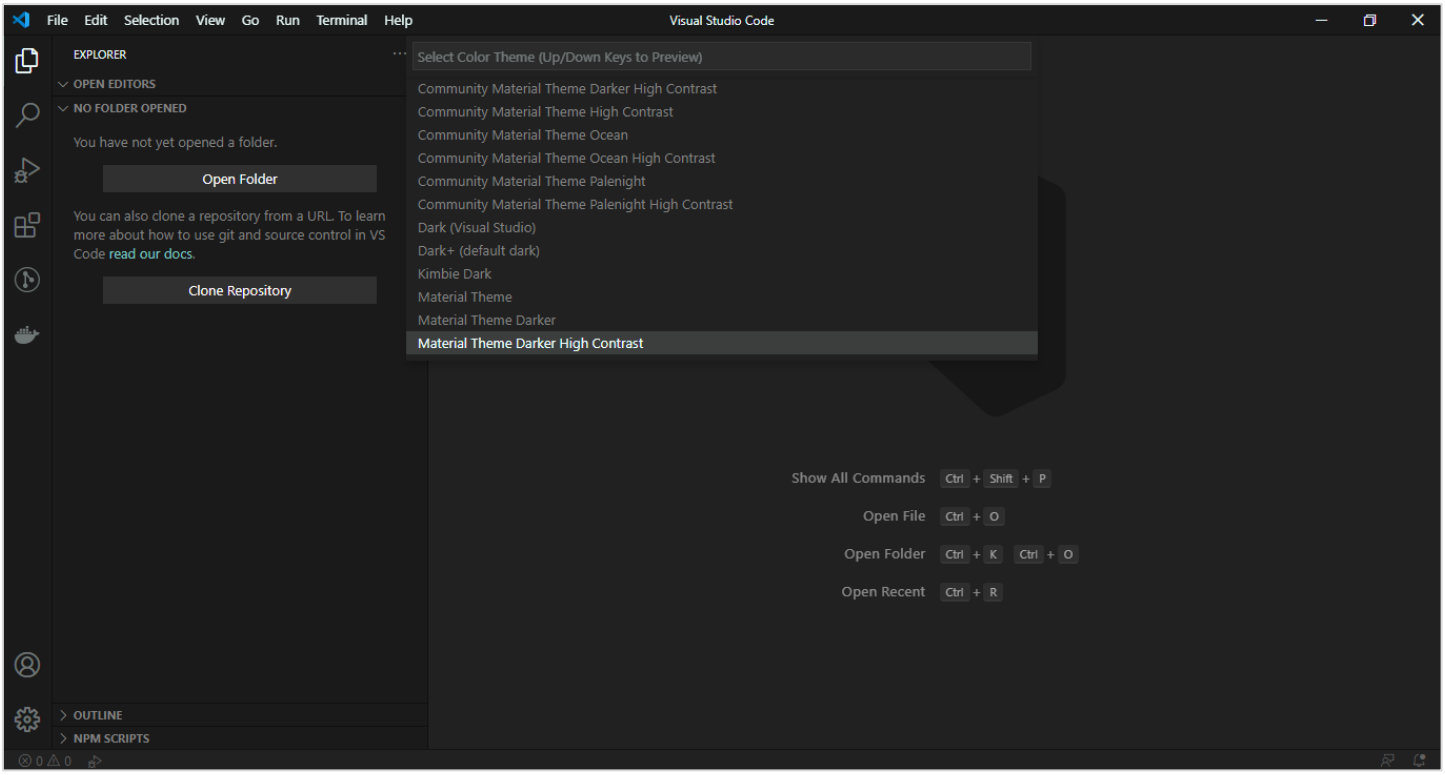
ونستفيد منها للوصول إلى متغير او خاصية او دالة معينة بحيث نكتب هذا الرمز ومن ثم (بدون مسافات) نكتب اسم الرمز الذي نُريد الانتقال له.

اما العلامة الأخرى وهي >، فعند كتابتها تظهر لنا القائمة التالية:

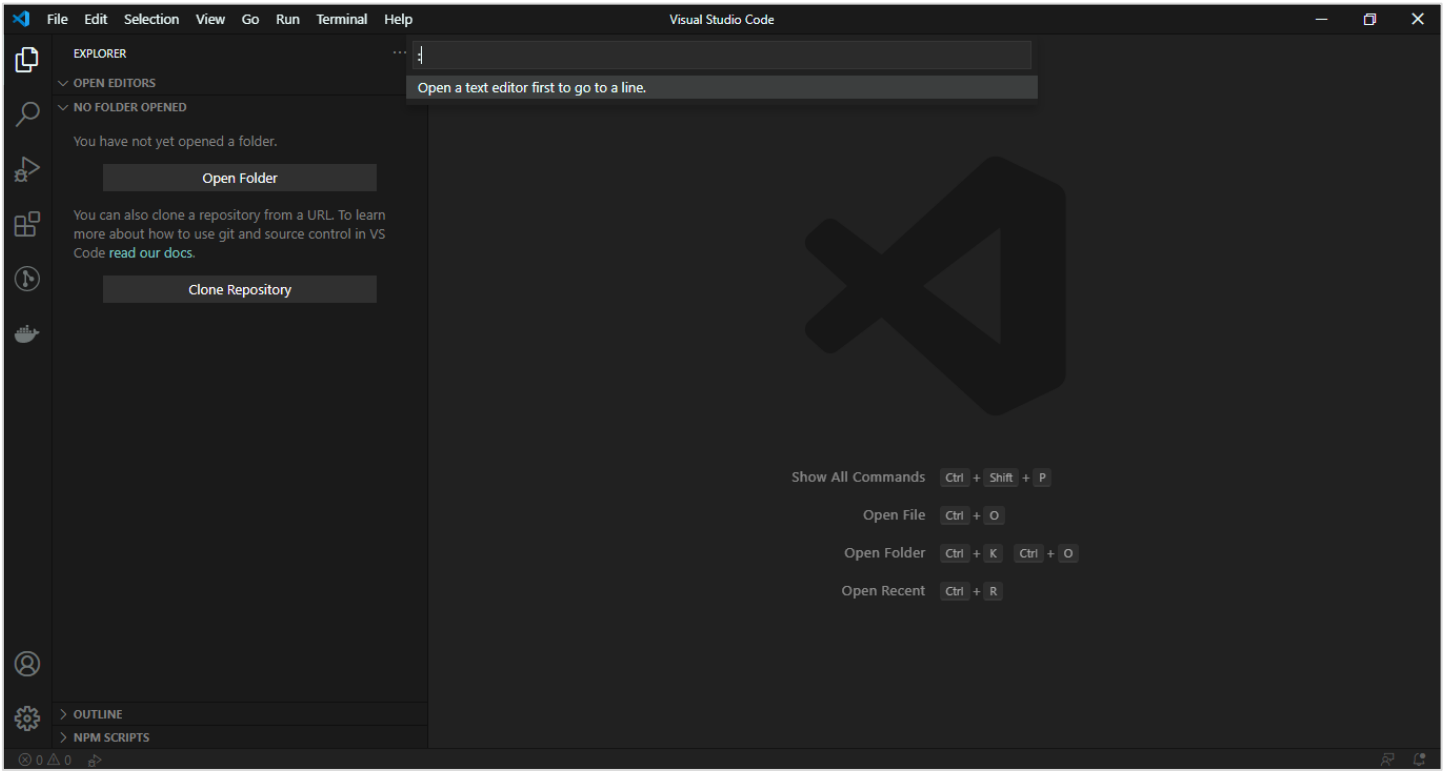


وهذه القائمة تحتوي على جميع الاختصارات والاوامر الخاصة بالتحكم بالتطبيق، ومنها على سبيل المثال theme والذي يُظهر قائمة بجميع الثيمات الموجودة في التطبيق مع إمكانية الاختيار منها، كالتالي:





ومن الأوامر ايضاً: حيث عن طريقها نضع رقم السطر لملف مفتوح وسوف ينتقل المؤشر إلى هذا السطر، كالتالي:



ومن الأوامر ايضاً @ حيث نستفيد منها للانتقال إلى دالة او خاصية او متغير لملف مفتوح، حيث لو افترضنا انه لدينا ملف مفتوح ويحتوي على كم من الدوال والخصائص والمتغيرات ولعرضها جميعا نقوم بكتابة الرمز @ وسوف تظهر لنا جميع هذه الرموز ونستطيع عندها اختيار أحدها، وايضاً الرمز :@ حيث يقوم بعمل تجميع بحسب نوع هذه الرموز بحيث يضع الدوال مع بعضها والخصائص مع بعضها وهكذا بقية الرموز الأخرى للملف المفتوح.

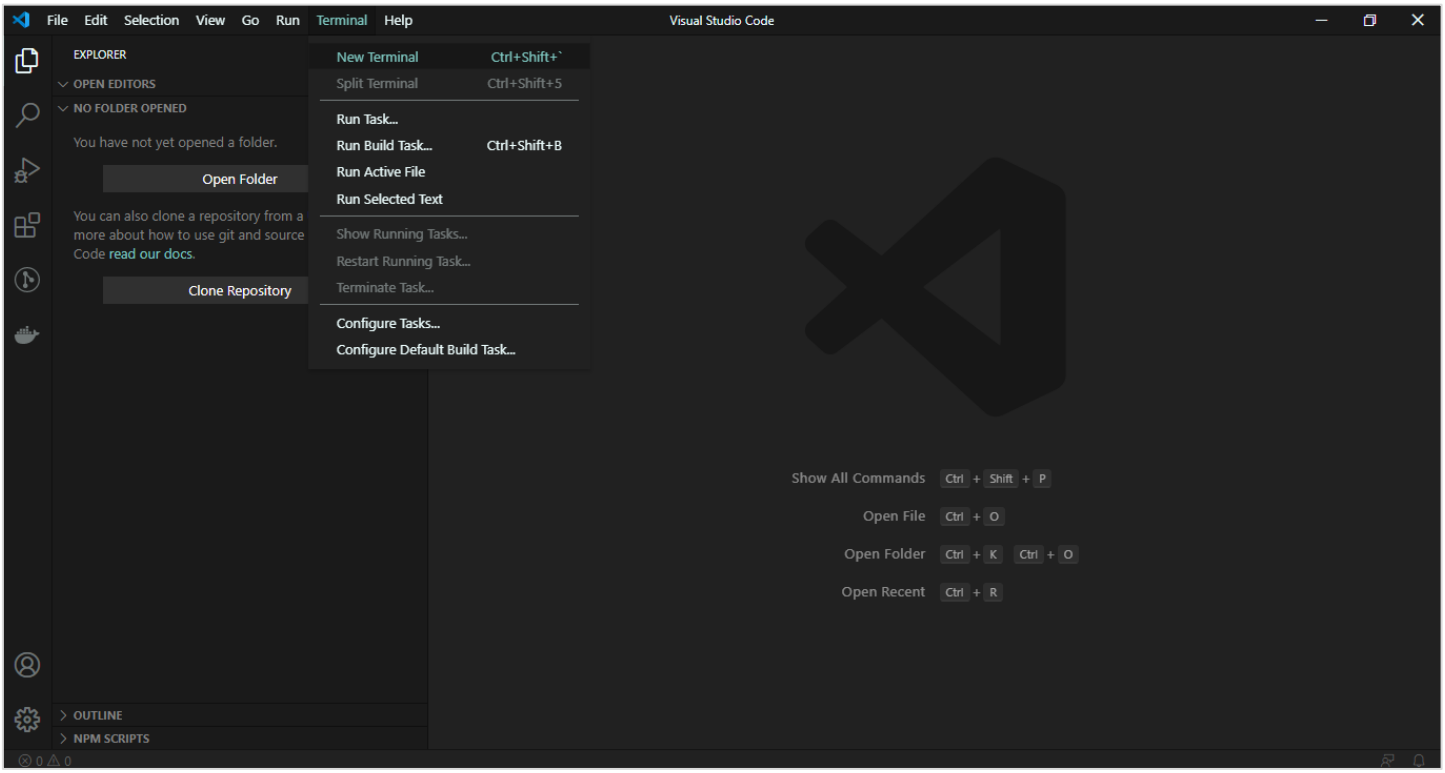
وتستطيع عزيزي المتعلم الاطلاع على باقي هذه الأوامر واكتشافها بنفسك، او البحث والاستزادة عن Command Palette من خلال شبكة الانترنت.

اما باقي الأوامر في القائمة View فهي ايضاً تستطيع الاطلاع عليها وتجربتها بنفسك.

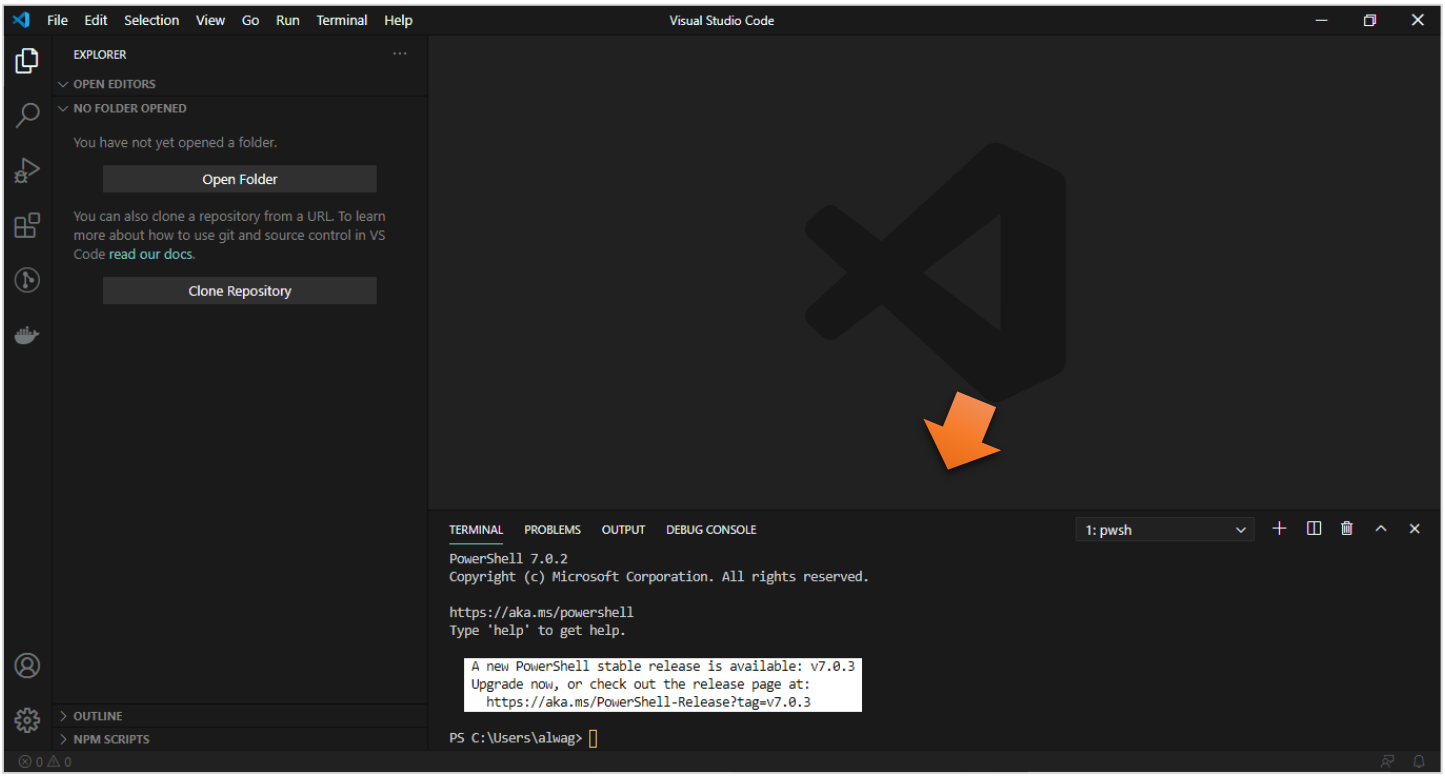
5.1.3.2.قائمة Terminal:

وهذه القائمة ايضاً من القوائم المهمة، حيث تقوم بجلب جميع terminals الموجودة عندك بالجهاز وعرضها لديك، وليس هذا فقط وانما سوف تعرض لك المسار كامل للمشروع الخاص بك بحيث تسهل لك التعامل مع المشروع الخاص بك، ولا اخفيك عزيزي المتعلم اننا في مشاريع Angular سوف نستخدم terminal بشكل متكرر ويكفي ان نقول ان Angular CLI نتعامل مع جميع أوامره عن طريق terminal.

اما الأوامر التي تحتويها هذه القائمة هي متعددة والذي يهمنا منها بالتحديد هو الامر New Terminal، كالتالي:

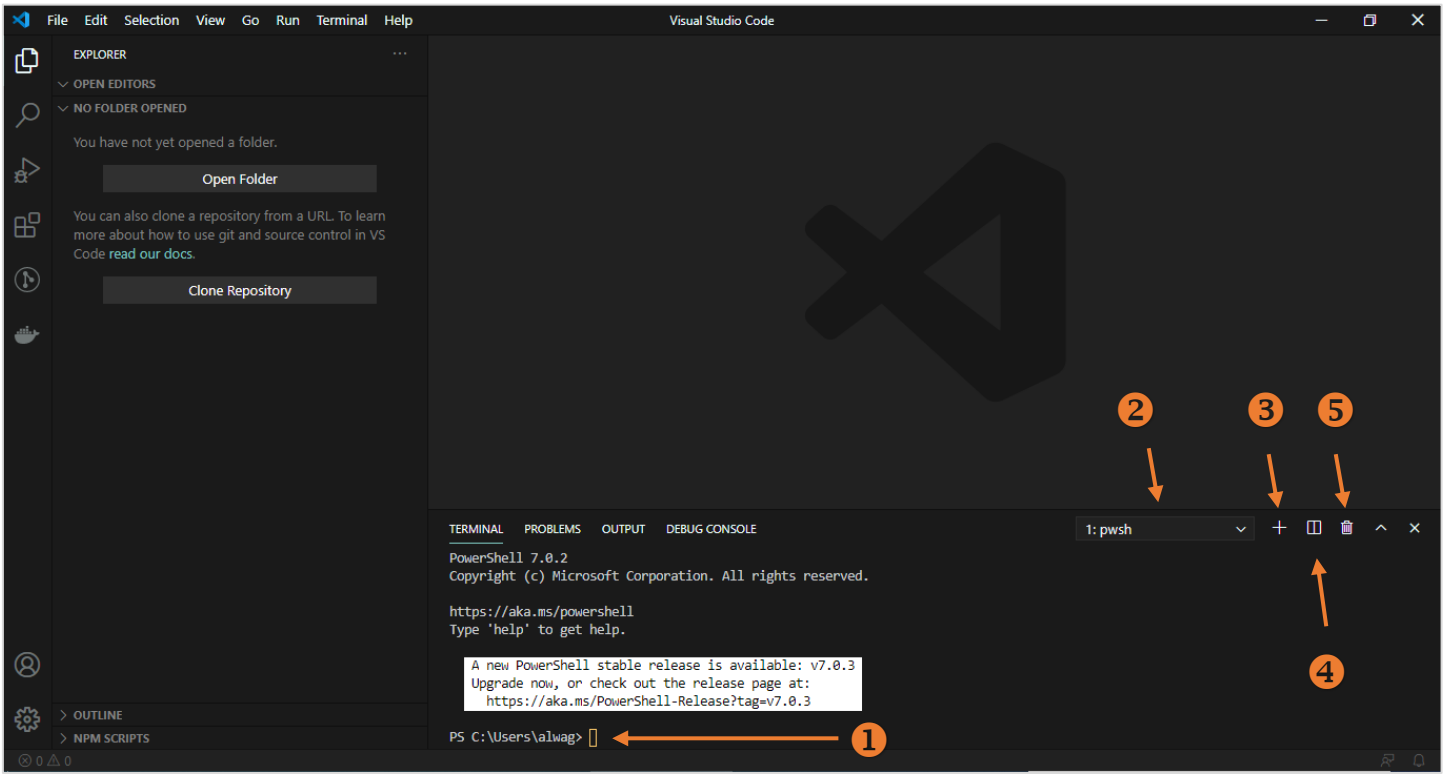


وعند الضغط على هذا الامر سوف تظهر لنا الشاشة التالية:



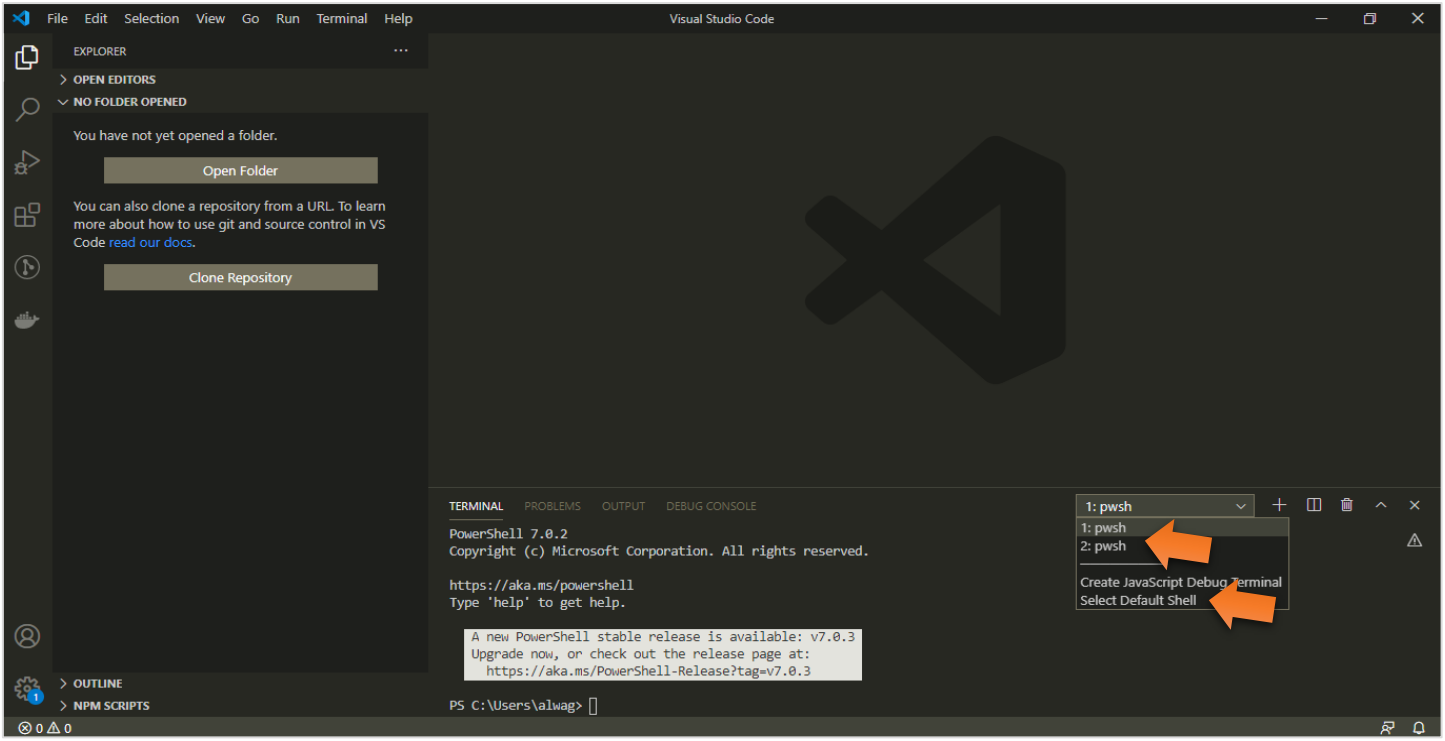
هذه الشاشة تحتوي على مجموعة من التبويبات هي TERNINAL وPROBLEMS وOUTPUT وDEBUG CONSOLE، مع العلم ان جميع هذه التبويبات نستطيع التحكم بإظهارها واخفاءها عن طريق القائمة View.

وما يهمنا من هذه التبويبات هو TERMINAL حيث عند اختيار هذه الشاشة سوف تظهر لنا شاشة تحتوي على مجموعة من الازرار والأدوات، كالتالي:

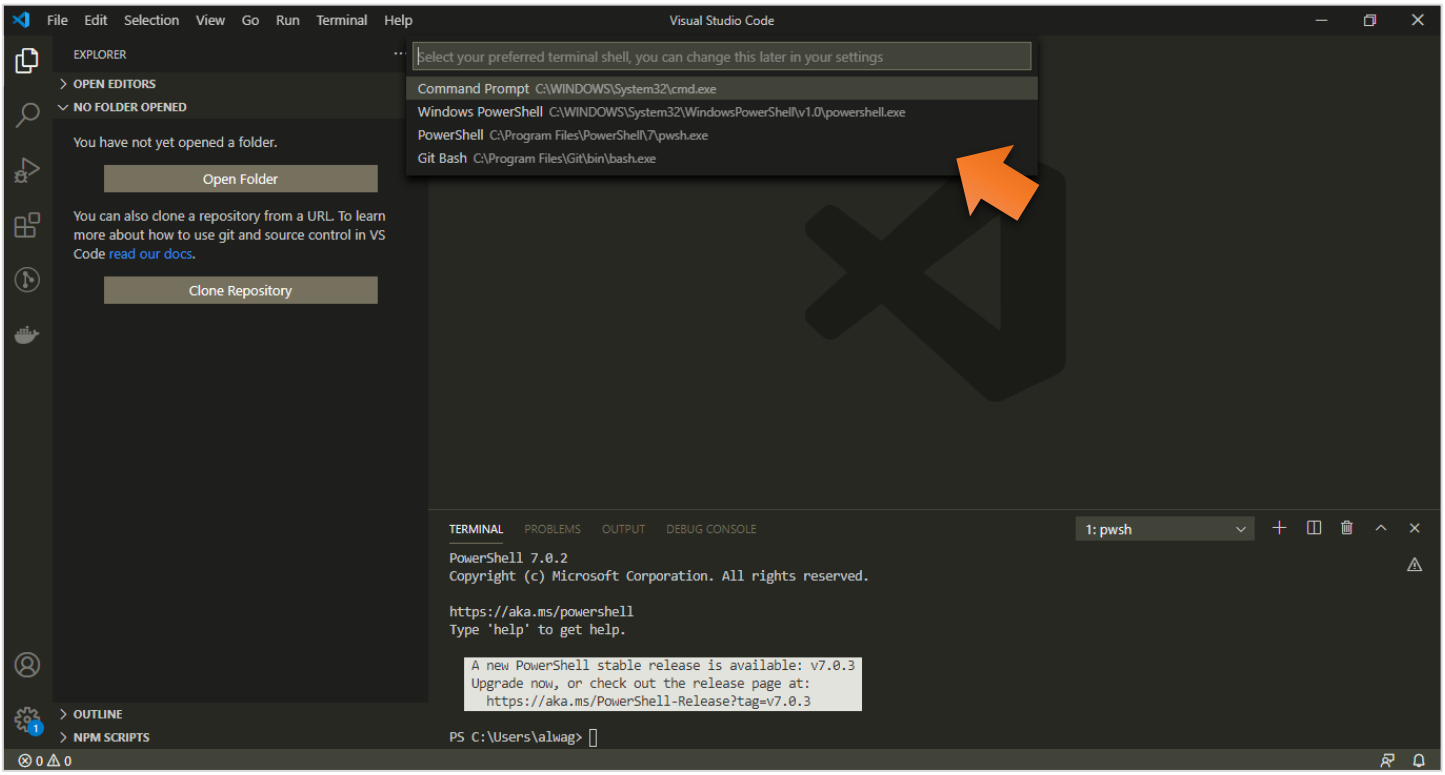


اول جزء هو المكان الذي نقوم بكتابة الأوامر ونلاحظ انه بشكل افتراضي اظهر لنا هذا الامتداد Path ولو اننا قمنا ببناء مشروع وفتحنا هذا المشروع عن طريق التطبيق سوف يتغير هذا الامتداد ليكون بنفس امتداد المشروع الخاص بنا، وهذه ميزة رائعة تسهل علينا كثير من الأمور لأننا عند التعامل مع سطر أوامر Angular CLI نحتاج إلى ان نكون بنفس امتداد المشروع لكي نقوم بتنفيذ هذه الأوامر.

اما رقم 2 فتعرض عدد terminals المفتوحة، فإي terminal نُضيفه في الرقم 3 يظهر لنا في هذه القائمة في الرقم 2، وايضاً نستطيع اختيار terminal الافتراضي في حال كان لدينا أكثر من terminal، والآن لنقوم بالضغط على زر علامة + في الرقم 3 ومن ثم نقوم بالضغط على القائمة في الرقم 2 ولنرى النتيجة، كالتالي:

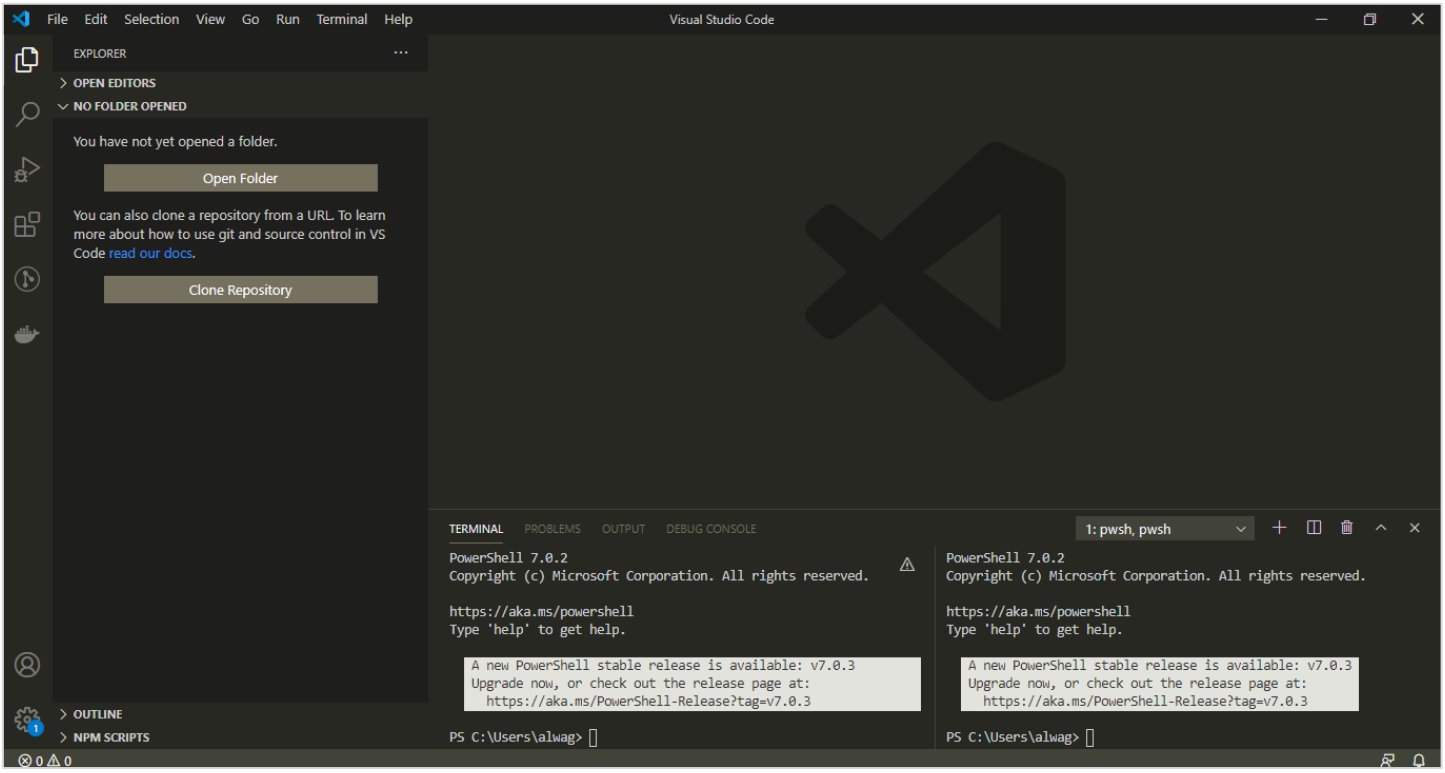


نلاحظ وجود نسختين من terminal نستطيع التبديل بينهما، وايضاً هنالك امر هو Select Default Shell ووظيفته يحدد terminal الافتراضي لذلك لنقوم بالضغط عليه ولنرى النتيجة، كالتالي:



نلاحظ ظهرت لنا جميع terminals التي تم تحميلها في جهاز الكمبيوتر في Command Palette، ونستطيع اختيار ما يناسبنا.

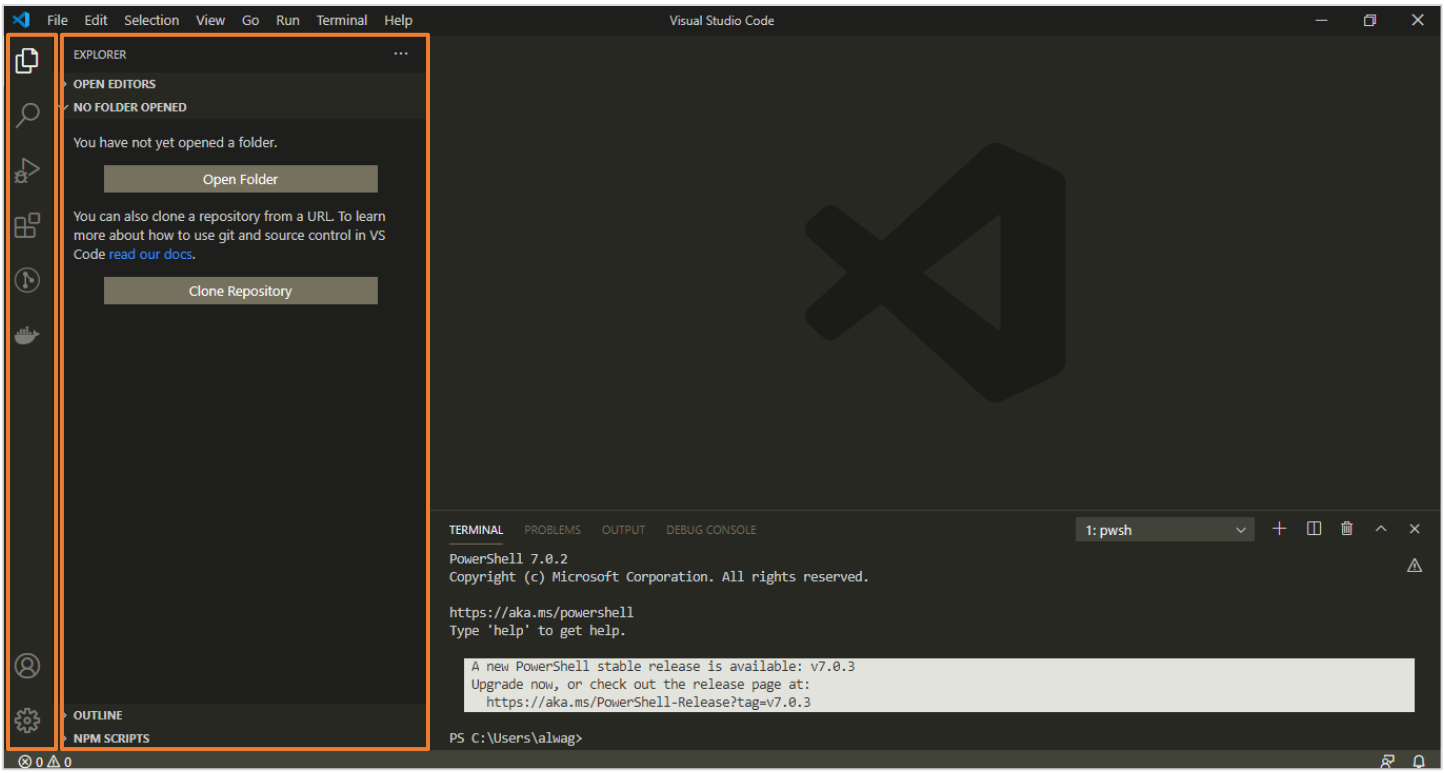
وفي حال أردنا ان نظهر اثنين terminal جنباً إلى حنب فنختار الزر في الرقم 4 حيث يقوم هذا الامر بوظيفة + بالإضافة على اظهار terminal الجديد بجانب القديم، كالتالي:



نلاحظ أضاف اثنين terminals بجنب بعض بالإضافة.

2.3.2.شريط Activity Bar:

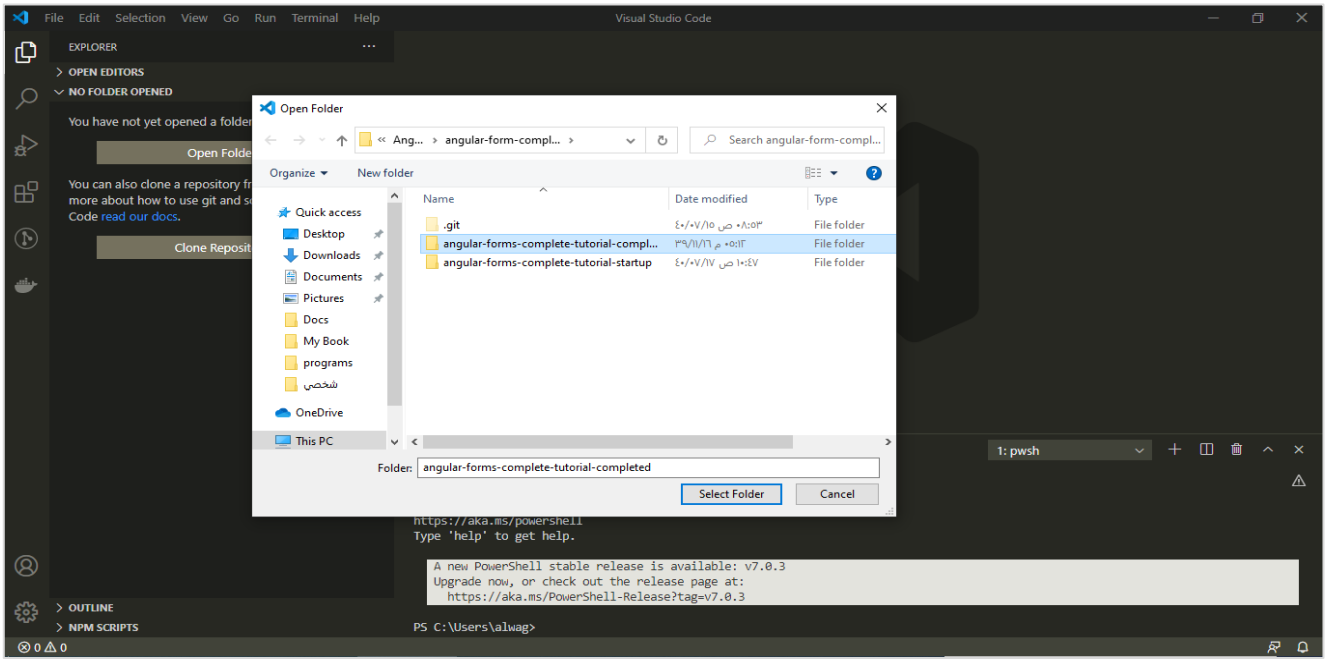
وهو الشريط الموجود في العادة في يسار الشاشة ويحتوي على مجموعة من القوائم كل قائمة تظهر لنا شاشة تحتوي على مهام معينة، كالتالي:

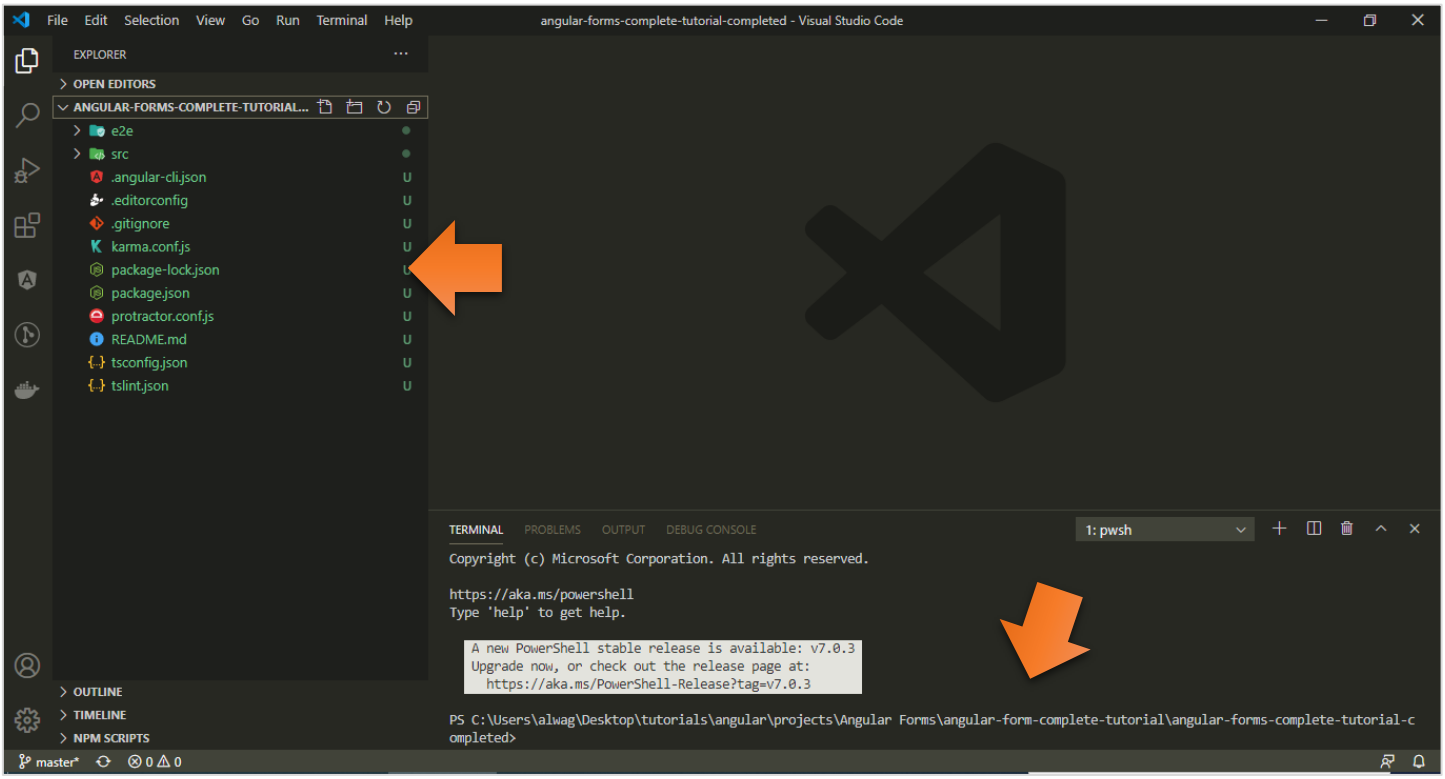


1.2.3.2.قائمة Explorer:

بشكل افتراضي عندما نقوم بفتح البرنامج لأول مرة فأنه سوف يظهر لنا زرين الأول باسم Open Folder لاختيار مشروع سابق او Clone Repository وهو لتحميل مشروع موجود في موقع GitHub عن طريق الرابط الخاص بهذا المشروع.

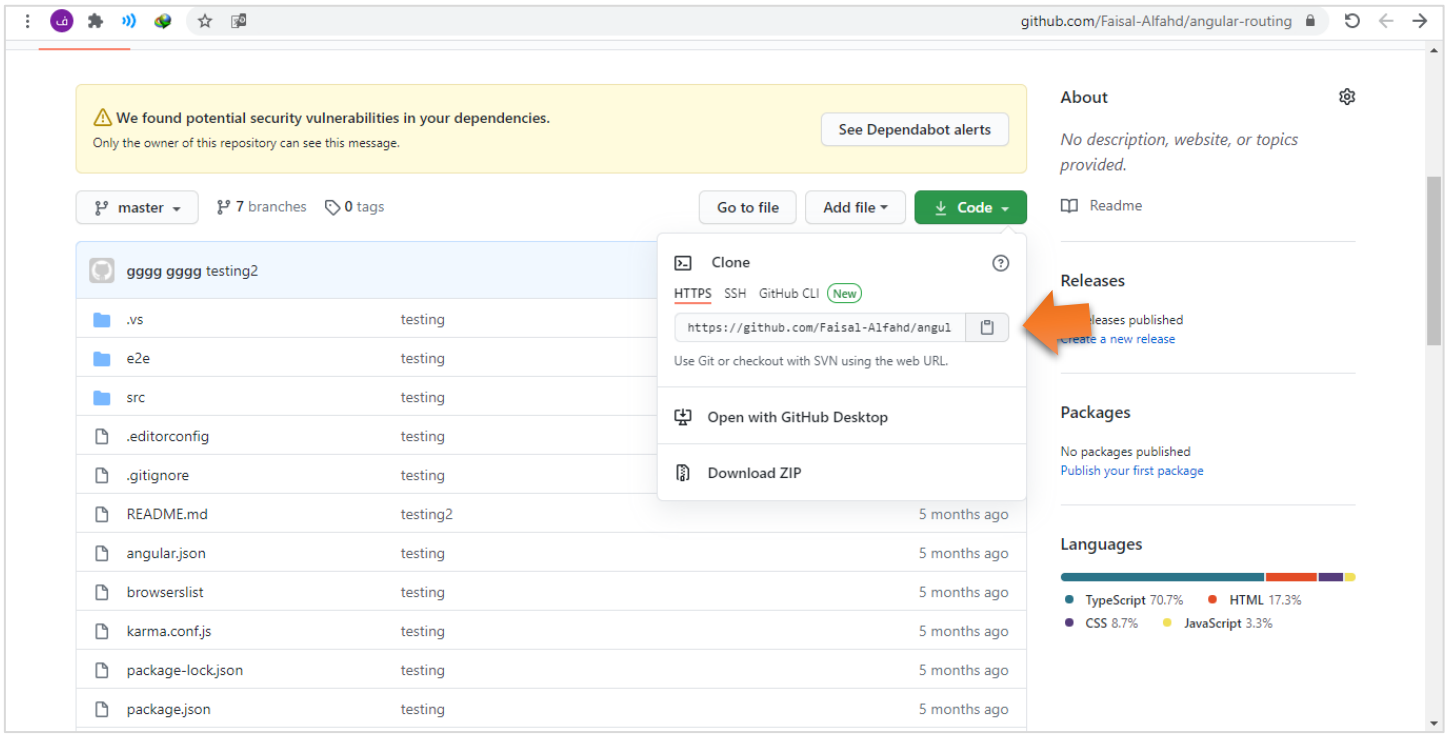
ويوجد لدي مشروع سابق من أحد مشاريع Angular لنقوم بفتحه وعرضه عن طريق الزر Open Folder، كالتالي:



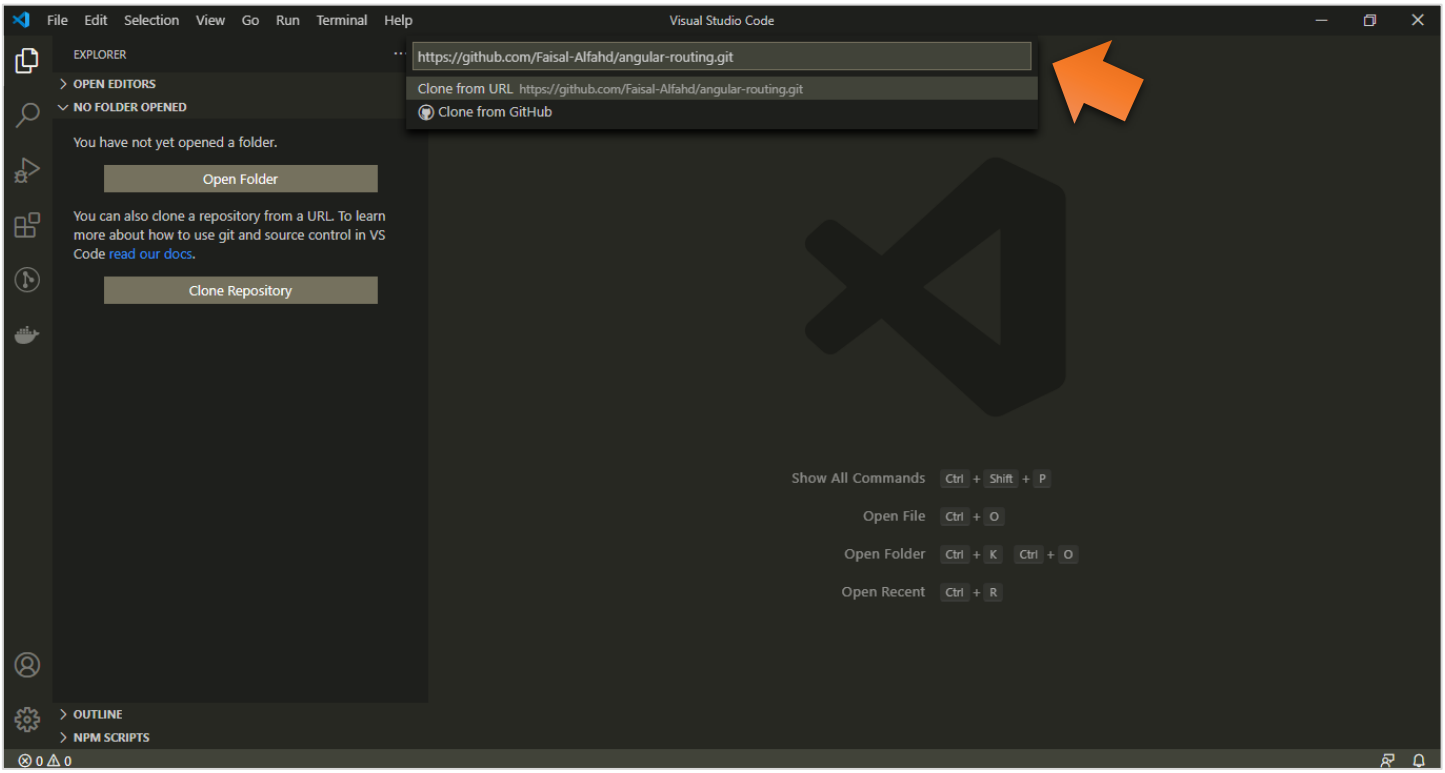


نلاحظ ظهرت لنا جميع أجزاء المشروع وبنفس الوقت في terminal بشكل تلقائي تم توجيهه إلى مسار هذا المشروع.

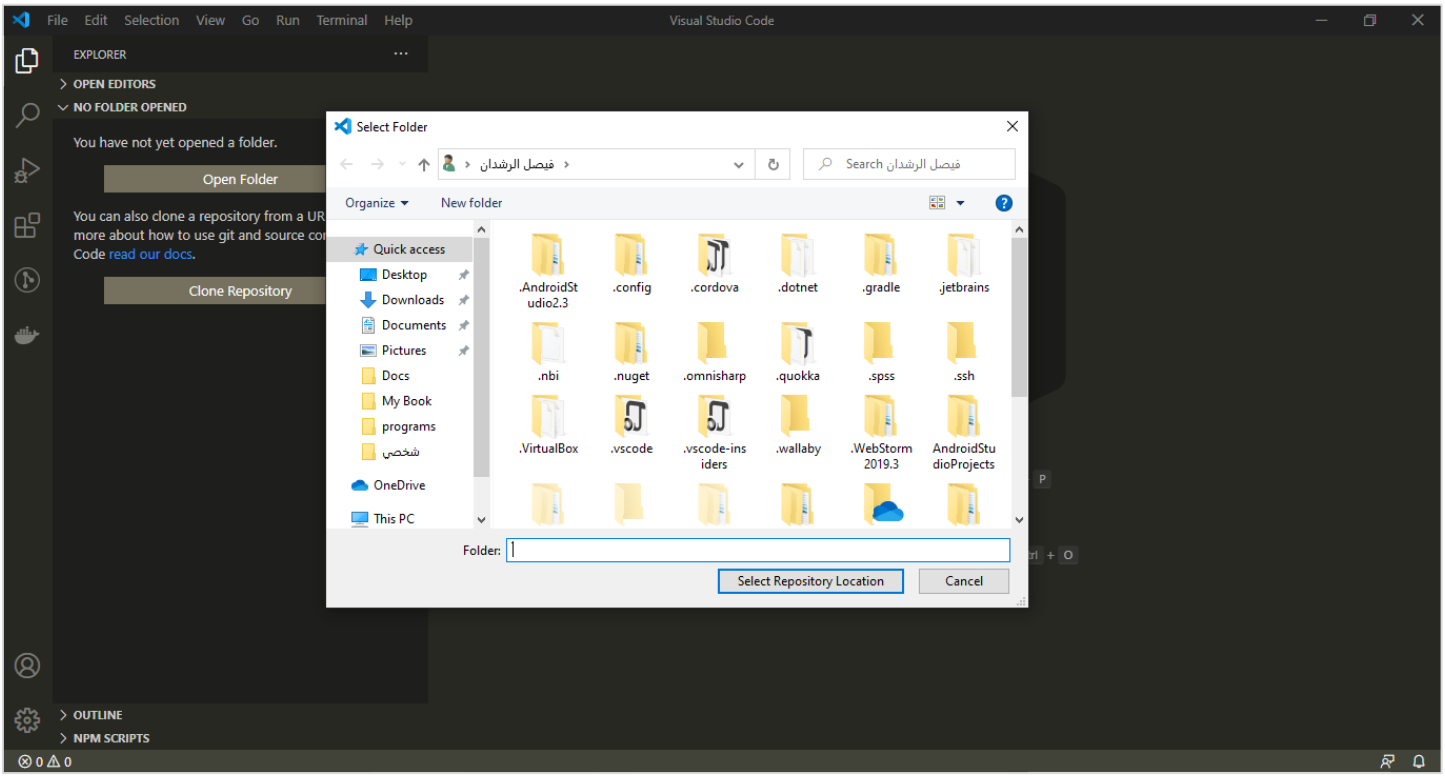
والآن لنقوم بالذهاب إلى قائمة File ونختار New Window لكي نفتح نافذة جديدة ونقوم بتجريب الزر الآخر Clone Repository، سوف أقوم بعرض مشروع بسيط قمت برفعه سابقاً على GitHub، كما في الشكل التالي:



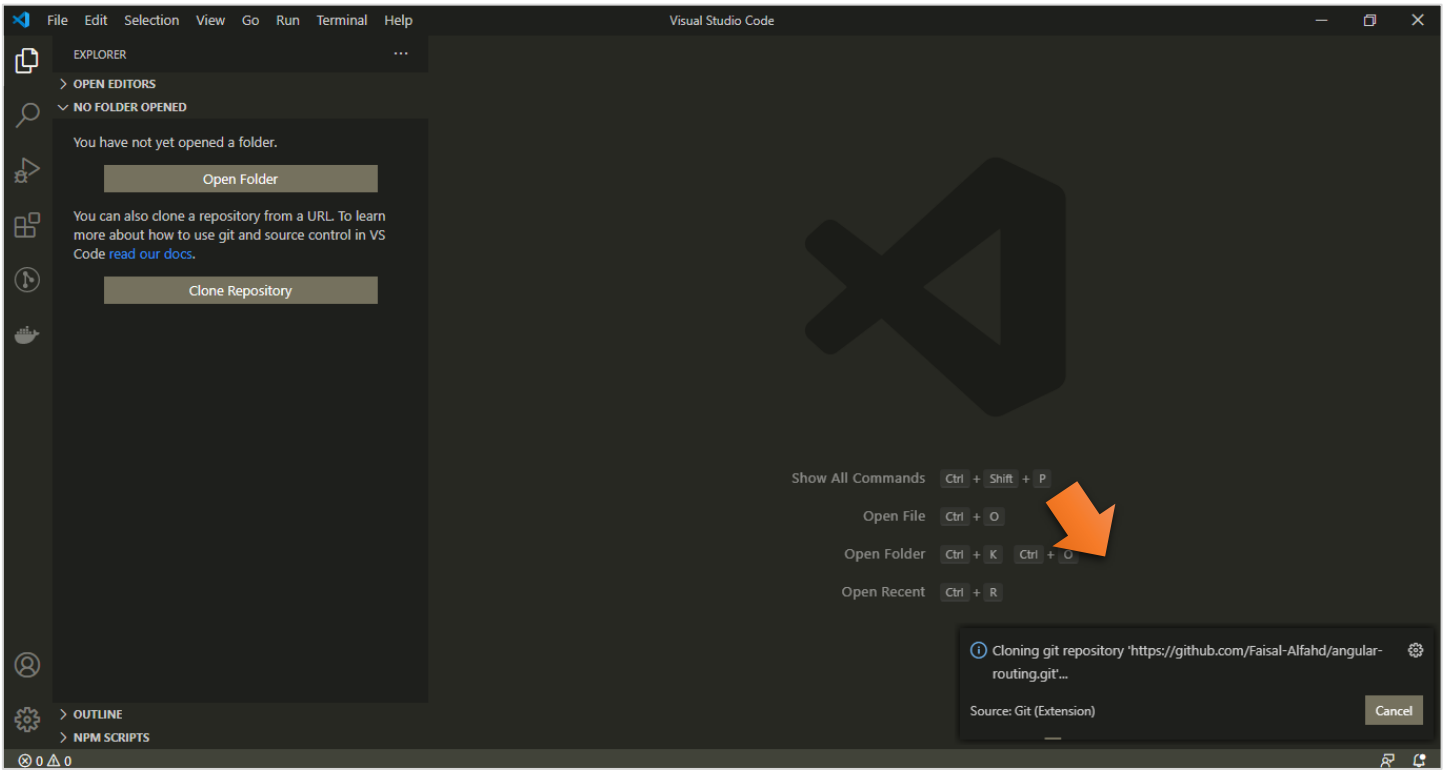
وبعدها نقوم بالضغط على زر نسخ لكي ننسخ الرابط: https://github.com/Faisal-Alfahd/angular-routing.git، وبعدما تحصلنا على الرابط سوف ننتقل الآن إلى تطبيق VS Code ومن خلال اختيار الزر Explorer في الشريط Activity Bar نضغط على الزر Clone Repository، كالتالي:



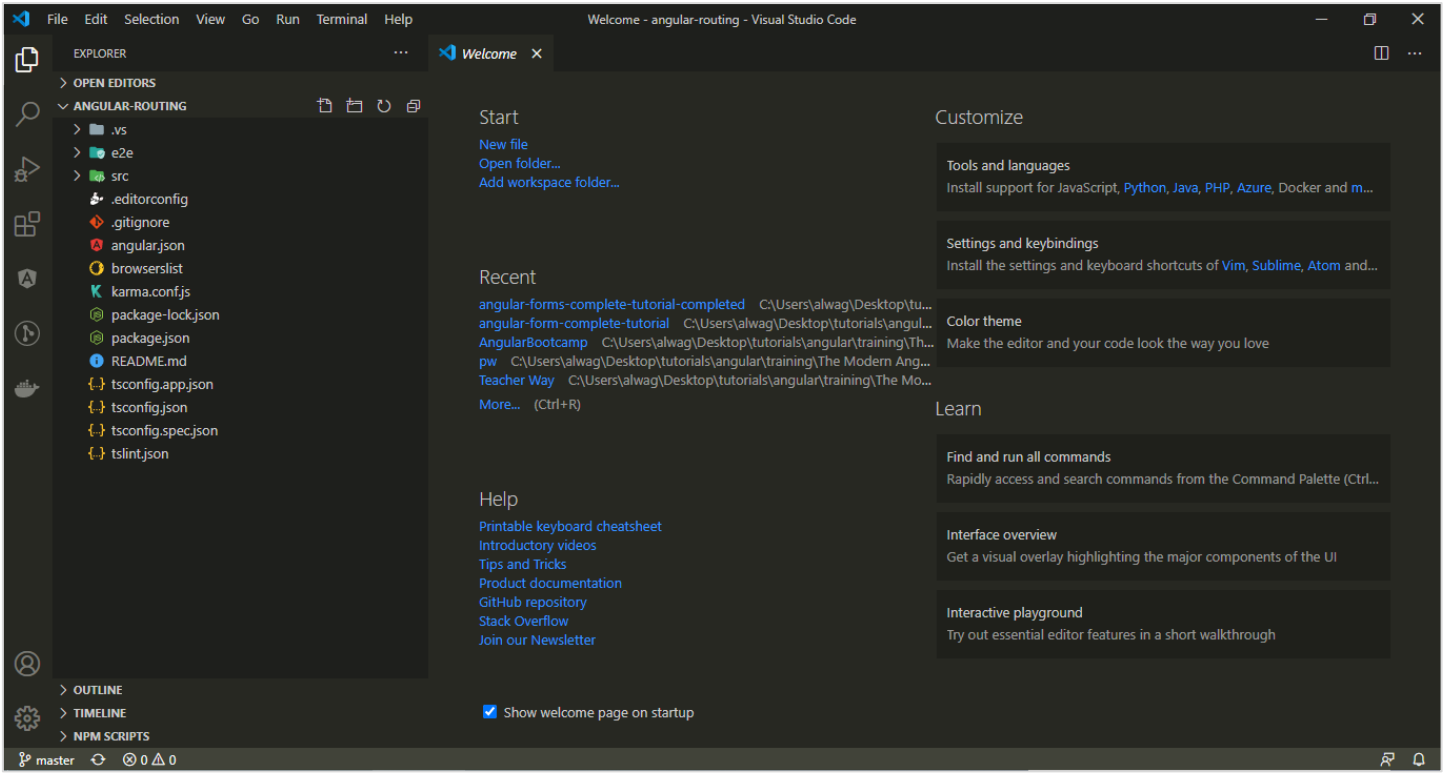
بعدما نلصق الرابط في المكان المحدد نضغط على زر Enter من لوحة المفاتيح، كالتالي:



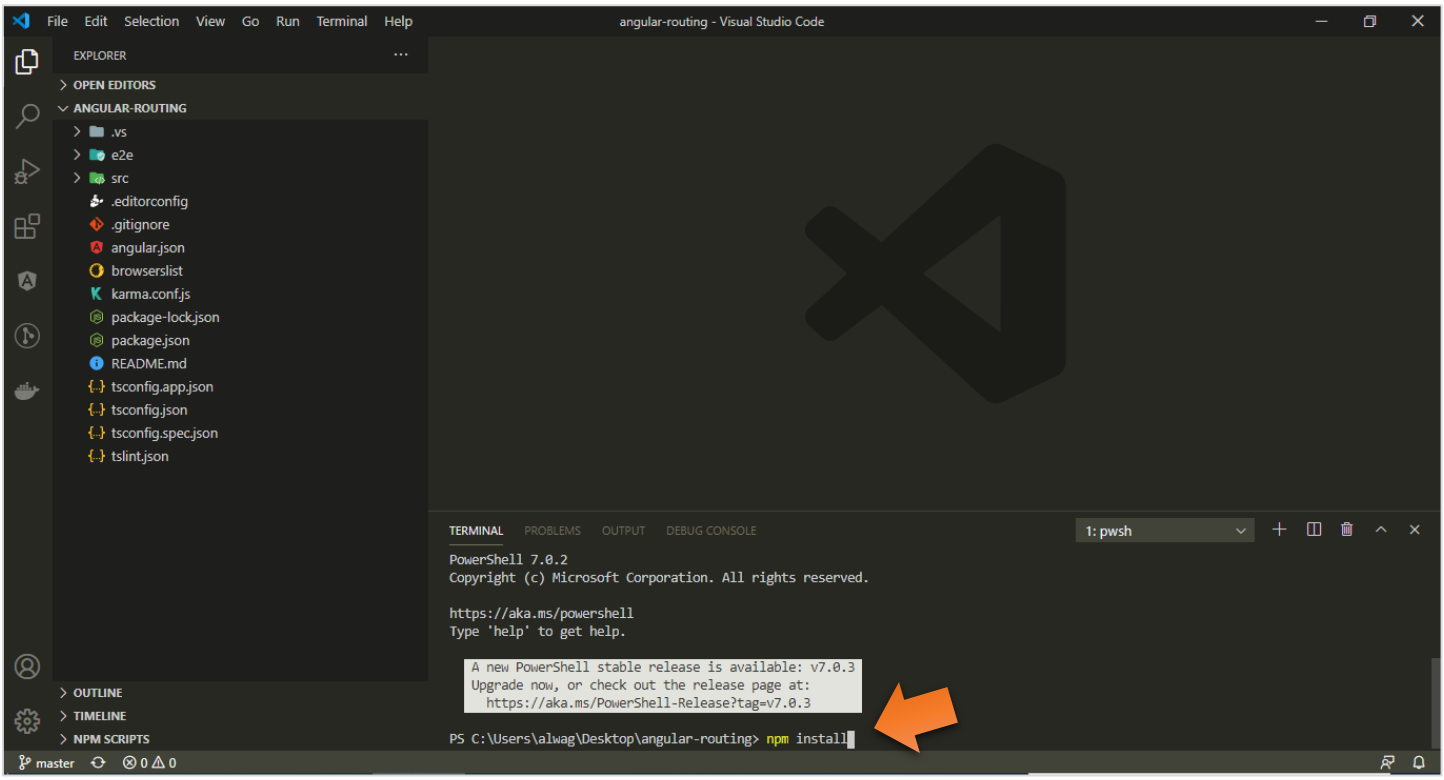
سوف يطلب منا مكان لنحفظ فيه هذا المشروع في الجهاز لدينا، لنقوم بتحديد مكان على اجهزتنا ومن ثم نضغط على الزر Select Repository Location، كالتالي:



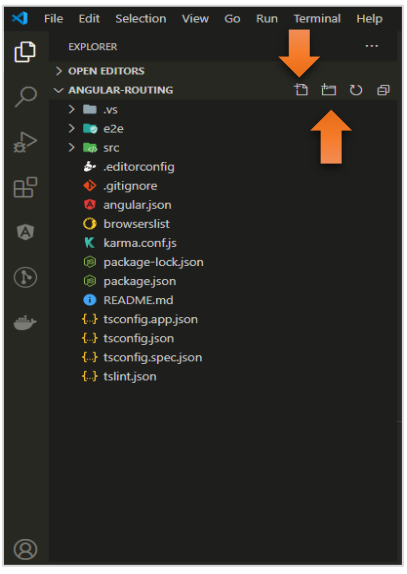
سوف يقوم بالاتصال بموقع GitHub ومن ثم يقوم بتحميل كافة ملفات هذا المشروع وحفظها في المكان الذي حددته له مسبقاً، اما الوقت الذي يستغرقه فيختلف بحسب حجم المشروع وسرعة الاتصال لديك، وعند الانتهاء سوف تظهر لنا رسالة تطلب منا ان نقوم بفتح هذا المشروع ويوجد زرين الأول open أي قم بفتحه بنفس النافذة الخاصة بتطبيق VS Code والثانية Open New Window أي قم بفتح هذا المشروع في نافذة جديدة، وبما اننا بالأساس قمنا سابقاً بفتح نافذة جديدة فلنقم باختيار الزر Open، وسوف تظهر لنا الشاشة التالية:



نلاحظ ظهرت لنا جميع ملفات المشروع، وبما اننا قمنا بتحميل هذا المشروع عن طريق الأنترنت وبالتحديد GitHub فنحتاج إلى خطوة أخيرة قبل تشغيل هذا المشروع وهي كتابة الامر npm install او اختصاراً npm i، وفائدة هذا الامر انه يقوم بتحميل جميع الملفات المساعدة التي يحتاجها هذا المشروع بالذات لكي تعمل، لأن مشاريع Angular عند رفعها على موقع GitHub او أي موقع آخر، فأنه يتم رفعها بدون هذه الملفات لكي لا يكون حجم المشروع ضخم، واي مبرمج آخر يقوم بتحميل هذا المشروع يقوم فقك بكتابة الامر السابق وسوف يتم تلقائياً قراءة ملف معين موجود من ضمن ملفات المشروع وبناءً عليه سوف يتم تحميل جميع هذه الملفات والمكتبات المساعدة، وسوف يكون لنا وقفة مع هذا الملف وشرحها بالتفصيل عندما نتكلم عن Angular Project Structure في الأقسام القادمة بإذن الله.

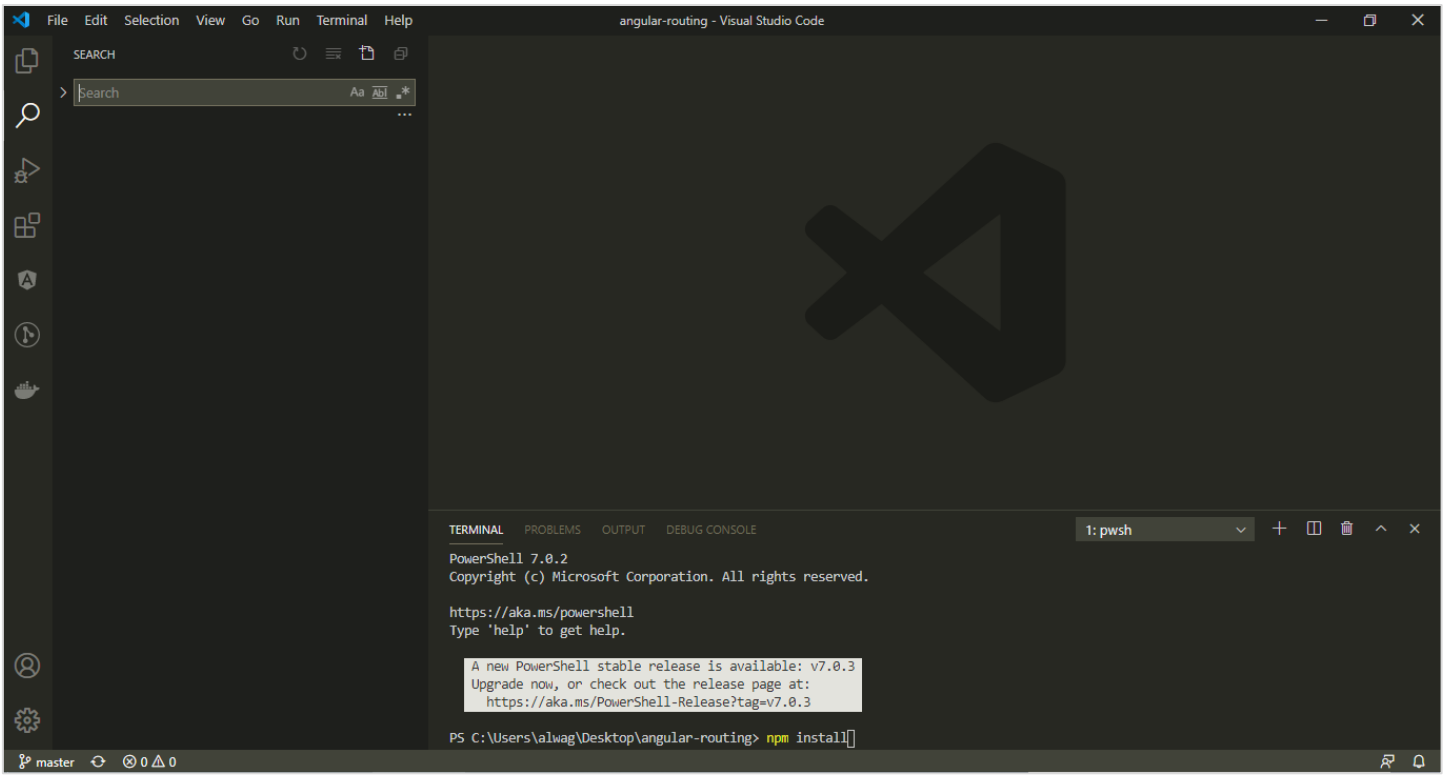


وآخر جزء اريد ان اوضحه هو عند فتح المشروع في هذه القائمة فأنه سوف تظهر لنا مجموعة من الأزرار في الأعلى، ما يهمنا بالتحديد هو زرين، الأول New File لإضافة ملف جديد والثاني New Folder لإضافة مجلد جديد، كالتالي:

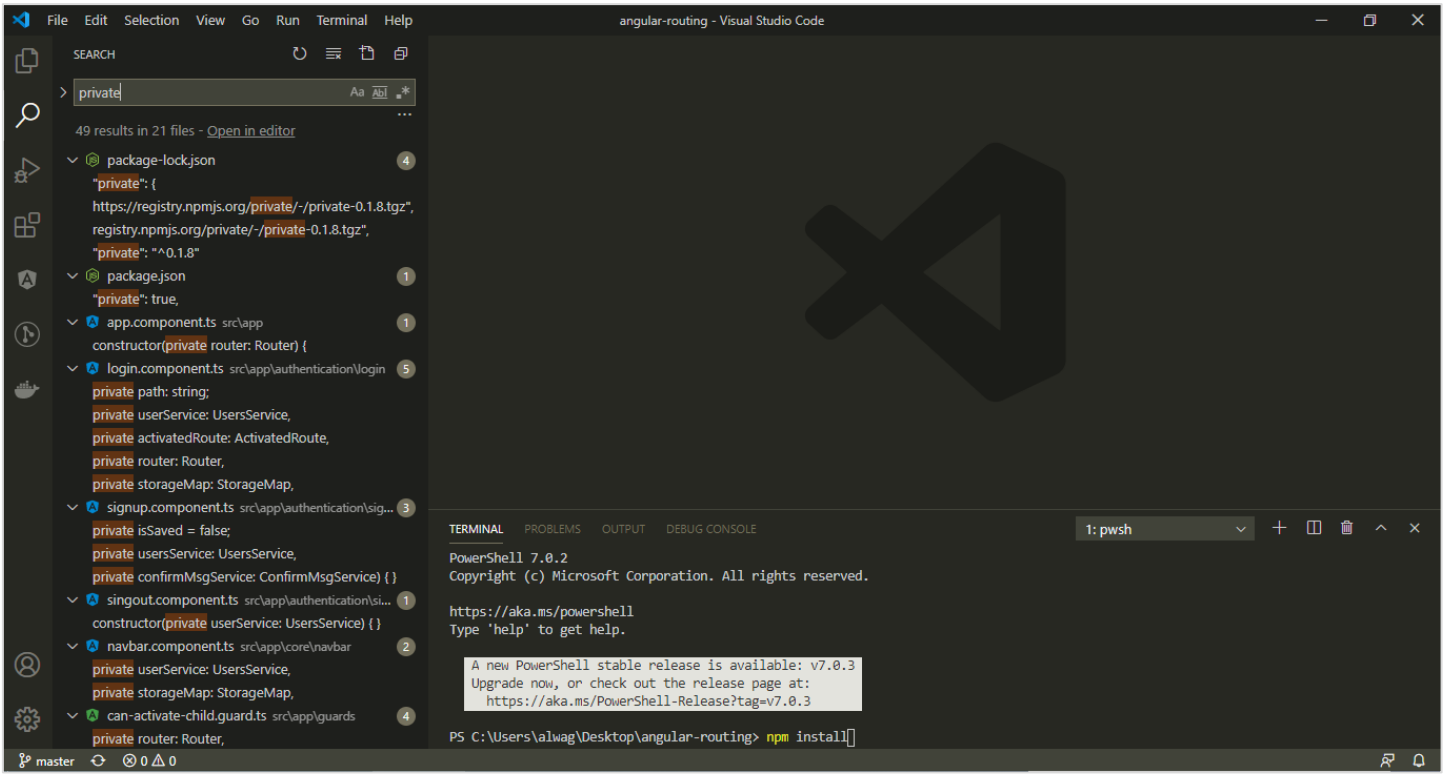


2.2.3.2.قائمة Search:

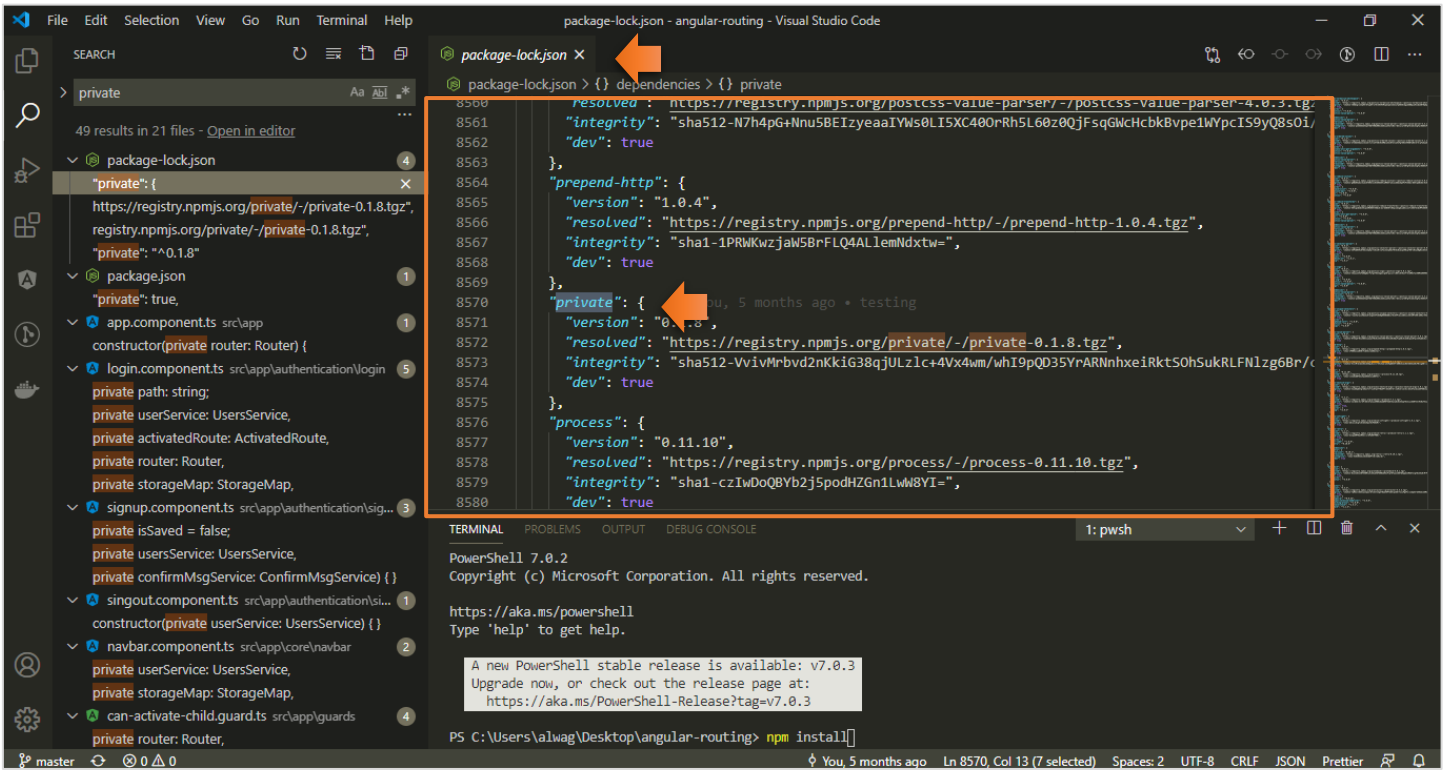
ومن اسمها تقوم هذه القائمة بالبحث في جميع أجزاء ملفات المشروع وبنفس الوقت نستطيع استبدال هذا البحث بقيم أخرى وسوف نطبق هذا الجزء على المشروع الذي قمنا بعمل له Clone Repository، كالتالي:



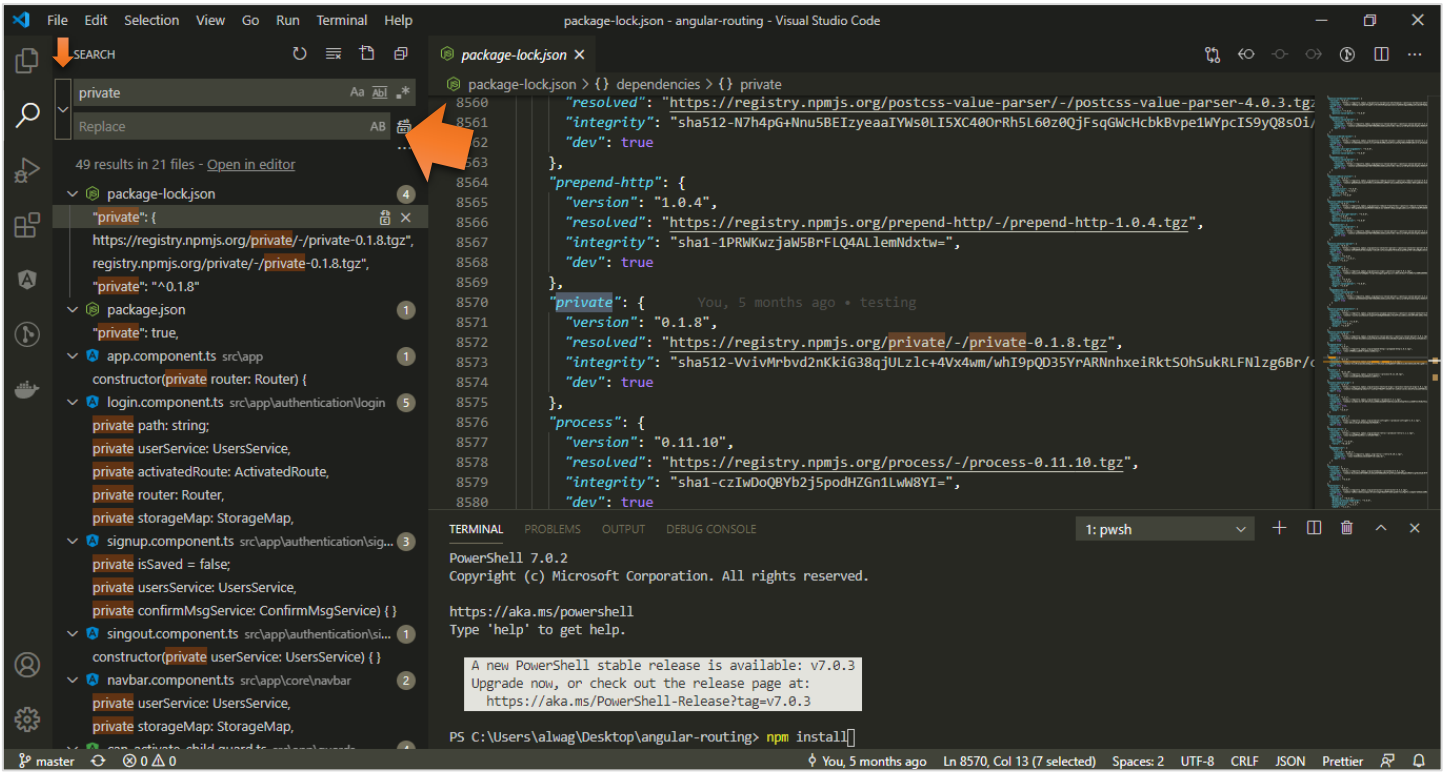
فمثلاً نستطيع البحث عن كلمة private، كالتالي:



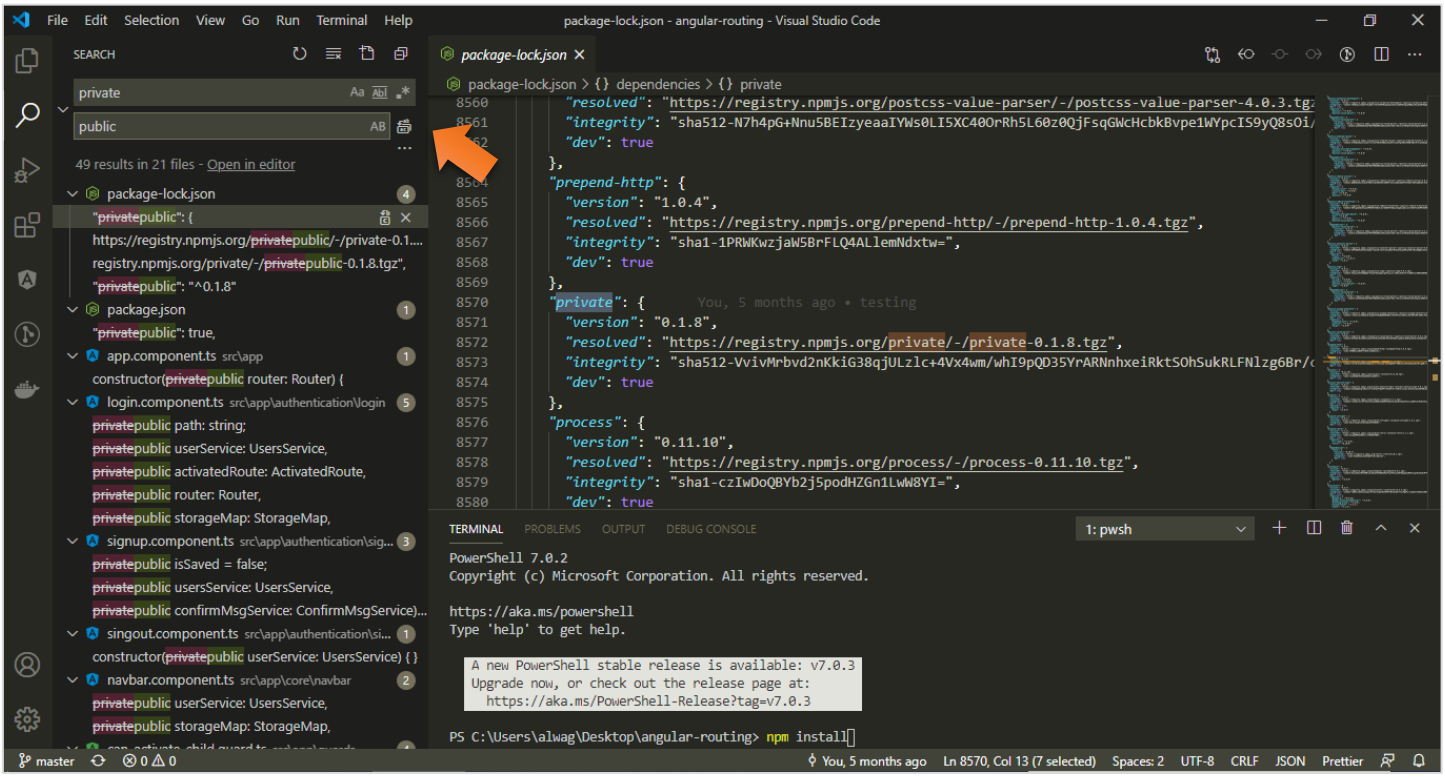
نلاحظ ظهرت لنا جميع الاسطر التي تحتوي على هذه القيمة بجميع أجزاء ملفات المشروع، ونستطيع اختيار أي منها وسوف يظهر لنا الملف الموجود فيه هذا السطر، كالتالي:



كما نستطيع استبدال هذه القيمة بقيمة أخرى وذلك بالضغط على علامة < الموجودة بجانب مربع البحث الذي اضفنا له القيمة private قبل قليل، كالتالي:



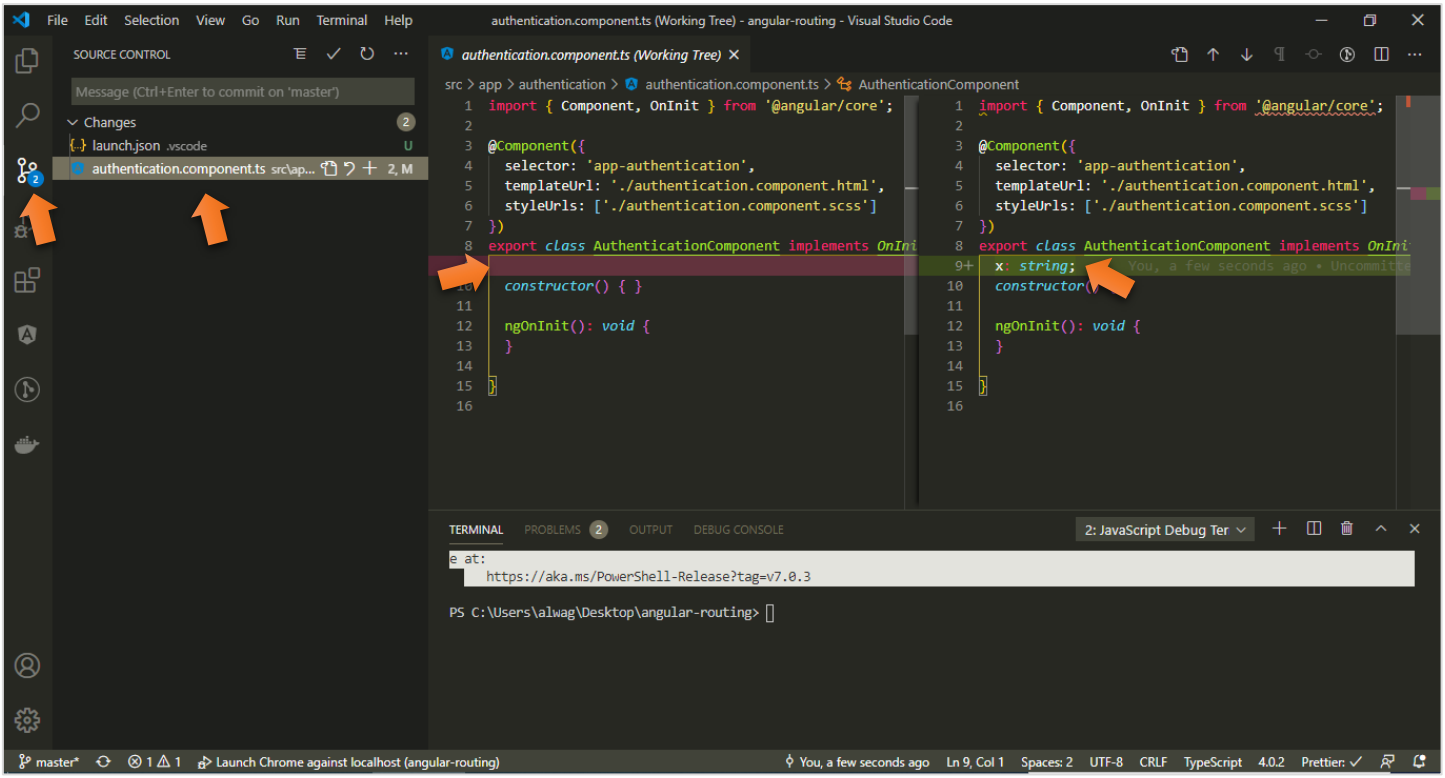
ولنكتب مثلاً public في مربع الاستبدال، كالتالي:



بعدما قمنا بكتابة القيمة public نضغط على الزر الذي تم التأشير عليه بالسهم في الشكل بالأعلى وسوف يتم استبدال كافة القيم private بالقيمة public.

3.2.3.2.قائمة Source Control:

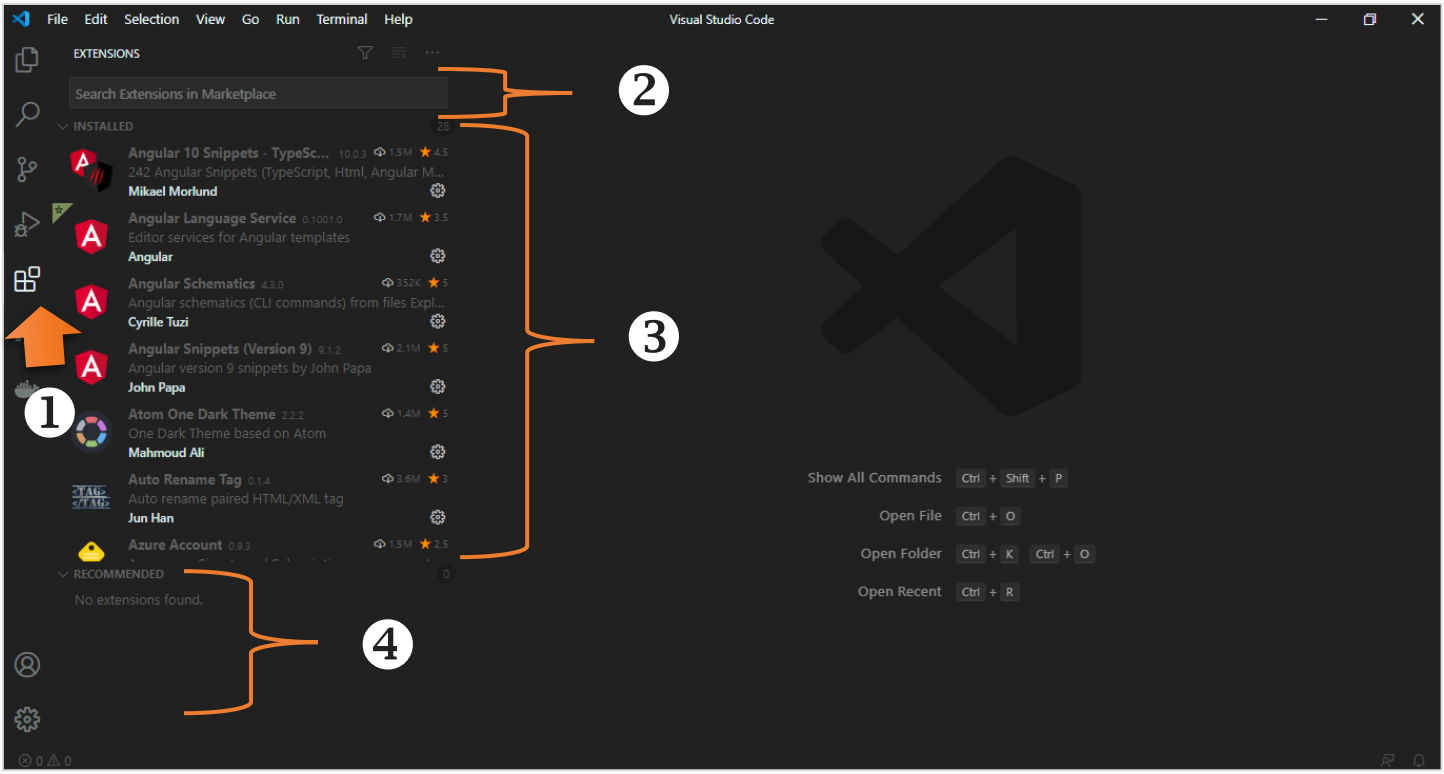
وهذه القائمة تظهر لنا أي تغيير يحدث بأي ملف مع اظهار الملف قبل التعديل وبعد التعديل، كالتالي:



حيث قمت بإجراء تعديل بسيط على احدى الملفات وذلك بإضافة متغير اسميته x وهو من النوع string، وعند اختيار هذه القائمة ظهر لنا هذا التعديل في القائمة وعند اختيار هذا التعديل ظهر لنا ملفين بجانب بعضهما البعض الأول من اليسار الملف قبل التعديل والثاني من اليمين الملف بعد التعديل.

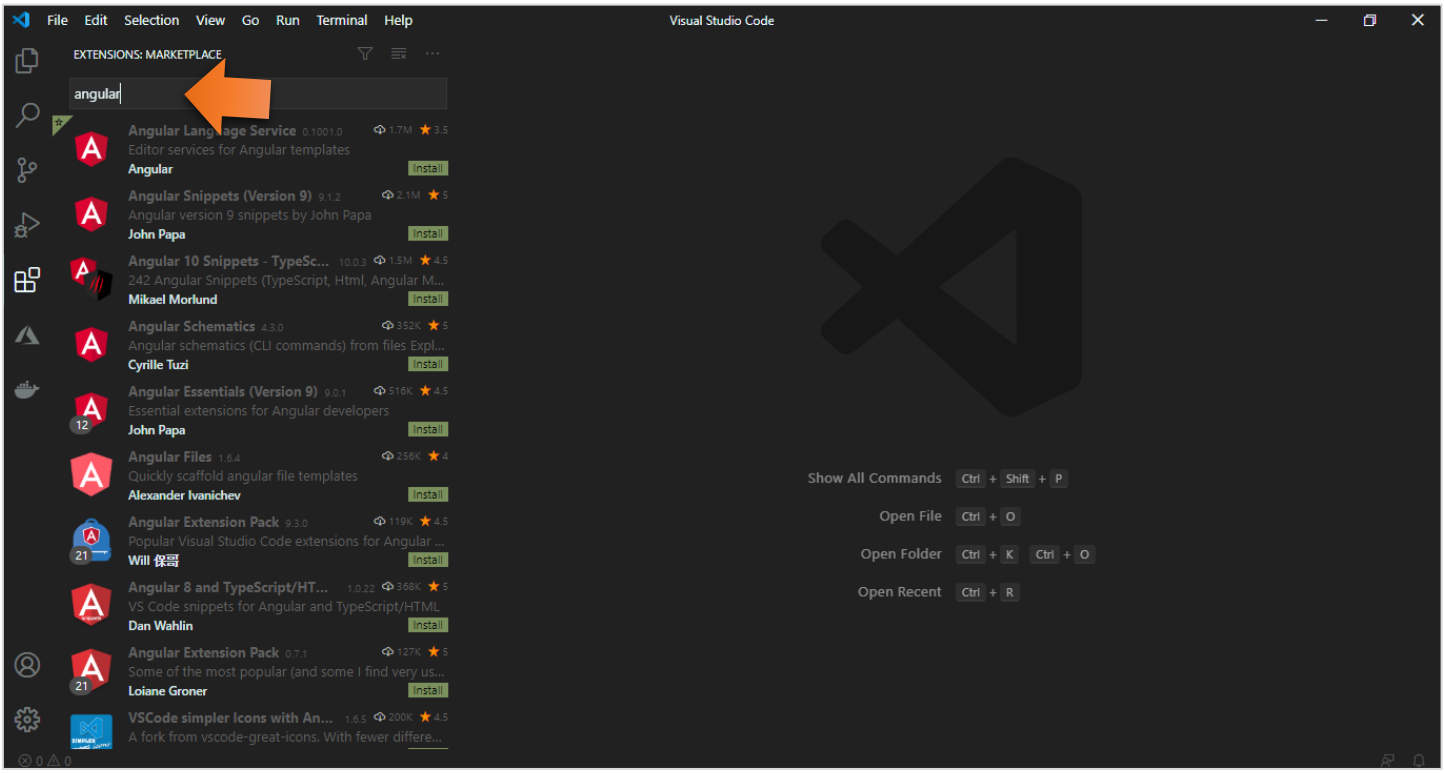
4.2.3.2.قائمة Extensions:

وهنا نشاهد قوة هذا المحرر من خلال إضافات قام ببنائها مجتمع المطورين بحيث تسهل وتبسط كتابة الشفرات البرمجية او تقوم بإضفاء اشكال جمالية على هذه الشفرات او حتى ترتيب هذه الاسطر البرمجية وإعادة تنسيقها، وفي الحقيقة هذه الإضافات بالآلاف ان لم تكن أكثر من هذا العدد بكثير وليس هنا المقام لشرحها ولكن سوف استعرض الإضافات الخاصة بمشاريع Angular بالإضافة إلى بعض الإضافات الأخرى التي تكون عامة لأي مشاريع برمجية أخرى.

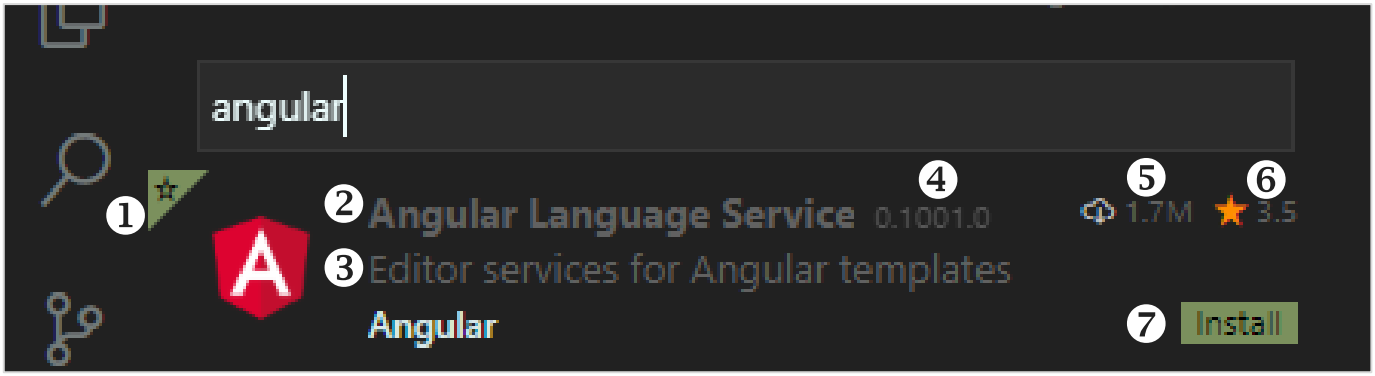


في الجزر المرقم بالرقم 1 نختار زر القائمة Extensions وفي الرقم 2 مربع بحث نقوم بكتابة اسم الإضافة للبحث عنها ويلزم ان تكون متصل بشبكة الانترنت، اما القسم ذو الرقم 3 فيظهر فيه جميع الإضافات التي قمت بتحميلها في الجهاز لديك، وأخيراً القسم ذو الرقم 4 يوجد فيه بعض الإضافات التي يقترحها لنا VS Code.

وسوف أقوم بإلغاء تثبيت جميع الإضافات الأربع الخاصة بي Angular الموجودة بالشكل السابق لكي نقوم بتثبيتها معاً، والآن لنكتب في محرك البحث angular، لكي تظهر لنا جميع Extensions الخاصة بإطار عمل Angular، كالتالي:



نلاحظ ظهرت لنا قائمة كبيرة تحتوي على عدد من الإضافات، وسوف نقوم بتكبير اول إضافة لمعرفة التفاصيل العامة والمشتركة بجميع هذه الإضافات، كالتالي:



(١) هذه العلامة تخبرك بإن VS Code يقترح عليك هذه الإضافة بناءً على استخدامك والمشاريع السابقة التي قمت بتعامل معها سابقاً.

(٢) اسم الإضافة.

(٣) نبذة عن الإضافة.

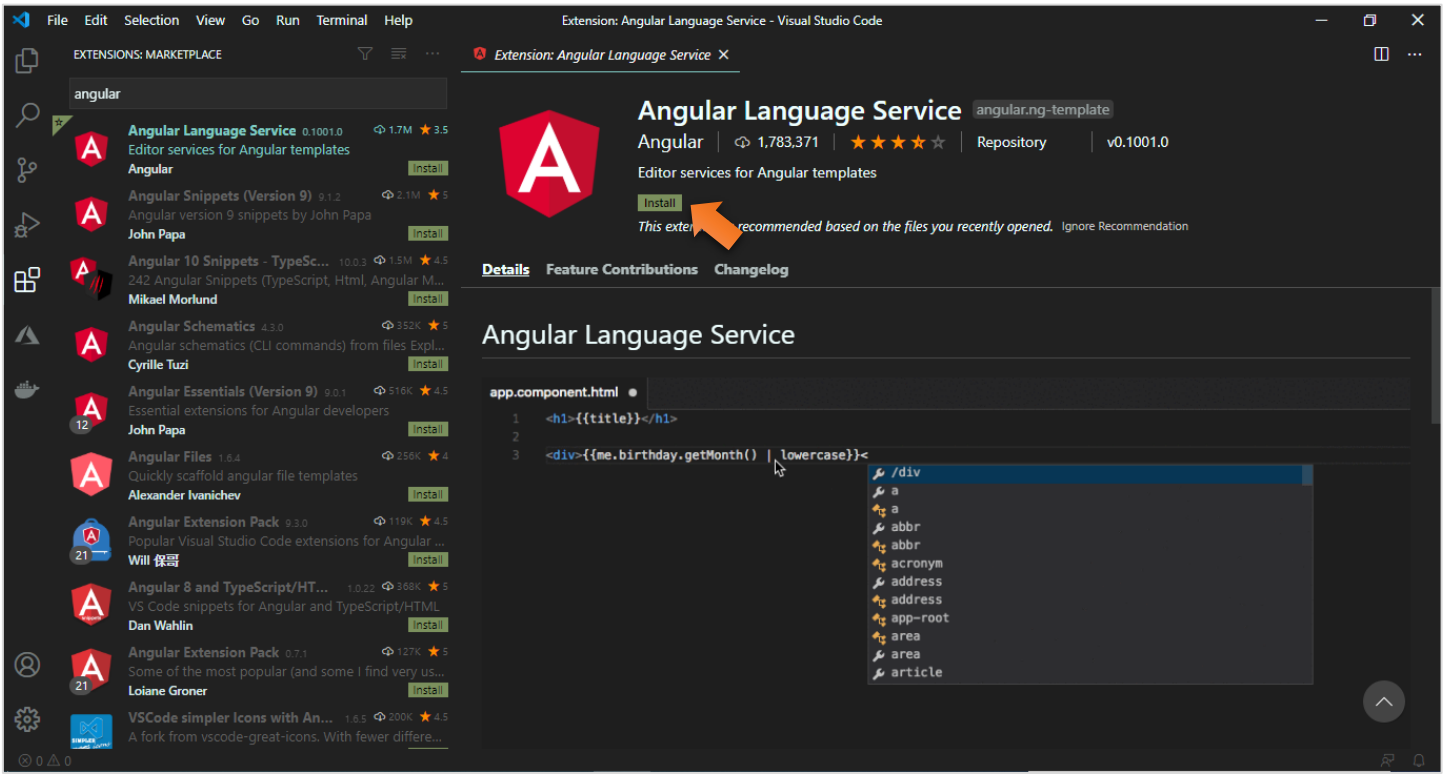
(٤) الاصدارة الخاصة بهذه الإضافة.

(٥) عدد من قام بتثبيت هذه الإضافة.

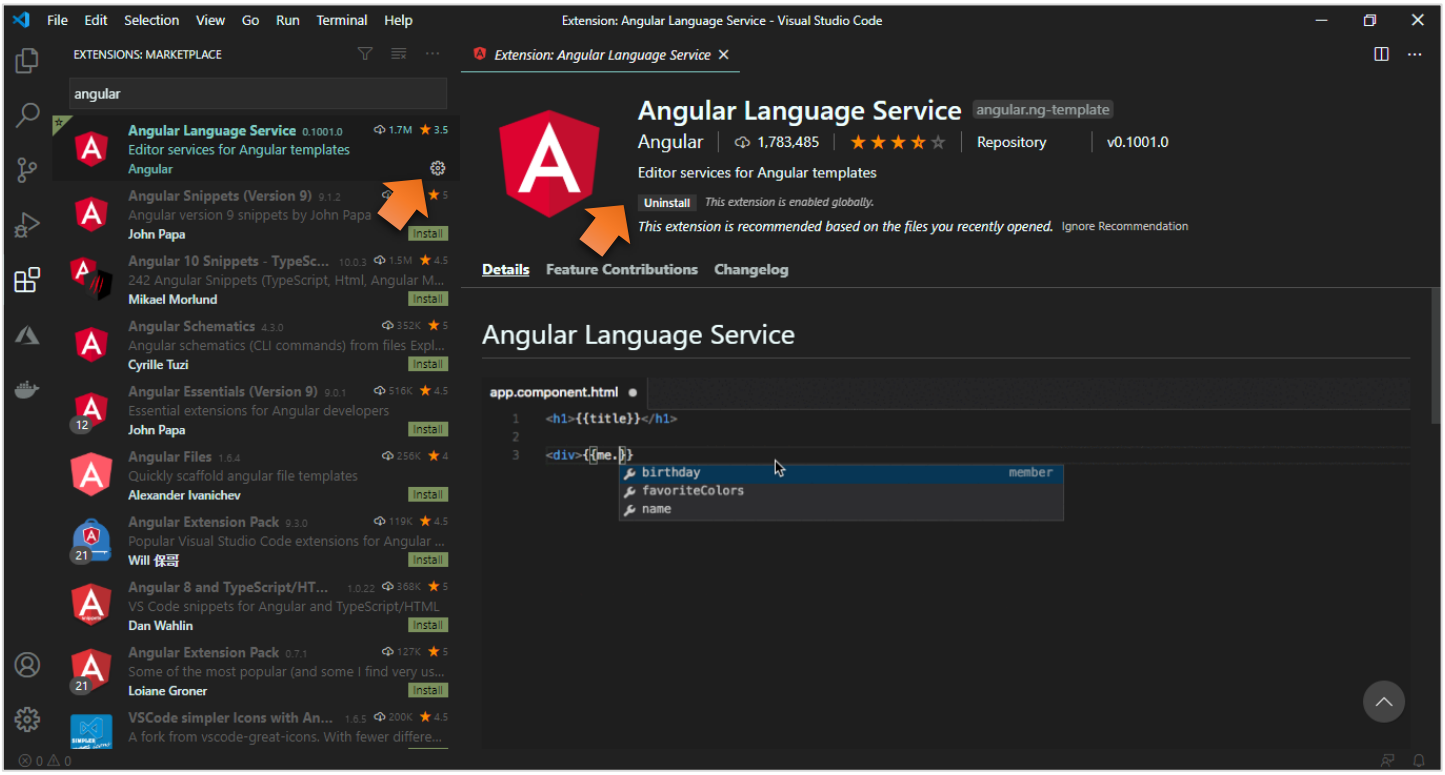
(٦) عدد مرات الاعجاب بهذه الإضافة.

(٧) تثبيت الإضافة.

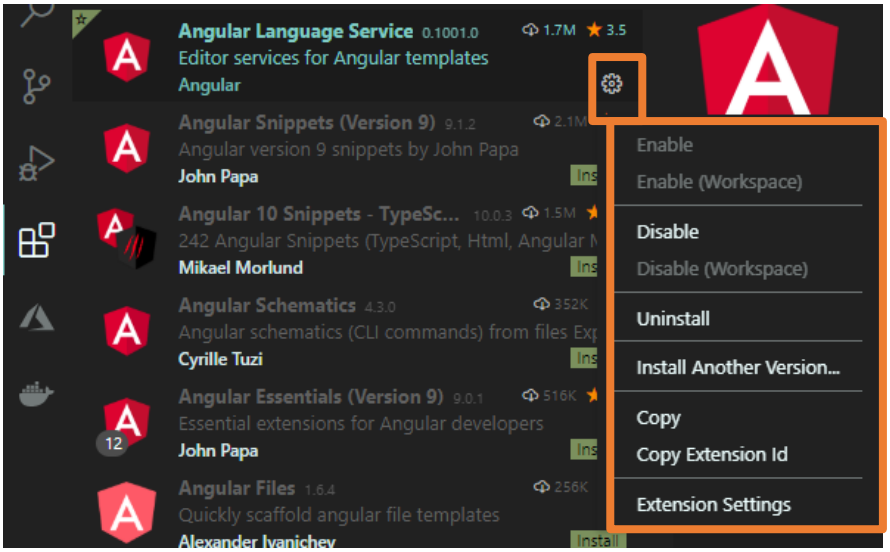
وفي حال أردنا ان نستعرض مزيد من التفاصيل عن هذه الإضافة نستطيع الضغط على اسم الإضافة لكي تظهر لنا تفاصيل أكثر مثل طريقة استخدام هذه الإضافة وموقع او Documentation الخاصة بهذه الإضافة، كالتالي:



نلاحظ ظهر لنا على يمين الشاشة بيانات وتفاصيل هذه الإضافة. وايضاً هذه الشاشة تحتوي على زر install بحيث نستطيع الضغط عليها لتثبيت هذه الإضافة، وبكل تأكيد تحتاج إلى الاتصال بشبكة الانترنت لكي تتعامل مع أي إضافة سواء كان بحث او استعراض او تثبيت، اما الآن لنقوم بالضغط على زر install لكي نقوم بتثبيت هذه الإضافة، كالتالي:



بعد الانتهاء من التثبيت سوف يتغير الزر من install إلى Uninstall، وهذا معناه أن هذه الإضافة تم تثبيتها بالإضافة إلى ظهور زر بمكان زر install في القائمة على يسار الشاشة، ولنقم بالضغط على هذا الزر لنرى محتوياته، كالتالي:



نلاحظ ظهرت لنا قائمة نستطيع عن طريقها إجراء بعض الإجراءات على هذه الإضافة، مع العلم ان هذه القائمة غير ثابته بحيث تختلف من إضافة إلى أخرى، وايضاً ليس كل الإضافات يتم تشغيلها مباشرة بعد تثبيتها لأنها قد تحتاج إلى بعض الاعدادات الأخرى التي تُلزمك ان تقرأ Documentation الخاص بهذه الإضافة، وايضاً يجب ان اشير أن بعض هذه الإضافات يجب ان نعمل Reload للتطبيق حيث تظهر لنا بعد تثبيت الإضافة زر مكتوب عليه reload وسوف يتم اغلاق تطبيق VS Code وإعادة فتحه مرة أخرى.

ولكن جميع الإضافات التي سوف اشير لها هنا لا تحتاج إلى أي اعدادات أخرى وإنما بمجرد تثبيتها سوف تعمل بشكل مباشر، وهي:

(١) Angular Language Service

(٢) Angular Snippets (Typescript, Html, Angular Material, Flex Layout, ngRx, Rxjs, PWA & Testing)

(٣) Angular version 9 snippets

(٤) Angular schematics

(٥) Auto Rename Tag

(٦) Beautify

(٧) Bracket Pair Colorizer

(٨) Debugger for Chrome

(٩) Debugger for Firefox

(١٠) Highlight Matching Tag

(١١) JSHint extension

(١٢) Material Icon Theme

(١٣) Material Theme

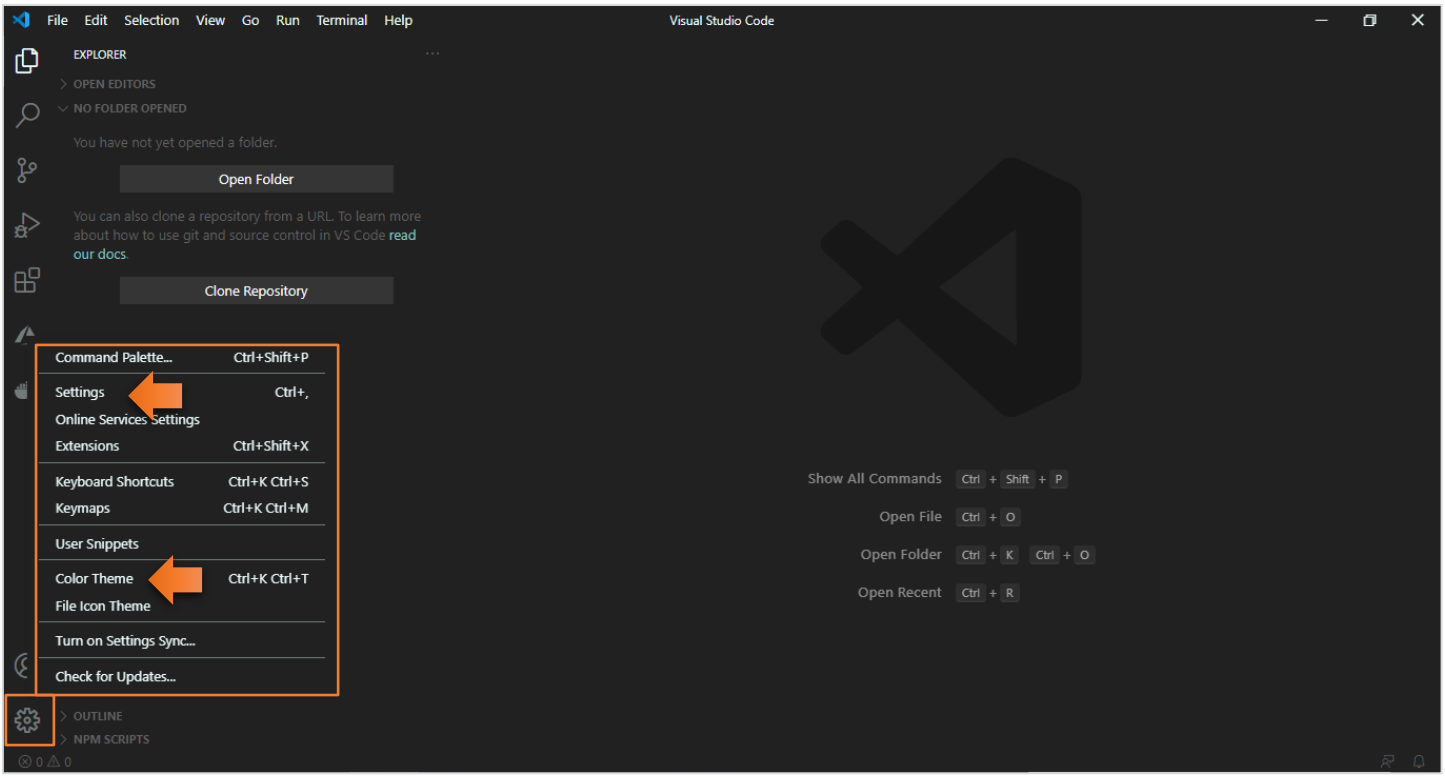
(١٤) Prettier Formatter

(١٥) TSLint

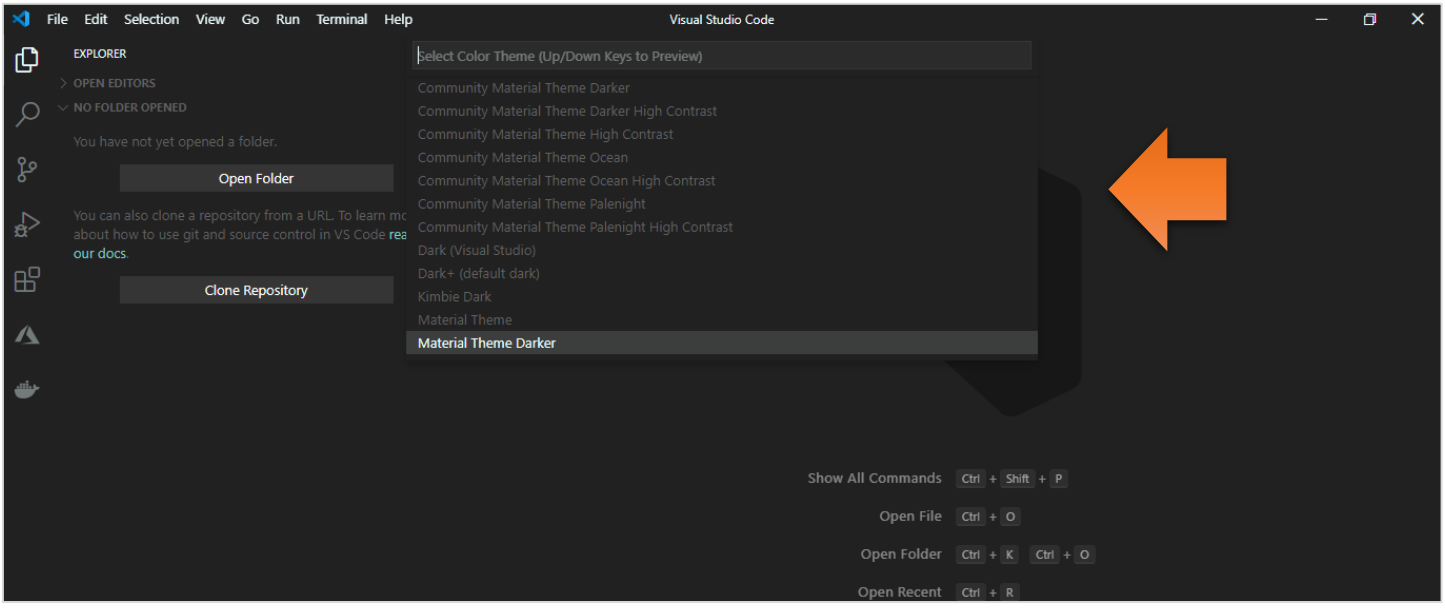
طبعاً القائمة السابقة هي قائمة مقترحة من واقع تجربة، وليس معناته انها هي الأفضل، ومن هذا المنطلق تستطيع عزيزي المتعلم اختيار أي إضافة بديلة لهذه الإضافات او تزيد او تنقص من هذه القائمة بحسب احتياجاتك.

5.2.3.2.قائمة Manage:

وهذه القائمة تحتوي بعض الأوامر كثيرة الاستخدام للوصول السريع لها، منها ما تكلمنا عنه سابقاً مثل Command Palette او Extensions ومنها ما هو جديد، كالتالي:

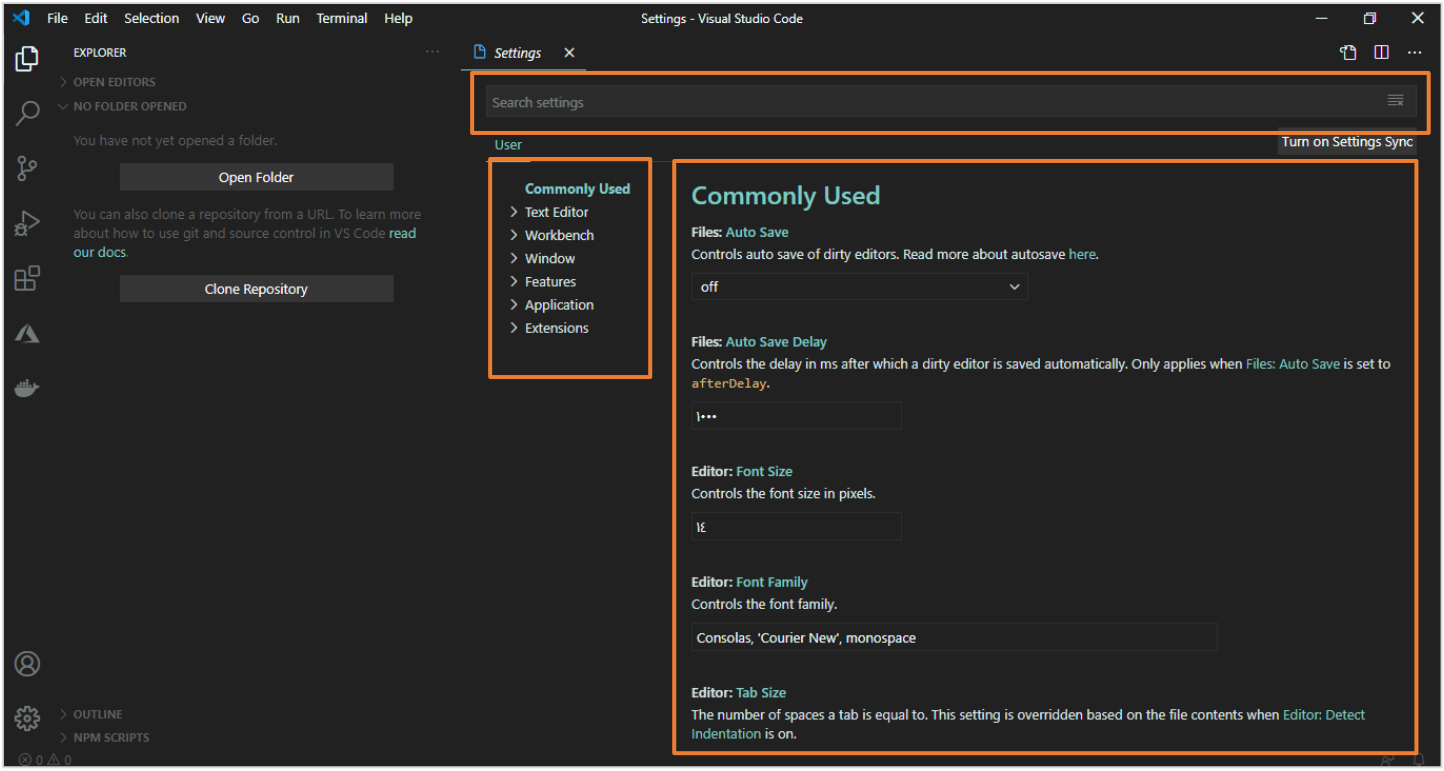


الأمر الأول وهو Color Theme ونستطيع عن طريقه تغيير السمة لتطبيق، سواء كان هذا theme من الموجود مع التطبيق او الذي قمنا بتثبيته عن طريق Extensions المختلفة، حيث عند الضغط على هذا الأمر سوف يتم فتح Command Palette وعرض جميع السمات الموجودة، كالتالي:



وعن طريق الأسهم ننتقل بين هذه السمات وعند الرغبة باختيار أحد هذه السمات نضغط على زر Enter.

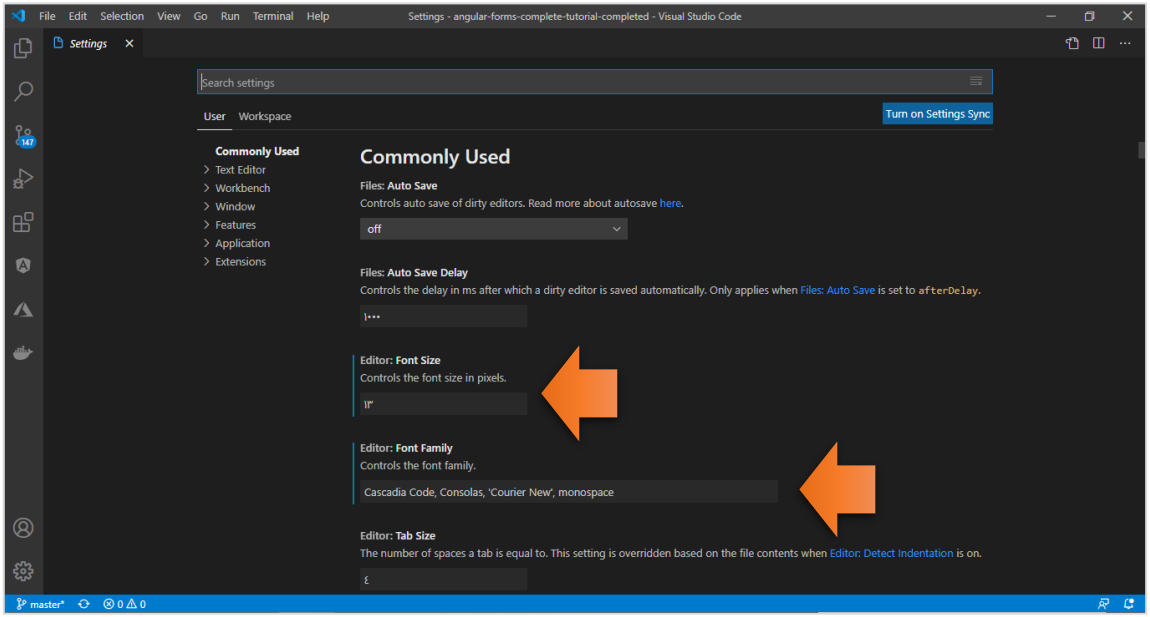
اما الأمر الثاني وهو Settings ومن اسمها نستطيع عن طريق تغيير جميع اعدادات التطبيق او الإضافات وغيرها الكثير، كالتالي:



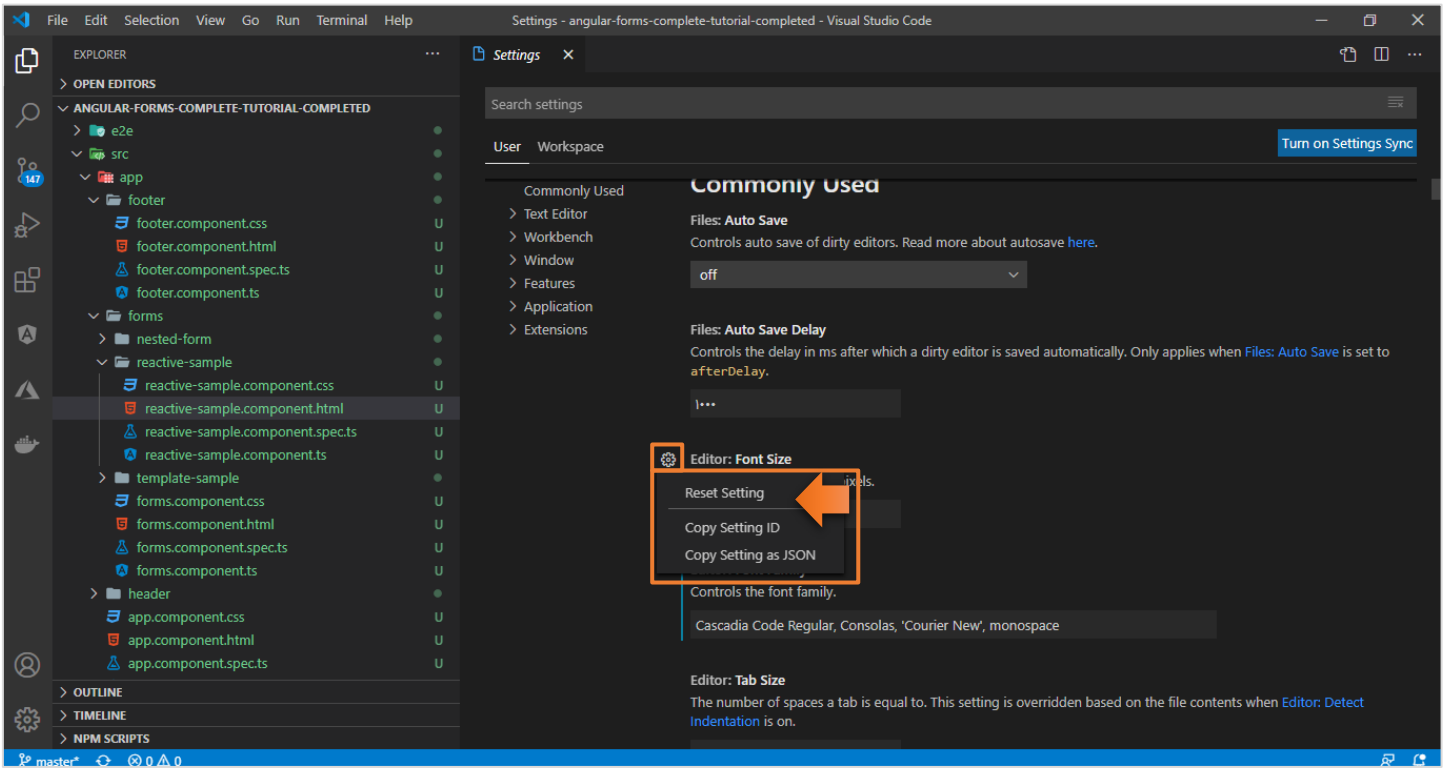
في الأعلى مربع بحث بحيث نكتب فيه أي خاصية او ميزة في التطبيق نُريد التعديل على قيمها، اما في اليسار فتم تقسيم جميع الاعدادات على شكل قائمة شجرية وكل مجموعة من الإعدادات المتشابهة تم تجميعها معاً بعكس اول قائمة Commonly Used حيث تعرض الاعدادات الأكثر استخداماً.

وبطبيعة الحال تستطيع عزيزي المتعلم أخذ جولة في جميع هذه الإعدادات وتجربتها ومعرفة كيفية التعامل معها، مع العلم انني سوف أوضح بعض هذه الإعدادات، ومنها:

الخط: ونجدها في Commonly Used حيث نستطيع التحكم بنوع وحجم الخط، كالتالي:

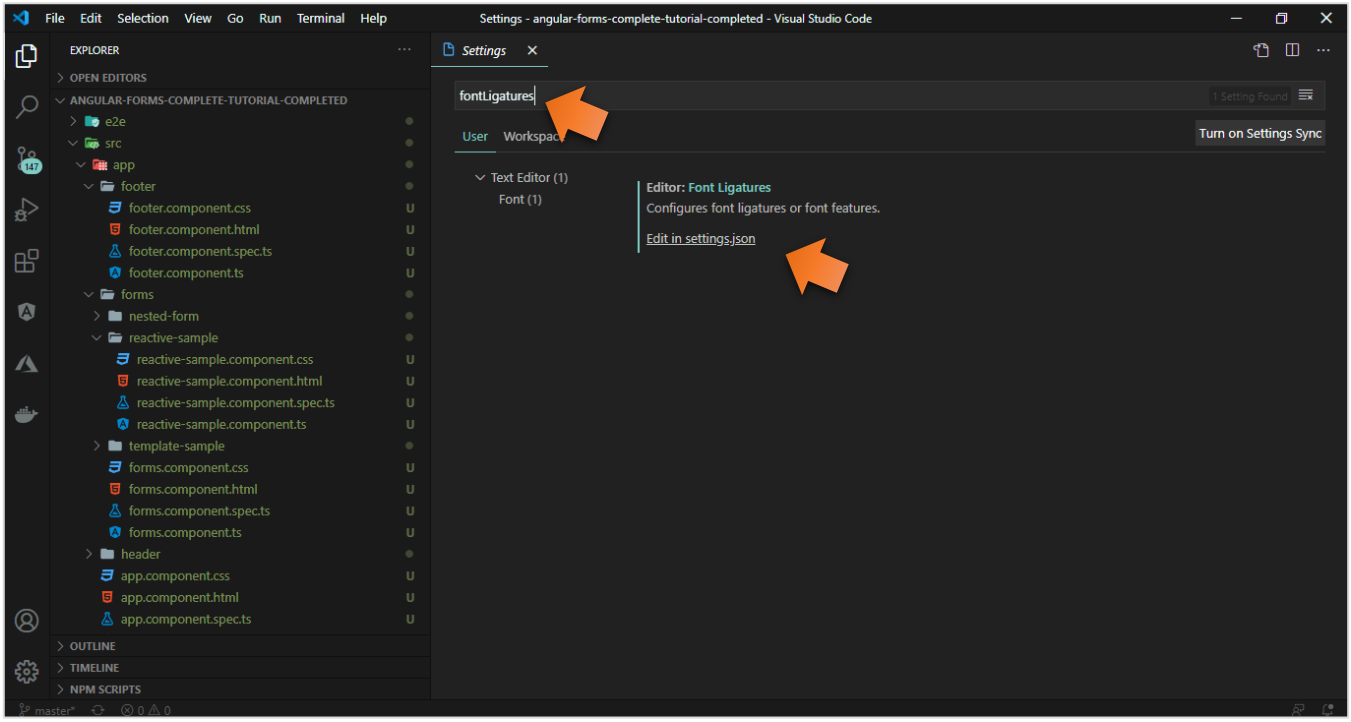


بحيث نستطيع تحديد الخطوط وحجمها، وفي حال أردنا ان نُرجع الإعدادات إلى وضعها الافتراضي فنقوم بوضع المؤشر بجانب اسم الإعداد وسوف تظهر لنا ايقونة نقوم بالضغط عليها وتظهر لنا قائمة نختار منها Reset Settings، كالتالي:

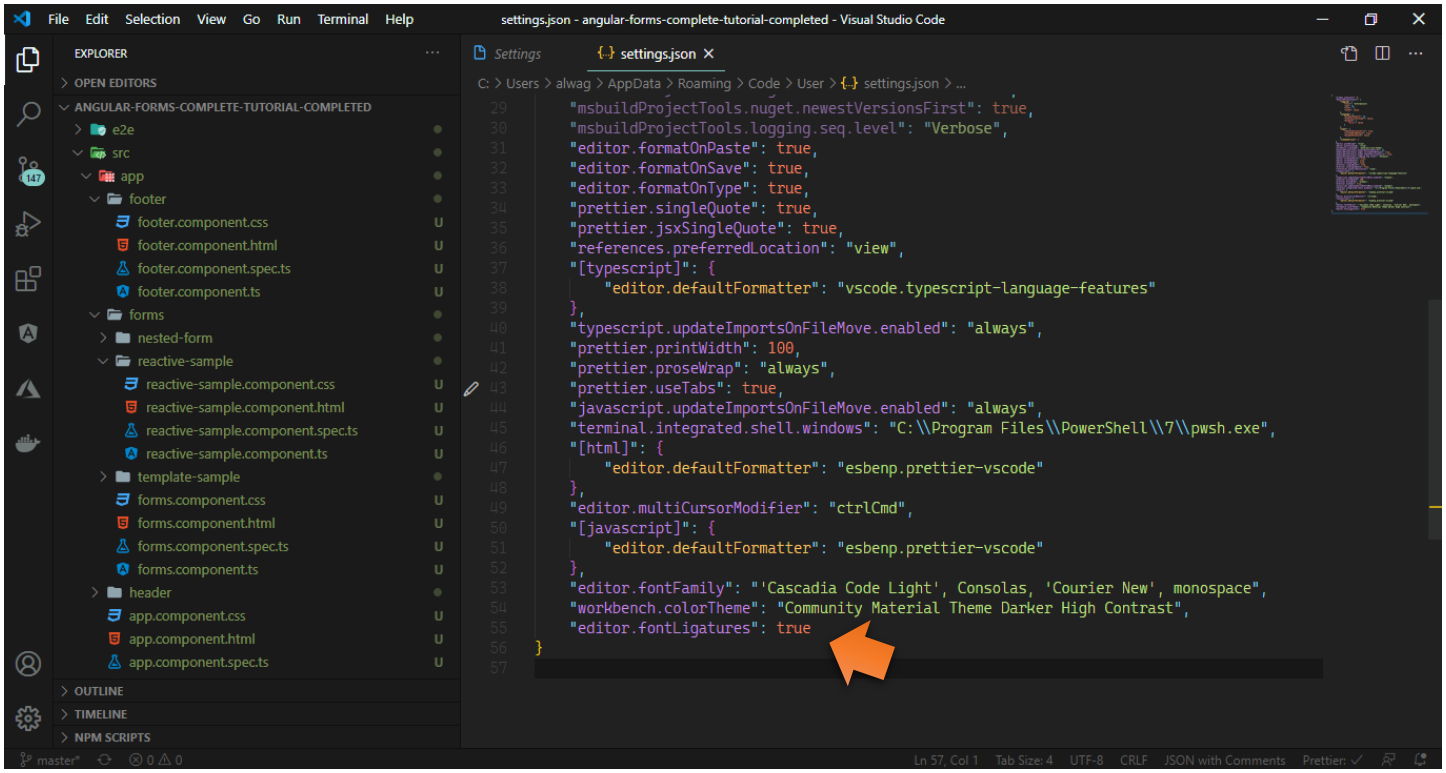


ومن الإعدادات ايضاً التي نستطيع التعديل عليها هي fontLigatures وتقوم هذه الخاصية بجعل المعاملات الرياضية اثناء كتابة الكود بشكلها الرياضي وليس بطريقة جهاز الحاسب حيث في حال كتبنا لا يساوي في لغة الحاسب (=!) فإنه سوف يقوم بتحويلها إلى ≠ والأكبر من او يساوي في لغة الحاسب (>=) فإنه سوف يقوم بتحويلها لشكل الرياضي ≥ وهكذا بقيمة المعاملات الرياضية.

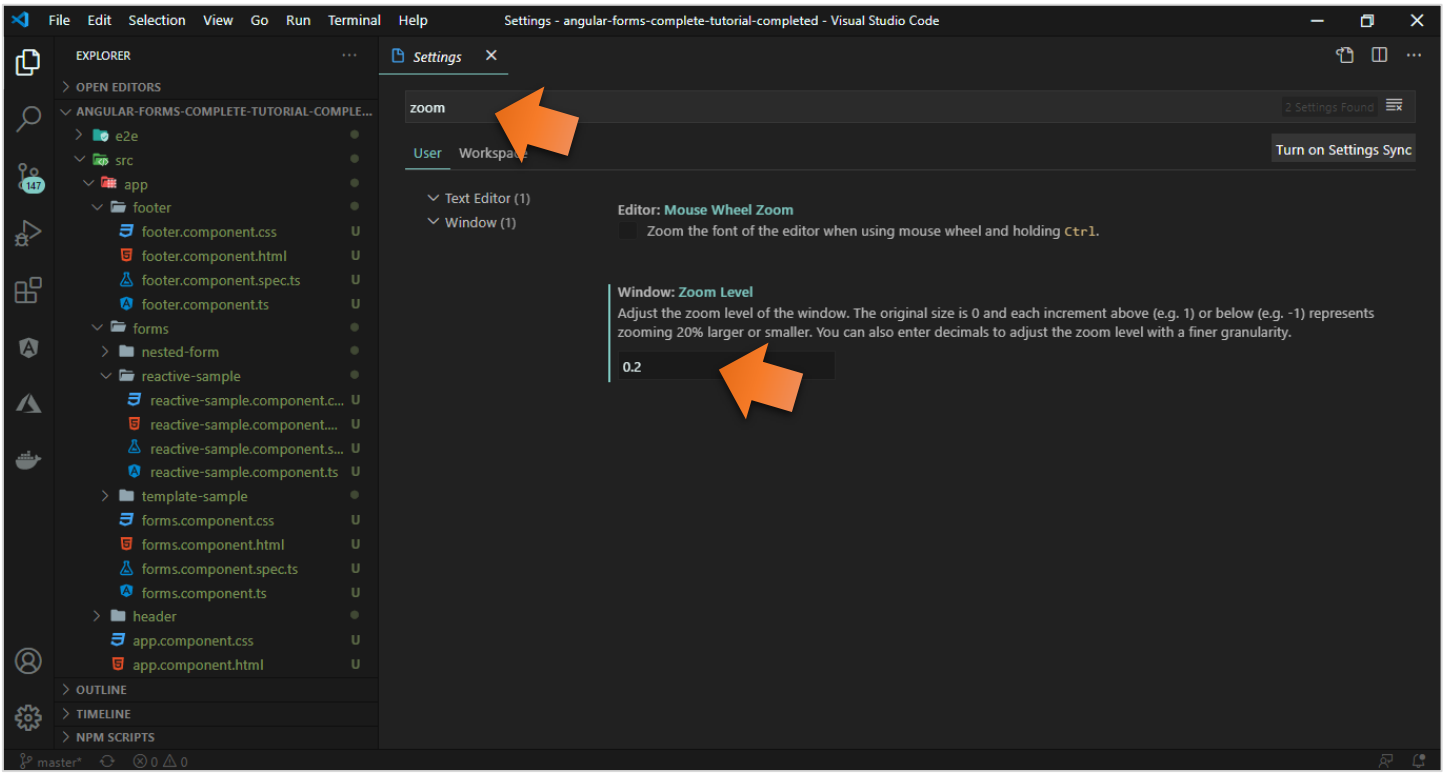
ونستطيع التعديل على هذه الخاصية بكتابة اسمها في مربع البحث ومن ثم نضغط على Edit in settings.json، كالتالي:



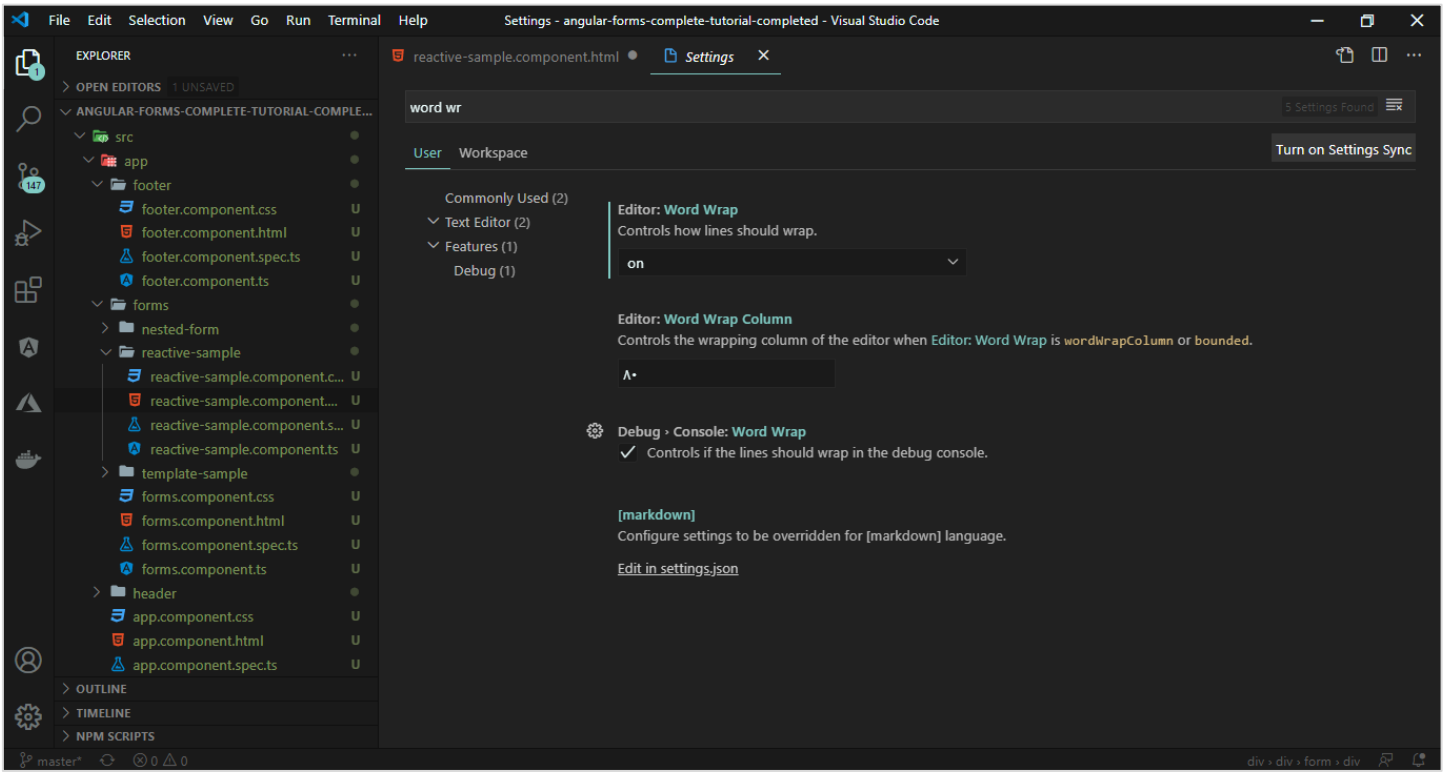
وسوف يُفتح ملف Json ونبحث عن هذه الخاصية ونقوم بتغيير قيمتها إلى true، كالتالي:



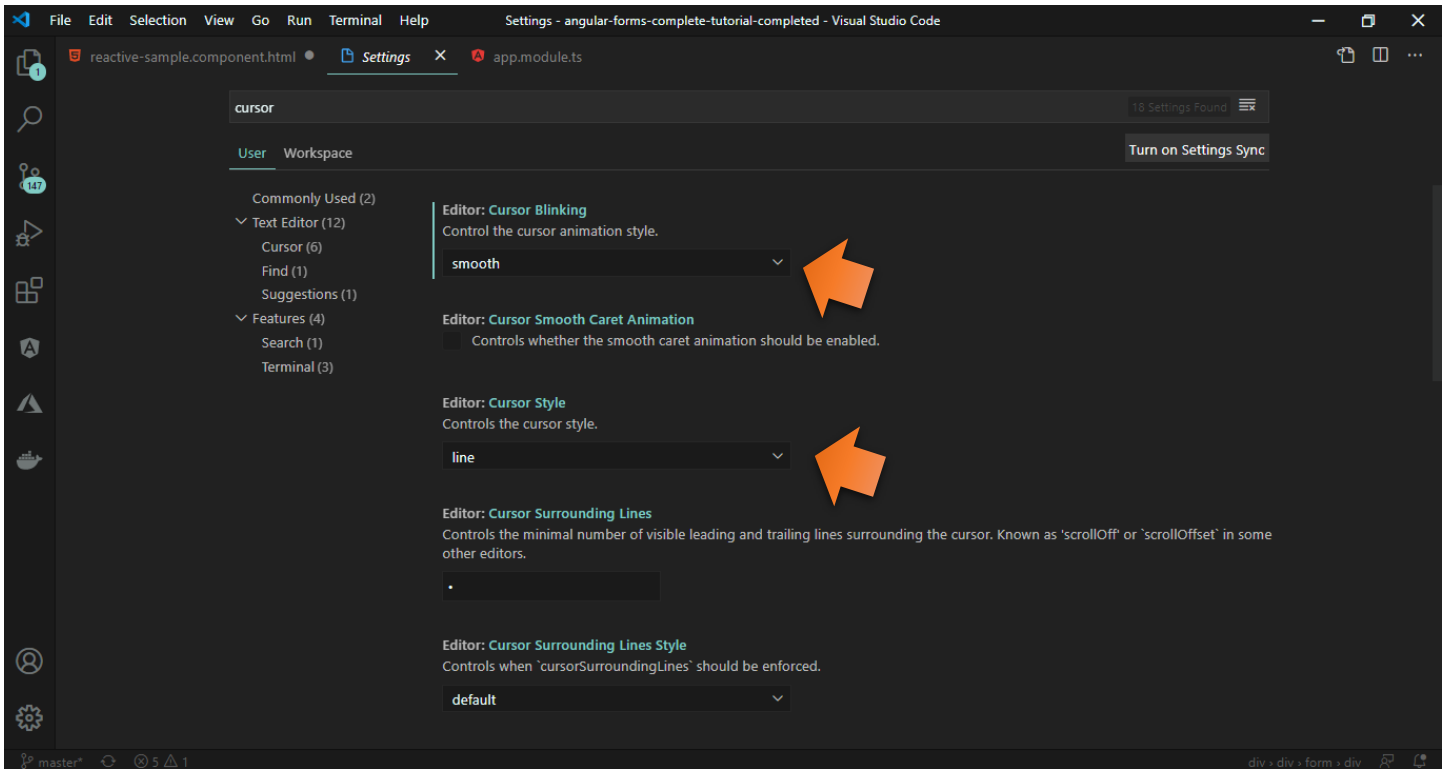
ومن الخصائص ايضاً zoom ونستطيع البحث عنها في مربع البحث ومن ثم نضع القيمة المناسبة لك، وهنا وضعت القيمة 0.2، كالتالي:



ومن الخصائص ايضاً Word Wrap حيث نستطيع تفعيلها وتحديد عدد الخانات في كل سطر، كالتالي:



ومن الخصائص ايضاً cursor والتي تتيح لنا التحكم بشكل وحركة مؤشر الكتابة، كالتالي:



هذه بعض الإعدادات البسيطة وكما أشرت سابقاً تستطيع التجول في هذه الإعدادات ومحاولة معرفة الفائدة منها وكيفية التعامل معها.

Shortcuts, Tricks and Tips.3.3.2 :

في هذا الجزء سوف نتكلم عن بعض الاختصارات المشهورة وكثيرة الاستخدام من قبل المطورين بالإضافة إلى بعض الحيل والنصائح التي قد تُساعد في تسريع كتابة الكود وإنجاز المهام.

Shortcuts.1.3.3.2 :

حقيقة الاختصارات كثيرة جد ونستطيع الاطلاع عليها جميعاً، من خلال القائمة Manage ومن ثم نختار الأمر Keyboard Shortcuts من القائمة المنسدلة، وسوف تظهر لك عزيزي المتعلم كم هائل من الاختصارات مع إمكانية البحث عن أحد الأوامر لمعرفة الاختصار الخاص به، وحقيقة ليس هنا المقام لشرح جميع هذه الاختصارات، ولكن سوف يتم التركيز على أهمها والمستخدمة بكثرة من قبل المطورين، وسوف اسردها على شكل جدول مع وضع الاختصار ووظيفته (جميع الاختصارات خاصة بنظام التشغيل Windows)، كالتالي:

	الاختصار	الشرح	ملاحظات
1	Ctrl + B	اظهار/إخفاء شريط Activity	
2	Ctrl + F	إظهار شاشة البحث في داخل ملف معين	يمكن اختصاراً للوقت وضع مؤشر الفأرة على كلمة معينة المراد البحث عنها او استبدالها – بدون الحاجة لتحديدها بالكامل – ومن ثم نضغط على هذه الاختصارات
3	Ctrl + H	إظهار شاشة البحث والاستبدال داخل ملف معين	
4	Ctrl + Shift + F	إظهار قائمة البحث Search على مستوى كامل التطبيق	
5	Ctrl + Shift + E	إظهار قائمة Explorer	
6	Ctrl + Shift + X	إظهار قائمة Extensions	
7	Ctrl + Shift + P	إظهار Command Palette جزء الأوامر	
8	Ctrl + P	إظهار Command Palette جزء استعراض ملفات المشروع	
9	(Ctrl + K) Z	وضع ملء الشاشة لجزء تحرير الكود فقط	نقوم بالضغط على الاختصار الذي بين الأقواس ومن ثم نضغط على حرف Z بشكل منفرد، كما نستطيع الضغط على الاختصار Ctrl + Shift + P ومن ثم نكتب الأمر zen mode ونختاره لوضع ملء الشاشة ونكرر نفس الأمر للخروج من هذا الوضع.
10	F11	وضع ملء الشاشة	

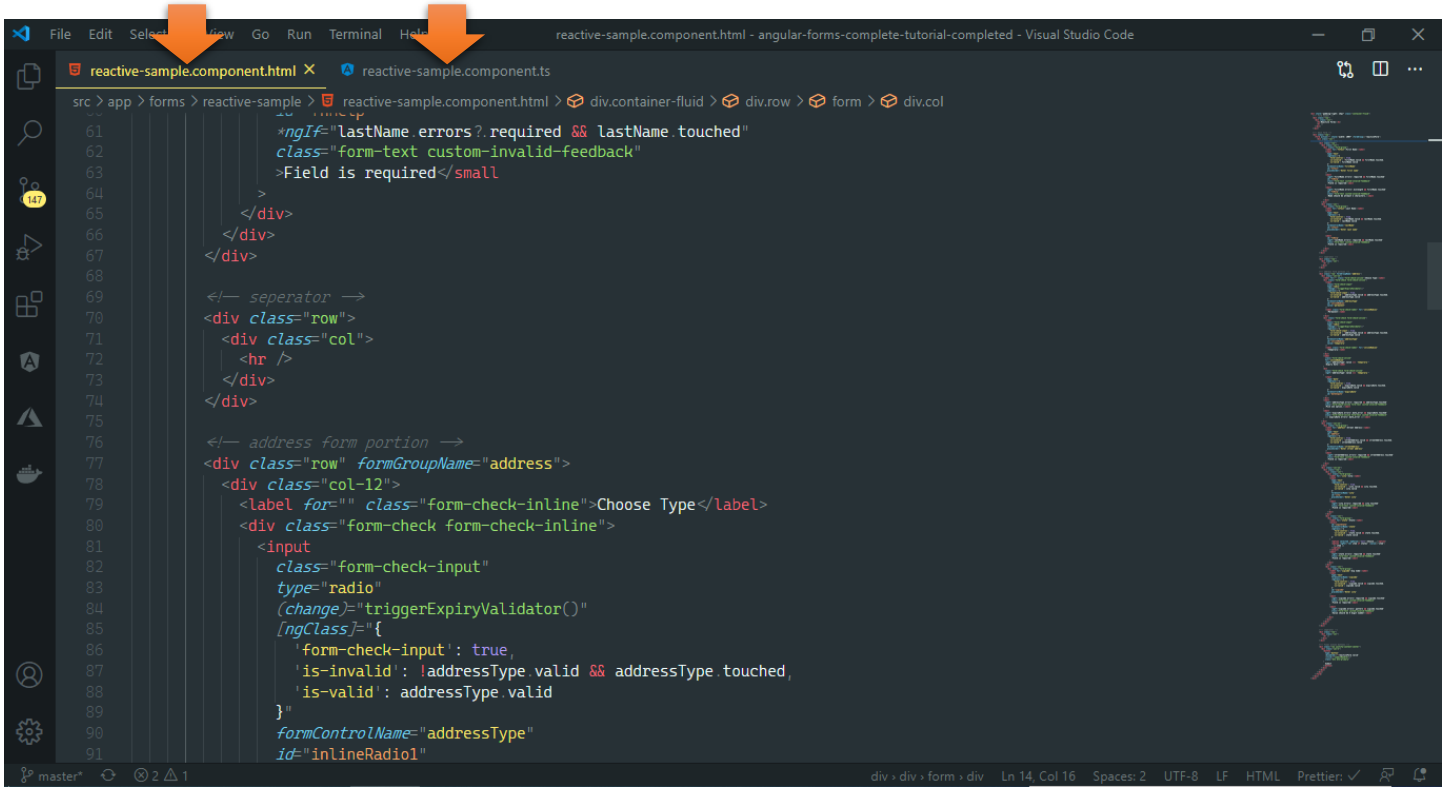
11	Ctrl + Shift + P ومن ثم الأمر minimap	إظهار/إخفاء صورة مصغرة لكل الملف على يمين الشاشة	يستخدم في الملف الكبير ذو الاسطر البرمجية الكثيرة بحيث يسهل الانتقال إلى أي سطر محدد
12	Ctrl + Tab	التنقل (التبديل) بين آخر ملفين تم فتحهما	مع الاشتراط ان لا يكون تم اغلاقهما
13	Ctrl + Shift + Tab	الانتقال بين الملفات المفتوحة بشكل متسلسل	
14	Ctrl + W	إغلاق الملف الحالي	
15	Shift + →	الانتقال بين الكلمات في نفس الملف من اليسار لليمين	
16	Shift + ←	الانتقال بين الكلمات في نفس الملف من اليمين لليسار	
17	Ctrl + →	الانتقال إلى نهاية السطر من اليسار لليمين	
18	Ctrl + ←	الانتقال إلى نهاية السطر من اليمين لليسار	
19	(Ctrl + K) S	حفظ جميع التعديلات على جميع ملفات المشروع	
20	Ctrl + S	حفظ التعديلات على الملف الحالي	
21	Alt + ↑	نقل السطر البرمجي للأعلى	يتم وضع مؤشر الكتابة على السطر المُستهدف بدون الحاجة لتحديده بالكامل ومن ثم يتم الضغط على أحد الاختصارين
22	Alt + ↓	نقل السطر البرمجي للأسفل	
23	Shift + Alt + ↑	نسخ السطر البرمجي للأعلى	
24	Shift + Alt + ↓	نسخ السطر البرمجي للأسفل	
25	Ctrl + D	تحديد الكلمة	وضع مؤشر الفأرة على الكلمة المحددة ومن ثم الضغط على الاختصار
26	Ctrl + C	نسخ السطر البرمجي	لا تحتاج إلى تحديد السطر البرمجي لكي تنسخه فقط ضع مؤشر الكتابة على السطر المستهدف ومن ثم اضغط على الاختصار

27	! + Tab	كتابة Markup الأساسي لأي ملف HTML	لا تعمل إلا مع ملفات HTML فقط
28	Ctrl + `	إخفاء/اظهار terminal	
29	Ctrl + Shift + `	إضافة terminal جديد	
30	Ctrl + Shift + 5	تقسيم شاشة terminal	

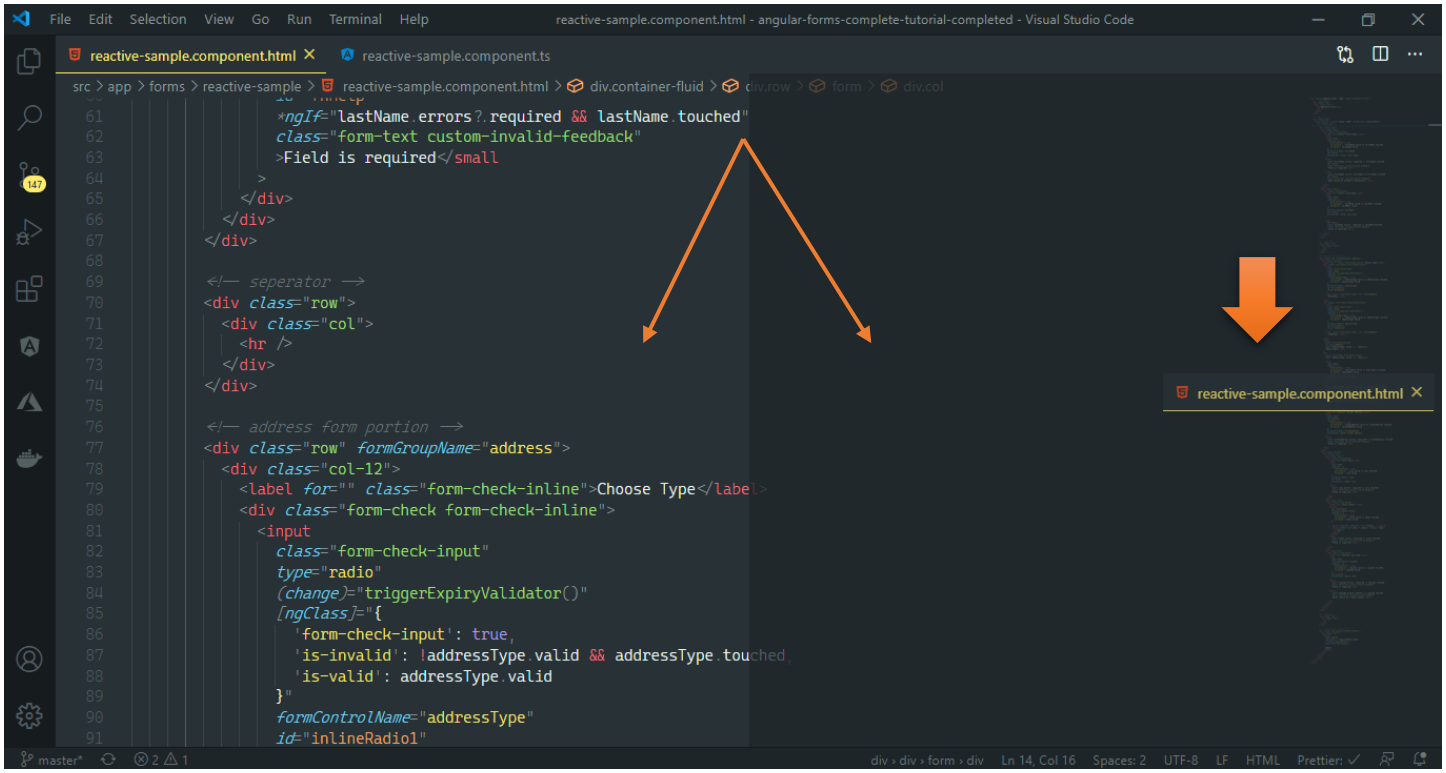
Tips and Tracks.2.3.3.2 :

اما من ناحية الحيل التي ممكن ان نستخدمها فهي تقسيم الشاشة، فنستطيع تقسيم الشاشة إلى جزئيين، ونستفيد من هذا الامر في حال كان لدينا ملفين ونريد ان نتعامل معهم بنفس الوقت.

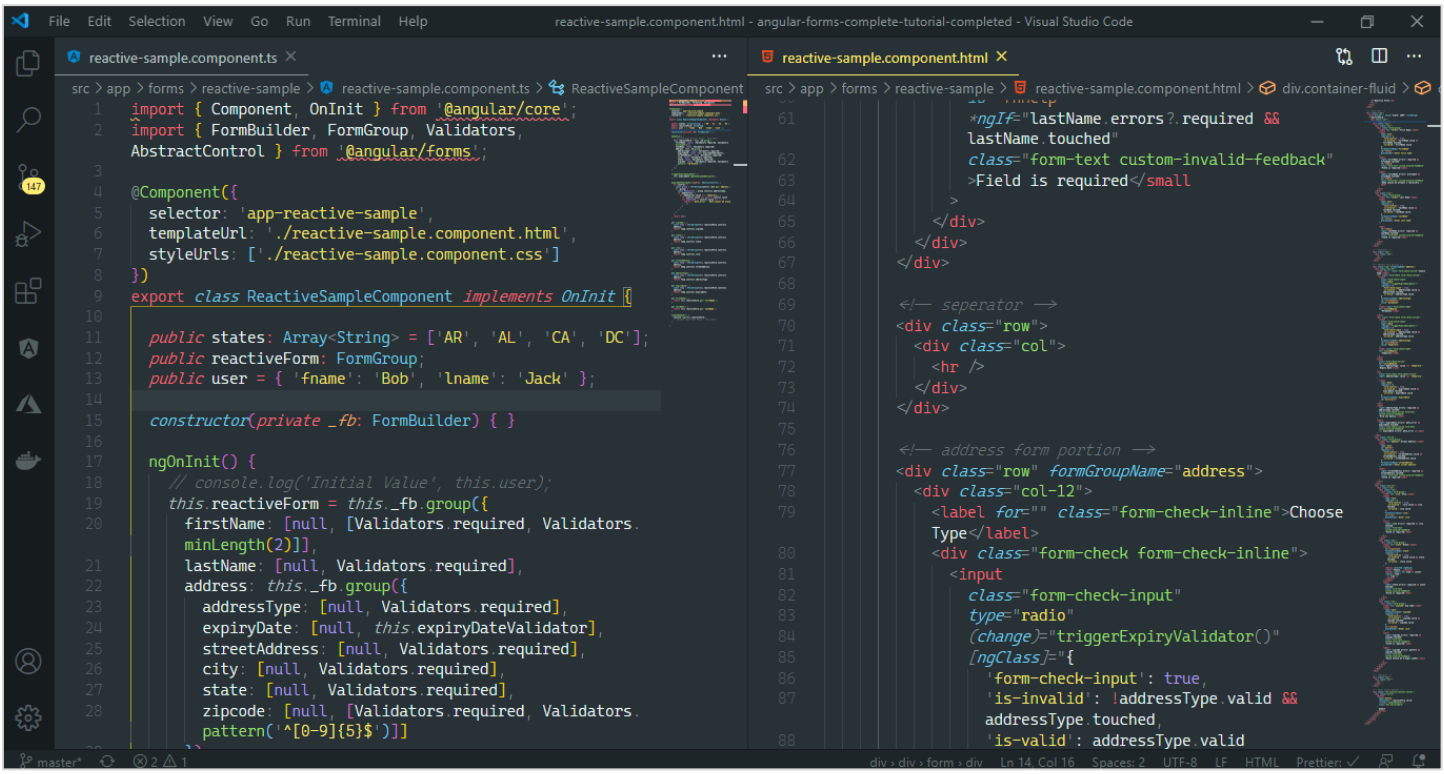
ونستطيع عمل هذا الأمر عن طريق فتح ملفين، كما في الشكل التالي:



وبعدها نقوم بالضغط بزر الفأرة الأيسر على أحد أسماء الملفات التي تم التأشير بها بالسهم في الشكل بالأعلى ومن ثم نسحب ونحرك إلى ان نصل إلى الجزء الأيمن من الشاشة، كالتالي:



عندما تصبح الشاشة مقسمة بلونين مختلفين كما هو موضح بالشكل بالأعلى عندها نقوم بالإفلات بمعنى رفع الاصبع من زر الفأرة الأيسر، بحيث يصبح الشكل كالتالي:



الفصل الثالث

Web Browser Developer Tools

(Google Chrome)

1.3.مقدمة:

في هذا الجزء سوف نشرح تقنية تعتبر من التقنيات التي يجب على أي مطور ويب وخصوصاً مطوري الواجهات الأمامية الإلمام بها ومعرفة كيفية التعامل معها، وهي ايضاً كما سبقها من فصول لا تختص بإطار عمل Angular وإنما تهم جميع مطوري الويب.

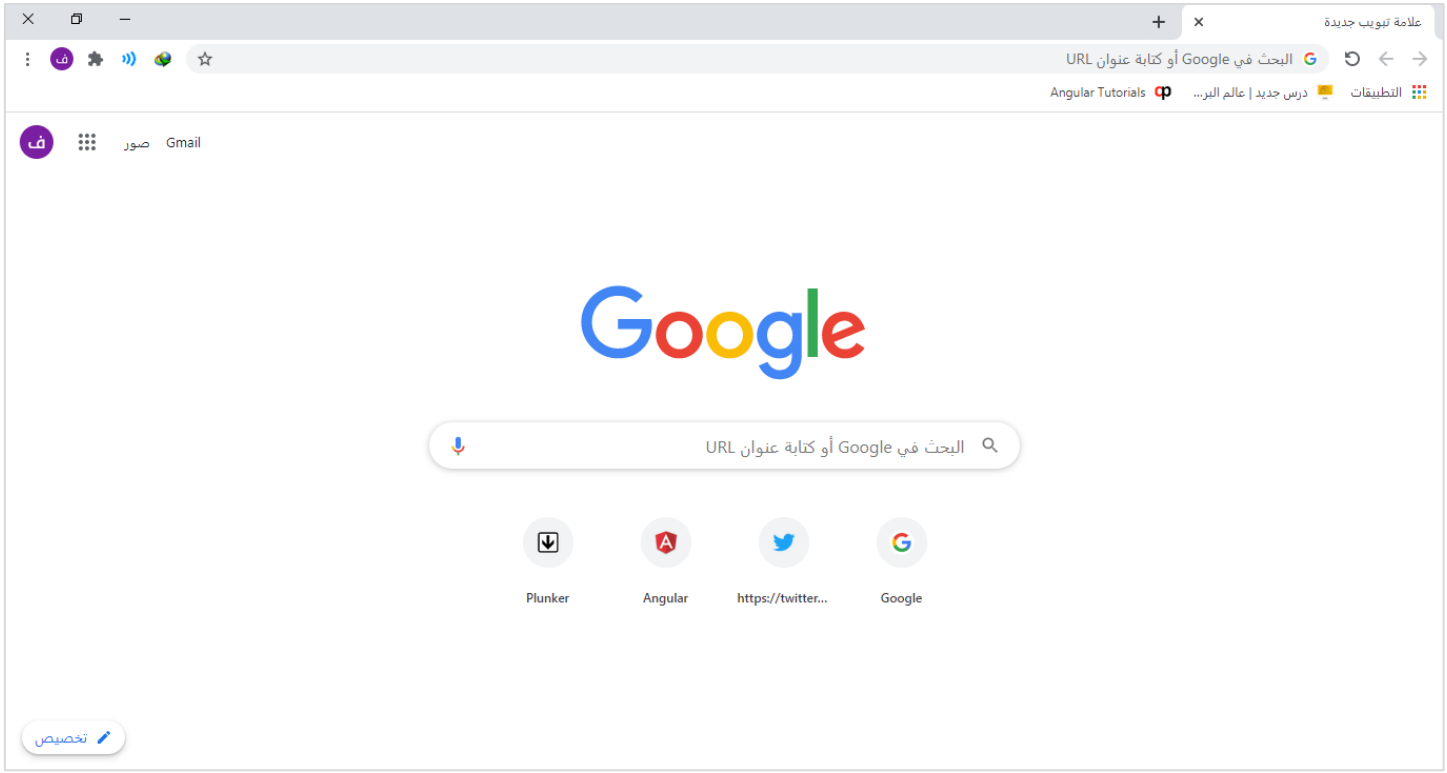
صحيح ان هذه التقنيات ليس بضخامة وكثرة التفاصيل والتعقيدات الموجودة في الفصلين السابقين، ولكن هي ايضاً لها تفاصيلها التي يصعب حصرها وشرحها في فصل واحد من كتاب، ولذلك سوف أقوم بالتركيز على الاجزاء الأهم والأكثر استخدام بين المطورين.

اما من ناحية ما هي هذه الادوات فهي عبارة عن مجموعة من الأدوات التي توفرها لنا المتصفحات مثل Chrome او Firefox او Edge (الإصدارات الحديثة من Edge)، تتيح لنا كمطورين التلاعب بي DOM والتنسيقات الخاصة بكل عنصر بالإضافة لمعرفة حجم الملفات التي تم تحميلها في بداية تشغيل التطبيق او اظهار رسالة بالإخطاء ان وجدت في console او حتى عمل Debugger، وغيرها الكثير، التي سوف نتكلم عنها ونعرف كيف نستفيد منها كمطورين للويب بشكل عام ومطوري Angular بشكل خاص.

وسوف نستخدم أدوات المُطور الخاصة بمتصفح Chrome مع العلم انه حالياً كلا المتصفحين الآخرين Firefox وEdge لهما نفس الأدوات ونفس الواجهات، ولكن أحببت ان اختصار متصفح Chrome لأنه الأشهر بين المُطورين.

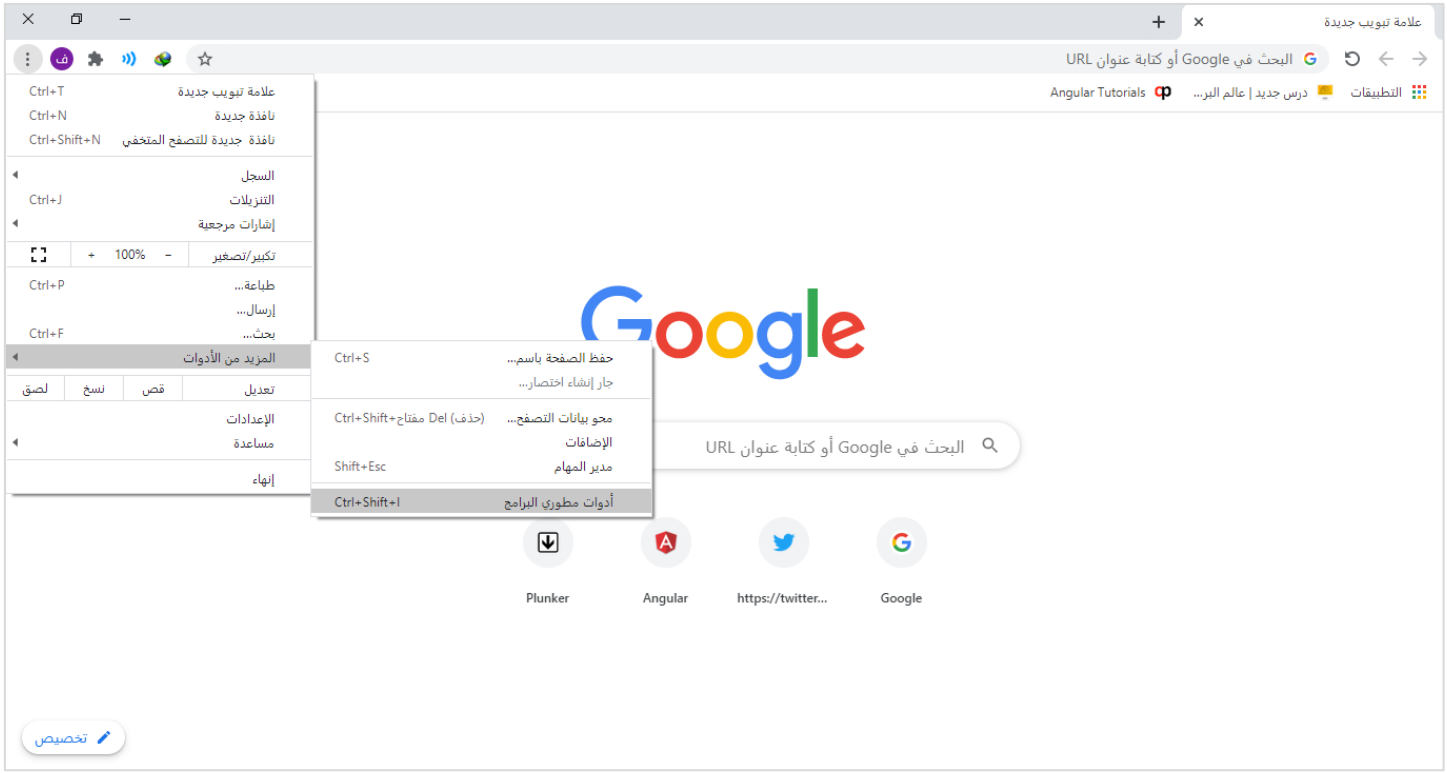
2.3.فتح Developer Tools، ونظرة عامة عليها:

لنقم بفتح المتصفح Chrome، كالتالي:

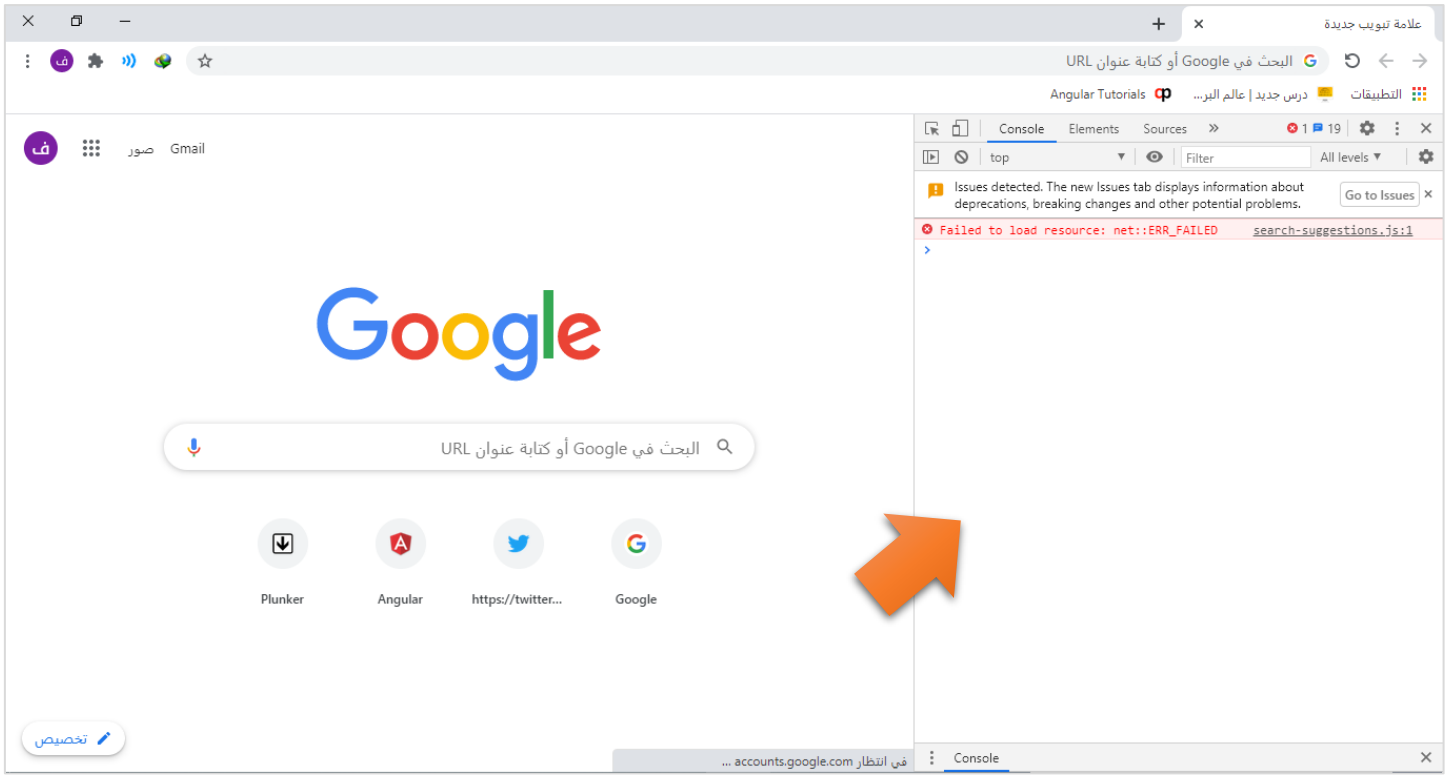


بطبيعة الحال قد تختلف الواجهة من جهاز إلى آخر بحسب الإعدادات الشخصية الخاصة بك عزيزي المتعلم.

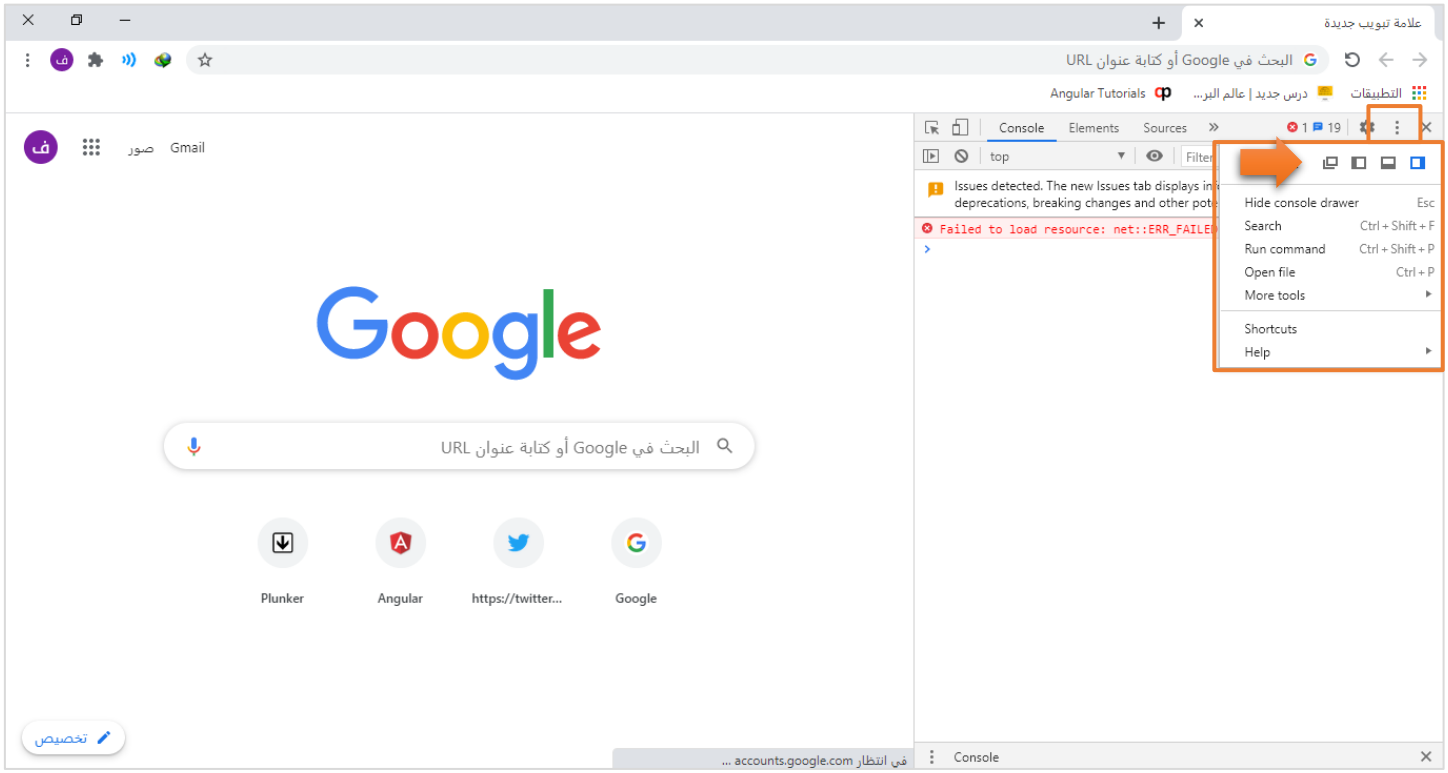
ونستطيع فتح أدوات المُطور Developer Tools بعدة طرق، منها الذهاب إلى الثلاث نقاط الشهيرة في شريط متصفح Chrome ومن ثم اختيار المزيد من الأدوات ومن ثم اختيار أدوات مطوري البرامج، اما الطريقة الأخرى فعن طريق كتابة الاختصار Ctrl + Shift + I، كالتالي:



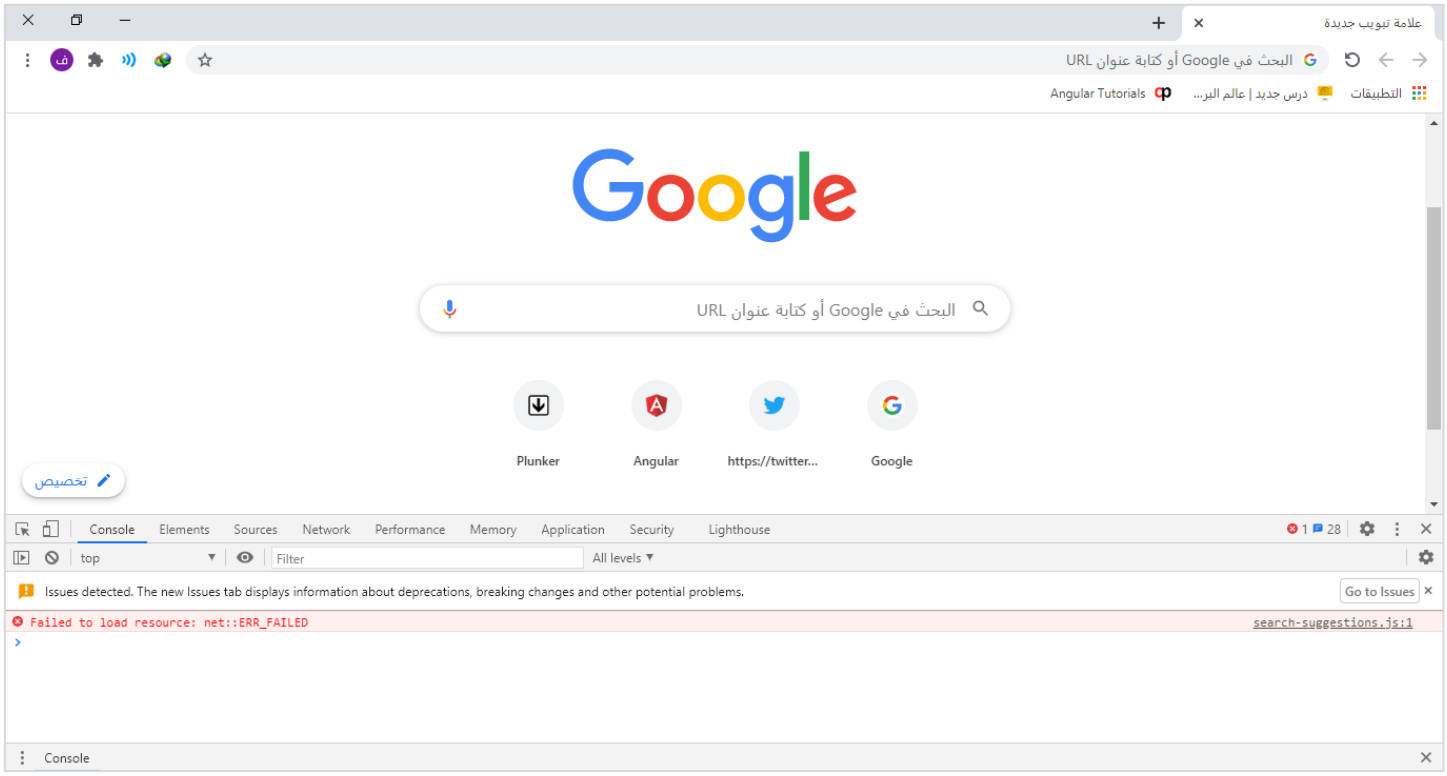
وعند اختيار هذا الأمر سوف تظهر لنا شاشة أدوات المطور، كالتالي:



هنا ظهرت على يمين الشاشة وقد تظهر لديك عزيزي المتعلم في الأسفل او الأعلى او حتى في نافذة جديدة وتستطيع التحكم في اتجاه ظهورها بالضغط على الثلاث نقاط حيث سوف تظهر لنا قائمة منسدلة نختار منها موقع ظهور هذه النافذة، كالتالي:

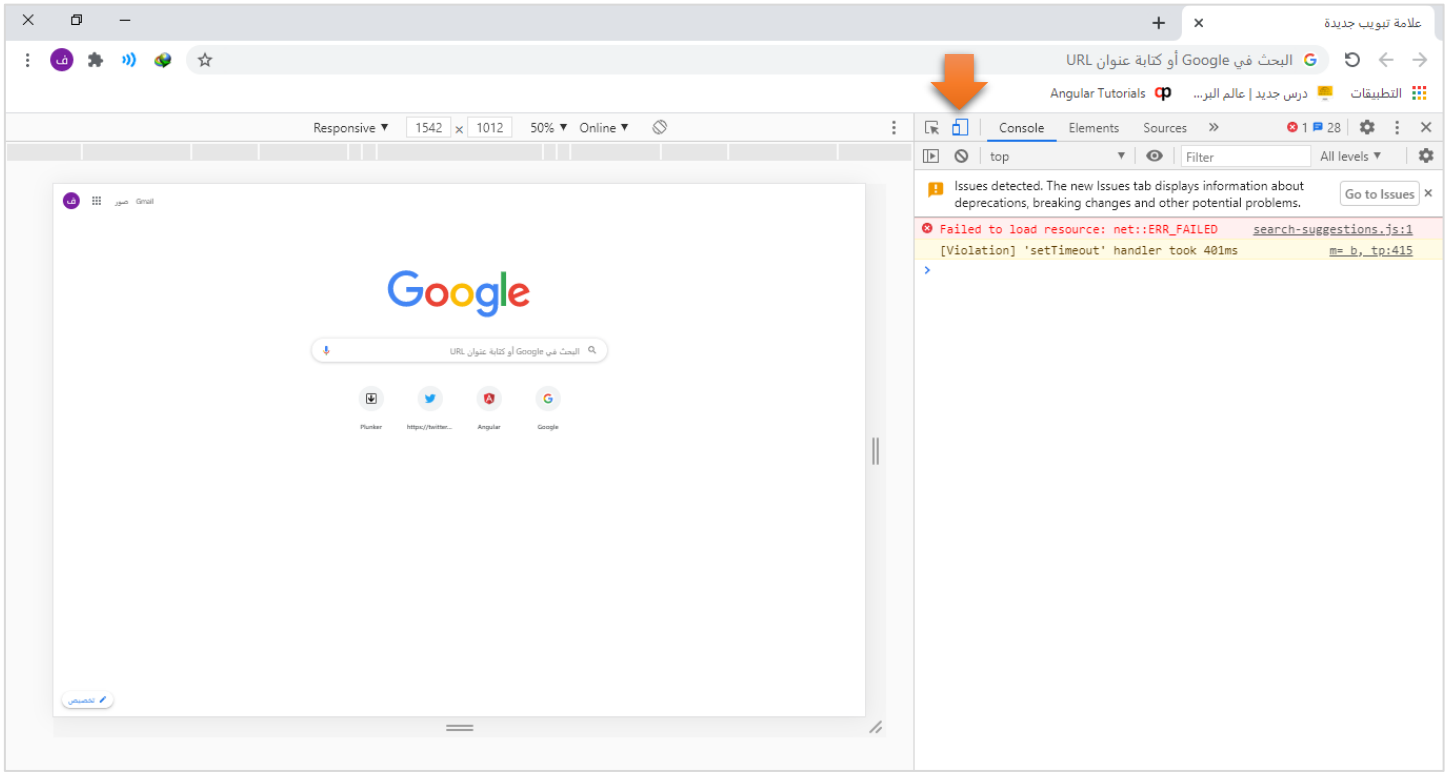


كما في الشكل السابق نستطيع اختيار موقع ظهور شاشة المطور في الصفحة، فمثلاً نستطيع وضعها في الأسفل، كالتالي:

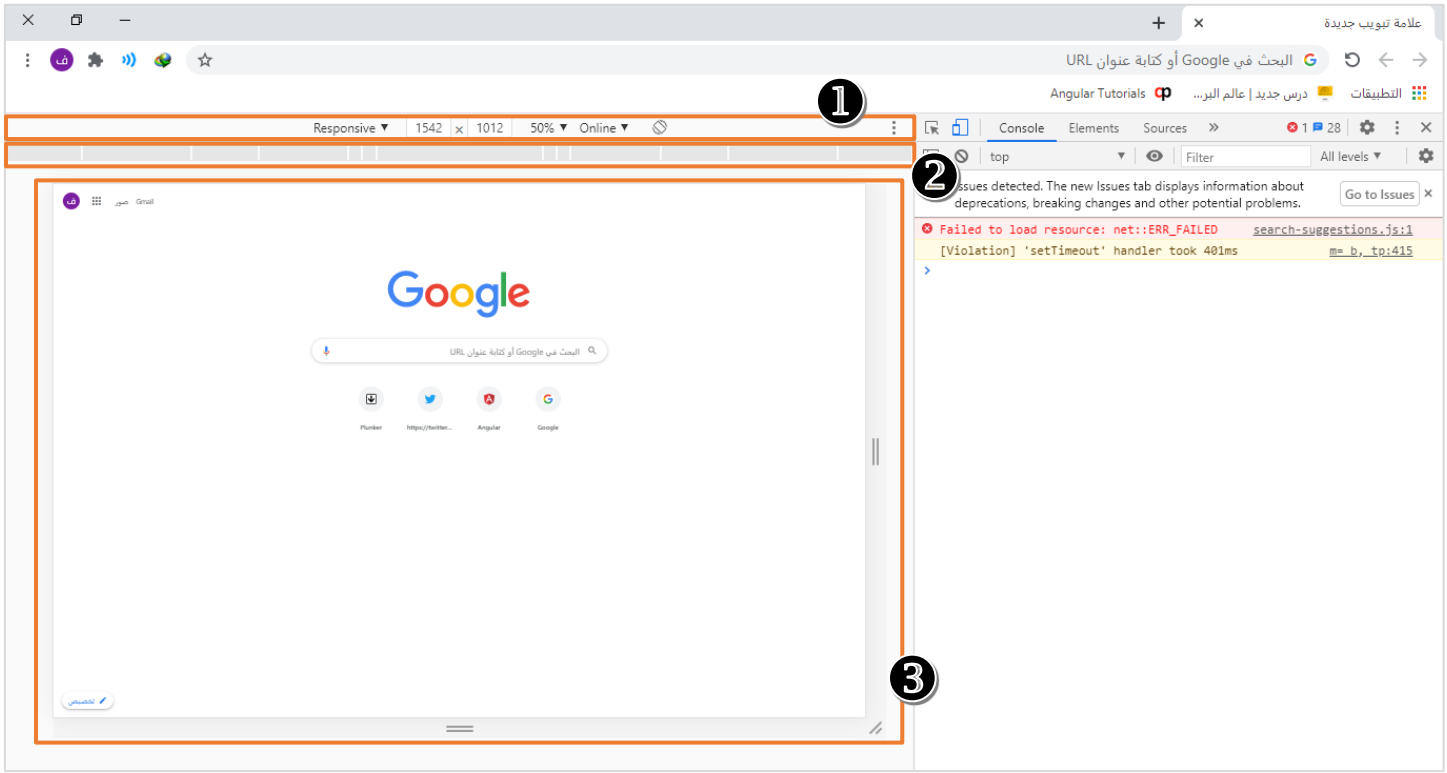


وفي الحقيقة وضعها في أي جزء يرجع لارتياحك معها، فإنا أفضل دائماً وضعها على يمين الشاشة، لذلك لنرجعها على يمين الشاشة، وذلك بإتباع نفس الخطوات السابقة.

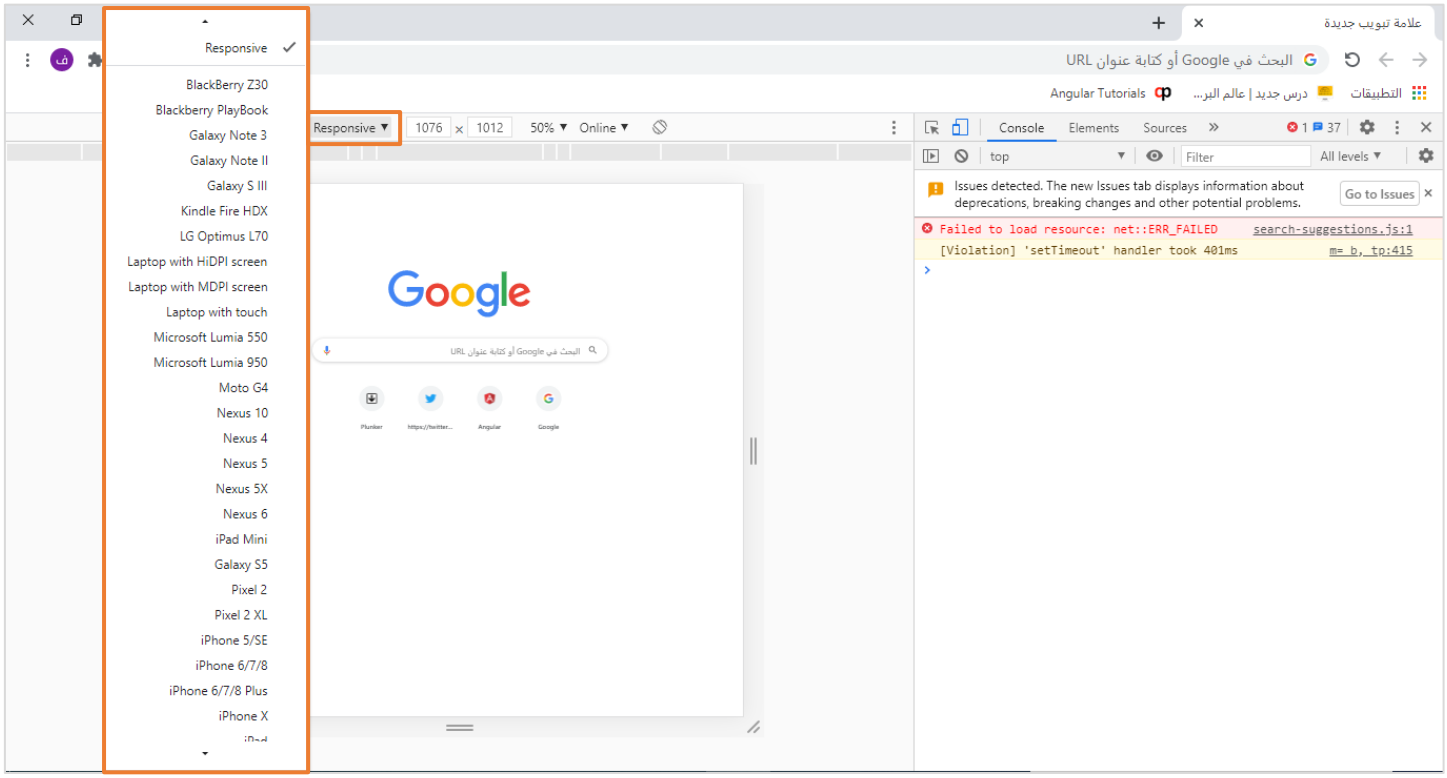
ومن الأمور التي نستطيع الاطلاع عليها هي محاكاة تجربة التطبيق على أكثر من جهاز لنتأكد من عدم وجود أي مشاكل ونستطيع القيام بهذا الأمر عن طريق الضغط على الموبايل، كالتالي:



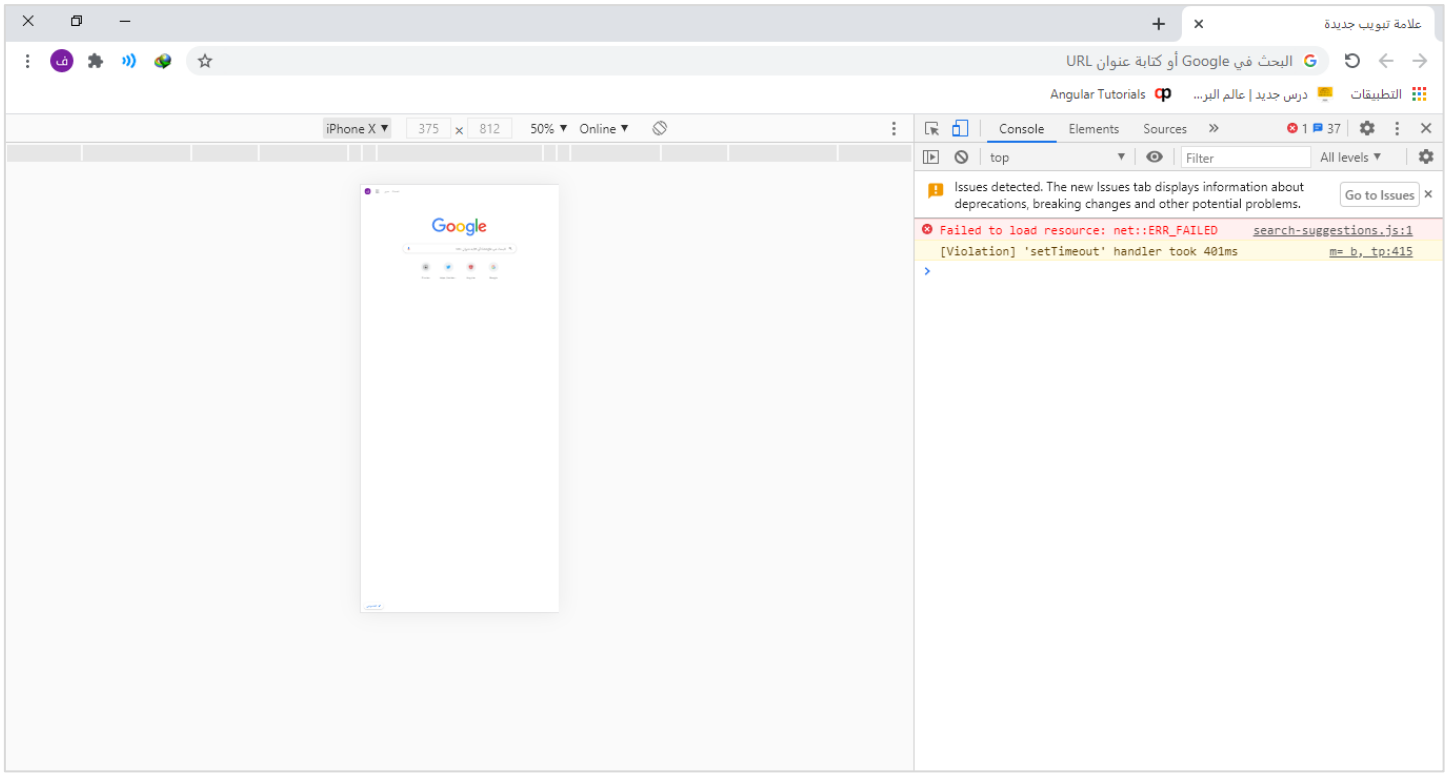
الشاشة السابقة مقسمة إلى مجموعة من الأجزاء، كالتالي:



وأول جزء هو الشريط الذي تم ترقيمه بالرقم 1 حيث يحتوي على مجموعة من الاجزاء هو الآخر، حيث من جهة اليسار سوف يواجهنا Device الذي نريد ان نحاكي التطبيق او الموقع عليه بحيث نريد ان نتأكد كيف سيكون شكل التطبيق او الموقع على هذا الجهاز، وعند الضغط عليه سوف تأتينا قائمة تحتوي على اغلب الأجهزة المشهورة، كالتالي:

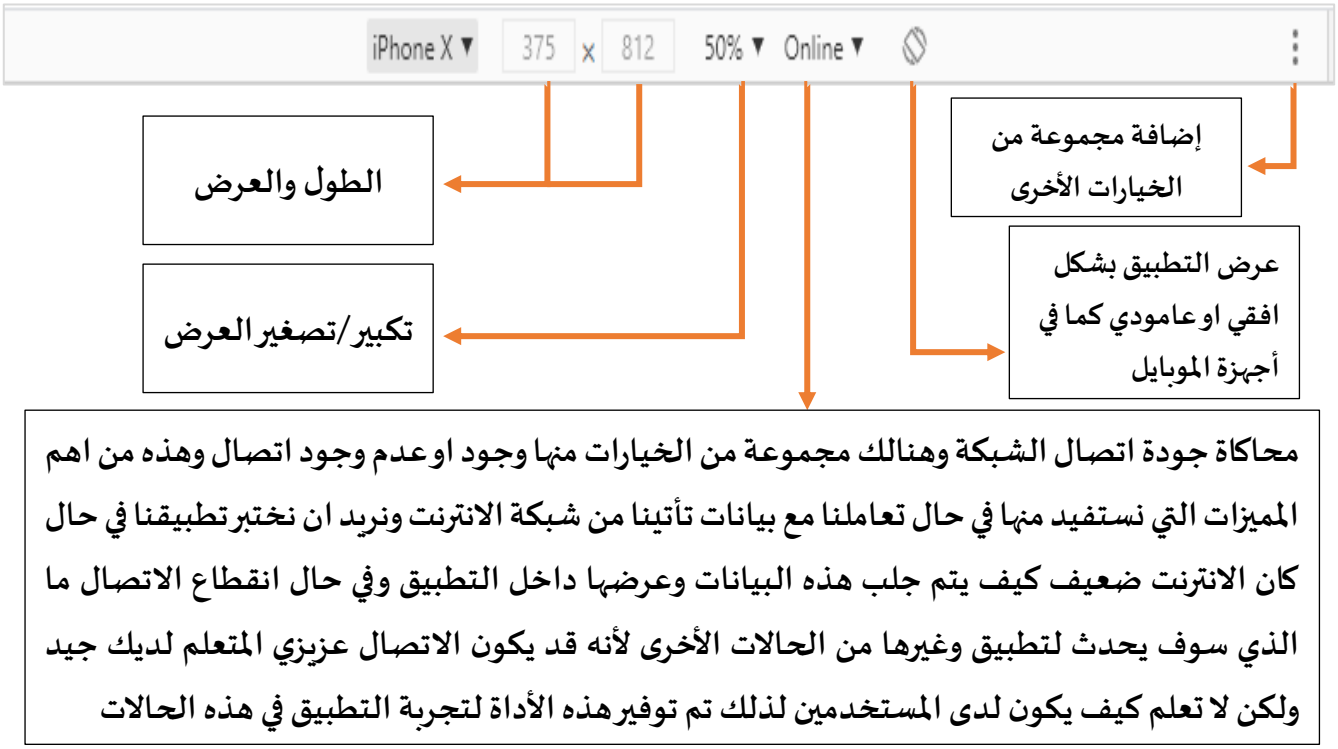


كما نلاحظ ظهرت لنا القائمة التي تحتوي على الأجهزة تستطيع اختيار الجهاز الذي تريد ان تُحاكي مشاهدة تطبيقه عليه وليكن iPhone X، كالتالي:

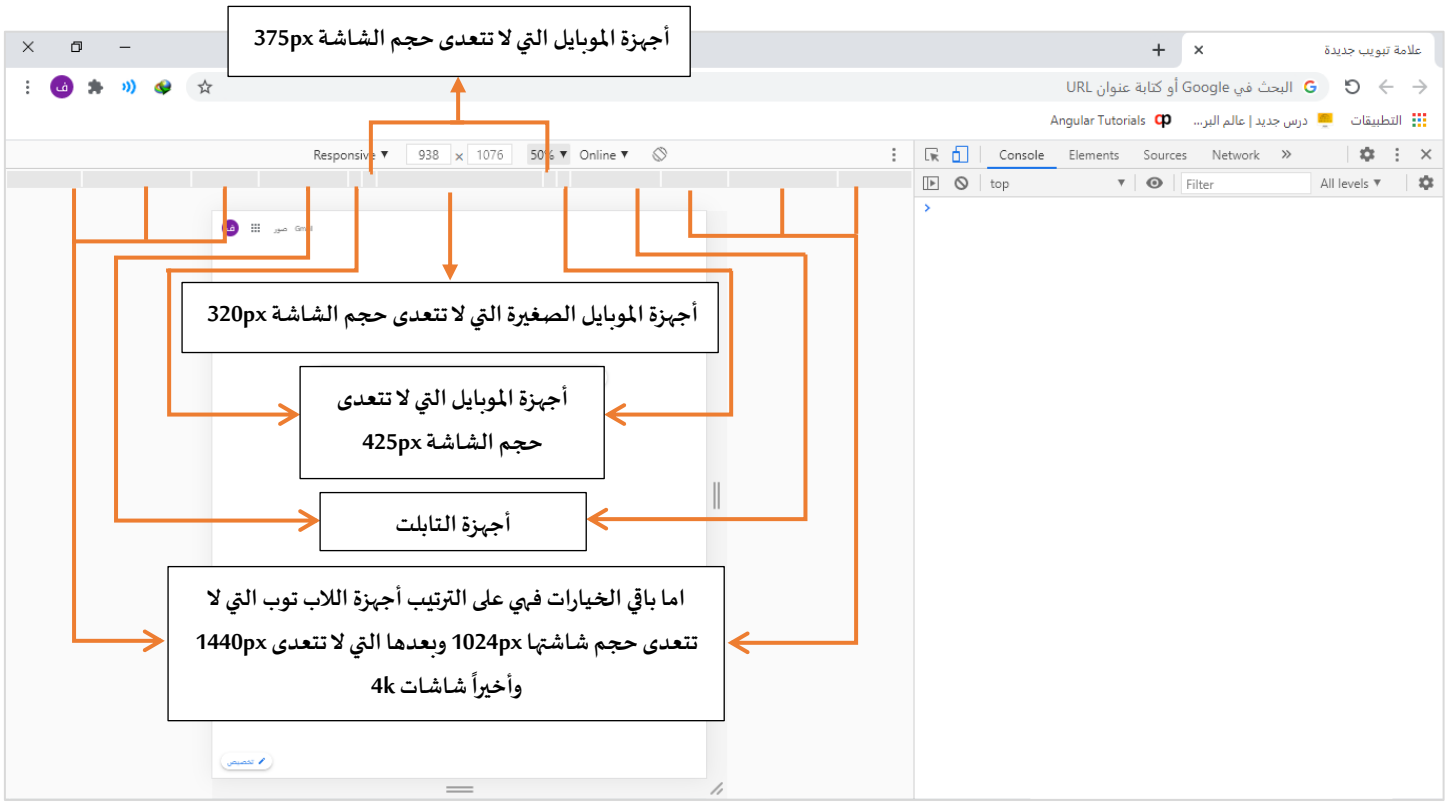


نلاحظ تغير شكل التطبيق ليتناسب مع ابعاد هذا الجهاز، وبذلك عن طريق هذه الميزة تستطيع عزيزي المتعلم معرفة كيف سيكون شكل التطبيق على اغلب الأجهزة لكيلا تتفاجأ بعدما تنتهي من بناء تطبيقك ونشره بان هنالك مشاكل معينة في التصميم والأبعاد الخاصة بتطبيقك على بعض الأجهزة، وفي حال أردت ان تُضيف أجهزة أخرى فتستطيع القيام بذلك عن

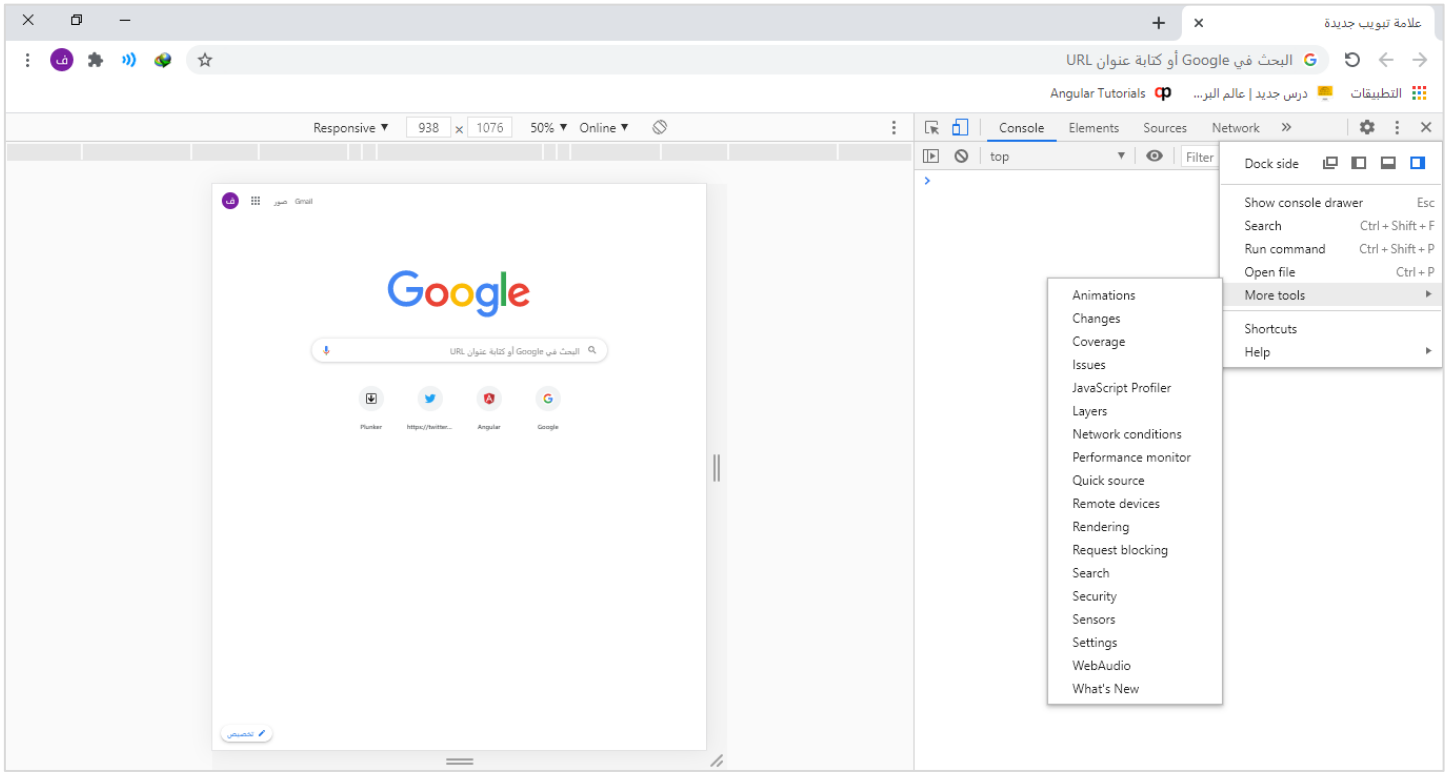
طريق الضغط على السهم ▼ في القائمة المنسدلة التي تحوي أسماء الأجهزة إلى أن تظهر لك كلمة Edit عندها قم بالضغط على هذه الكلمة وسوف تظهر لك شاشة أخرى تستطيع عن طريقها إضافة أجهزة أخرى.



اما الجزء الثاني فهو يعطيك خيارات مباشرة بغض النظر عن نوع الجهاز فهو يعتمد على أبعاد الشاشات، كالتالي:



هذا من ناحية المحاكاة وكيفية استخدامها وطرق الاستفادة منها، اما الخيارات فهي مقسمة على شكل تبويبات وهي كثيرة منها console وElements ...الخ وفي حال اردت الإطلاع على كامل هذه الخيارات تستطيع اختيار الثلاث نقاط ومن ثم More Tools، كالتالي:



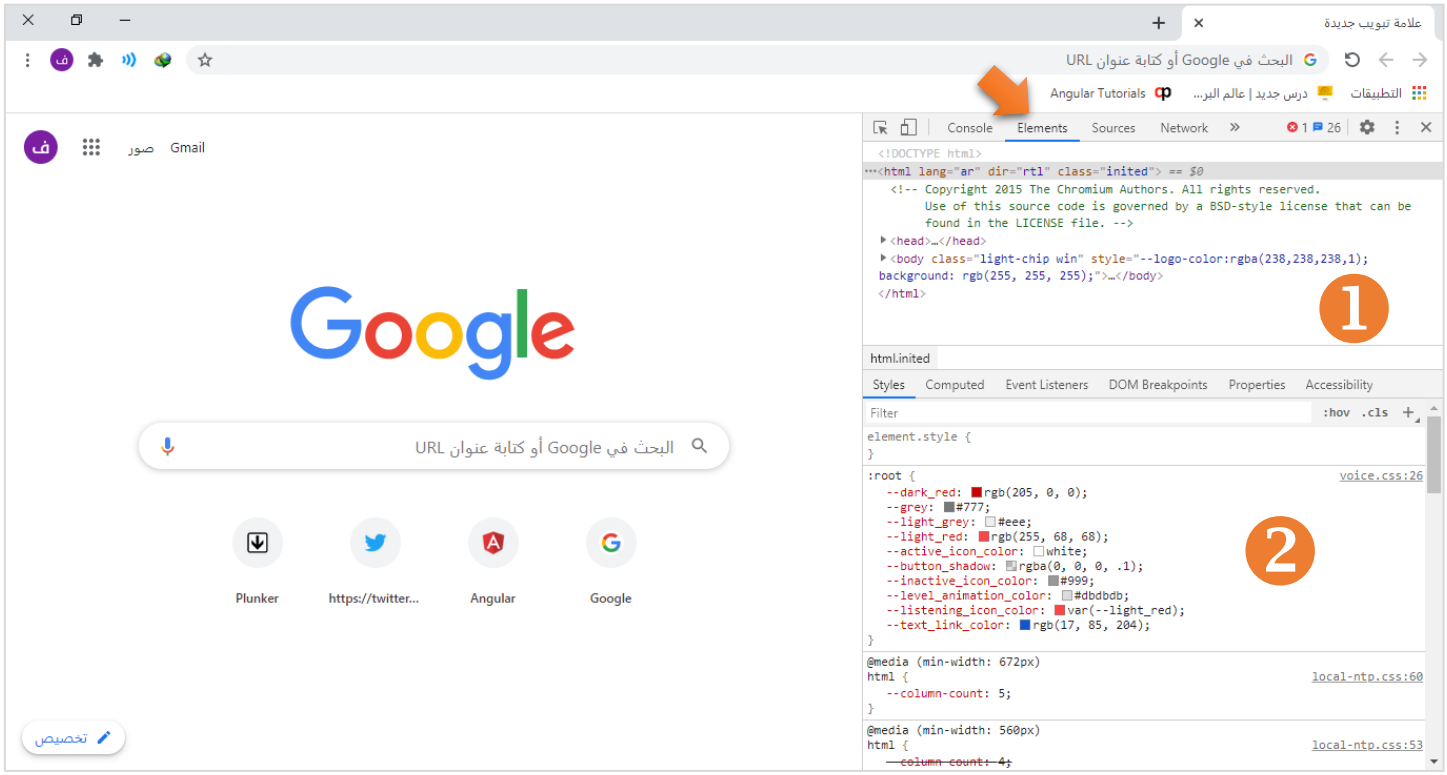
وكما أشرنا سابقاً فليس هنا المقام لشرح جميع هذه الخيارات ولكن سوف أقوم بشرح أهمها وأكثرها استخداماً لدى مطوري الويب.

Element.3.3 :

عن طريق هذا الخيار نستطيع التلاعب والتعامل مع DOM (DOM هو عبار عن إعادة بناء جميع عناصر صفحة HTML الخاصة بك عند تشغيل الموقع على شكل شجري ومتداخل مع تفرعاته العديدة تبعاً لعدد وتداخل العناصر في صفحة الموقع، بحيث كل عنصر من هذه العناصر يتم تحويلها إلى كائن في الذاكرة RAM وله خصائصه واحداثه وجميع تنسيقات الخاصة به).

ويجب التنويه انه في حال تحديث الصفحة فإن جميع هذه التعديلات سوف تذهب، والسبب ان هذا الخيار معمول لتجربة التصميم والتعديل عليه او معرفة اين الخطأ بالتصميم الخاص بنا وعند الانتهاء نأخذ هذه التعديلات إلى محرر الأكواد لكتابتها هناك وعندئذً سوف يتم حفظ هذه التعديلات.

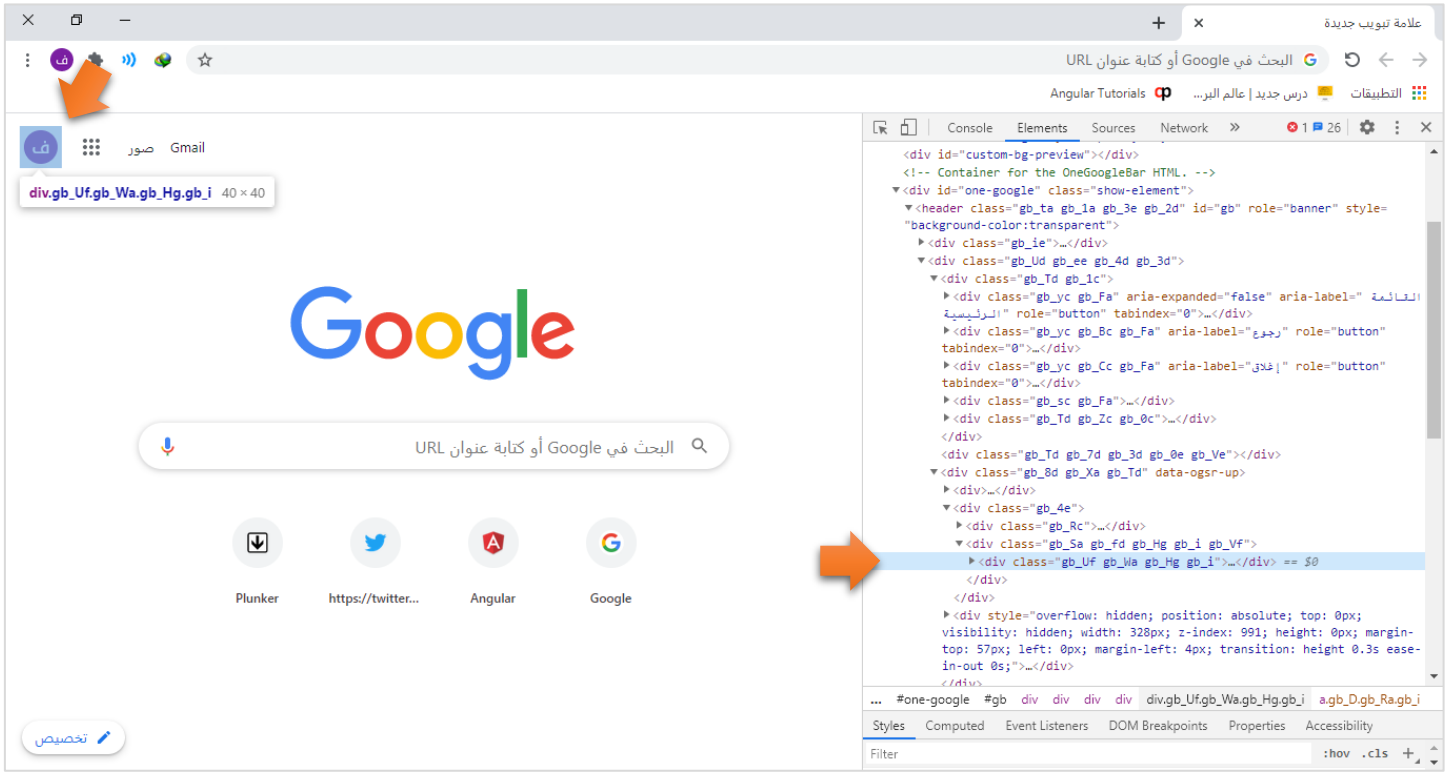
اما الآن لنستعرض محتويات هذا الخيار ولنرى ما فيه، كالتالي:



(1) في هذه المنطقة نستطيع استعراض جميع العناصر في DOM والتلاعب بها والتحكم فيها.

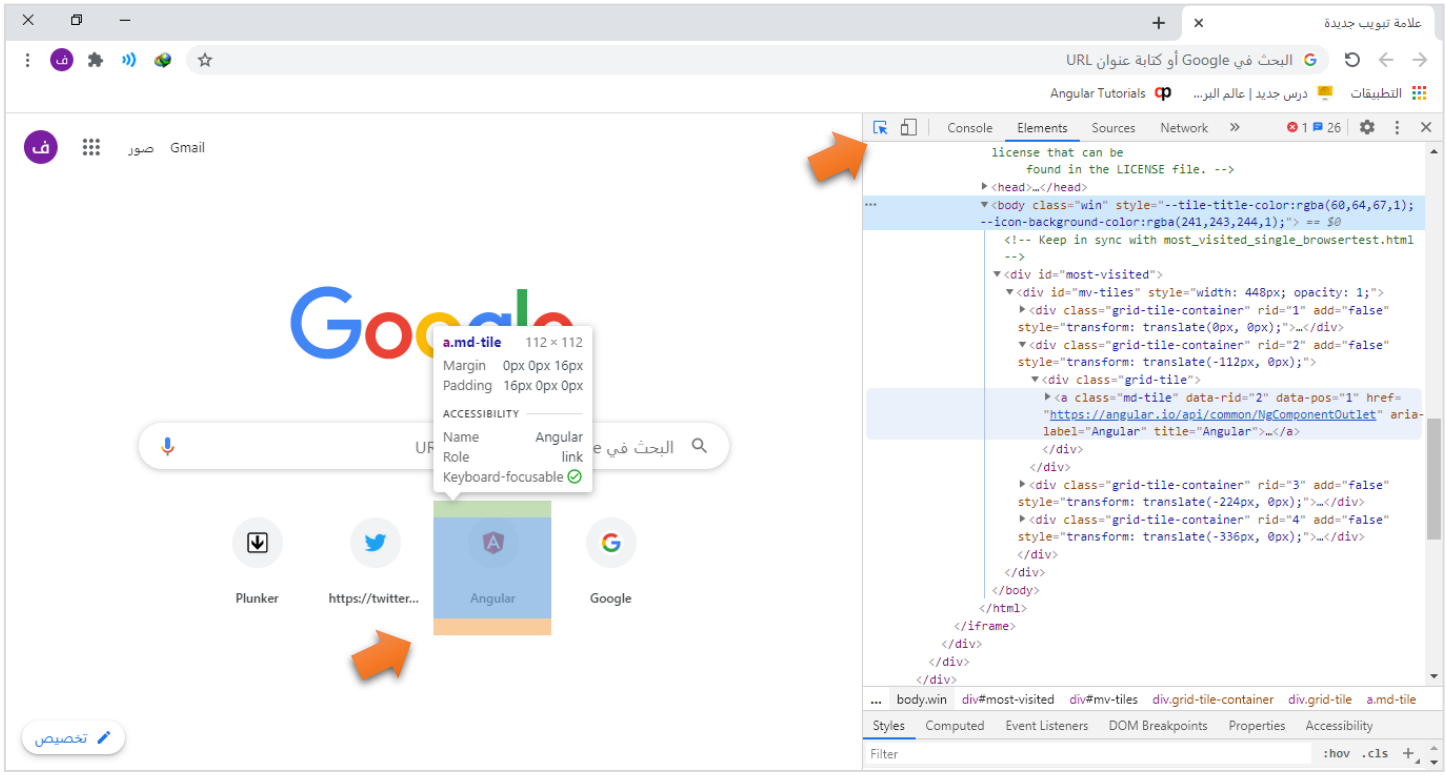
(2) اما هنا فإي عنصر يتم اختياره في رقم (1) نستطيع التحكم بجميع تنسيقات css الخاصة به من إضافة او تعديل او غيره.

ونستطيع الوصول لأي عنصر بالصفحة بطريقتين اما بالطريقة التقليدية وهي البحث عنه في جميع التفرعات الشجرية في DOM، مع العلم ان أي عنصر نضع عليه مؤشر الفأرة فإن المتصفح تلقائياً سوف يقوم بعمل تضليل عليه، كالتالي:

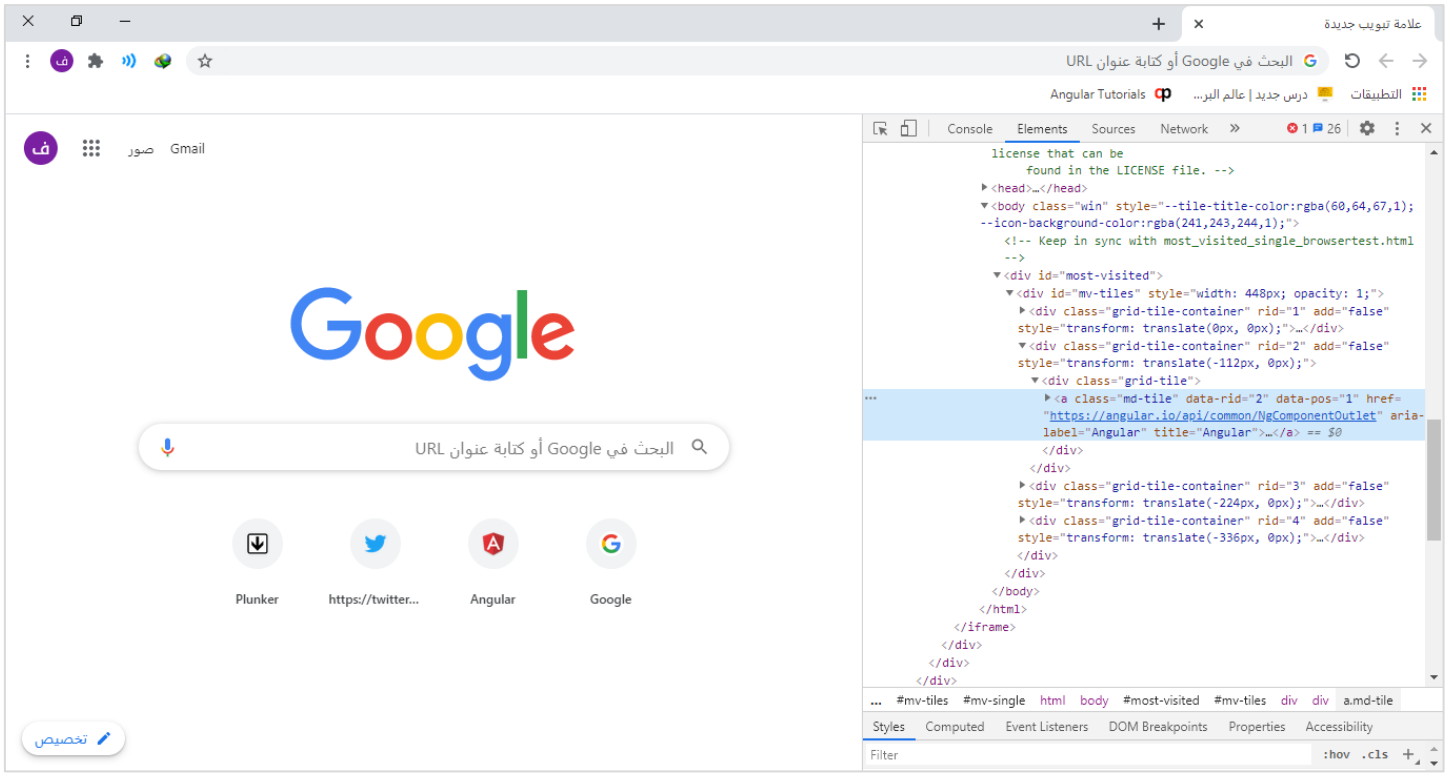


نلاحظ استخدمنا الطريقة التقليدية إلى ان وصلنا إلى العنصر المحدد وبمجرد اختيارنا له في DOM تم تحديديه في الصفحة.

هذا بالنسبة لطريقة الأولى، اما الطريقة الثانية فهي أسهل واسرع وذلك عن طريق الضغط على السهم في الصفحة ومن ثم اختيار أي عنصر تريده في الصفحة وسوف يتم تحديده تلقائياً في DOM، كالتالي:

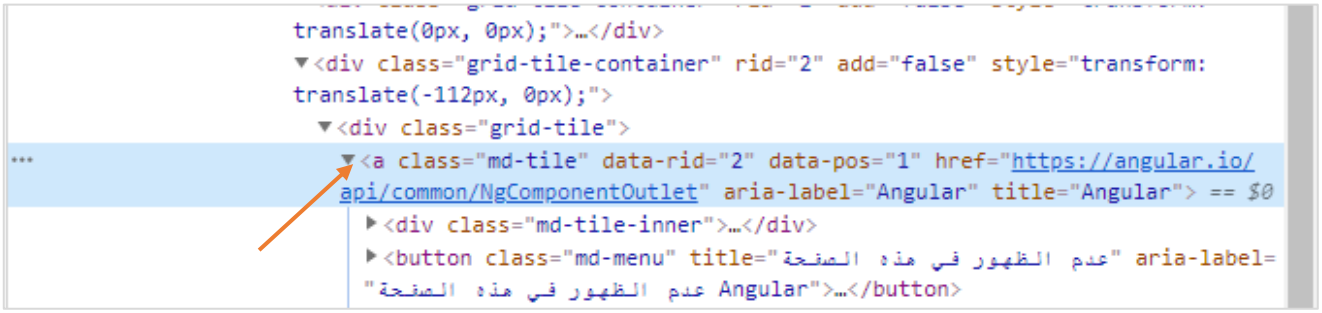


نلاحظ بمجرد اننا قمنا بوضع مؤشر الفأرة (لم يتم الضغط فقط مرور المؤشر فوق العنصر في الصفحة) على العنصر وهو صورة ورابط Angular تم تضليلها، والآن لنقم بالضغط ولنرى النتيجة في DOM، كالتالي:

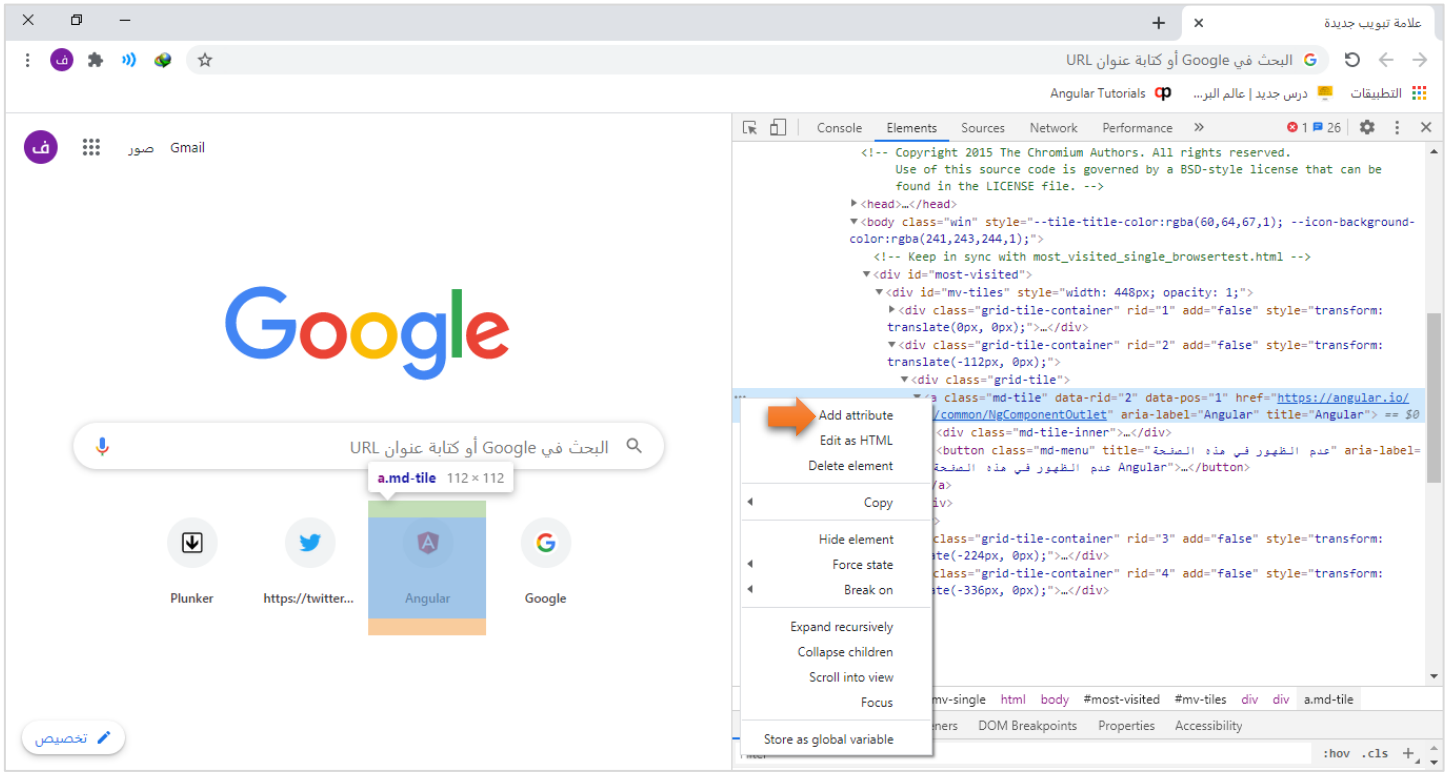


بعدما تعرفنا كيف نقوم بالوصول إلى أي عنصر، لنعرف الآن كيف نقوم بالتحكم بهذا العنصر سواء بتعديل خصائصه في DOM او تعديلها او حتى حذفه، ونستطيع القيام بهذا الأمر بعدة طرق منها ان نقوم بالضغط بزر الفأرة الأيمن على العنصر

المراد إضافة الخاصية له، ولنفرض اننا نريد ان نقوم بإضافة خاصية للعنصر <a> فنقوم في البداية بالذهاب إلى العنصر ونتأكد اننا نكون على نفس العنصر وعندها نضغط زر الفأرة الأيمن، كالتالي:



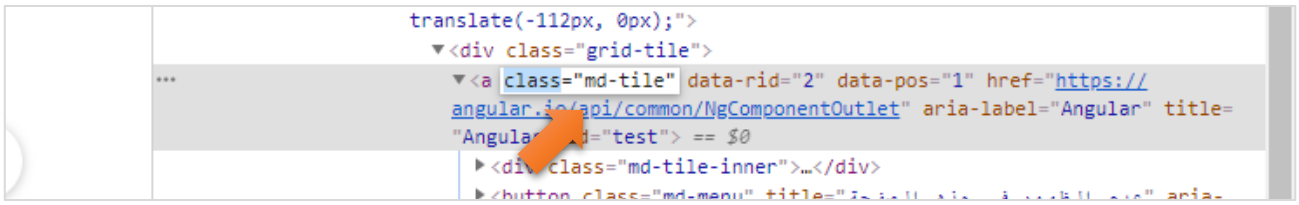
نتأكد أننا على نفس العنصر <a> وعندها نضغط زر الفأرة الأيمن، كالتالي:



سوف تظهر لنا قائمة منسدلة نختار منها الخيار الأول إضافة خاصية، ولتكن الخاصية المضافة id والقيمة test، كالتالي:

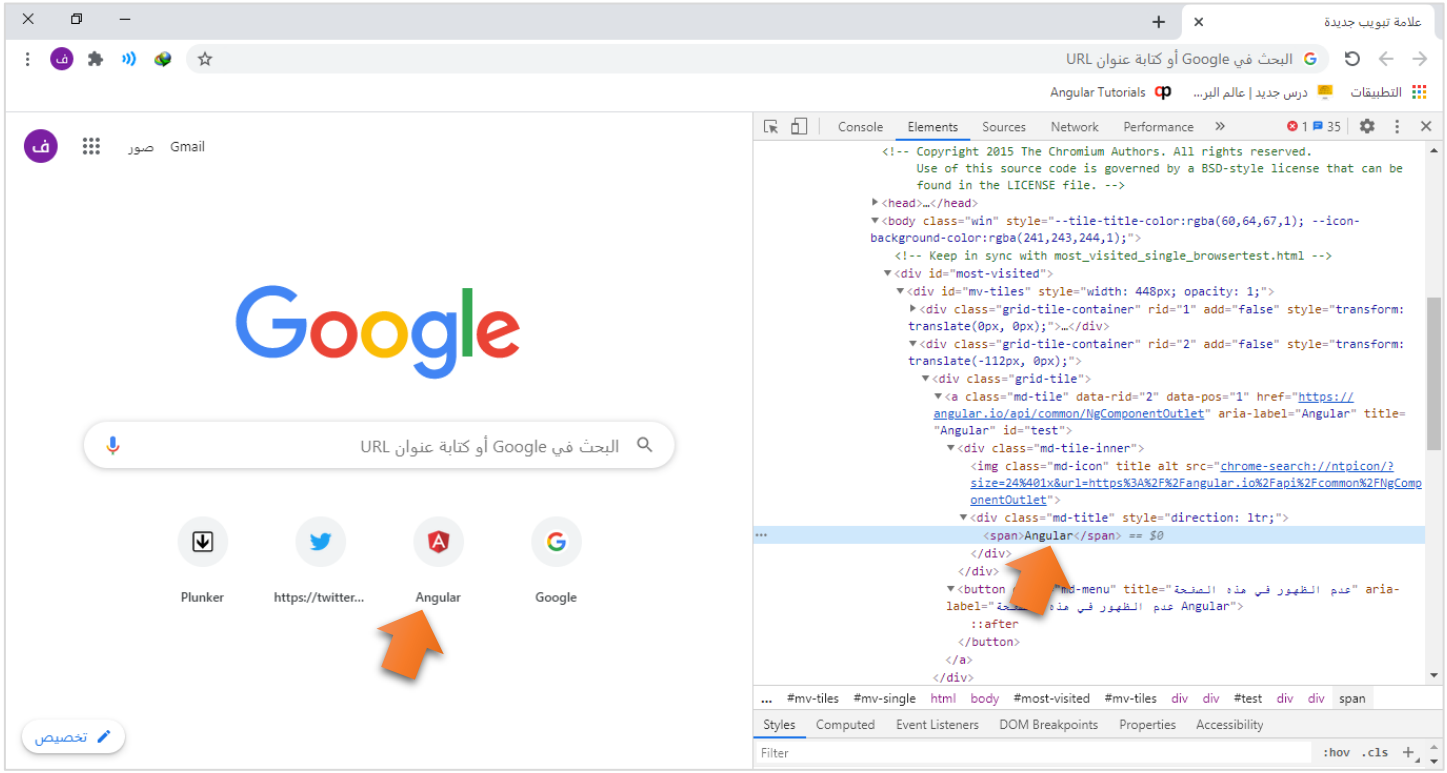
او نستطيع الضغط على النقاط الثلاثة المؤشر عليها بالشكل السابق وسوف تظهر لنا نفس القائمة.

اما في حال الرغبة بالتعديل على خاصية معينة، ولتكن على سبيل class نستطيع اما بالضغط عليها ضغطتين متتاليتين بزر الفأرة الأيسر او بوضع مؤشر الفأرة عليها ومن ثم نضغط بزر الفأرة الأيمن، وسوف تظهر لنا نفس القائمة المنسدلة السابقة ولكن هنا فيها امر أضافي هو Add Attribute، كالتالي:

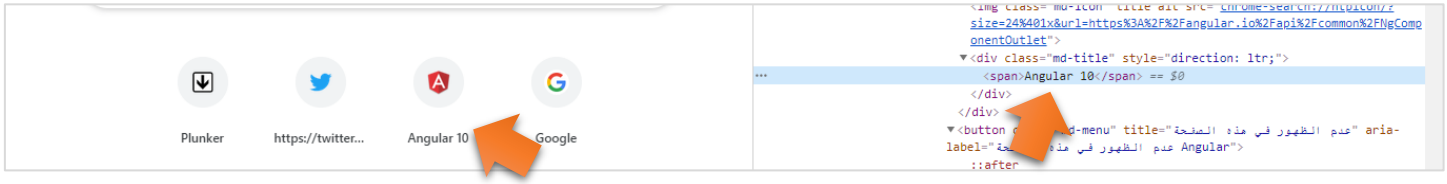


نلاحظ ان الخاصية أصبحت بوضع التحرير وعندها نستطيع تغيير قيمتها او حذفها لتجريب.

كما نستطيع تغيير نص داخل عنصر، ولنفرض مثلاً أردنا تغيير الكلمة Angular، كما في الشكل التالي:



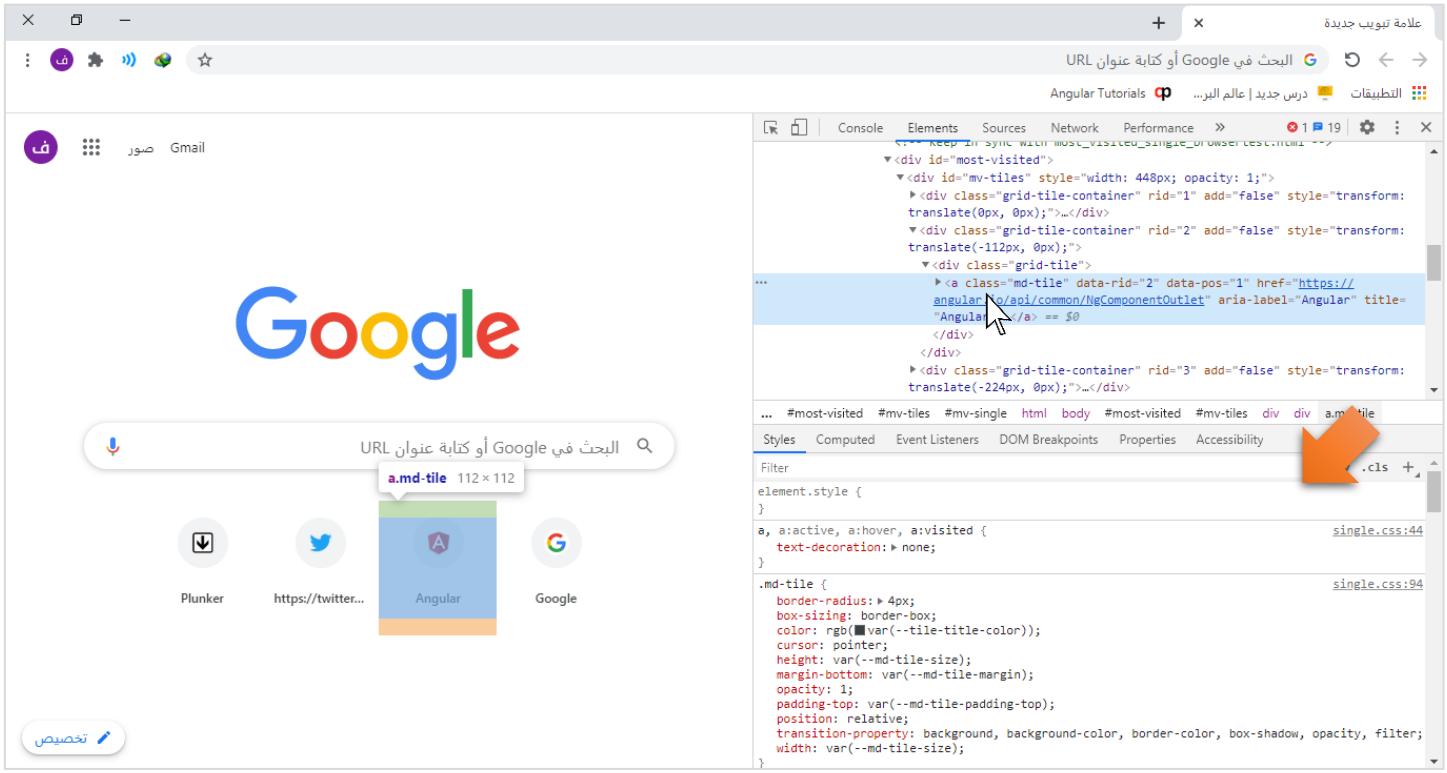
ولتعديل عليها نقوم في البداية بالوصول إلى العنصر الذي توجد به هذه الكلمة وهو ومن ثم لدينا طريقتين اما بالضغط بزر الفأرة الأيسر ضغطتين متتاليتين في العنصر في DOM ومن ثم القيام بالتعديل او بالضغط بزر الفأرة الأيمن فوق الكلمة وسوف تظهر لنا قائمة نختار منها Edit Text، كالتالي:



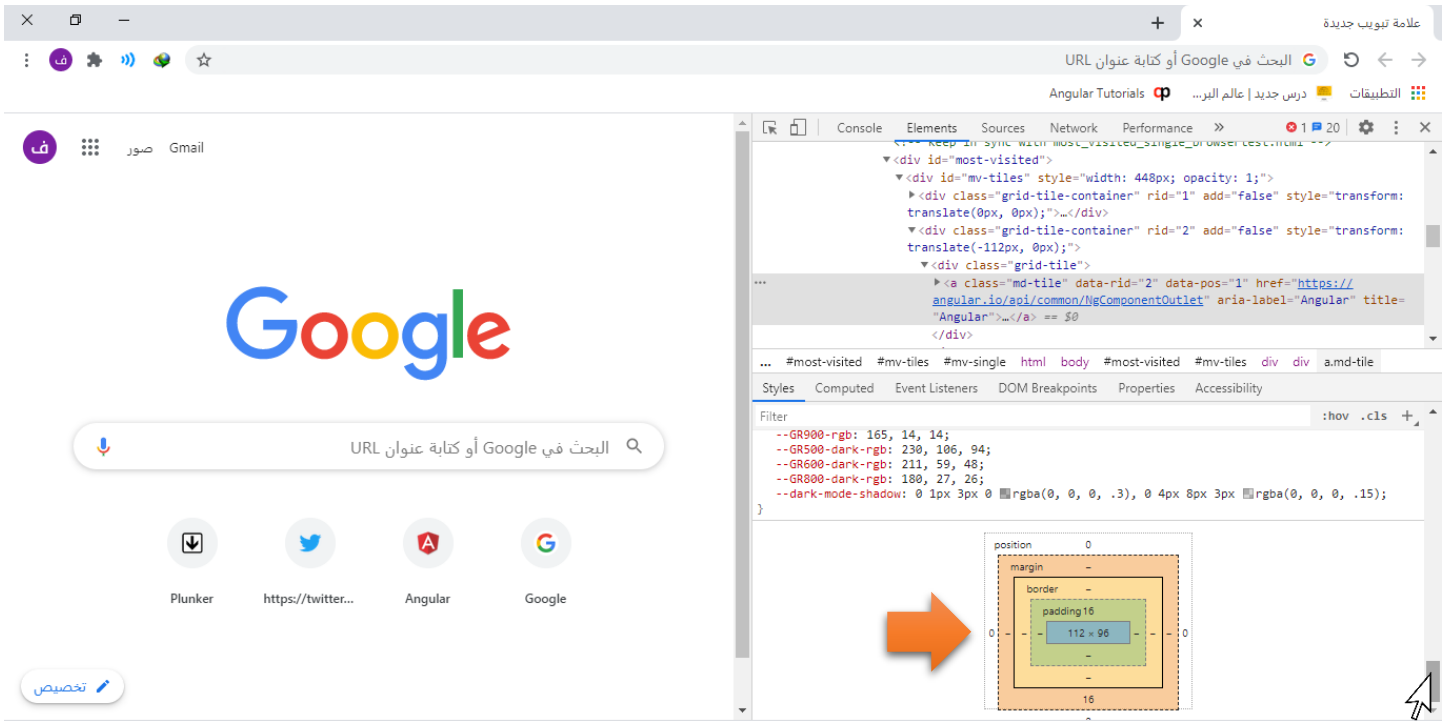
نلاحظ استطعنا تعديل القيمة ولو قمنا بتحديث الصفحة فإن جميع هذه التعديلات سوف تُحذف، كما أشرنا لهذا سابقاً.

نكتفي بهذا القدر من ناحية التعامل مع العناصر في DOM، وتستطيع عزيزي المتعلم الاستزادة من مراجع أخرى، في حال الرغبة بذلك.

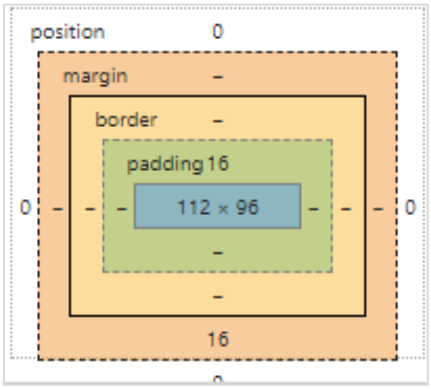
قلنا سابقاً أن أي عنصر يتم تحديده في DOM سوف تظهر لنا في شاشة أخرى جميع تنسيقات CSS والإحداث Events الخاصة به، لذلك لنقوم بتحديد أحد العناصر كما تعلمنا سابقاً وليكن نفس العنصر السابق، كالتالي:



نلاحظ عند تحديدنا واختيار هذا العنصر ظهرت جميع تنسيقات css الخاصة به في الجزء المؤشر بالسهم، ولنقوم بتحريك شريط التمرير إلى الأسفل، كالتالي:



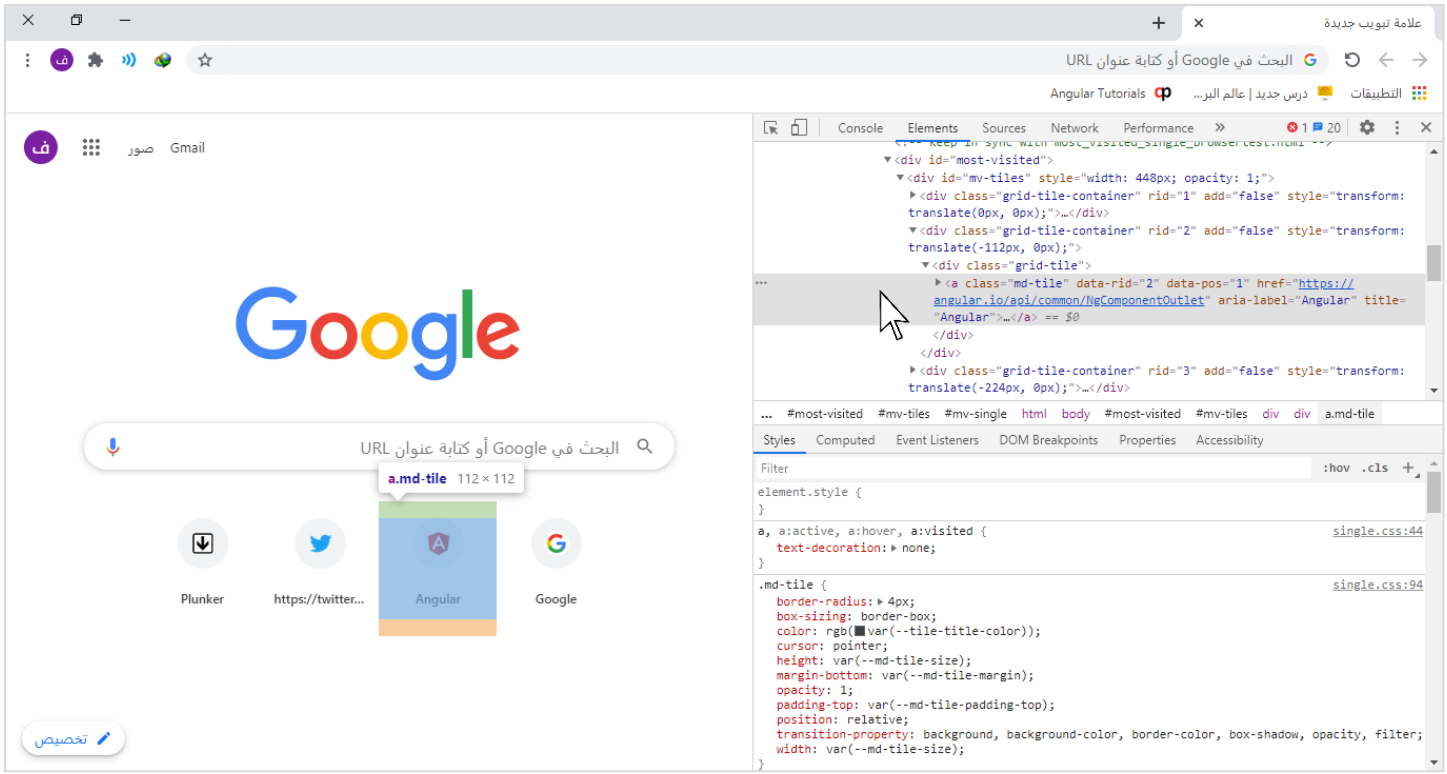
نلاحظ الشكل المؤشر بالسهم هو في الحقيقة عبارة عن العنصر المحدد ويوجد به مجموعة من القيم التي نستطيع التعديل عليها، لذلك لنقم بتكبير هذا الشكل ومن ثم نشرح هذه القيم وكيف يمكن الاستفادة منها، كالتالي:



المربع الأزرق يمثل العرض والطول للعنصر، اما الأخضر فيمثل خاصية css ذات الاسم padding حيث نستطيع تحديد padding من جميع الاتجاهات (top – bottom – left – right) حيث في الشكل بالأعلى تم تحديد padding لهذا العنصر لي top فقط بقيمة 16px ونستطيع تغيير القيمة بالضغط بزر الفأرة الايسر ضغطتين متتاليتين على القيمة ومن ثم سوف تصبح القيمة في وضع التحرير وعندها تستطيع تغيير القيمة للقيمة التي تُريدها، واللون الذي يليه وهو احدى تدرجات اللون البني يمثل حدود العنصر والذي يمثل بالخاصية border ونفس ما قيل من ناحية الاتجاهات ايضاً يُقال هنا، وبعدها يأتي المربع البرتقالي حيث يمثل خاصية margin وايضاً نستطيع التعديل عليها بكلا الاتجاهات.

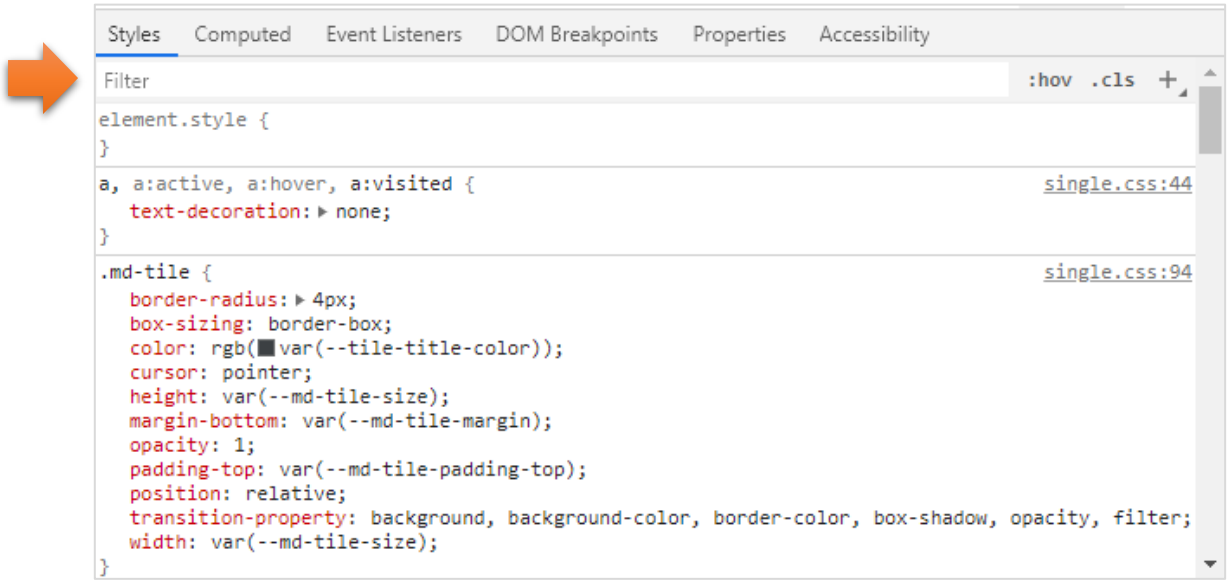
ونستفيد من هذا الشكل بضبط هذه الخصائص للعنصر ومشاهدة التعديلات بشكل مباشر في المتصفح وعند الانتهاء نأخذ هذه القيم ونضعها في المحرر لكي نعتمدها في التصميم.

لنرجع الآن إلى بداية هذا الجزء، كالتالي:



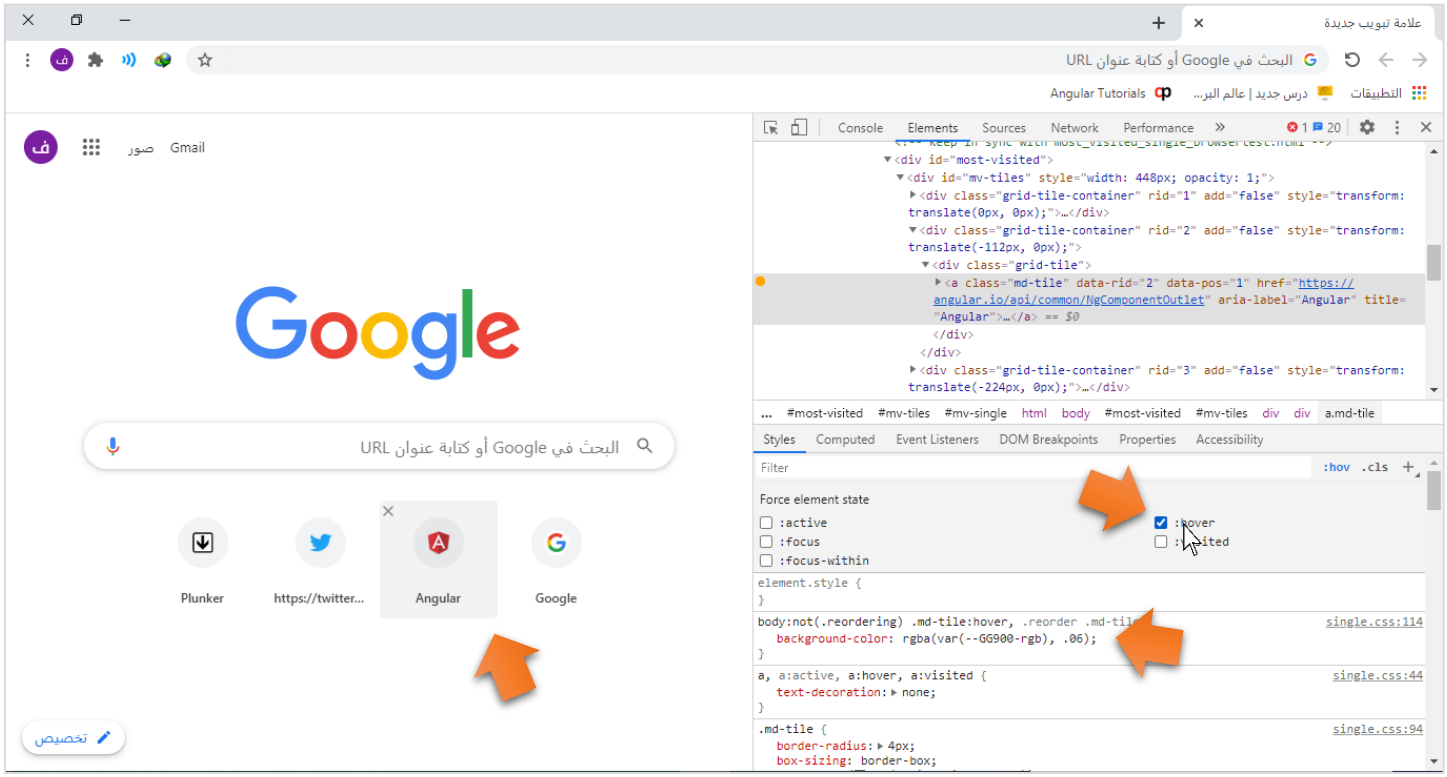
سوف نلاحظ انه بمجرد وضع مؤشر الفأرة على العنصر في DOM تم تضليل العنصر المقابل في الصفحة وليس هذا فحسب وانما تم تحديد padding وmargin لهذا العنصر باللونين الأخضر والبرتقالي بحسب القيم المُعطاة له.

ولنقم بتكبير الجزء الخاص بالتنسيقات ونشرح اهم الأجزاء مع العلم ان العنصر المحدد هو نفسه السابق وجميع التنسيقات سوف تتم عليه، كما أشرنا سابقاً اننا نقوم بتحديد العنصر وتظهر لنا جميع التنسيقات الخاصة به، كالتالي:



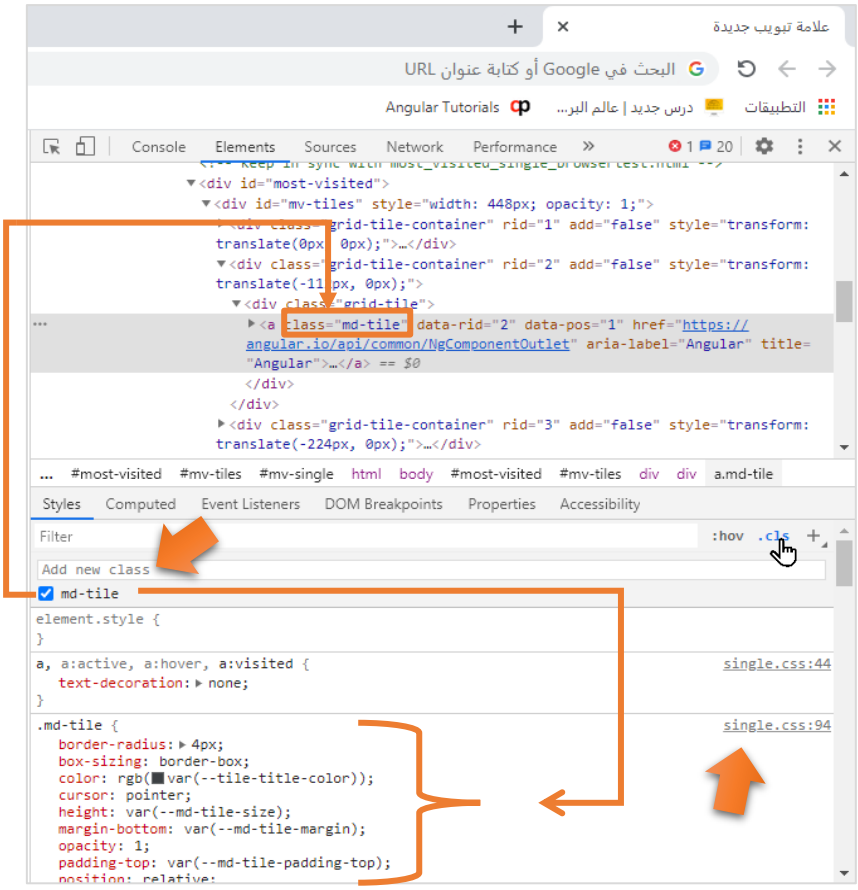
أول جزء سوف نتكلم عنه هو الشريط المؤشر عليه بالسهم في الشكل بالأعلى، حيث نستطيع كتابة اسم أي خاصية css هنا ومن ثم نقوم بالتعديل عليها، ويوجد ايضاً زر باسم hov:، ولنقم بالضغط عليها ولنرى ما سوف يحدث، كالتالي:

نلاحظ عند الضغط على هذا الزر سوف تظهر لنا قائمة نستطيع عن طريقها تحديد حالة وشكل العنصر إذا تم تطبيق عليه مجموعة من الأحداث Events، فمثلاً لو تم تحديد hover: فهذا معناه نُريد ان نعرف ما هو التأثير الذي يتم في حال مرور مؤشر الفأرة فوق هذا العنصر، كالتالي:



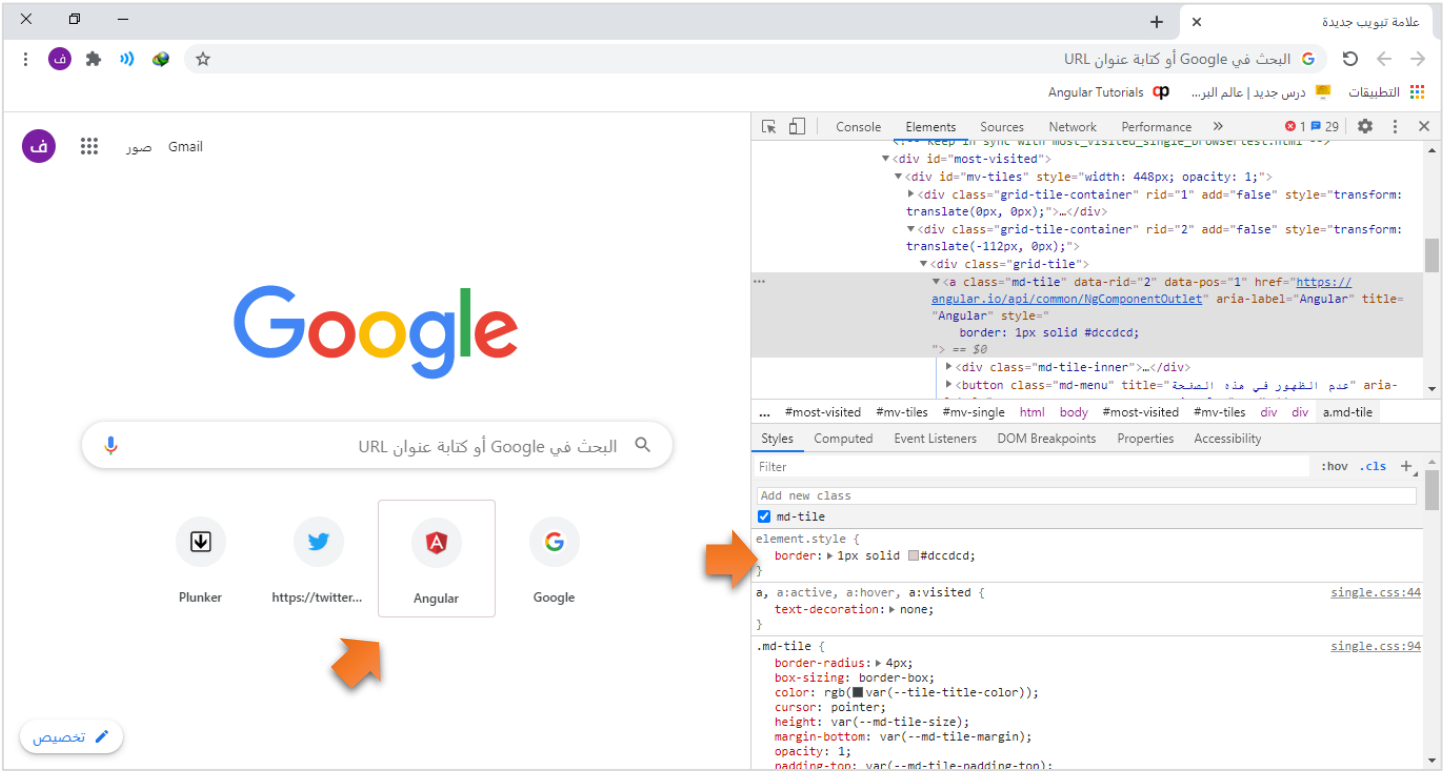
نلاحظ عند تحديد هذا الخيار تم اظهار الخاصية التي يتم تطبيقها على هذا العنصر عند مرور مؤشر الفأرة وهو لون الخلفية وعندها نستطيع تغييرها والتحكم بها وعند الانتها نأخذ القيمة الجديدة ونضعها في محرر الأكواد، لكي يتم تطبيقها بشكل دائم على هذا العنصر.

وبجنب هذا الزر يوجد زر آخر هو cls. والمقصود فيه تجربة إضافة كلاسات css، كالتالي:



نلاحظ عند الضغط على هذا الزر تم إظهار مربع نستطيع عن طريقه كتابة اسم الكلاس الجديد ونلاحظ انه لا يوجد في العنصر إلا كلاس واحد وهو md-tile ونستطيع تحريره والتعديل عليه وليس هذا فحسب فأيضاً يوفر لنا المتصفح الملف الموجود فيه هذا الكلاس وهو single.css ورقم السطر ايضاً وهو 94.

كما نستطيع ايضاً كتابة التنسيقات بشكل مباشر وسوف نشاهد التأثير مباشرة على العنصر في المتصفح، كالتالي:



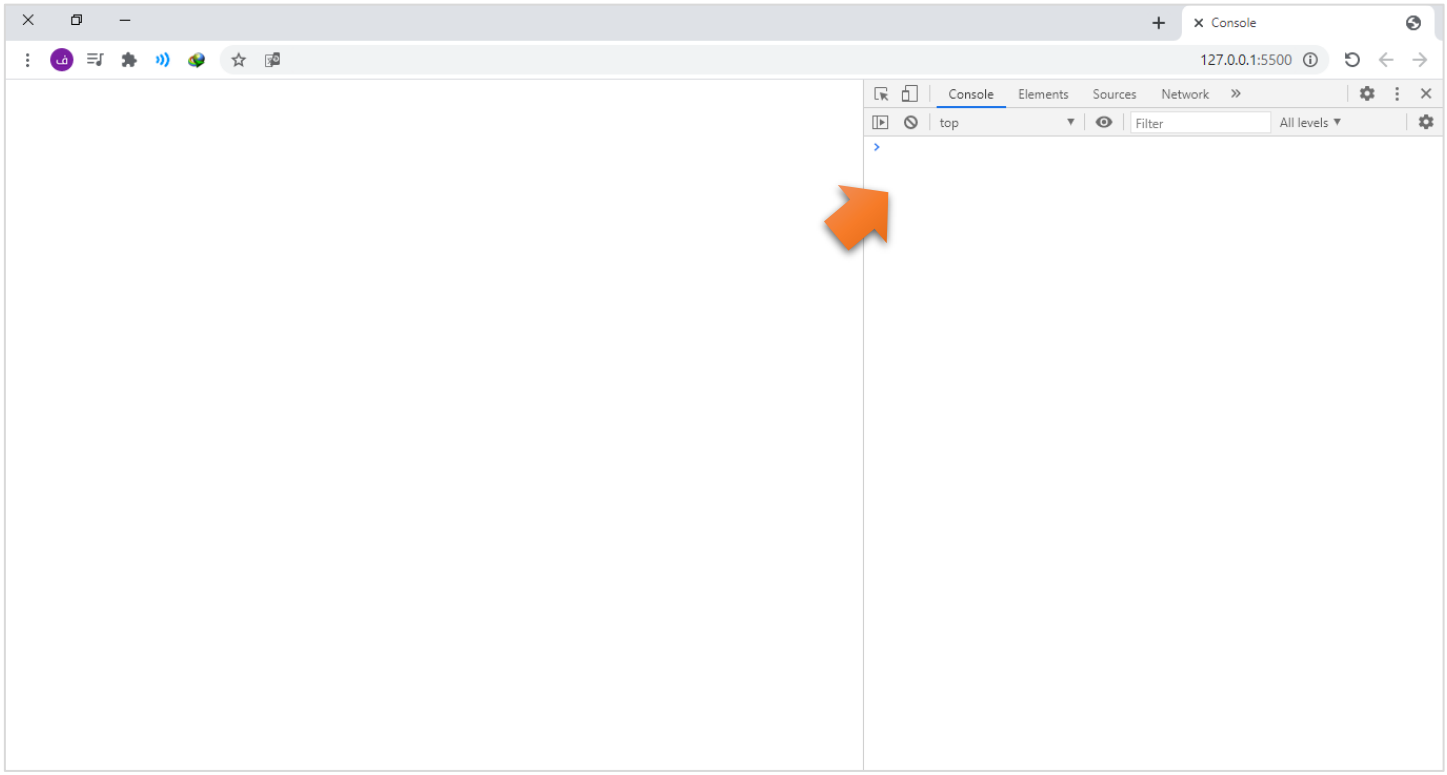
نلاحظ قمنا بكتابة التنسيق وهو ووضع border على العنصر وبنفس الوقت تم تطبيق هذا التنسيق على العنصر، وعند الانتهاء والرغبة بجعل هذه التنسيق او مجموعة التنسيقات دائمة نقوم بنسخ هذه التنسيقات ونقلها إلى محرر الأكواد لتطبيقها بشكل دائم.

ونكتفي بهذا القدر من الخيار Elements والذي يقدم خيارات متعددة لصممي الصفحات وليس له علاقة بكتابة الأكواد، وفي حال أردنا ان نتعامل مع الشفرات البرمجية ففي هذه الحالة نحتاج أن ننتقل إلى الخيار التالي وهو console.

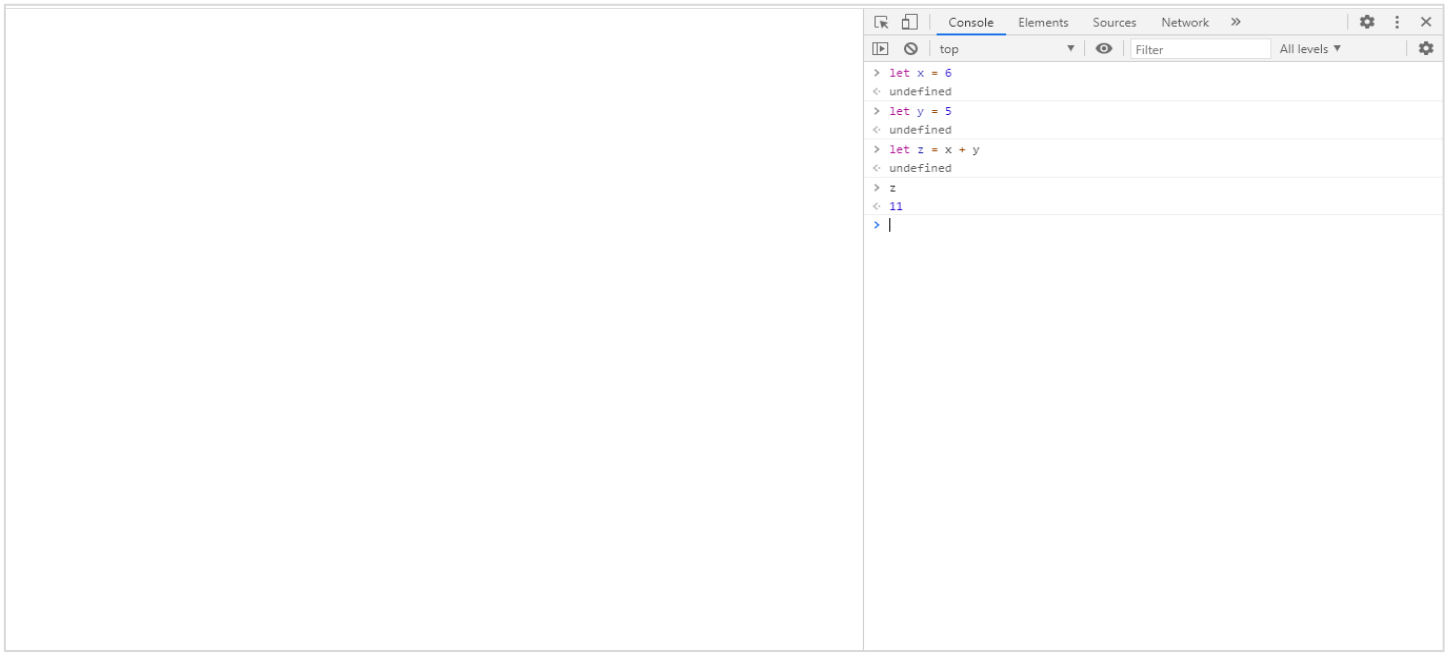
Console.4.3 :

كل مطور ويب Front End لا يستغني عن Console بأشكال متعددة منها معرفة قيمة متغير معين او متابعة خطأ ومعرفة سبب الخطأ، وغيره من الأمور التي يستخدمها المَطور لإظهار output معين، لأسباب خاصة به لمعرفة نتيجة جزء معين في الشفرة الخاصة به، وليس هذا فحسب فتستطيع عن طريق Console كتابة أي شفرة JavaScript تريدها من تعريف متغيرات إلى دوال واجراء عمليات حسابية ...الخ، نستطيع ان نقول بصيغة أخرى كانه محرر أكواد مصغر لي JavaScript.

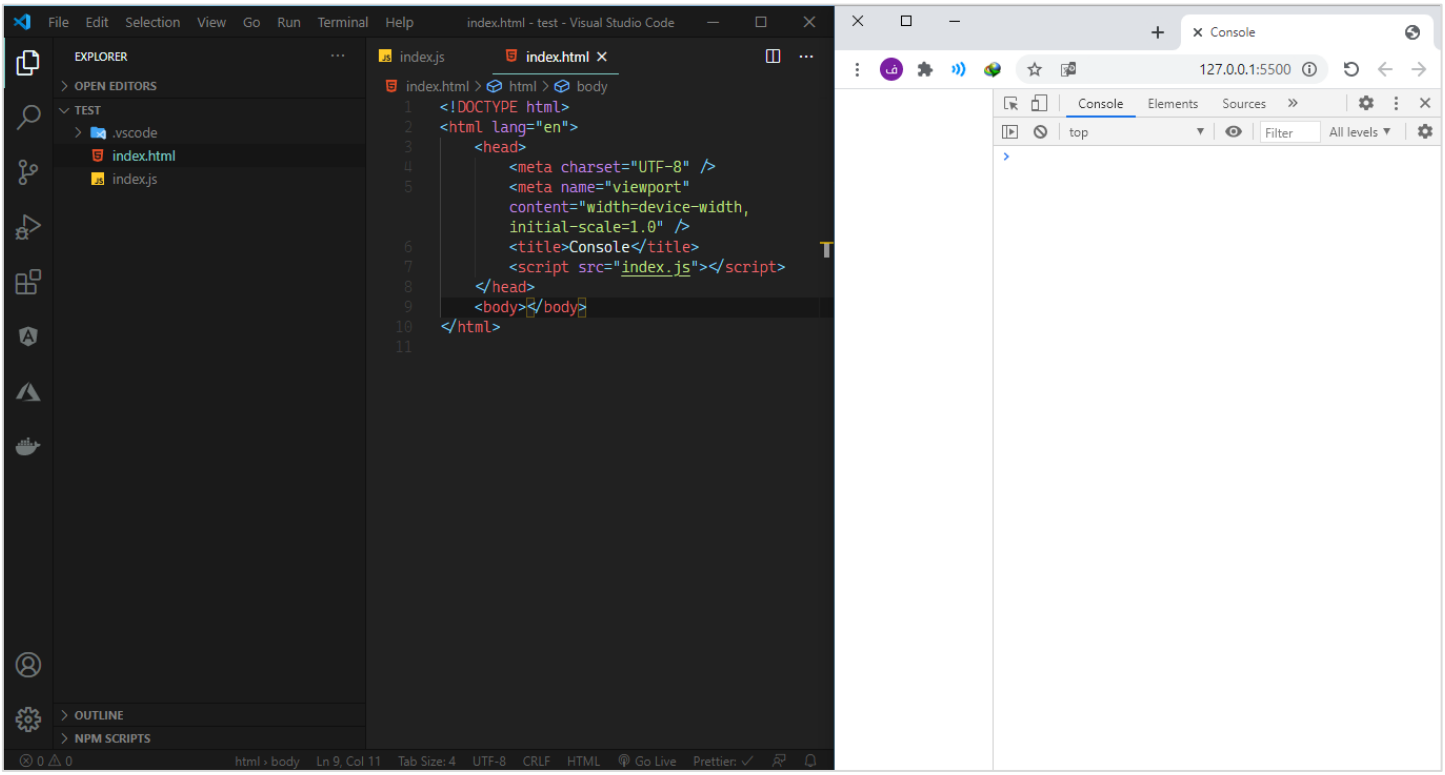
ونستطيع اختيار التبويب Console، في جزء أدوات المطور لكي تظهر لنا شاشة console، كالتالي:



وكما أشرنا سابقاً نستطيع عن طريق Console كتابة أي Logic في لغة JavaScript، كالتالي:

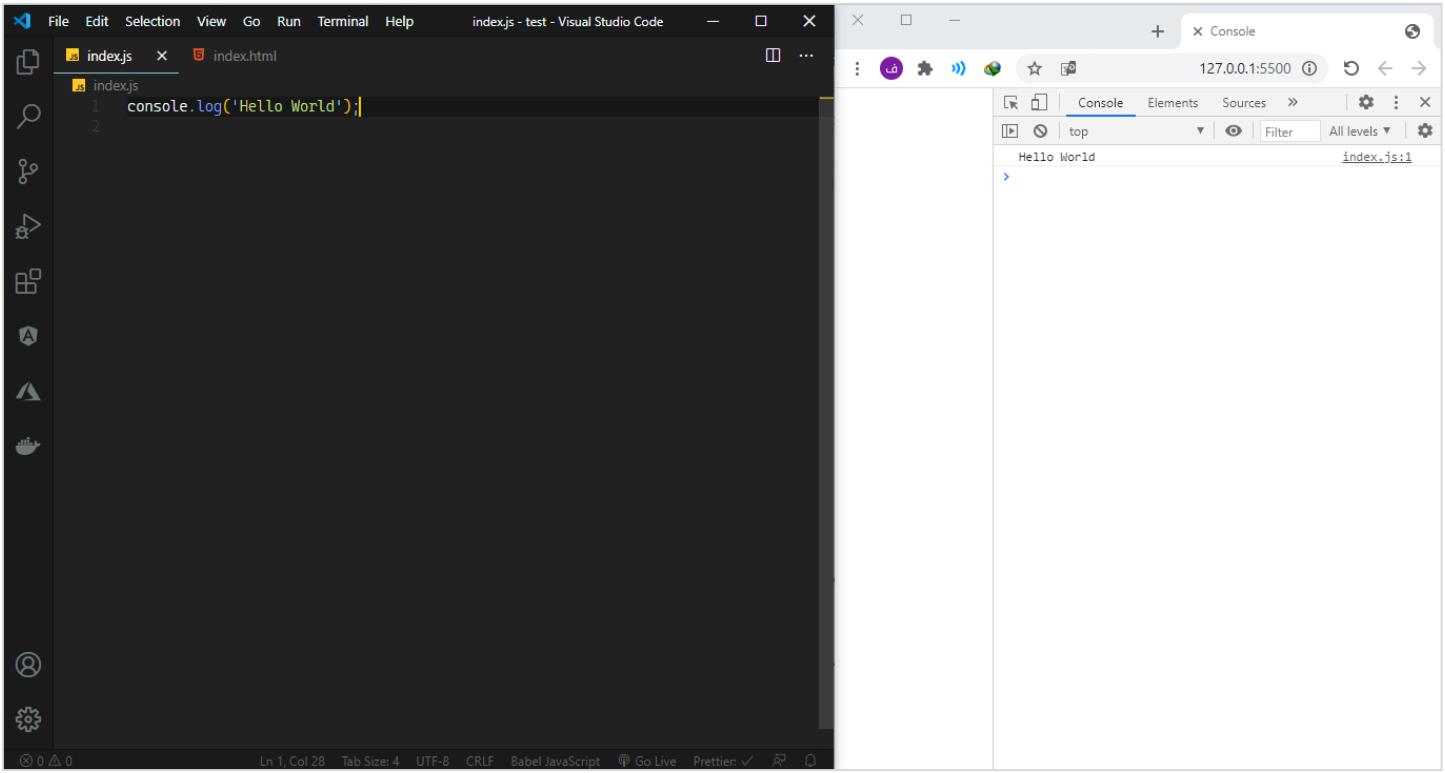


ولكن في الحقيقة هذا لا يهمنا كثيراً وخصوصاً في هذه السلسلة، والذي يهمنا هو كيف نستفيد من Console في طباعة مخرجات معينة لتجربة ومعرفة نتيجة المخرج لي Logic معين، لذلك قمت بإنشاء مجلد على سطح المكتب وانشأت بداخله ملفين الأول باسم index.html والثاني باسم index.js ومن ثم في محرر الأكواد VS Code قمت بفتح هذا المجلد عن طريق القائمة ملف ومن ثم اختيار الامر فتح مجلد أو open folder وأخيراً قمت باستدعاء ملف index.js في ملف index.html، وايضاً قمت باستخدام الإضافة Live Server لكي أقوم بتشغيل هذين الملفين في المتصفح في البورت 5500 والهوست 127.0.0.1، والفائدة من هذه الإضافة هو لاكتشاف أي تغيير يحدث بالكود عن طريق محرر الأكواد وعرضه مباشرة في المتصفح بدون الحاجة لإعادة تحديث الصفحة يدوياً مع كل تعديل في الكود، كالتالي:



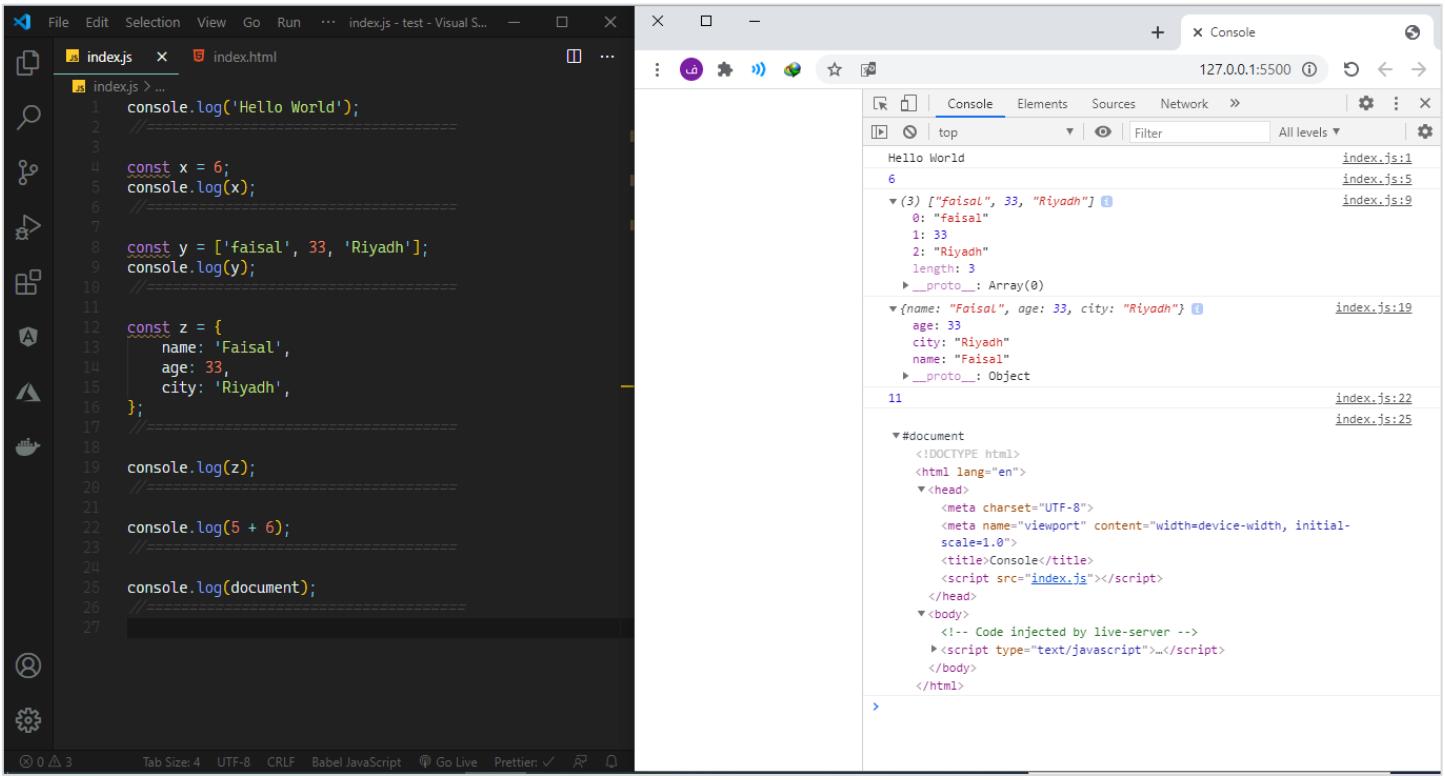
الشكل السابق هو عبارة عن نافذتين قمت بتغيير احجامهما لكي يتسنى لي مشاهدة نتيجة أي تعديل اجريه على الكود في محرر الأكواد في المتصفح مباشرة.

عموما لنقوم بفتح index.js، ونكتب السطر البرمجي التالي:

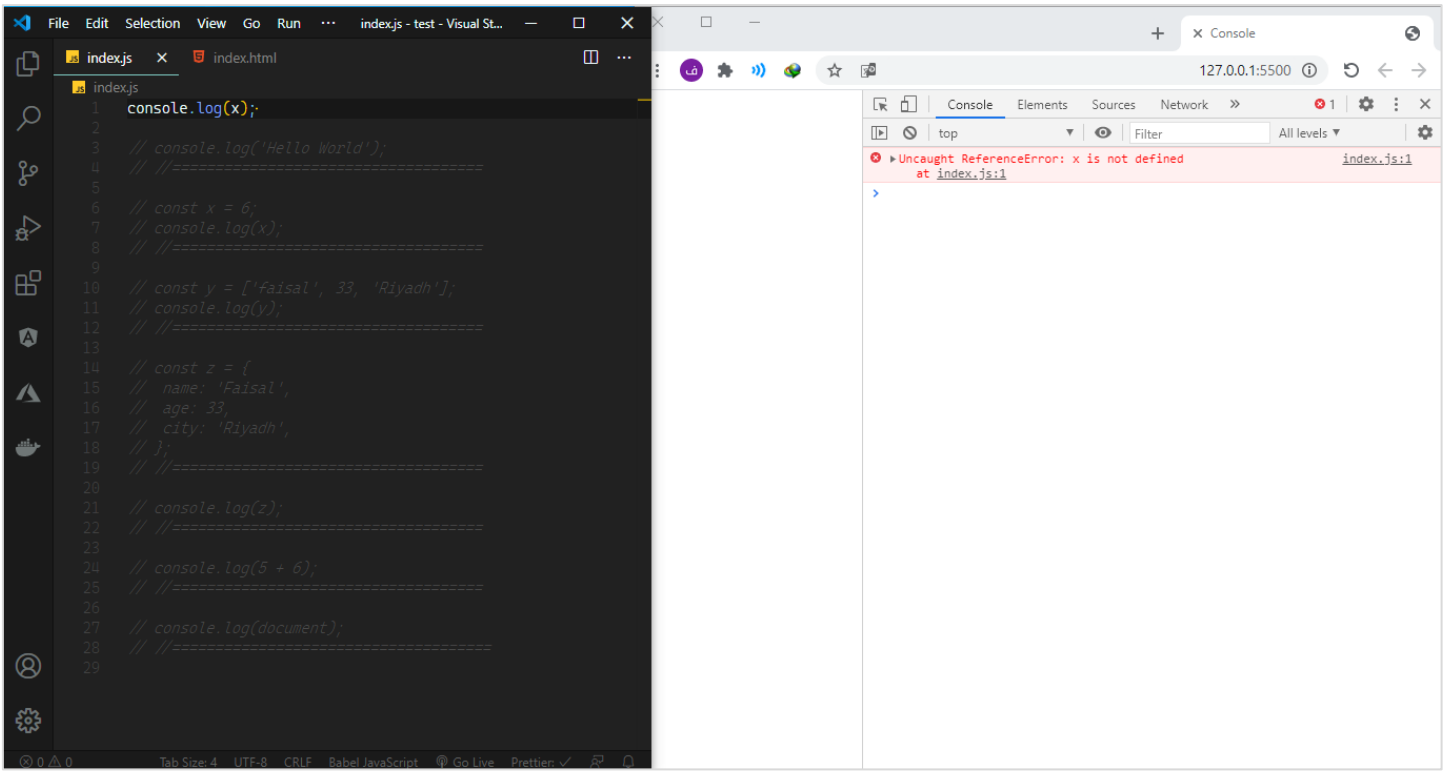


نلاحظ اننا إذا أردنا استخدام Console فلابد أن نستخدم الأمر console اما الدالة log فنستخدمها لطباعة قيمة معينة في Console، ومن هذا المنطلق قد يتبادر إلى ذهنك عزيزي المتعلم هل هنالك دوال أخرى موجودة في الأمر console والإجابة بالطبع هنالك دوال كثيرة ولكن في الحقيقة قليلة الاستخدام وأكثر ما يتم استخدامه هو log فقط.

ونستطيع الطباعة في Console كل شيء تقريباً بداية من نص إلى عملية حسابية بحيث يتم اجراء العملية وطباعة النتيجة فقط او مصفوفة او كائن او متغير عادي او حتى document نفسه لأنه في الأخير هو عبارة عن كائن.



وليس هذا فحسب فأيضاً إذا كان هنالك خطأ في الكود فسوف يظهر في Console، ولنفرض اننا نريد طباعة متغير وهذا المتغير لم يتم تعريفه، كالتالي:

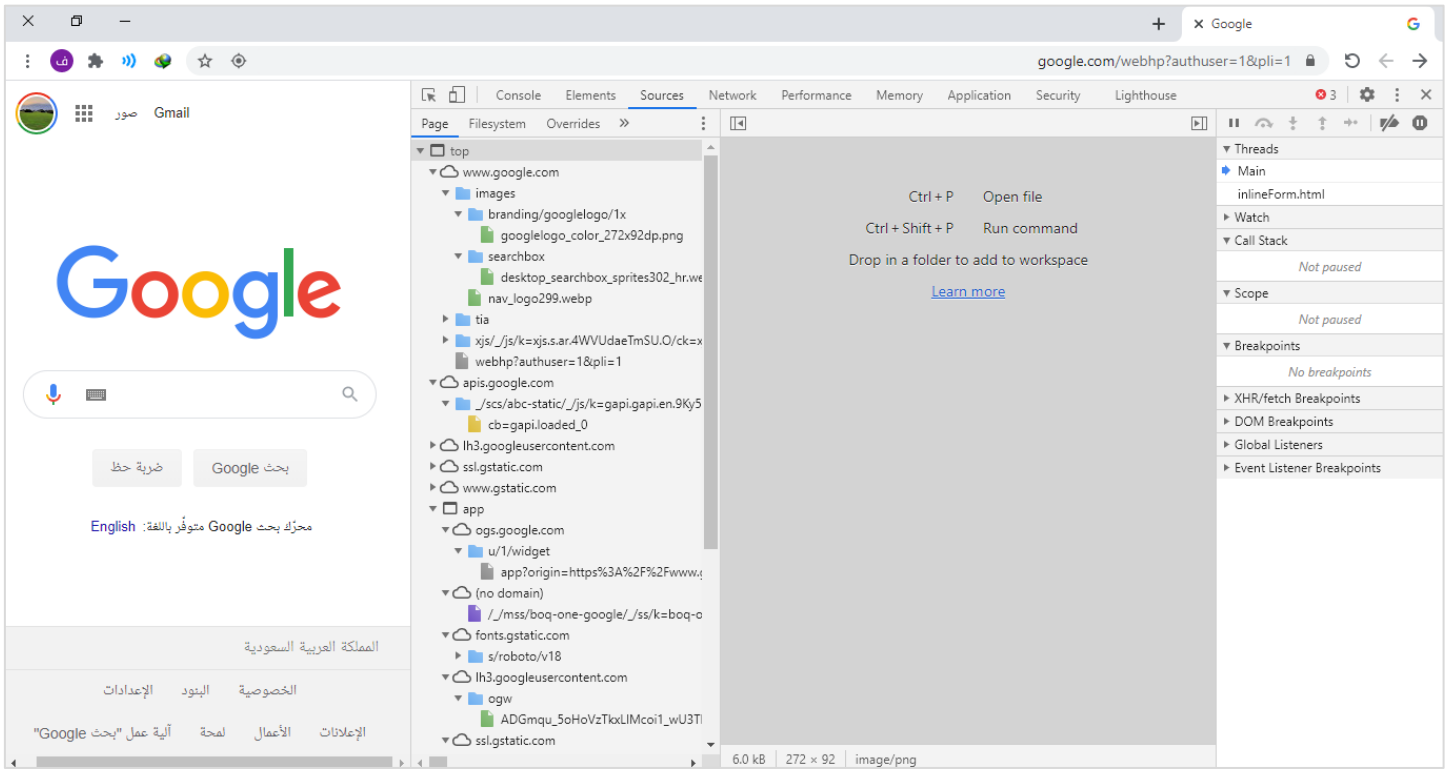


نلاحظ رسالة الخطأ التي ظهرت في Console والتي تُفيد بأن المتغير لم يتم تعريفه، ونكتفي في هذا الجزء في Console ولن أُسهب اكثر في استخداماته، لأنني اريد التركيز فقط على ما يخدمنا في هذه السلسلة، مع العلم اننا سوف نستخدمه كثيراً.

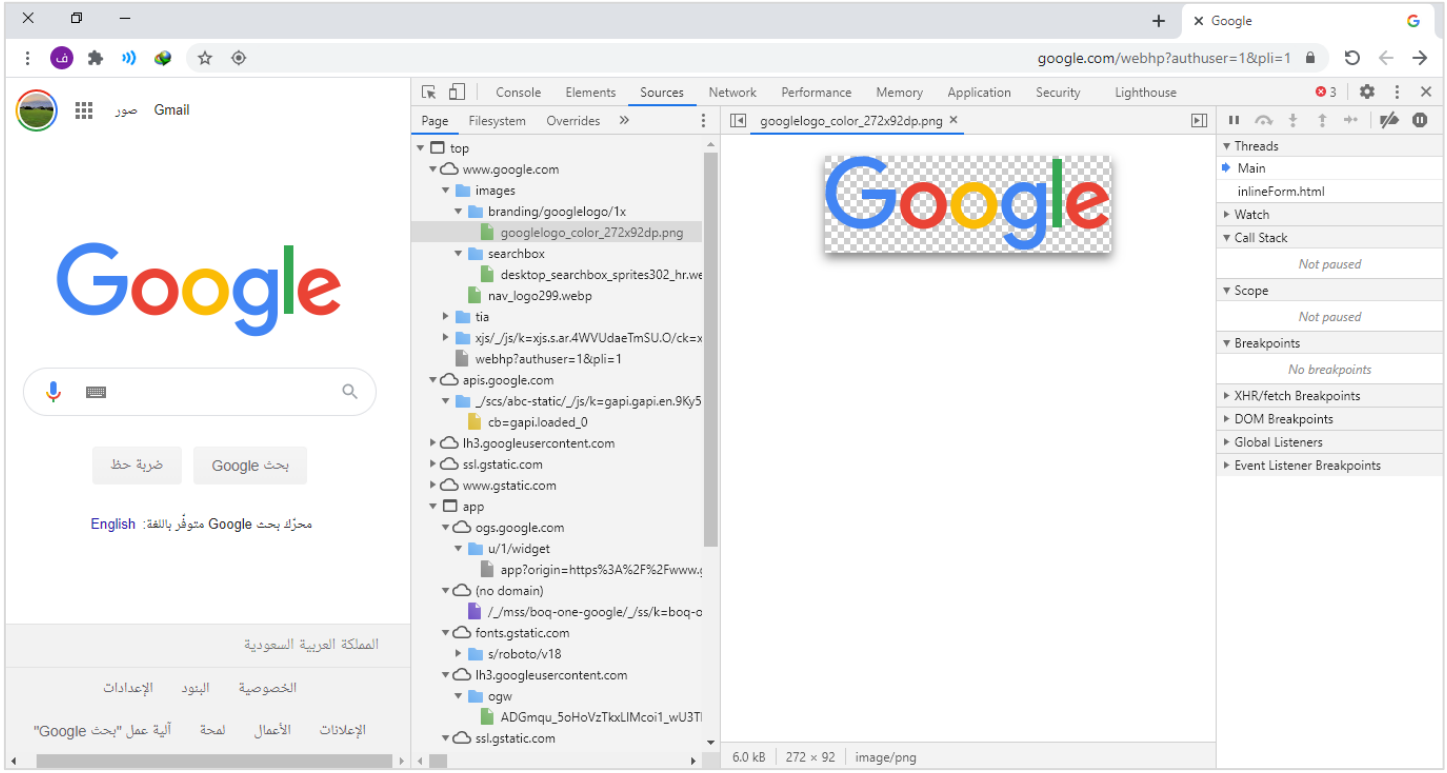
Sources.5.3 :

هذا الخيار يتيح لنا مشاهدة جميع ملفات التطبيق بالشفرة المصدرية الخاصة بها، بالإضافة يمكن استخدامه كمحرر أكواد نستخدمه لتحرير وكتابة الأكواد.

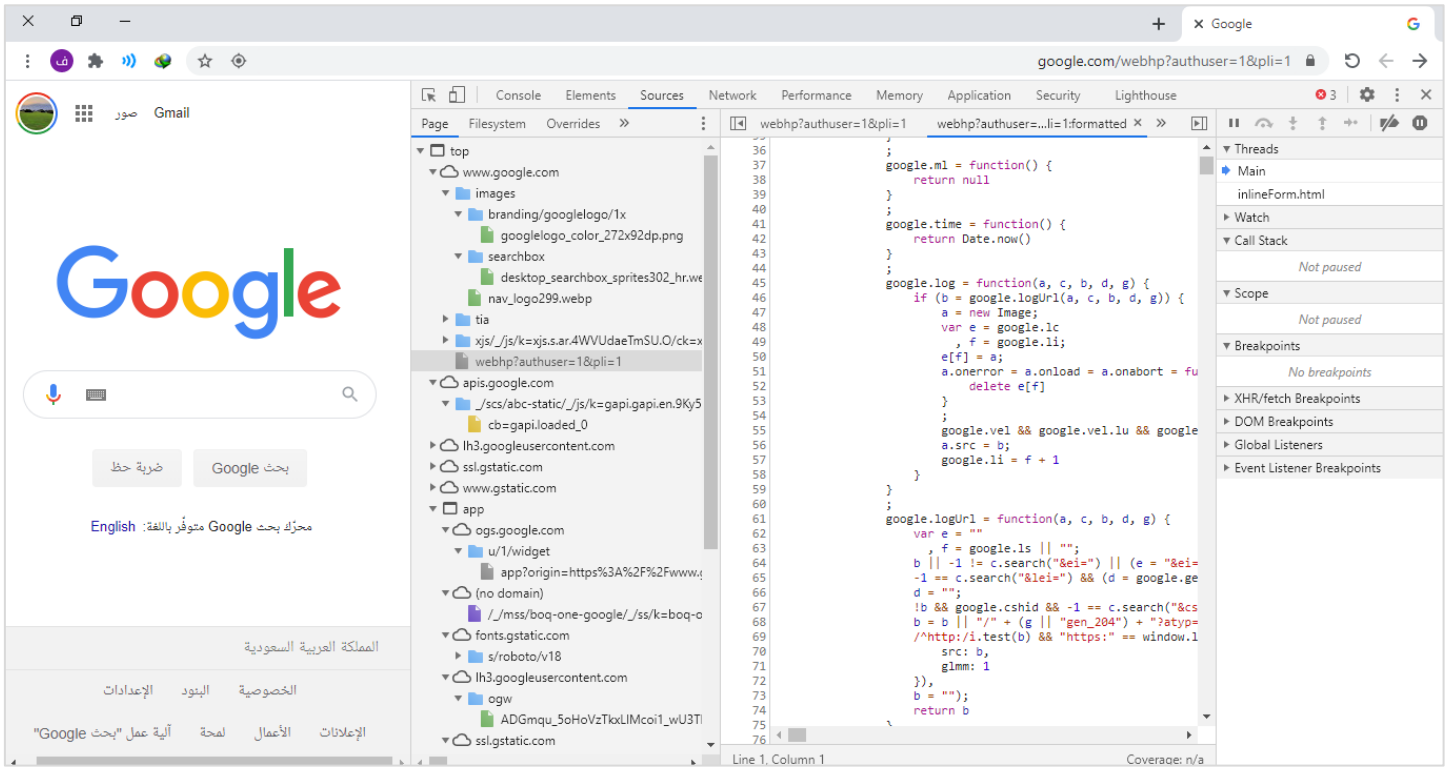
مثلاً نستطيع ان نذهب إلى موقع google وفتح أدوات المطور واستعراض ملفات الموقع، كالتالي:



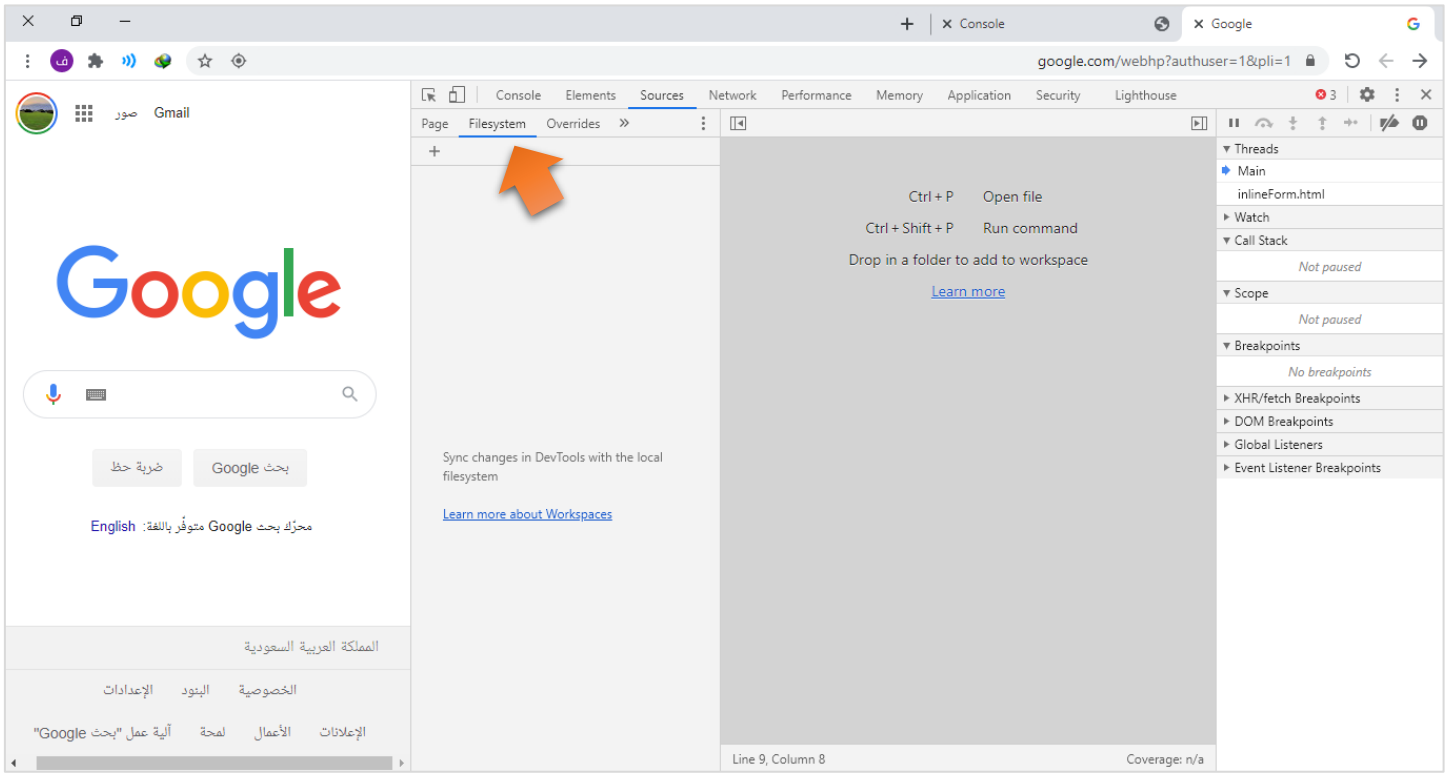
فنستطيع مثلاً استعراض الصور الخاصة بالموقع بشرط ان تكون static، كالتالي:



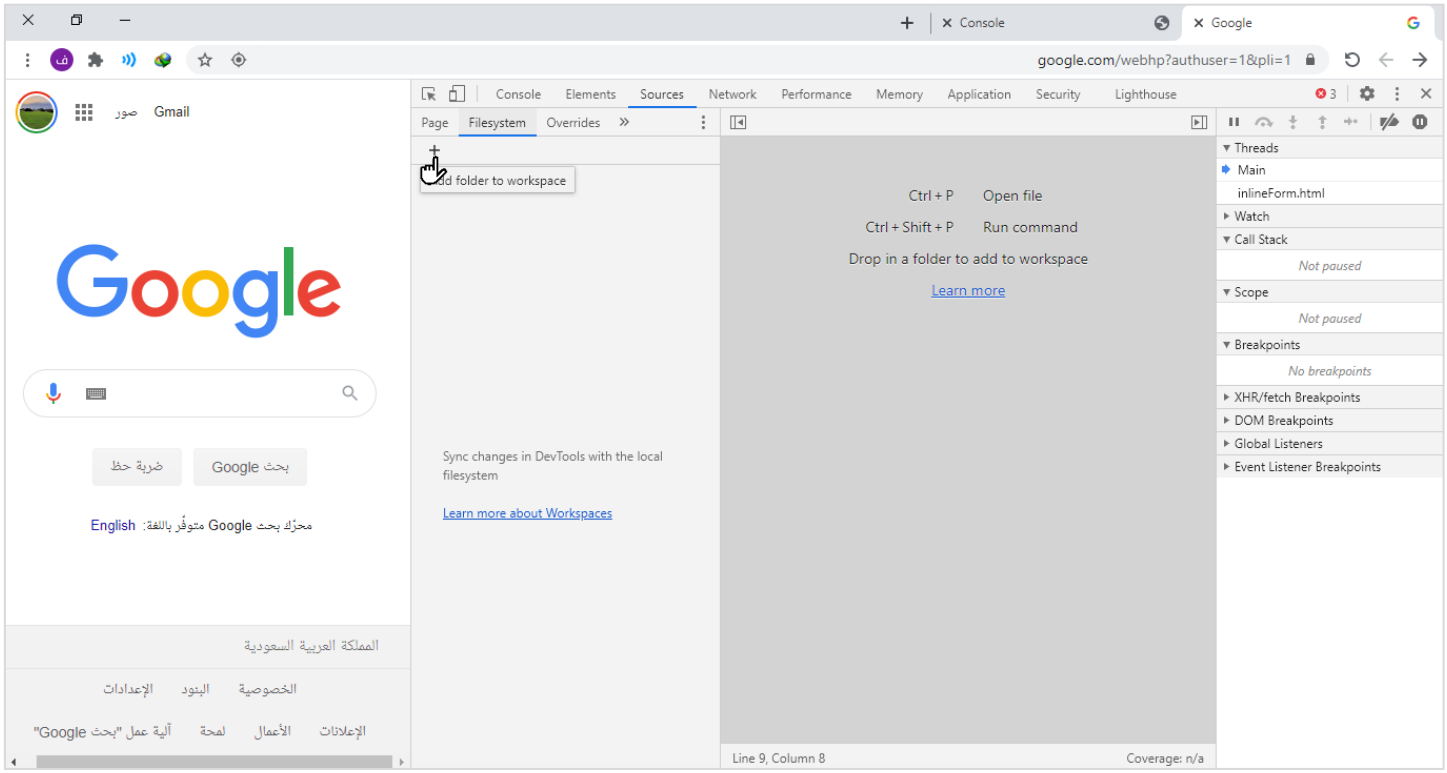
كما نستطيع استعراض الشفرات المصدرية للموقع، كالتالي:



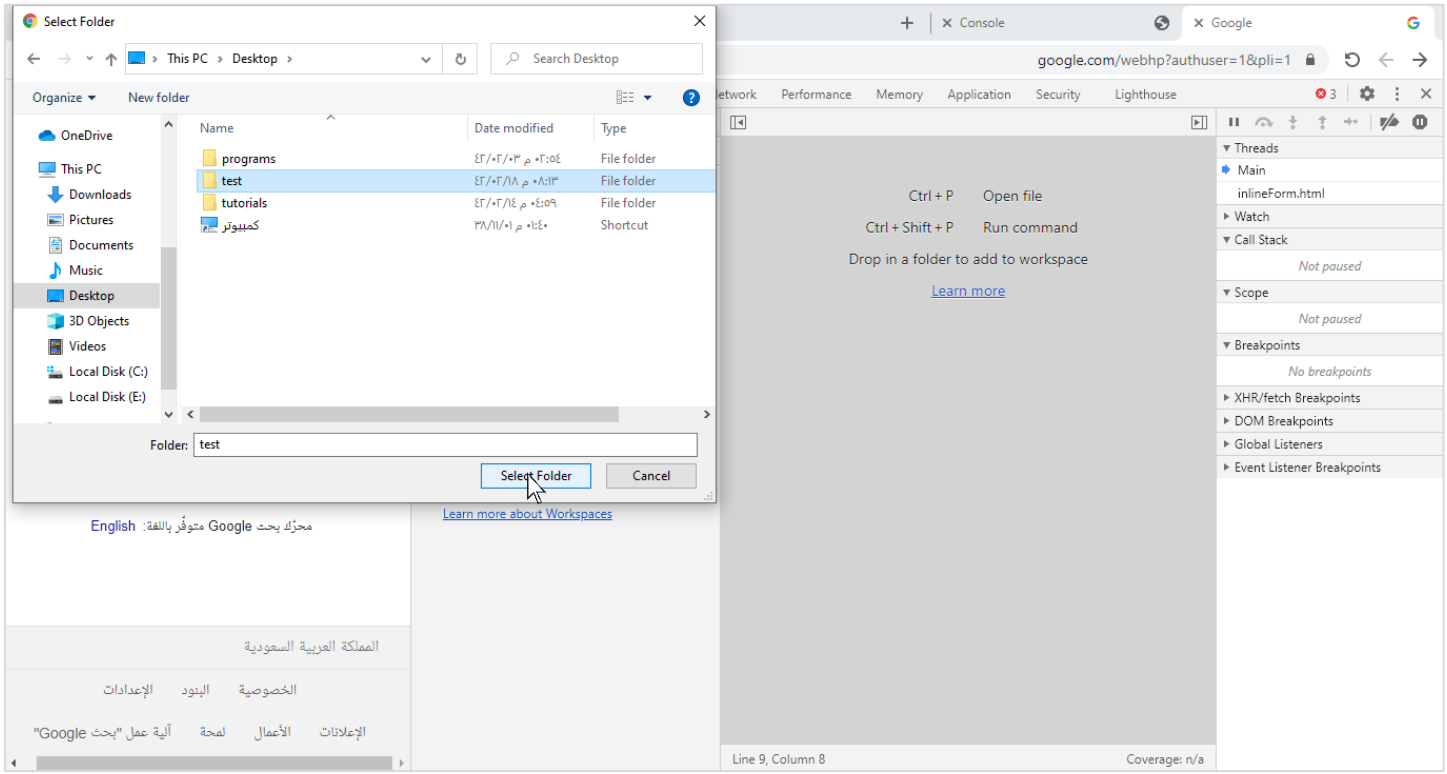
وكما أشرنا سابقاً نستطيع الاستفادة من هذا الخيار كمحرر أكواد لذلك سوف نختار Filesystem، كالتالي:



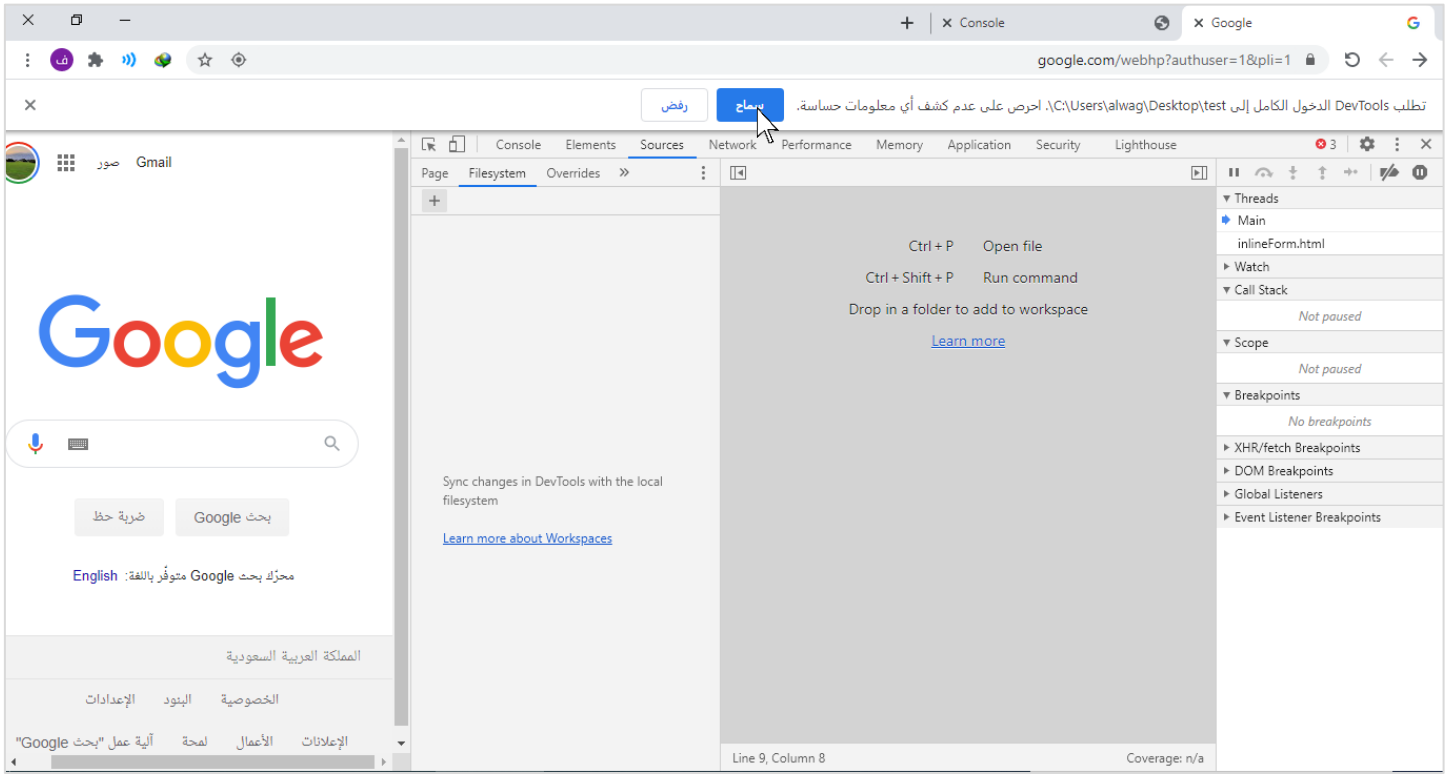
ومن ثم نختار علامة + لإضافة مجلد جديد، مع العلم ان المجلد الذي سوف نختاره هو نفس المجلد test الذي قمنا بإنشائه سابقاً والذي يحتوي على ملفين هما index.html وindex.js، كالتالي:



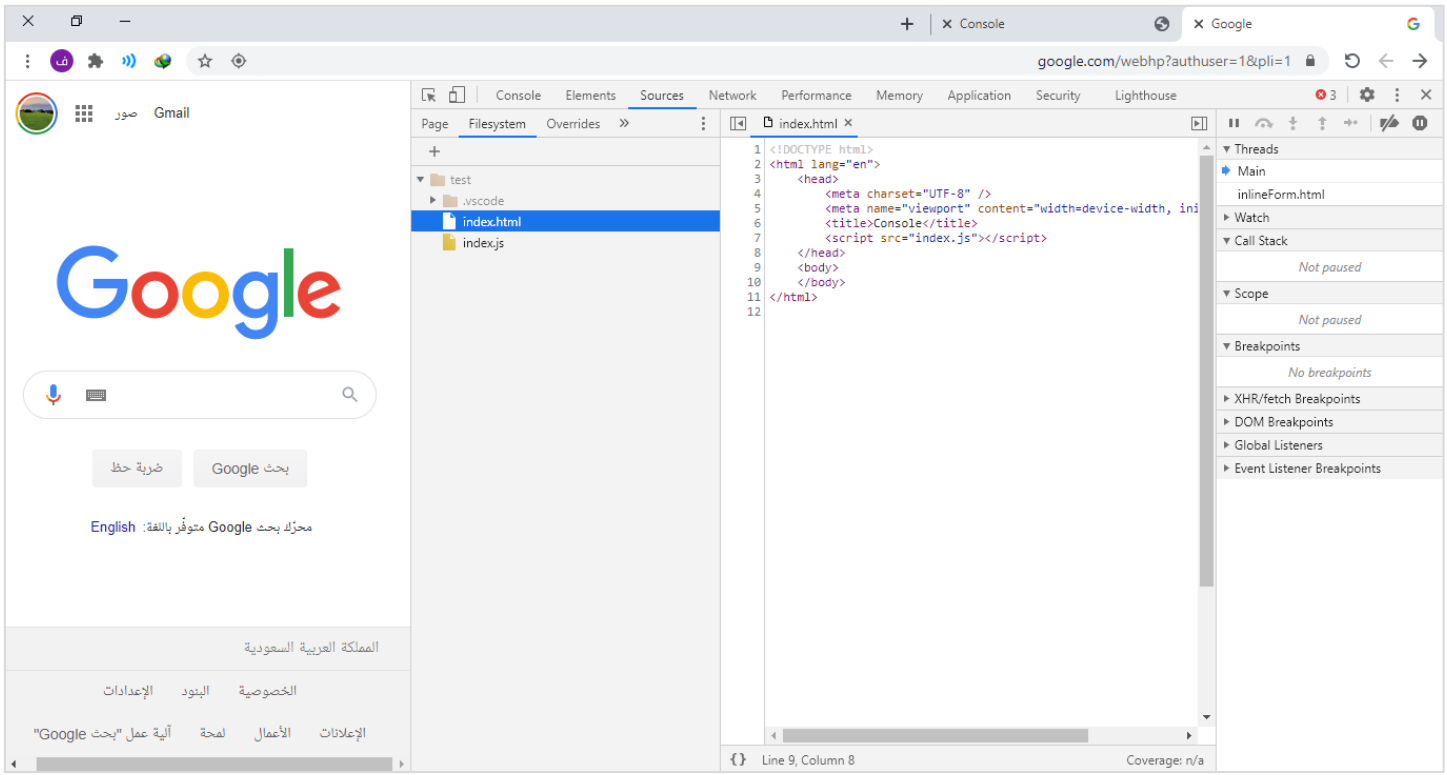
عند الضغط على هذا الزر سوف تظهر لنا شاشة نختار منها المجلد test، كالتالي:



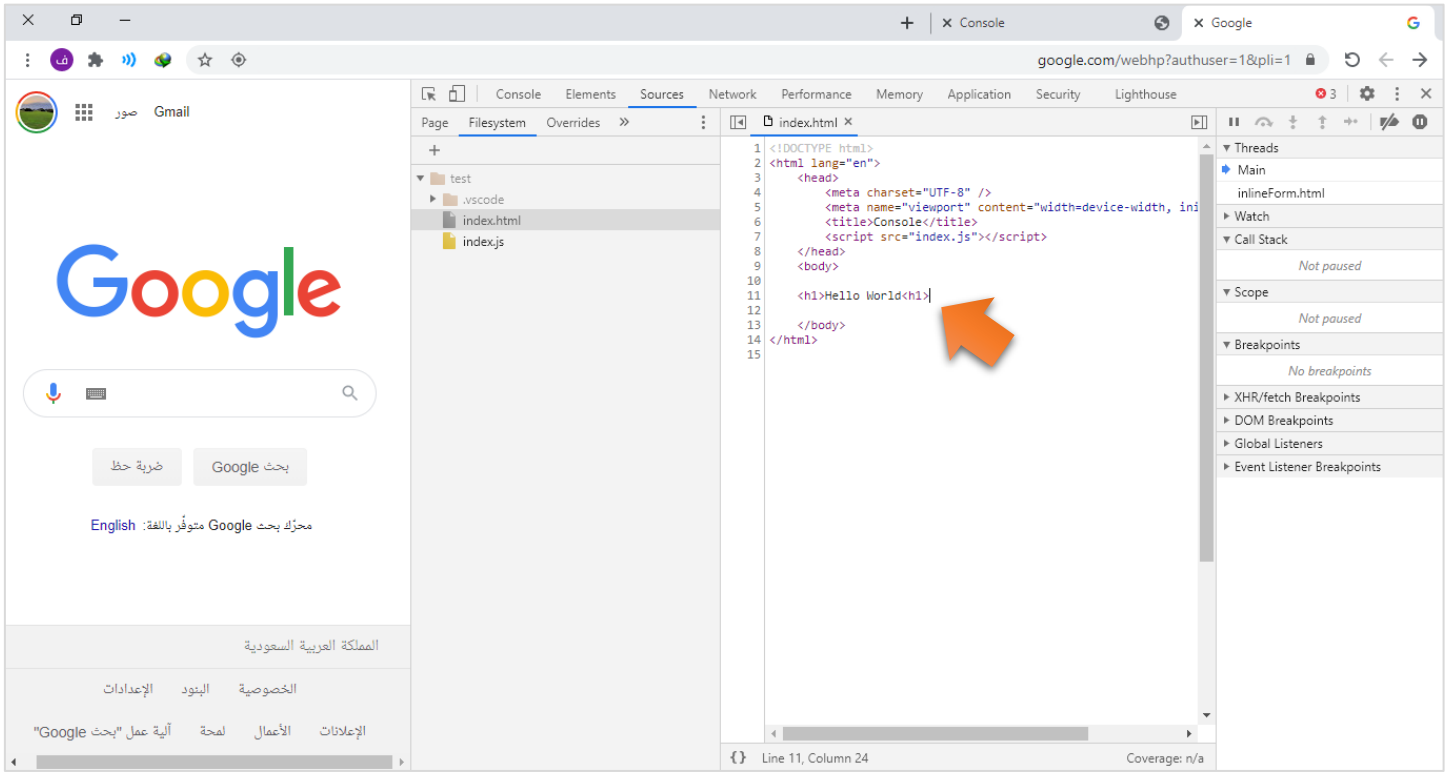
وبعد الضغط على زر Select Folder سوف تظهر لك رسالة تطلب منك السماح بالمتصفح بقراءة وتعديل هذه الملفات، كالتالي:



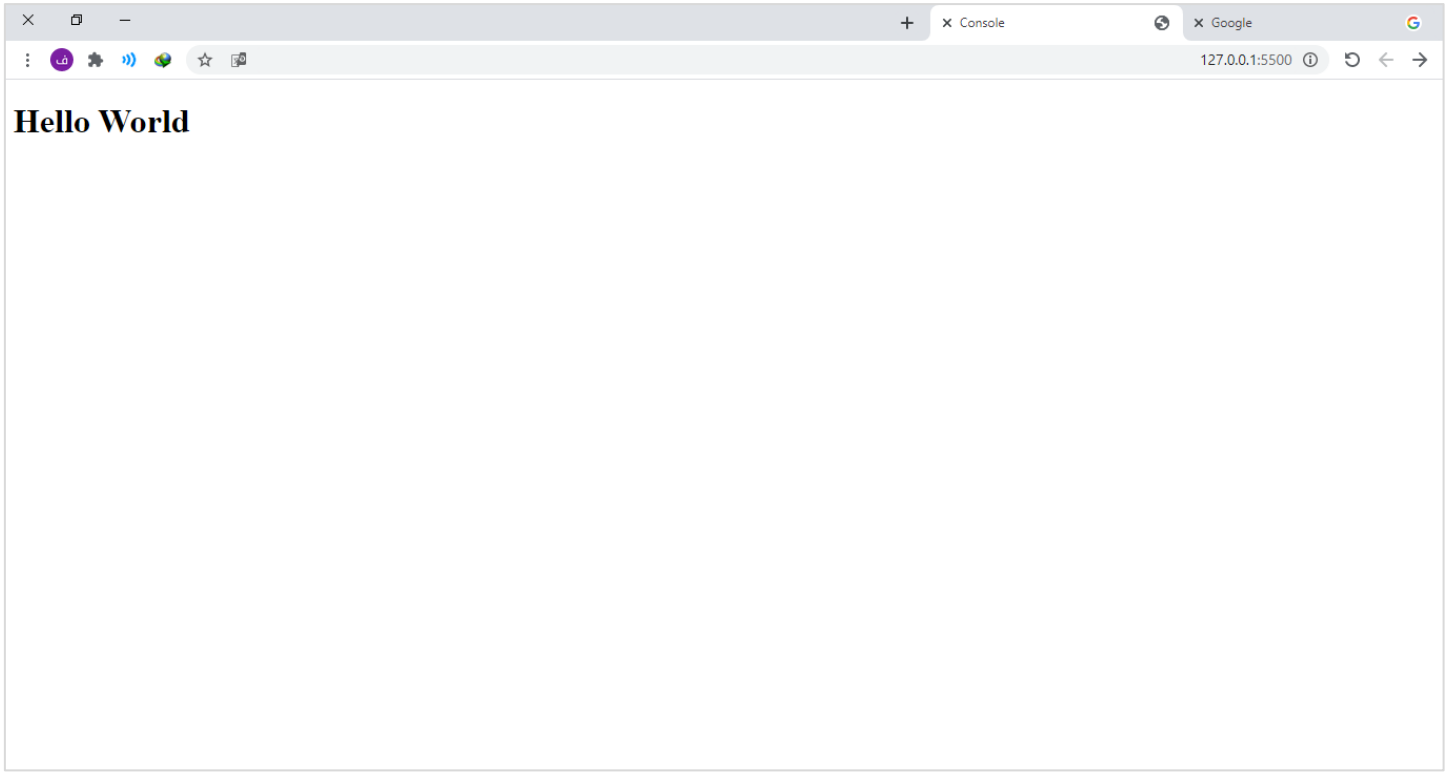
بعد الضغط على زر سماح سوف يتم ادراج هذا المجلد في Sources، كالتالي:



الآن لنقم بإجراء تعديل بسيط ومن ثم الضغط على الاختصار Ctrl + Save للحفظ، كالتالي:



والآن لنقوم بالذهاب إلى هذه الصفحة في المتصفح عن طريق نفس البورت 5500 والهوست 127.0.0.1 الخاص بالإضافة Live Server لكي نشاهد النتيجة ويجب التأكد من أن Live Server يشتغل، كالتالي:

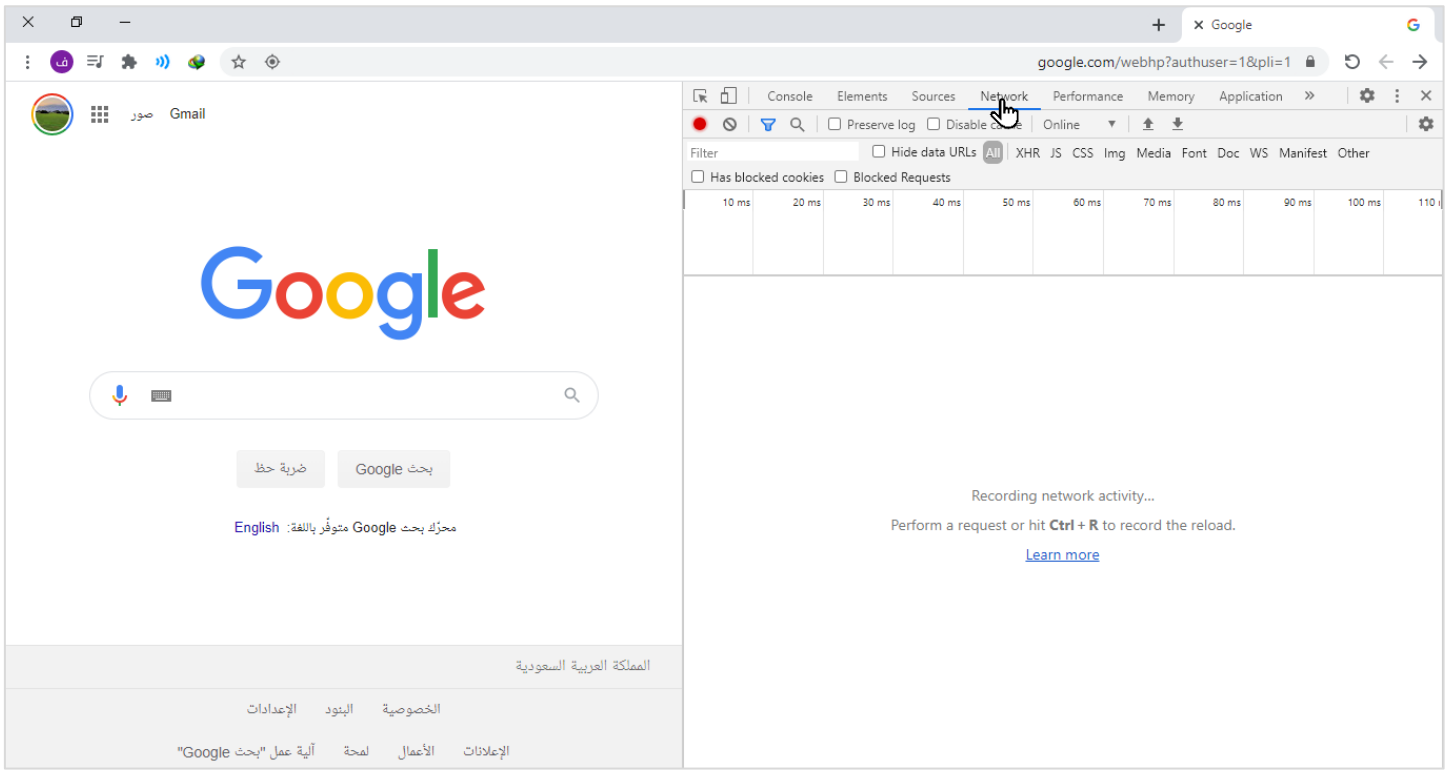


نلاحظ انه تم تعديل الكود كما هو ظاهر في النتيجة، ولو قمنا بعمل تحديث لصفحة فإن التعديل لن يختفي كما كان الحال في Elements.

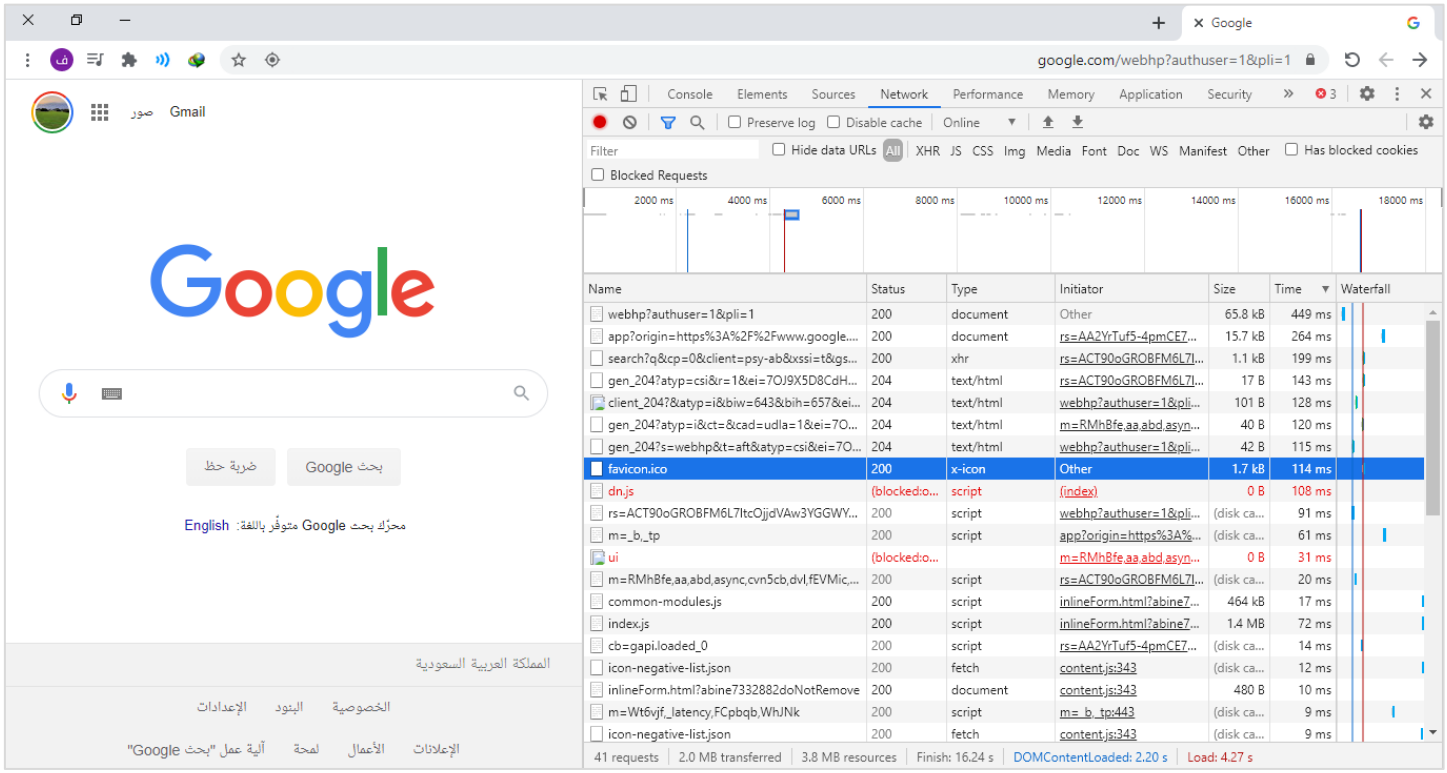
Network.6.3 :

في هذا الخيار نستطيع معرفة معلومات عن ملفات تطبيق الويب او الموقع من ناحية عدد الملفات التي تم تحميلها في بداية تشغيل هذا التطبيق وحجم كل ملف والوقت الذي قضاه لتحميل هذا الملف من server وعرضه في المتصفح، بالإضافة إلى بعض المعلومات الأخرى مثل حالة status لهذا الملف هل تم الرد من server وجلب هذا الملف بنجاح والتي تمثل بالرقم 200 ام لم يتم إيجاد الملف المطلوب والتي تمثل بالرقم 404 (هذه الأرقام تسمى HTTP Status Code وتستطيع الاطلاع أكثر من خلال هذا الرابط https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)، بالإضافة نستطيع الاطلاع على Response القادم من Server سواء كانت بيانات على شكل Json كما في xhr او ملف بأي شكل كان، وسوف نتعامل مع هذه الخيار بإذن الله في كتاب Angular Routing وبالتحديد في الجزء الخاص Lazy Loading.

ونستطيع الذهاب مثلاً إلى موقع google.com والذهاب إلى خيار Network في أدوات المطور، كالتالي:

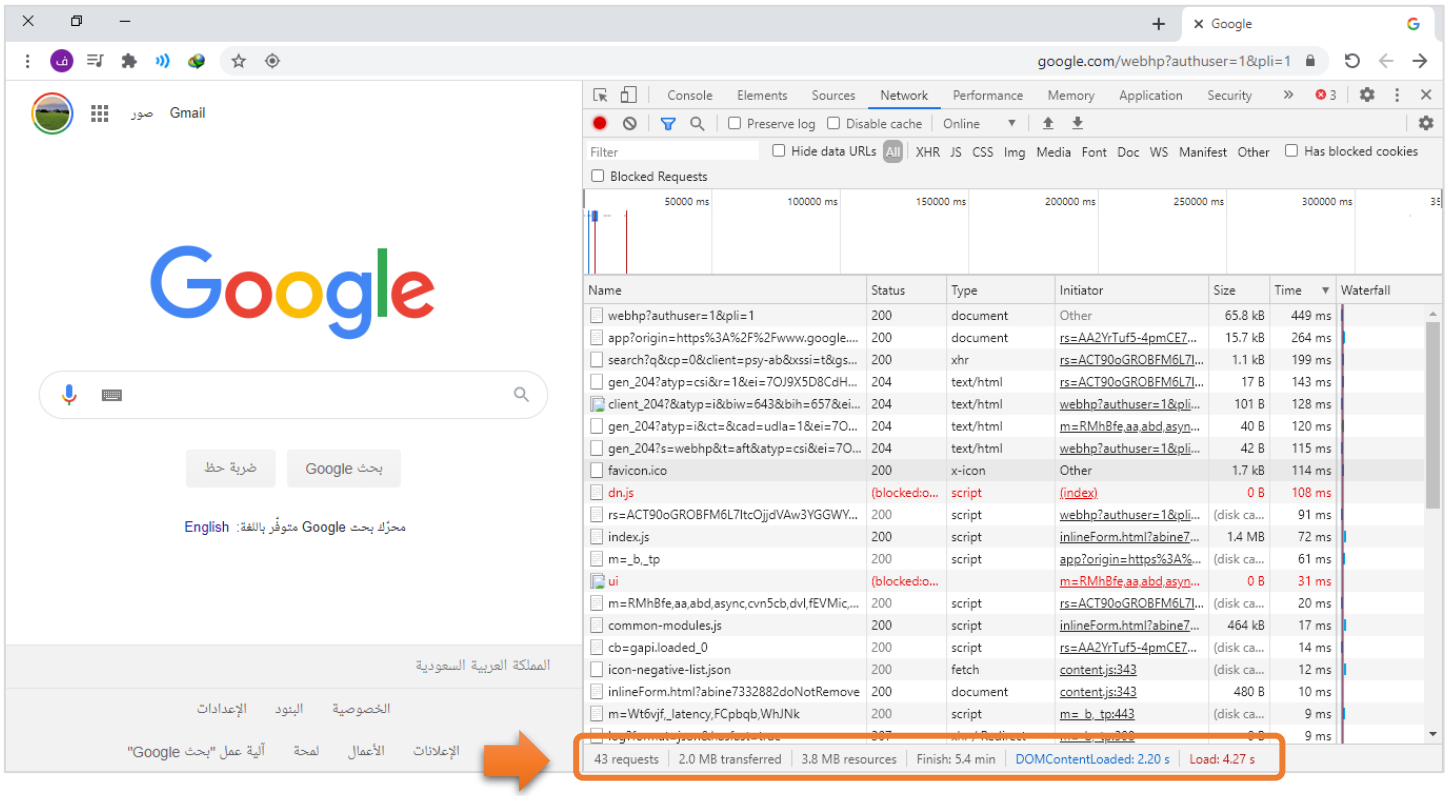


في حال عدم وجود الملفات قم بإعادة تحميل الصفحة، كالتالي:



نلاحظ ظهرت لنا جميع الملفات وكل ملف يحتوي على مجموعة من المعلومات مثل اسم هذا الملف وstatus والحجم والوقت Time الذي قضاه server لجلب هذا الملف وعرضه في server.

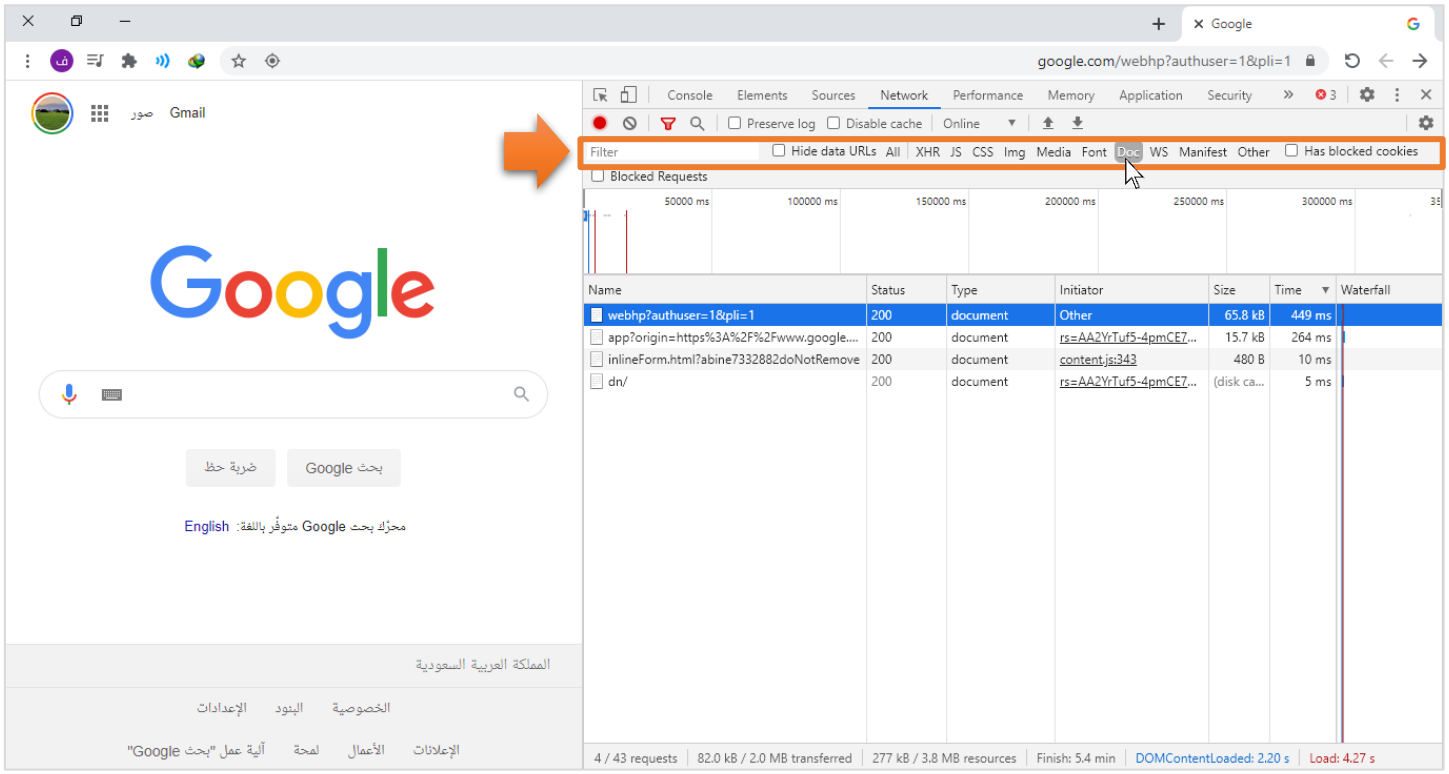
وايضاً نستطيع القاء نظرة عامة على سرعة التطبيق الخاص بنا وأداءه من خلال شريط status في الأسفل، كالتالي:



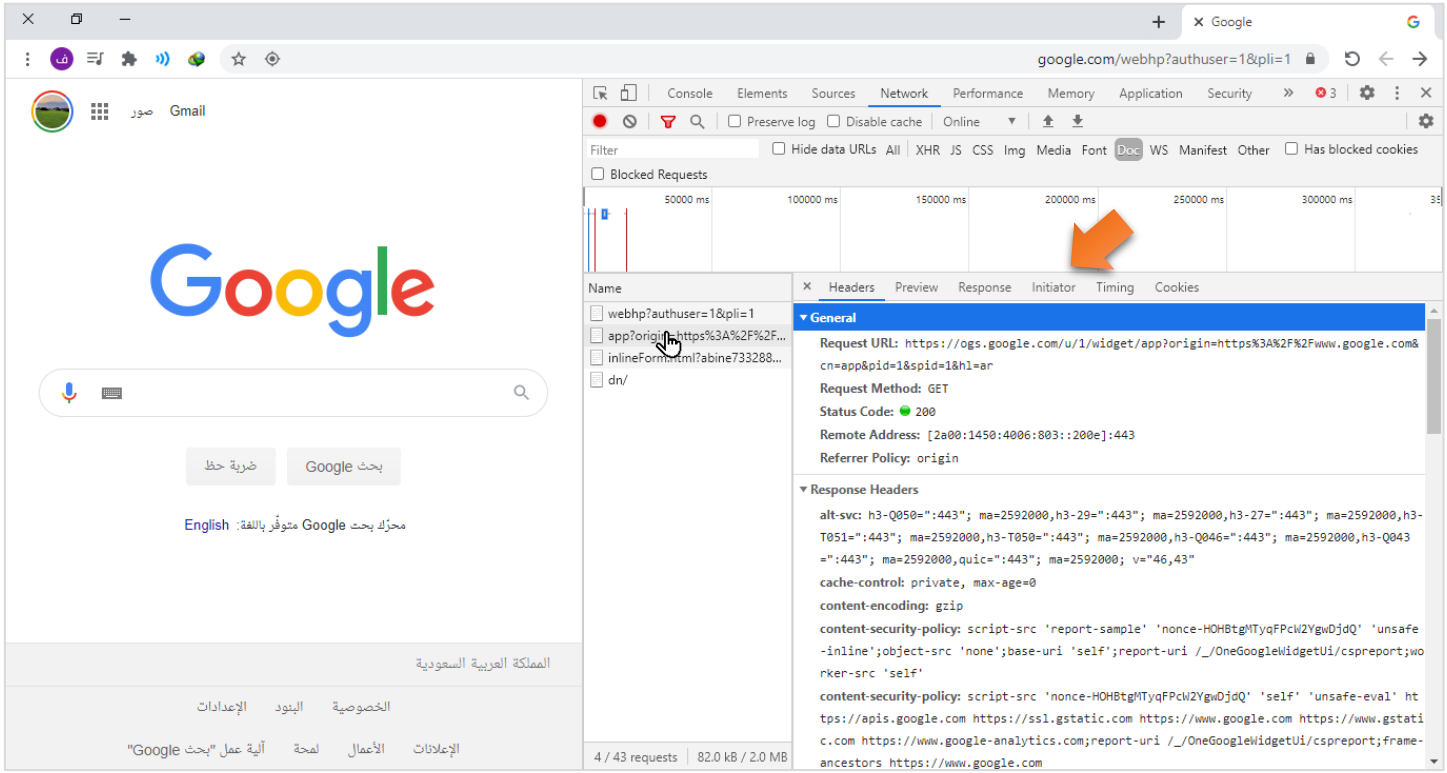
نستطيع معرفة كم طلب أرسل من المتصفح لي server وحجم الملفات المصدرية التي تم تحميلها عند بداية تشغيل هذا الموقع بالإضافة إلى الوقت الذي احتاجه المتصفح لتشغيل الموقع بالكامل، وبناءً على هذه المعلومات والملفات السابقة نستطيع الرفع

من أداء التطبيق الخاص بنا، على سبيل المثال بالاستغناء عن الملفات التي ليس من الضروري تحميلها عند بدأ بداية التشغيل واستدعائها عند الحاجة إليها وهو ما سوف نعمله عند كلامنا عن Lazy Loading بإذن الله.

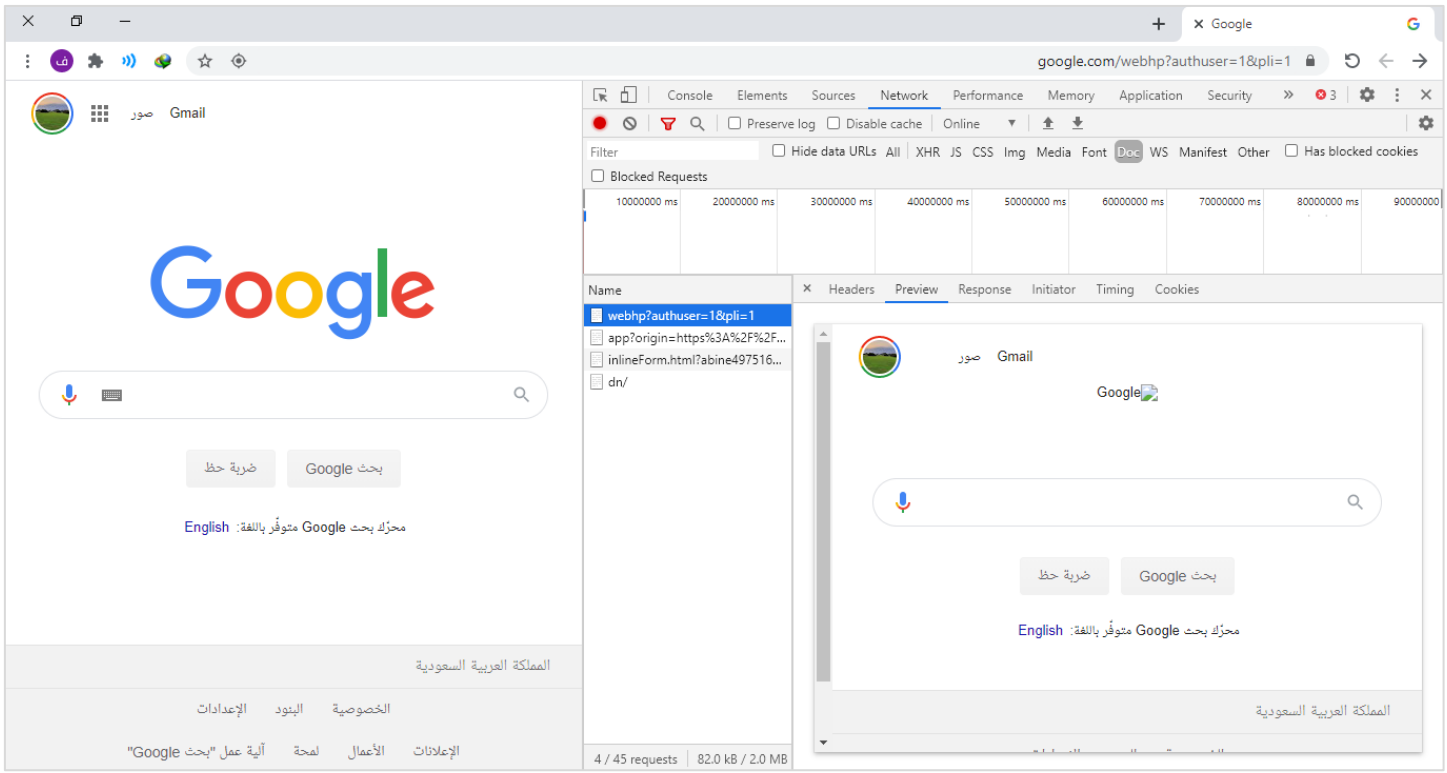
كما نستطيع عمل فلترة لهذه الملفات بحيث نُريد مثلاً اظهار ملفات معينة فقط، كالتالي:



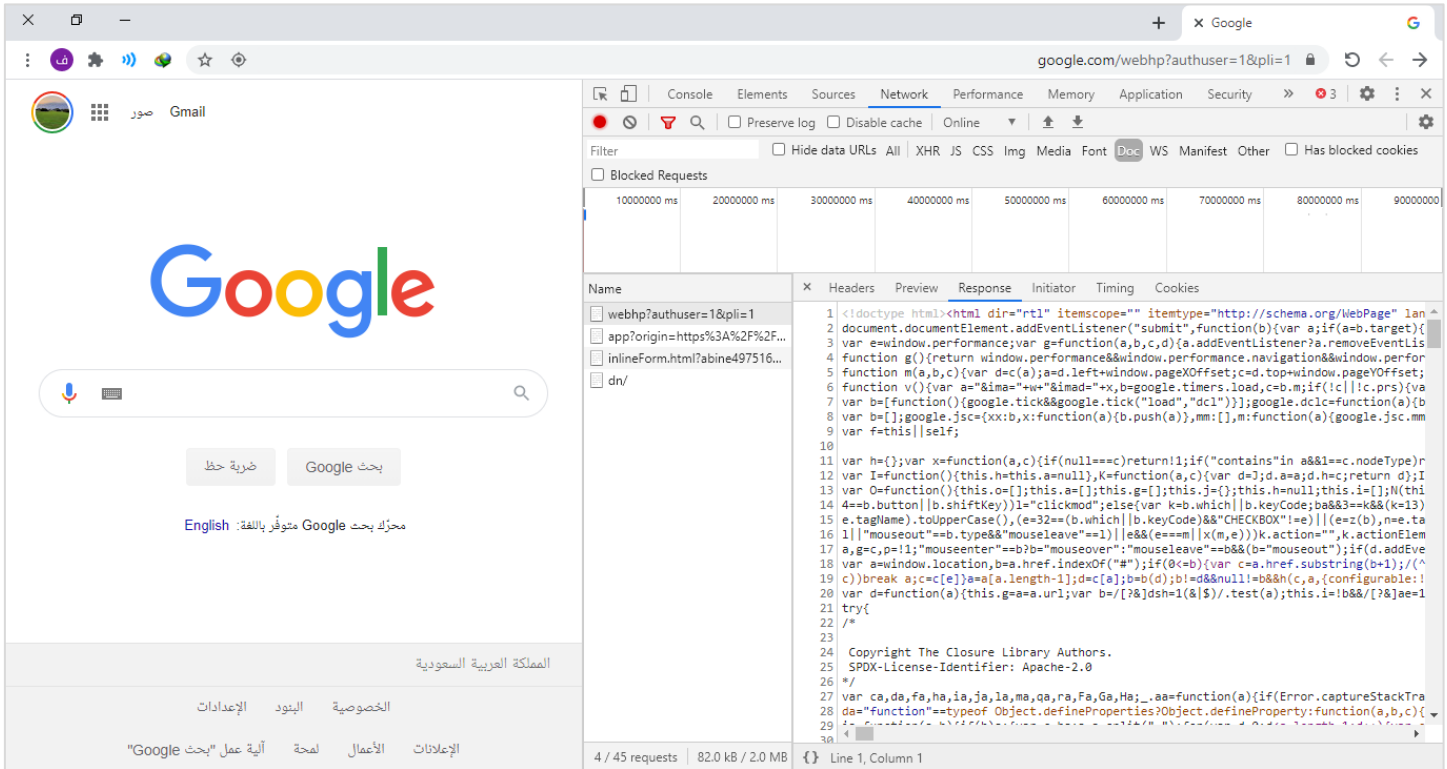
نلاحظ اخترنا Doc وتم عرض ملفات Document وهي ملفات HTML فقط، لنقم باختيار أحدها بالضغط عليه لمعرفة بعض تفاصيله، كالتالي:



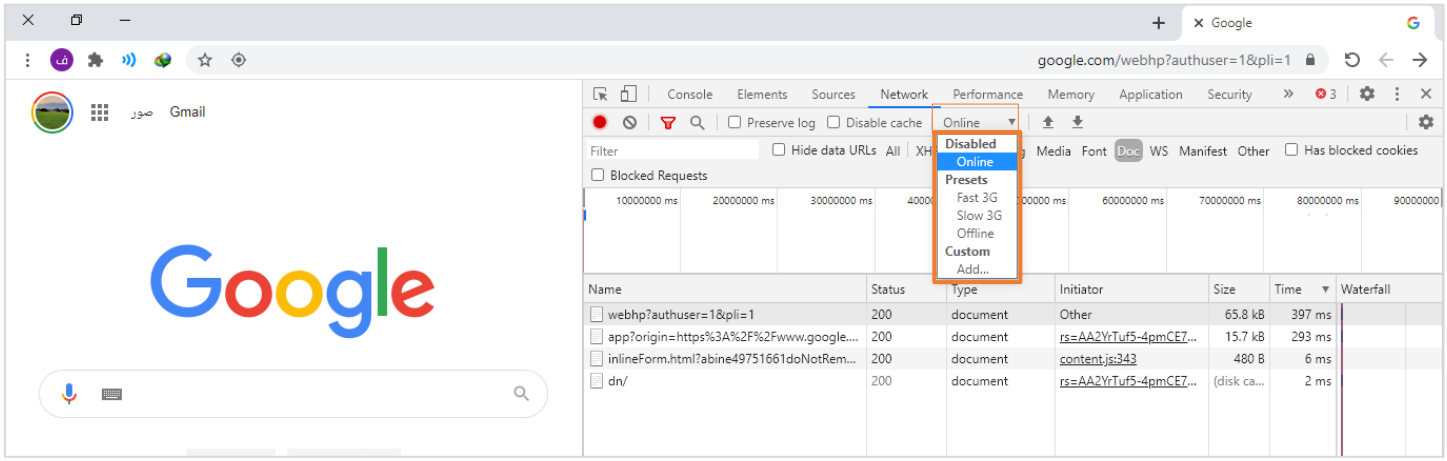
نلاحظ ظهر لنا شريط بالأعلى يحتوي على مجموعة من التبويبات بشكل افتراضي سوف يتم تفعيل او تبويب وهو Headers حيث هذا التبويب يظهر بعض التفاصيل الخاصة عن الاتصال بالشبكة وCode Status بالإضافة إلى Response القادم من Server في Header، اما الثاني فهو Preview وهو عبارة عن معاينة لي Response القادم من Server، كالتالي:



والتبويب الثالث وهو Response وهو عبارة الكود المصدري القادم من Server في حال كان ملف او البيانات على شكل Json في حال كانت Response عبارة عن بيانات Data او بصيغة أخرى xhr، كالتالي:



وأخيراً يجب أن أُشير إلى قائمة مهمة في هذا الخيار Network، كالتالي:



حيث عن طريق هذه القائمة المنسدلة نستطيع تجربة التطبيق عن طريق محاكاة الاتصال بشبكة الانترنت في بيئات مختلفة من online وهو بحسب سرعة الاتصال لديك، او محاكاة الاتصال بالشبكة بسرعة بطيئة او قطع الاتصال بالشبكة.

Application.7.3 :

وهذا الخيار يحتوي على مجموعة من الخيارات للمطور لتخزين البيانات داخل المتصفح، ومنها:

١) Local Storage: ويتم فيها تخزين البيانات بشكل دائم في المتصفح مالم يتم حذف هذه البيانات سواء من قبل مستخدم التطبيق او عن طريق المطور نفسه في حال عدم الرغبة في هذه البيانات، ولها استخدامات متعددة منها على سبيل المثال حفظ بيانات تسجيل الدخول للمستخدم بحيث لا يحتاج في كل مرة يزور فيها الموقع او التطبيق لإجراء عملية تسجيل الدخول، مع العلم ان اقصى حجم للبيانات التي يُسمح بتخزينها هي 250 ميغابايت، وقد تختلف من متصفح إلى آخر.

٢) Session Storage: وهي تشابه Local Storage في جميع الدوال وايضاً في أقصى حجم للبيانات المُخزنة ولكن الفرق انه هنا يتم فيها تخزين البيانات بشكل مؤقت بحيث يتم حذف هذه البيانات بمجرد اغلاق المتصفح او انتهاء الوقت المحدد للجلسة Session، او قيام المستخدم بحذفها يدوياً من المتصفح، ومن استخداماتها على سبيل المثال في حال كان المستخدم قام بفتح أكثر من تبويب في تطبيق الويب الخاص بك عزيزي المتعلم، وتريد ان تشارك بعض البيانات في جميع هذه التبويبات ولكن لا تُريدها ان تبقى في المتصفح وانما في حال اغلاق التطبيق يتم حذف هذه البيانات.

٣) IndexedDB: وتُسمى قاعدة بيانات الجافا سكربت وتعتمد تقنية قواعد البيانات الغير علائقية NoSQL ويتم التخزين في المتصفح الخاص بمستخدم التطبيق ولكن يستطيع المستخدم حذفها في حال الرغبة بذلك، لذلك يجب الحذر عند استخدام هذا النوع من قواعد البيانات.

٤) Web SQL: وهي مشابهه IndexedDB ولكن هنا الفرق انها تعتمد على تقنية قواعد البيانات العلائقية SQL.

ويجب الاشارة أننا هنا لسنا بصدد كيفية تعامل JavaScript مع جميع هذه التقنيات، وانما ما يهمنا هو ان تعرف عزيزي المتعلم هذه الأنواع والفائدة منها نظرياً بحيث إذا احتجنا إلى أي منها في أي جزء من أجزاء هذه السلسلة سوف نتطرق إلى كيفية استخدامها عملياً مع Angular دون الخوض بالتفاصيل النظرية التي يفترض أنك أصبحت ملم بها.

الفصل الرابع

Create New Project

And

Project Structure

1.4.مقدمة:

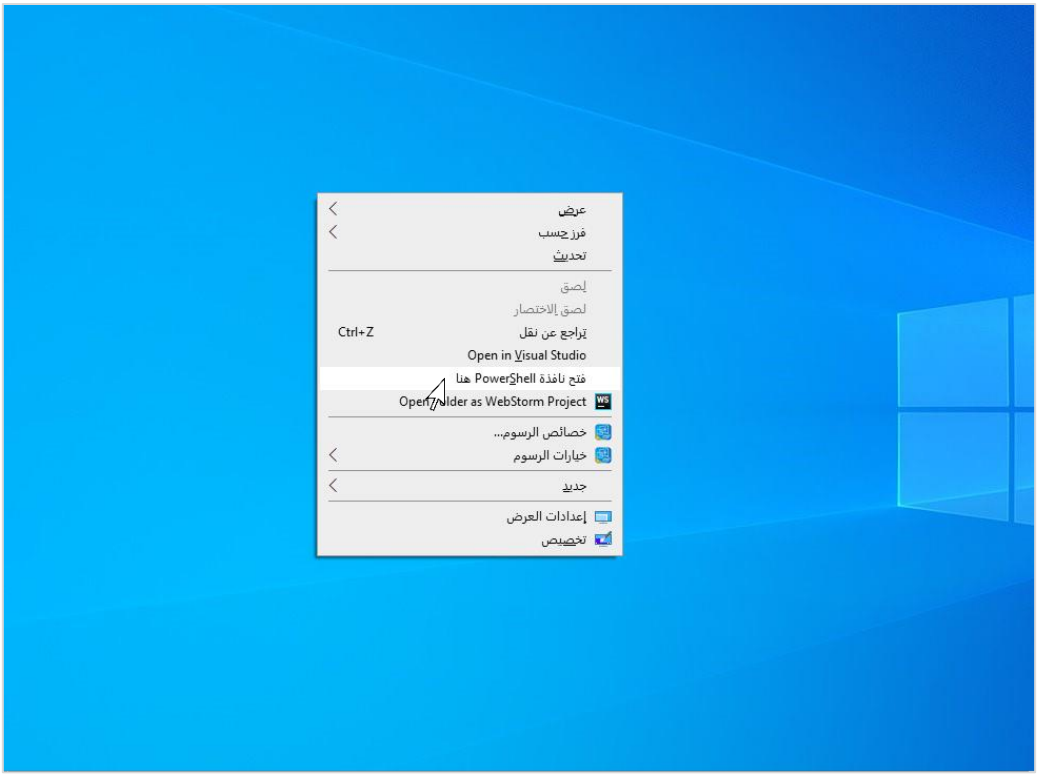
جميع الفصول السابقة ليس مخصصة لإطار عمل Angular فقط وانما معلومات عامة لأي مطور وخصوصاً مطوري الويب، ومن هذا الفصل والفصول التي تليه سوف نركز على Angular حيث في هذا الفصل سنتكلم بإذن الله عن كيفية بناء مشروع Angular جديد وكيفية مع شرح لبنية هذا المشروع من ناحية استعراض الملفات التي يتم انشائها تلقائياً مع أي مشروع Angular جديد، وفهمها، ومتى يتم استخدامها.

2.4.إنشاء مشروع Angular جديد:

قبل ان ننشأ مشروع Angular جديد نحتاج إلى تحميل وتثبيت Angular CLI (سوف نشرحه بالتفصيل في فصل خاص به بإذن الله) حيث تتيح لنا سطر الأوامر التعامل بسهولة في تحميل وتثبيت جميع الذي نريده.

1.2.4.تحميل وتثبيت Angular CLI:

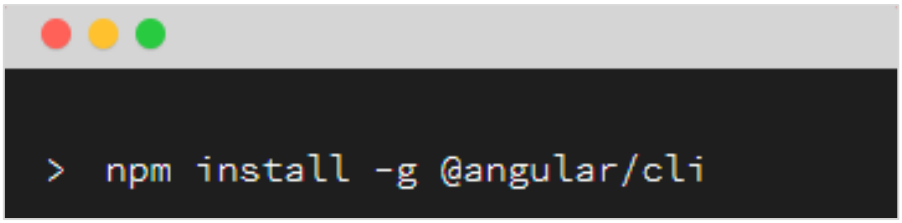
نستطيع ذلك عن طريق terminal، لذلك لنقوم بتشغيل terminal بحسب نظام التشغيل لديك، انا نظامي Windows لذلك سوف أقوم بتشغيل Power Shell وهو terminal الافتراضي في نظام التشغيل Windows، وهنالك أكثر من طريقة لتشغيل هذا terminal، منها الضغط على سطح المكتب بأي مكان فارغ بزر الفأرة الأيمن مع الضغط على زر Shift في لوحة المفاتيح بنفس الوقت، وسوف تظهر لنا قائمة منسدلة نختار منها فتح نافذة PowerShell هنا، كالتالي:



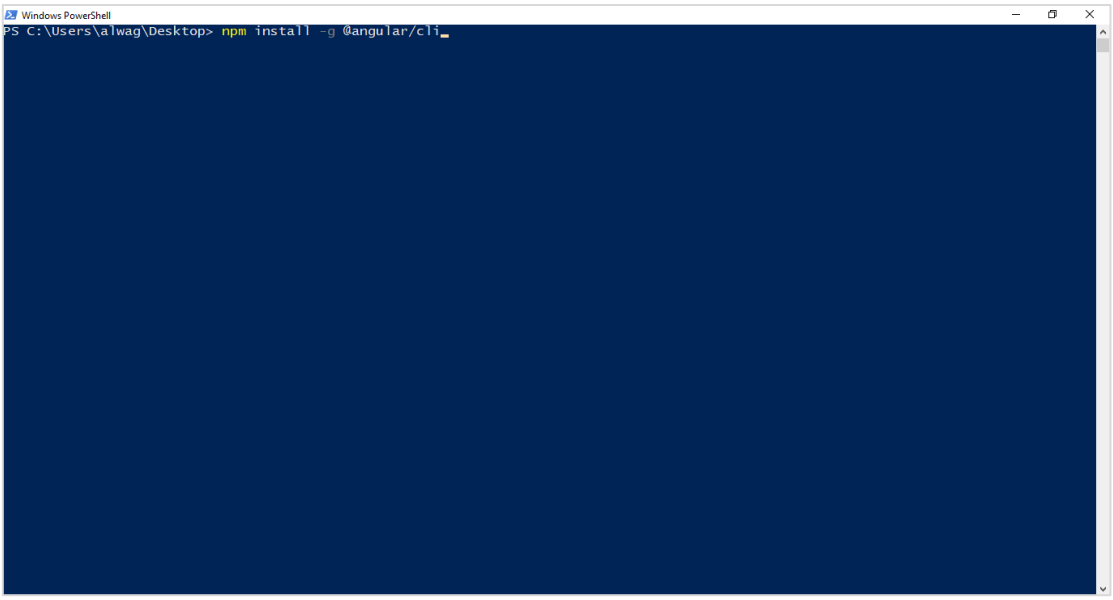
بعد اختيار هذا الأمر سوف تظهر لنا شاشة PowerShell، كالتالي:

نلاحظ فتح لنا شاشة terminal بنفس المسار الذي فتحنا فيه القائمة المنسدلة، ومن هذا المنطلق تستطيع عزيزي المتعلم الذهاب إلى أي مجلد تريده وتنفيذ نفس الخطوات وسوف يتم فتح terminal بنفس المسار المحدد، بالطبع هنالك طرق أخرى وتستطيع البحث عنها في شبكة الانترنت في حال الرغبة بذلك.

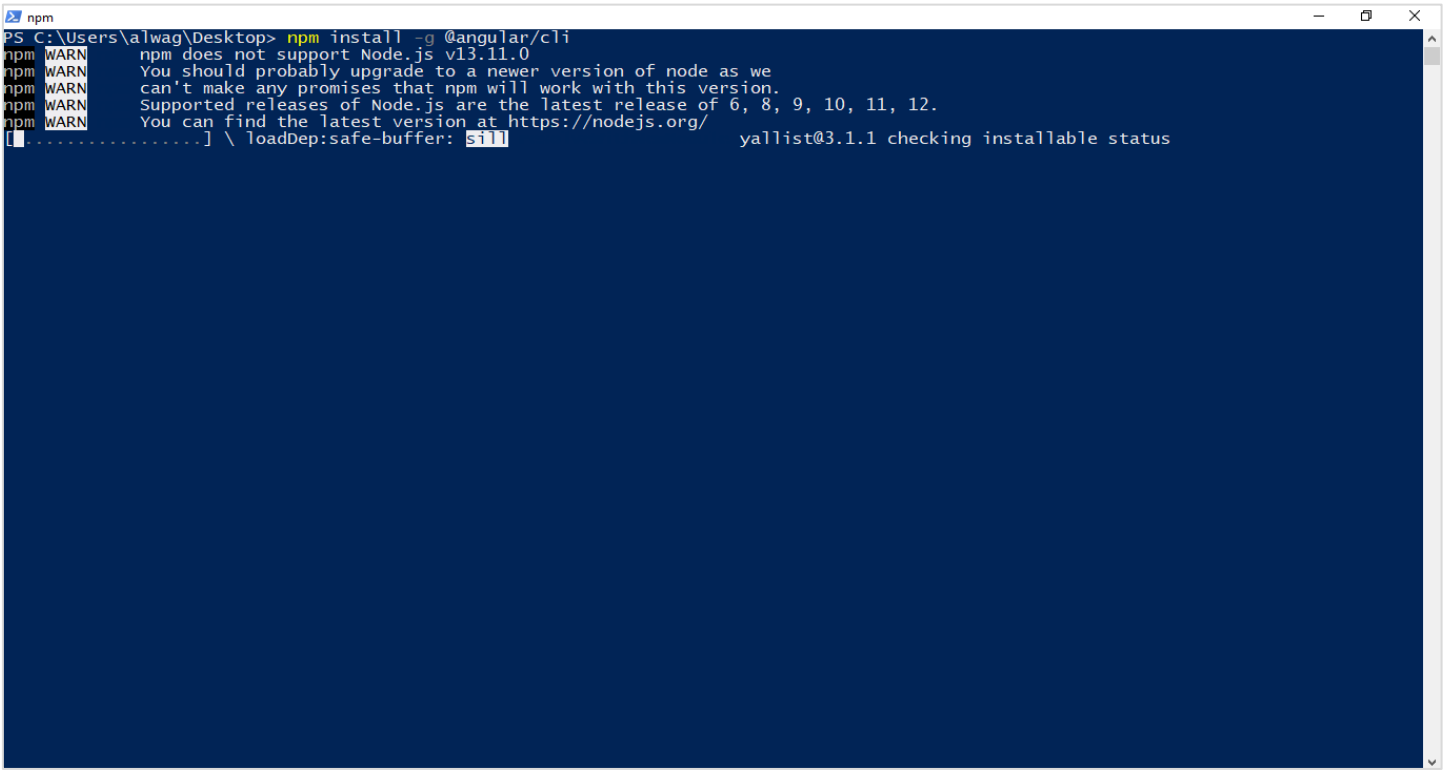
أما الآن لنستفيد من NPM التي قمنا بتثبيتها في الفصل الأول في تحميل وتثبيت Angular CLI عن طريق كتابة الأمر التالي:



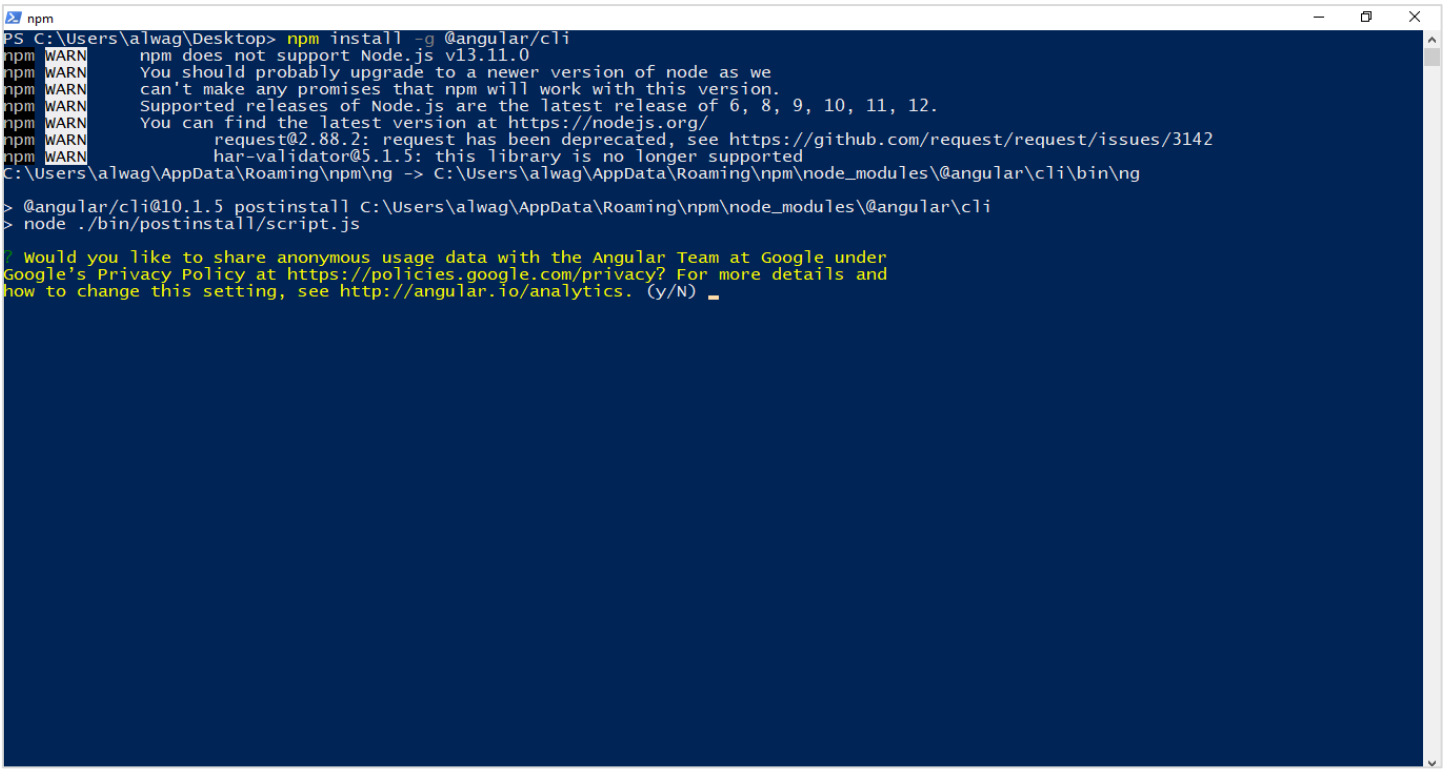
نكتب الأمر السابق في terminal، كالتالي:



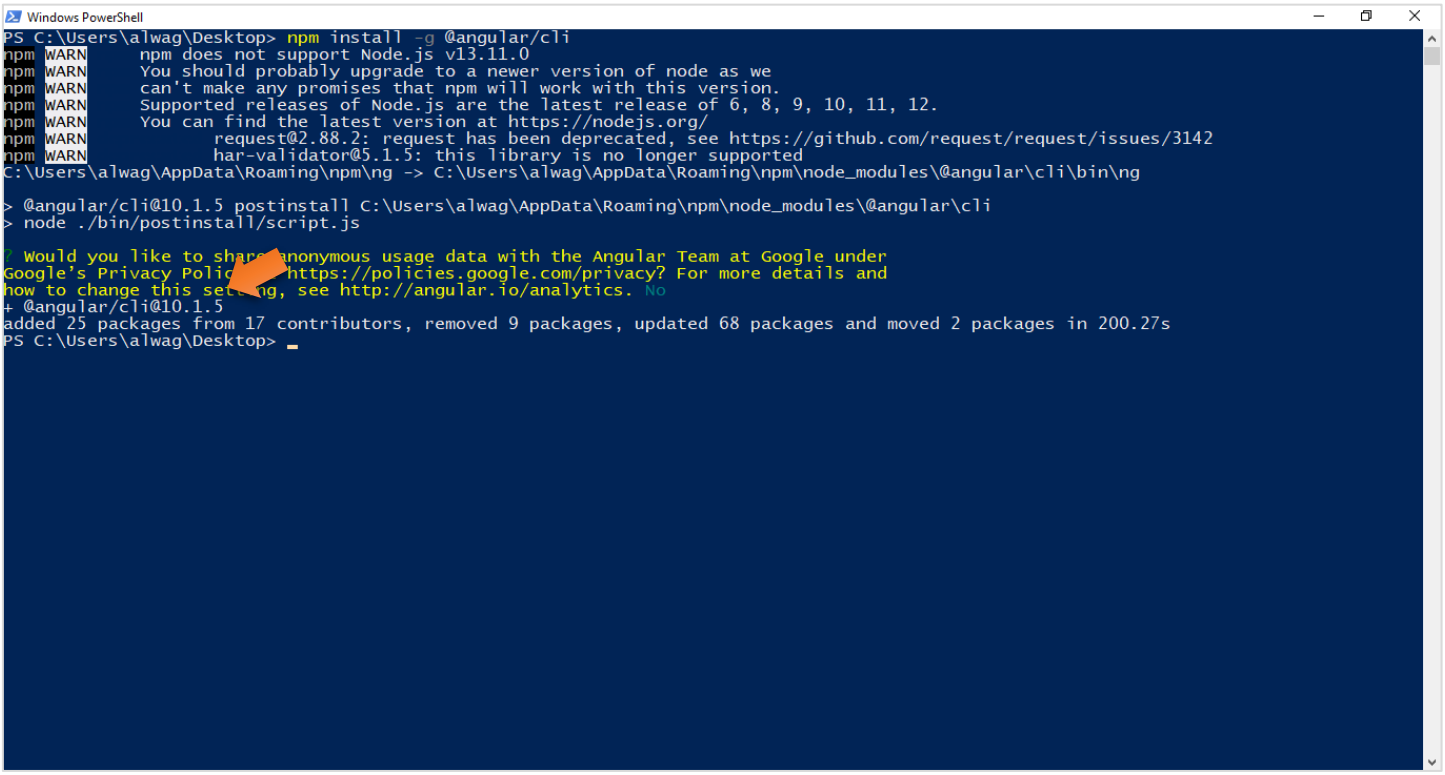
وبعدها نضغط Enter من لوحة المفاتيح وسوف يتم تجميل وتثبيت Angular CLI في جهاز الحاسب لديك وهذا الأمر قد يستغرق بعض الوقت، كالتالي:



عند الانتهاء سوف تظهر رسالة مفادها في حال أردت مشاركة بعض البيانات مع فريق مطوري Angular ولك حرية الاختيار عزيزي المتعلم بالموافقة او الرفض، كالتالي:



وبعدها سوف تظهر لك الإصدار وهذا يدل على انه تم تحميل Angular بشكل سليم في جهازك، كالتالي:



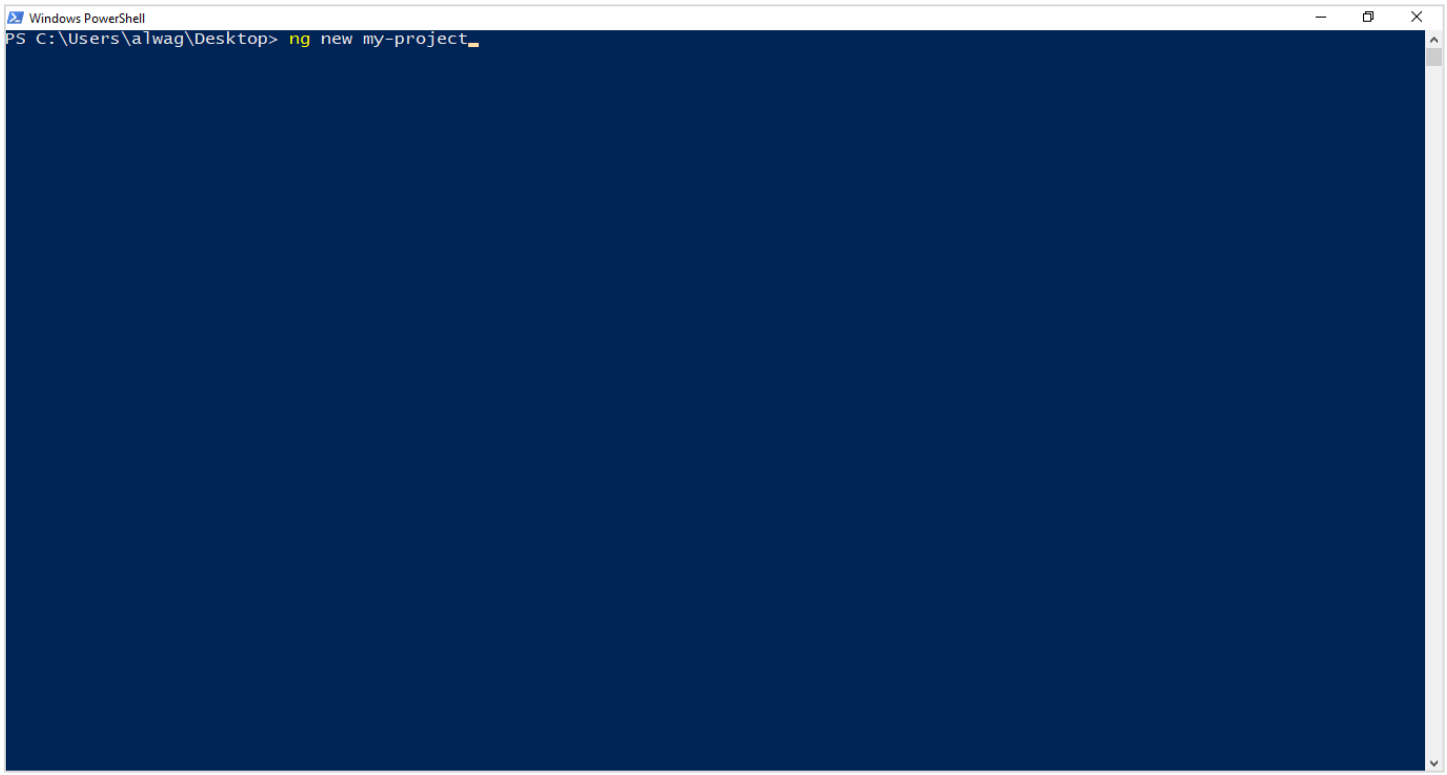
نلاحظ ان الإصدار هو 10 وهو آخر اصدار عند كتابة هذه السلسلة، والتي قد يختلف عند قراءتك لهذه السلسة.

وبذلك نكون انتهينا من تحميل وتثبيت Angular CLI وأصبحنا جاهزين لإنشاء مشروع Angular جديد.

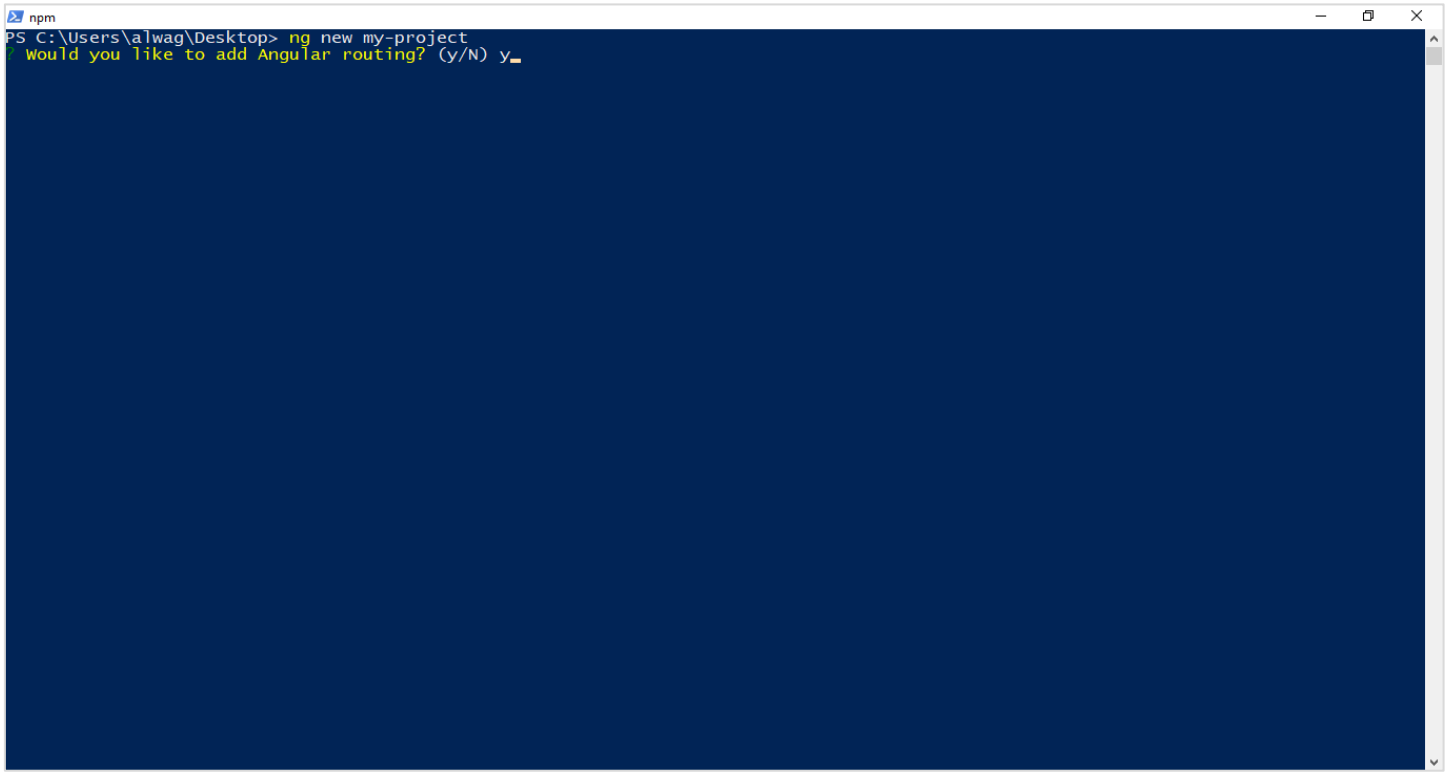
2.2.4.إنشاء مشروع Angular:

لإنشاء مشروع Angular جديد نحتاج إلى terminal لذلك لنقوم بفتح terminal كما فعلنا سابقاً وايضاً بنفس المسار وهو سطح المكتب، كالتالي:

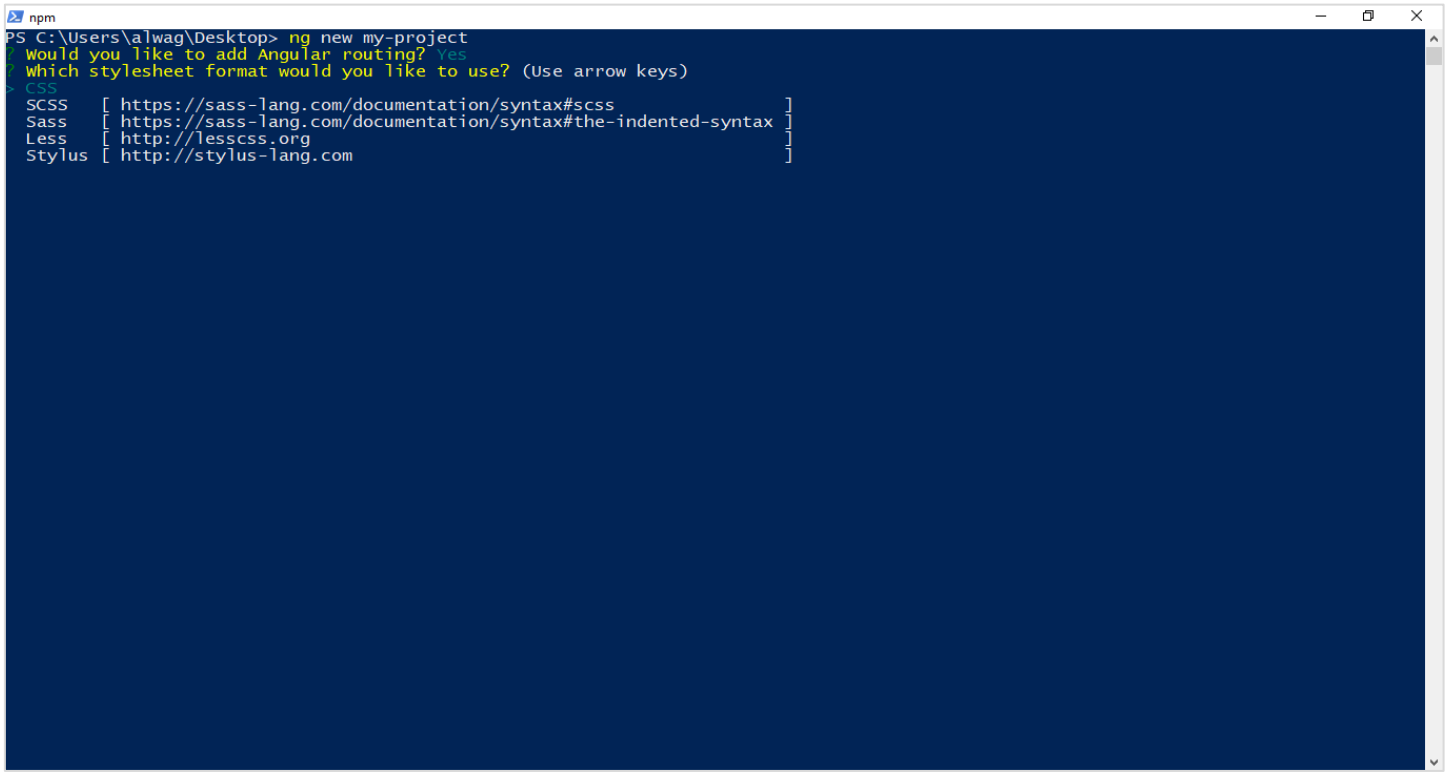
ومن ثم نكتب الأمر التالي:



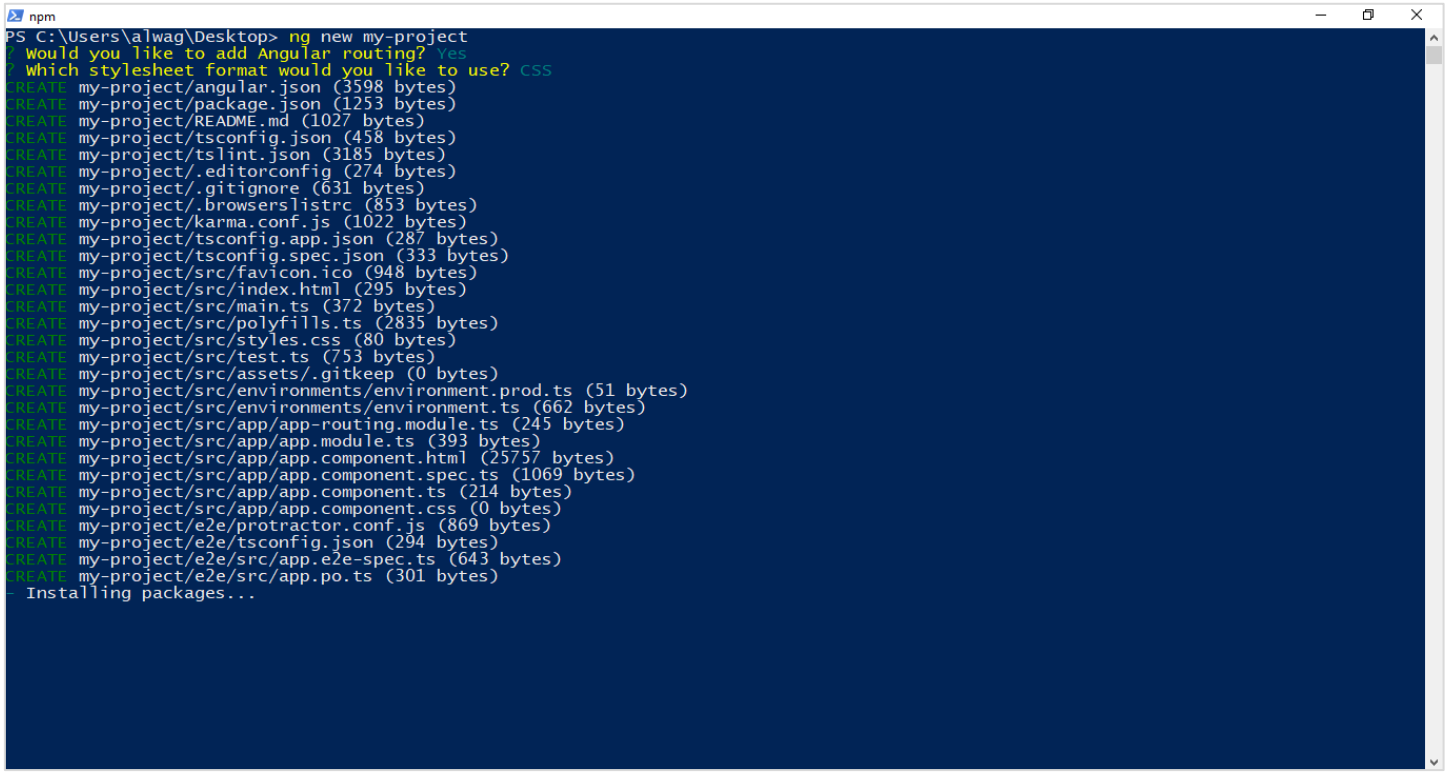
ng هذا يرمز إلى أن الأوامر التي تليه تخص Angular ولو لم يتم تحميل Angular CLI فلن يتعرف terminal على هذا الأمر، اما new فنحن بذلك نخبر Angular اننا نُريد أن ننشأ مشروع جديد وأخيراً my-project هذا هو اسم المشروع ولك عزيزي المتعلم حرية اختيار الاسم الذي تُريده، وبعد كتابتنا لهذا الأمر نضغط Enter وسوف ننتظر قليلاً ومن ثم سوف تظهر لنا رسالة توضح هل نُريد تضمين Angular Routing في هذا المشروع وانا هنا سوف اختار Y بمعنى نعم، كالتالي:



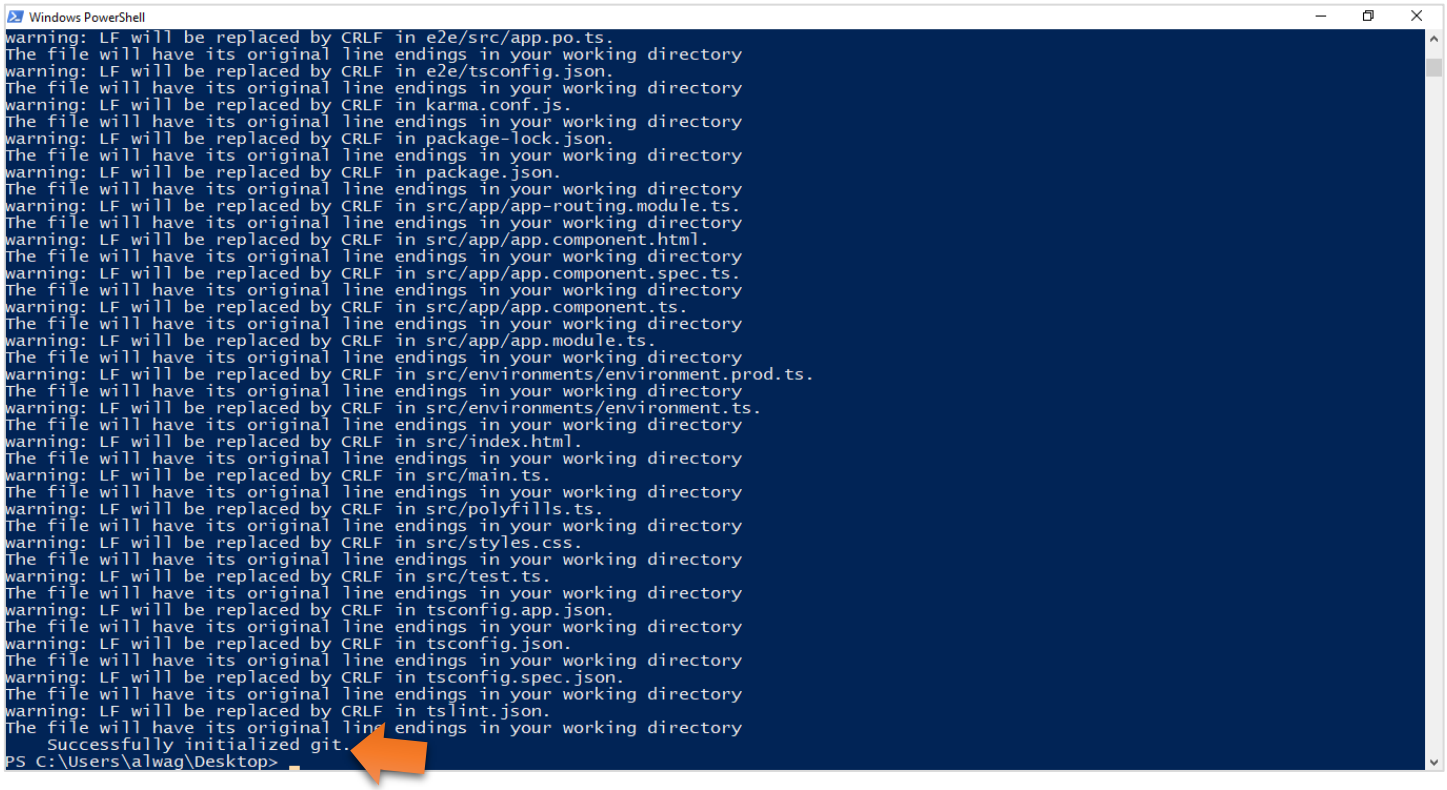
بعدها سوف يطلب منا نوع التنسيق الذي نريده في المشروع هل هو css او scss او sass ونستطيع الاختيار من بينها عن طريق ازرار الأسهم في لوحة المفاتيح ومن ثم نضغط الزر Enter لكي نختار التنسيق الذي نُريده، كالتالي:



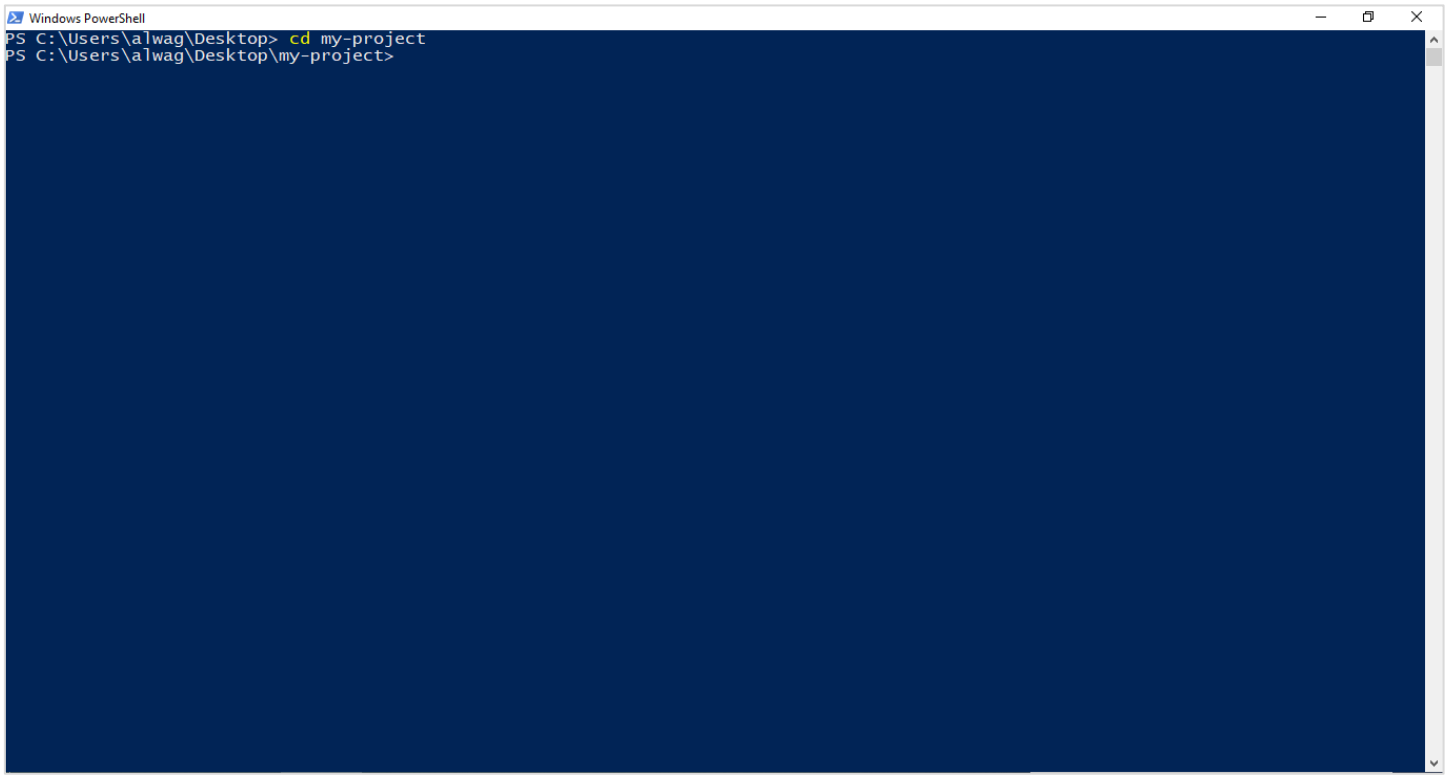
بعد اختيارنا لاحد التنسيقات سوف يبدأ Angular CLI بإنشاء المشروع، وهذا قد يستغرق بعض الوقت، كالتالي:



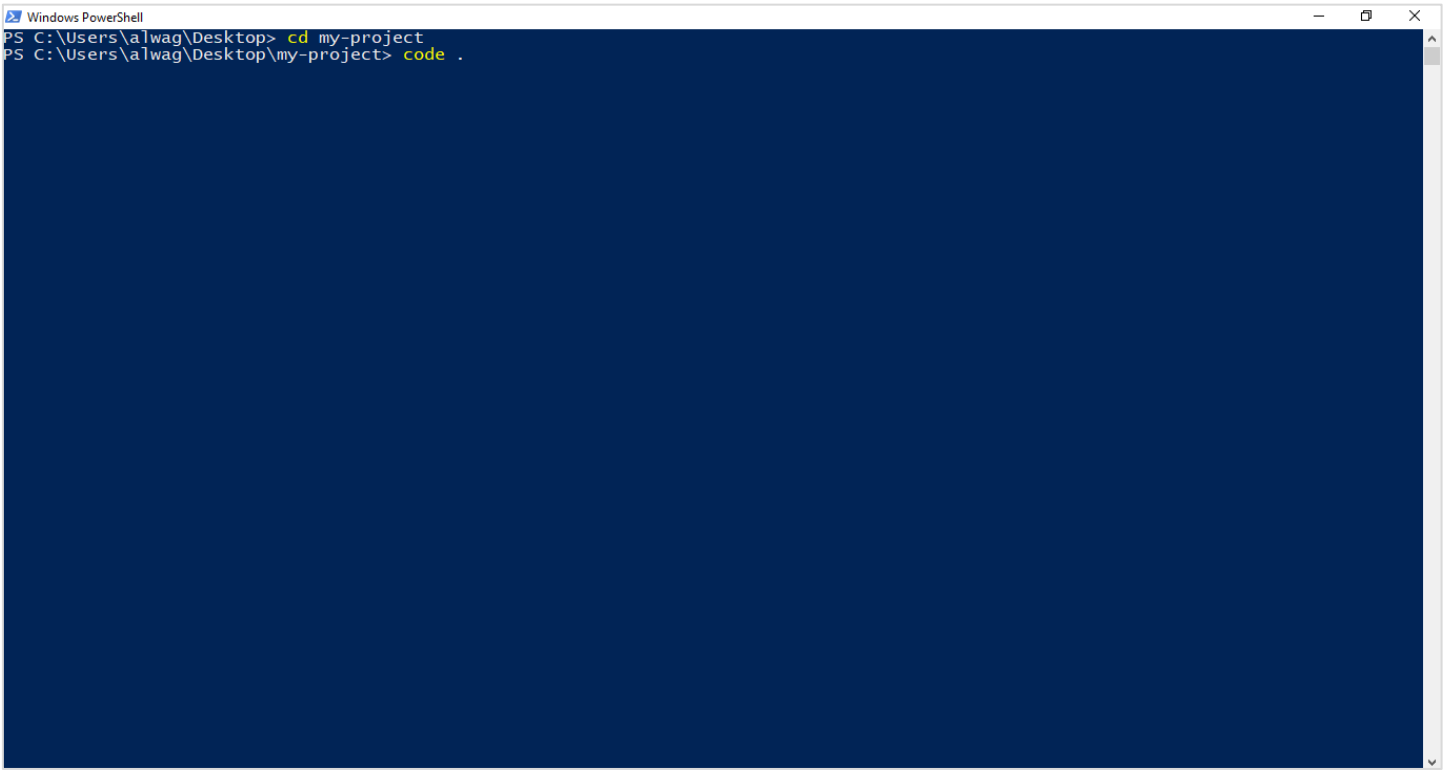
عند الانتهاء سوف تظهر لك الرسالة التالية:



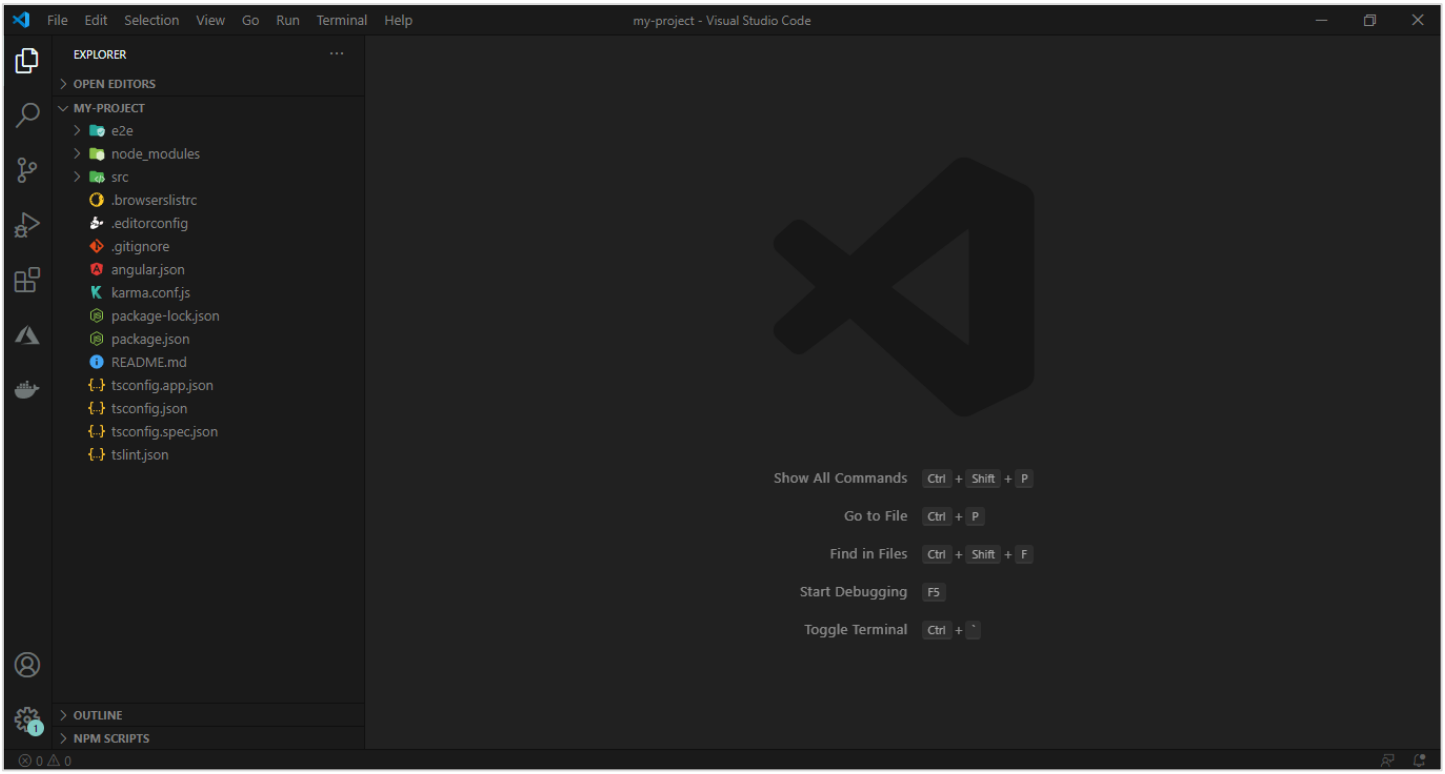
والآن سوف ننتقل إلى أهم نقطة وهي ضرورية وغالب المبرمجين المبتدئين يُخطئون بها، وهي الدخول إلى مجلد المشروع المُنشأ، حيث اننا عندما قمنا بإنشاء مشروع Angular جديد فإن Angular CLI سوف ينشأُ مجلد بنفس اسم المشروع ويبني جميع ملفات هذا المشروع بداخله، لذلك نحتاج ان نقوم بتوجيه terminal لدخول إلى مجلد المشروع بكتابة الأمر التالي:



وآخر خطوة هي فتح هذا المشروع في محرر الأكواد VS Code ونستطيع عمل ذلك عن طريق كتابة الأمر التالي:



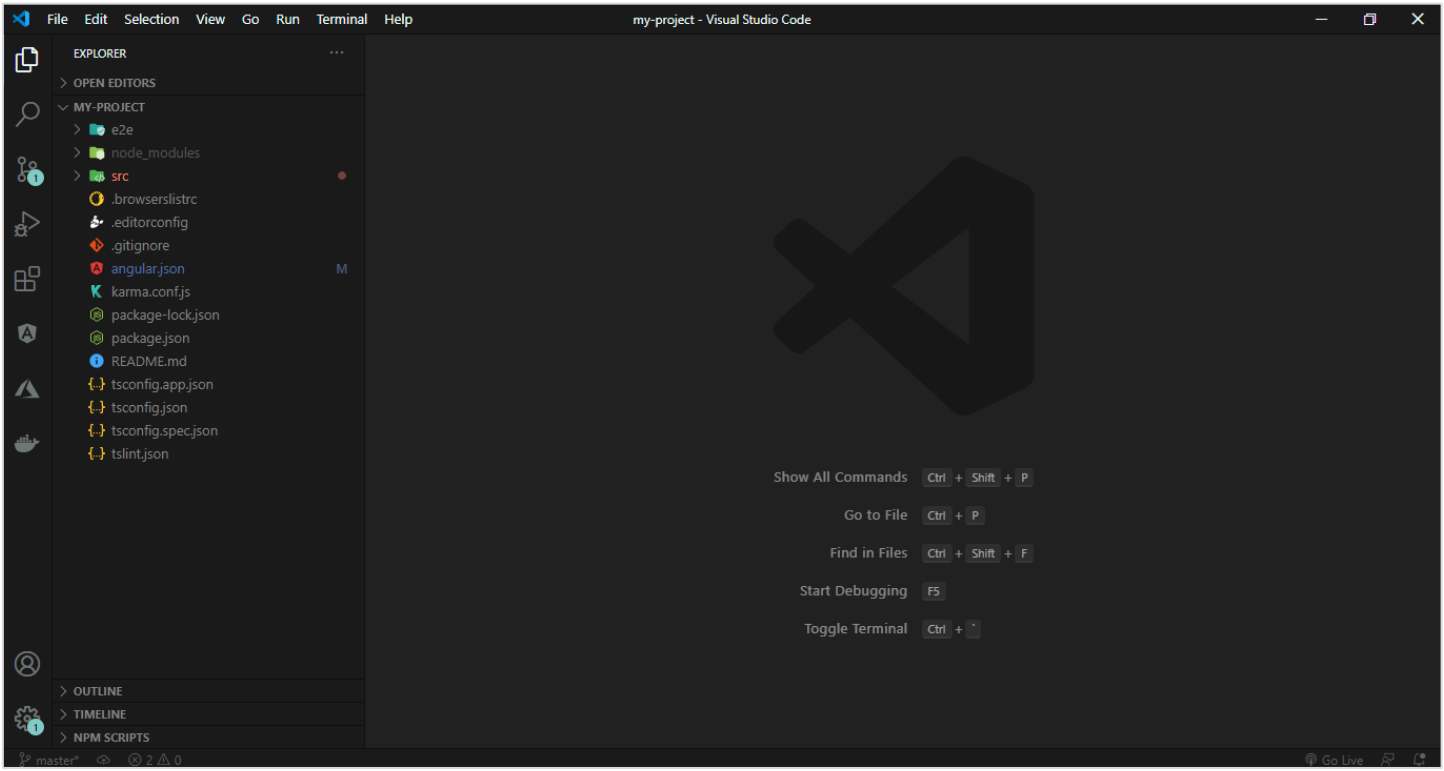
نلاحظ كتبنا الأمر code ومن ثم نقطة وبعدها نقوم بالضغط على زر Enter من لوحة المفاتيح وسوف يتم فتح محرر الأكواد VS Code وبداخله المشروع، كالتالي:



وبذلك أصبح المشروع جاهز، وفي الخطوة التالية سوف أقوم بشرح اهم ملفات بنية المشروع.

3.4.بنية مشروع Angular:

لو استعرضنا ملفات مشروع Angular كما في الشكل التالي:



سوف نلاحظ انه تم انشاء مجموعة من الملفات الضرورية لتشغيل أي مشروع Angular وهذه الملفات هي ما نسميه بنية المشروع او Project Structure وبطبيعة الحال لن نشرح جميع هذه الملفات بالتفصيل وانما سوف نستعرض اغلبها وبعض هذه الملفات سوف نتكلم عنها بالتفصيل والبعض الآخر مجرد استعراض لمحتوياتها والفائدة منها.

1.3.4.مجلد e2e:

مهمة هذا المجلد في التعامل مع Testing على مستوى التطبيق كامل، ولعلنا نفرد كتاب يتكمل فقط عن Unit Testing.

2.3.4.مجلد node_modules:

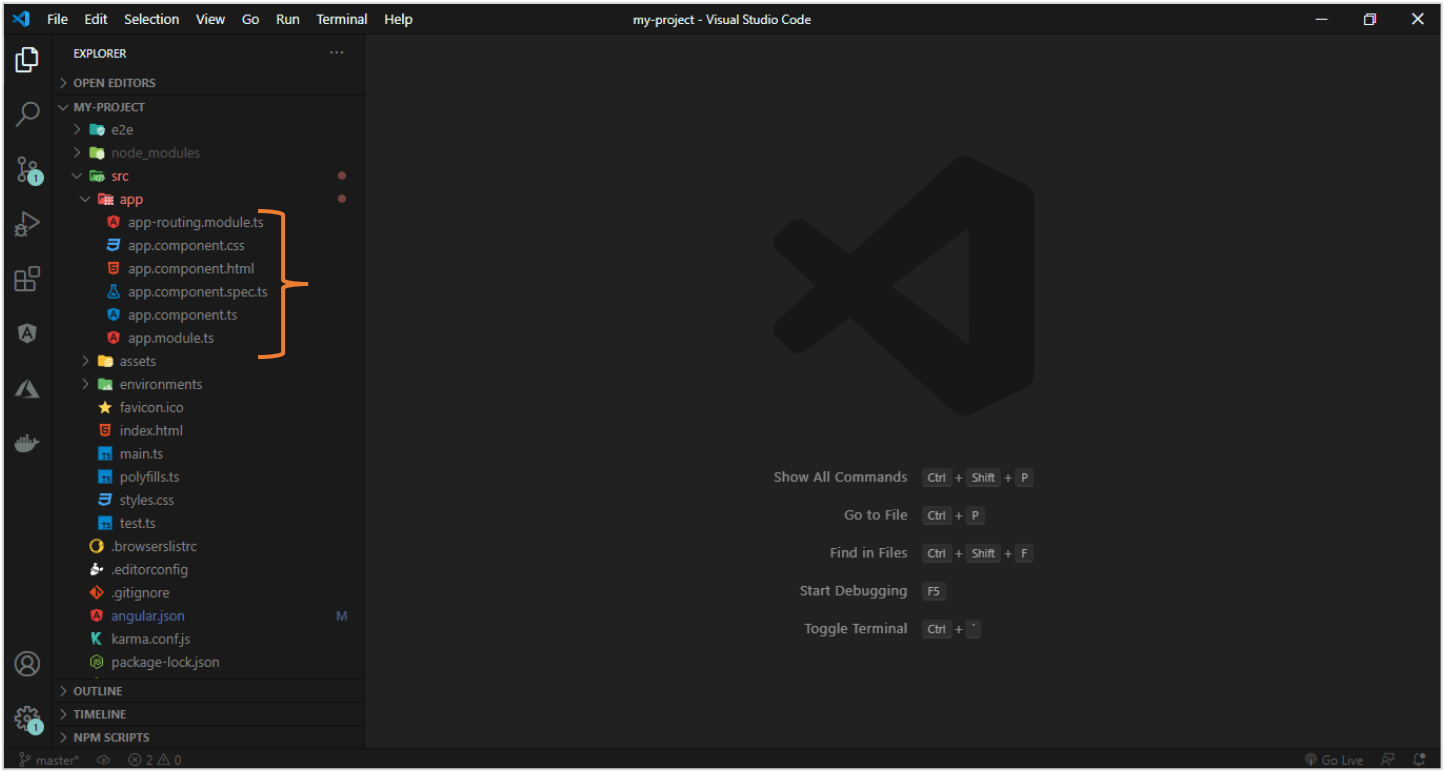
ويحتوي هذا المجلد على جميع الملفات الضرورية التي تنزل مع التطبيق لضمان عمل جميع مميزات Angular على أكمل وجه، هذا من جهة ومن جهة أخرى أي مكتبة خارجية يتم تضمينها في مشروع Angular سيتم اضافتها في هذا المجلد، مع العلم ان هذا المجلد حجمه كبير حيث في بداية المشروع يصل إلى أكثر من ٢٥٠ ميغابايت فما رأيك عزيزي المتعلم في حال كان مشروعنا يحتوي على عشرات ان لم يكن المئات من Components وقد نحتاج إلى مكتبات خارجية لكي نستفيد منها ولا نُعيد اختراع العجلة مرة أخرى، لذلك لا يتم رفع هذا المجلد في العادة لمشاركة مشاريع Angular وانما يتم استخدام طريقة أخرى سوف نتكلم عنها بعد قليل عندما نصل pakage.json.

3.3.4.مجلد src:

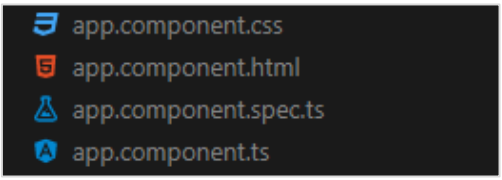
ولعل هذا المجلد يعتبر من أهم المجلدات في مشروع Angular، وعند بناء مشروع Angular بشكله النهائي فأننا نقوم ببناء هذا المجلد بما يحتويه من مجلدات وملفات فرعية، وجميع المجلدات والملفات خارج هذا المجلد تساعدنا في مرحلة التطوير فقط، أما من ناحية المجلدات والملفات الفرعية التي يحتويها هذا المجلد فهي كالتالي:

1.3.3.4.مجلد app:

وهذا المجلد يحتوي على جميع أجزاء التطبيق الخاص بك عزيزي المتعلم سواء كانت components او directives او pipes او classes او services او ...الخ، حيث يرتكز تقريباً 90% من عمل أي مطور Angular في هذا المجلد، وبشكل مبدئي عند انشاء أي مشروع Angular جديد بالعادة فإن هذا المجلد يحتوي على ستة ملفات، كما في الشكل التالي:



اول ملف وهو خاص Routing وبإذن الله سوف نفصل فيه في الكتاب الخاص بي Angular Routing and Modules اما الملفات الأربعة التالية:



فهي تمثل Component الأب او الرئيسي لجميع Components الأخرى وايضاً عزيزي المتعلم لا تقلق من هذه المصطلحات فقد قمت بشرح جميع مفاهيم components في الكتاب ذو الاسم Angular Components and Services، ونلاحظ ان هنالك ملف يحتوي على كلمة spec وفيه نكتب Logic الخاص بعمل Testing على مستوى هذا component فقط.

اما آخر ملف وهو app.module.ts، فيعتبر بمثابة نقطة الوصل التي يتم فيها تعريف أي components او Modules أخرى سواء كانت تنتمي لمكتبات خارجية او المبنية ضمناً في نفس إطار عمل Angular، حيث يستخدم Angular هذا الملف لكي يقرأ جميع أجزاء المشروع الخاص بك عزيزي المتعلم وضمان عمله على الوجه الأمثل، مع العلم انه يمكن تقسيم هذا الملف إلى ملفات أصغر وخصوصاً في التطبيقات المتوسطة إلى الكبيرة ولكن في النهاية لابد ان يتم تعريف هذه الأجزاء الصغيرة بهذا الملف. وقد تكلمت بالتفصيل عن Modules وطريقة تقسيمها إلى أجزاء صغيرة في كتاب Angular Routing and Modules.

2.3.3.4.مجلد assets:

في هذا المجلد يتم إضافة المجلدات الثابتة static مثل الصور والميديا الأخرى او ممكن نضف الخطوط بمعنى أي ملف نريد تضمينه ويكون ثابت ويتم بنائه مع التطبيق في المرحلة النهائية نضعه هنا، وما يميز هذا المجلد ان Angular يتعرف عليه بمجرد كتابة اسمه ومن ثم مسار الملف المستهدف ولا نحتاج إلى كتابة اسمه بالكامل، كالتالي:

```
<img src="assets/images/image1.png" />
```

نلاحظ اننا افترضنا اننا انشأنا مجلد باسم images وبداخله صورة باسم image`.png، حيث بمجرد كتابة اسم مجلد assets تلقائياً Angular سوف يتعرف على مكانه ولا نحتاج ان نحدد المسار بالكامل بالنسبة للملف الموجود بع العنصر <img />.

3.3.3.4.مجلد environments:

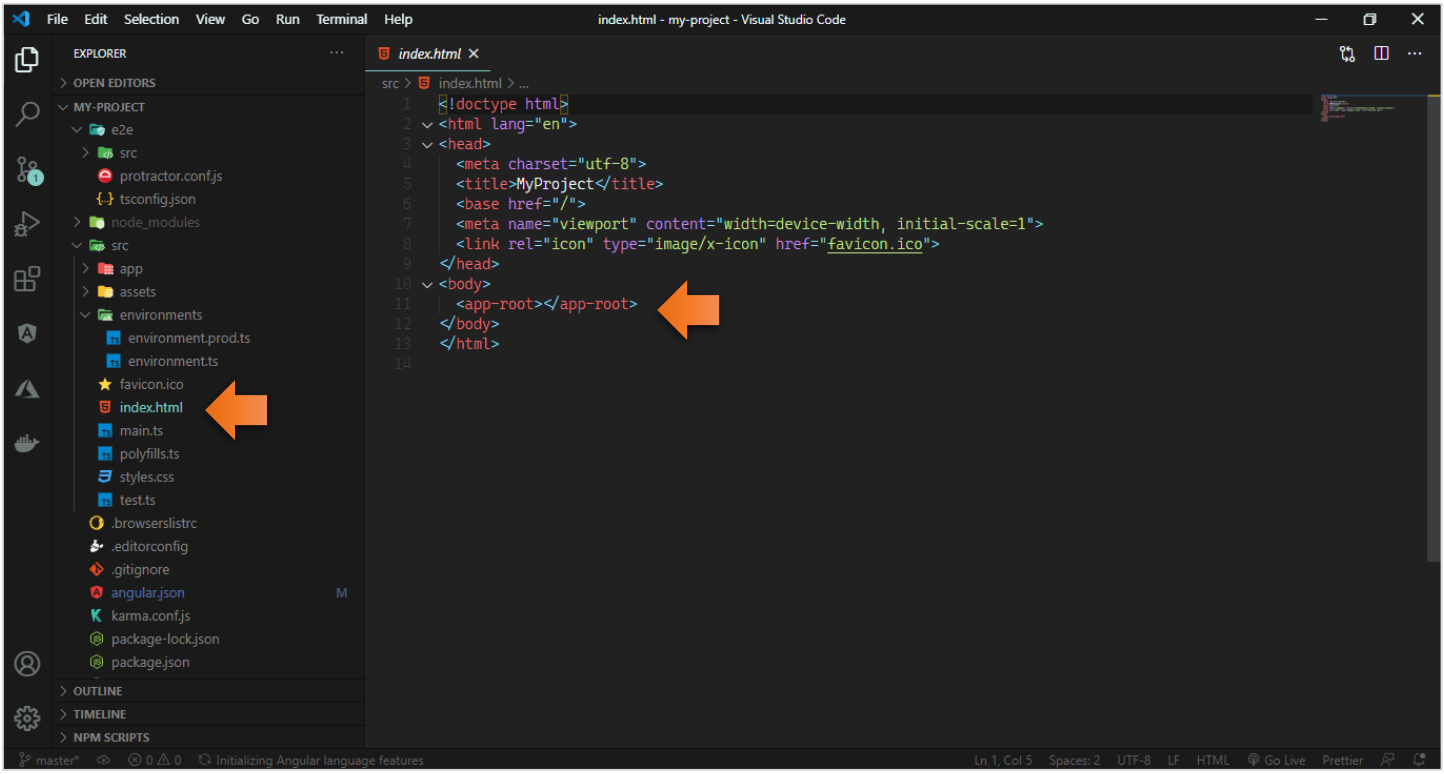
هذا المجلد يحتوي على ملفين هما environment.ts وملف environment.prod.ts ويتم فيهما إضافة بعض الإعدادات مثل اعدادات قاعدة البيانات، وAngular في مرحلة التطوير يستخدم environment.ts اما في مرحلة البناء والمُنتج النهائي لتطبيق فأنه يستخدم environment.prod.ts.

4.3.3.4.ملف Favicon.ico:

وهو ملف أيقونة التطبيق وبشكل افتراضي Angular مضمن في أي مشروع هذه الأيقونة 🅰 وتستطيع تغييرها بالأيقونة الخاصة بك ولكن لابد ان يكون اسمها favicon والامتداد ico.

5.3.3.4.ملف Index.html:

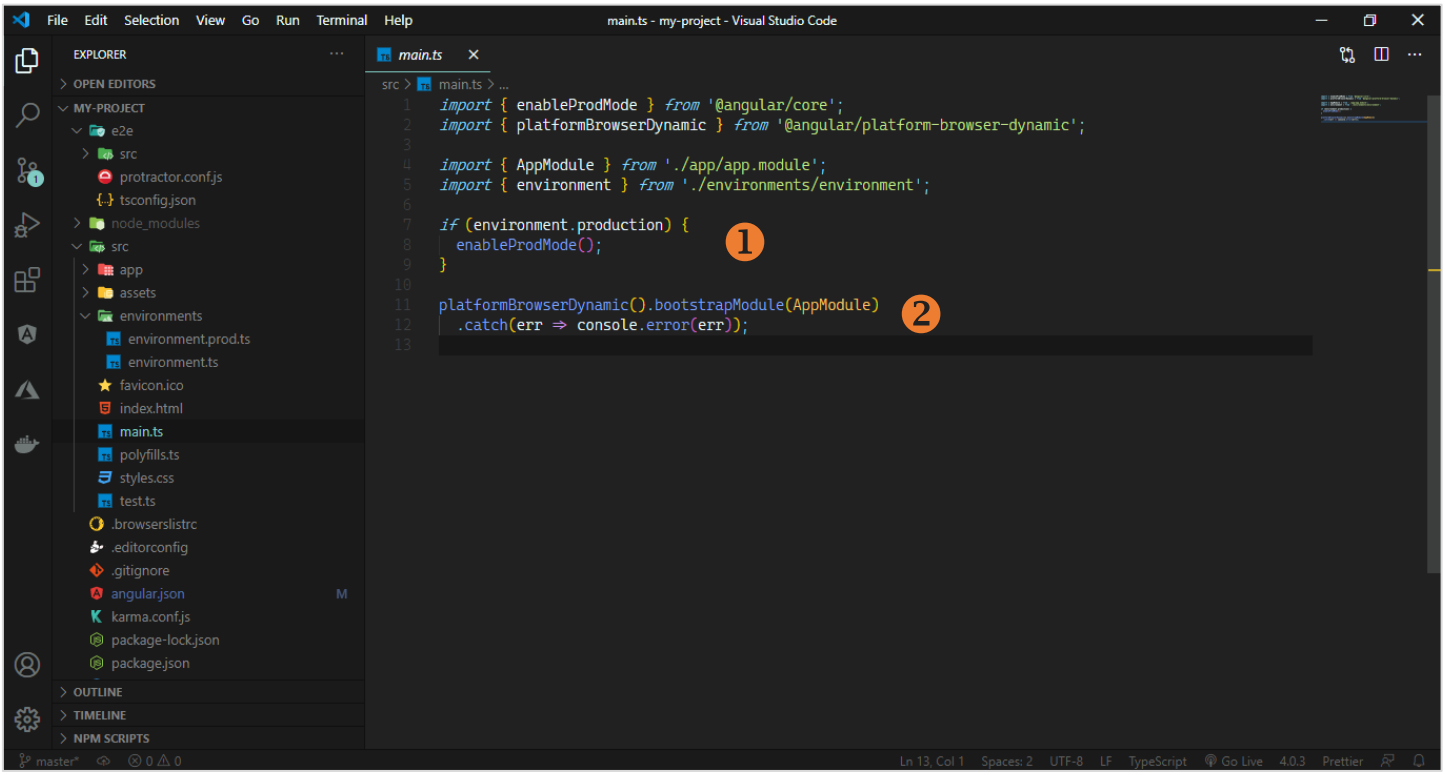
وهذا الملف هو ملف HTML الرئيسي الذي يقرأه Server عند بداية تشغيل تطبيق الويب، وهو بمثابة الصفحة الرئيسية التي يتم تقديمها (عرضها) لأي زائر لتطبيق الخاص بك في بداية تشغيله. ولو استعرضنا محتويات هذا الملف لوجدنا التالي:



نلاحظ ان هذا الملف هو عبارة عن ملف HTML عادي يحتوي على Markup الأساسي لأي ملف HTML، الشيء الوحيد الجديد هو ما تم التأشير عليه بالسهم حيث يوجد به عنصر جديد وهو <app-root></app-root> وهذا العنصر ليس من ضمن عناصر HTML وانما هو عنصر خاص بي Angular وهو عبارة عن Selector الخاص بالComponent الأب الذي أشرنا له في مجلد app، بحيث عند بداية تشغيل التطبيق فإن هذه الصفحة هي الرئيسية وسوف يقرأ هذا العنصر ويقوم بعرضه للمستخدم، وايضاً اكرر انه ليس المطلوب منك فهم جميع هذه التفاصيل وانما فقط عليك ان تعرف فائدة ومهمة هذا الملف، وفي الكتب القادمة سوف نتطرق إلى جميع هذه الأمور بالتفصيل إن شاء الله.

6.3.3.4.ملف main.ts:

وهذا الملف هو نقطة البداية التي يبدأ في التطبيق الخاص بك حيث سوف يقرأ هذا الملف، ولكي لا يحدث لك لبس بين هذا الملف وملف Index.html، الملف الأخير هو صفحة البداية التي تكون يقرأها server ويقدمها إلى أجهزة المستخدمين عند طلب عرض موقعك، اما هذا الملف main.ts فهو الملف الذي يعتبر نقطة الانطلاق للتطبيق الخاص بك حيث يتم قراءة جميع Functionality اللازمة لعمل هذا التطبيق، وابسطها هو التعرف على العنصر <app-root></app-root> ويعرف انه عبارة عن Selector الخاص بالComponent الأب لكي يقوم بعرضه، ولو استعرضنا محتويات هذا الملف لوجدنا التالي:



(1) نلاحظ اننا استفدنا من ملف environment لتأكد هل نحن في وضع التطوير فإذا كنا في وضع التطوير فإن Angular سوف يقوم بتشغيل دالة وهذه الدالة تفعل جميع الخدمات التي تؤدي في النهاية لإكمال تطوير تطبيقه على الوجه الأمثل، مع العلم ان الوضع الافتراضي هو وضع التطبيق النهائي.

(2) هذا هو Logic الخاص بقراءة التطبيق ونلاحظ قمنا بتمرير AppModule وهو Module الرئيسي الذي أشرنا لها سابقاً عندما تكلمنا عن المجلد app.

7.3.3.4.ملف Polyfills.ts:

وهذا الملف يُستفاد منه لضمان عمل تطبيق الويب الخاص بك على جميع المتصفحات، لأن قد يكون هنالك بعض المميزات الحديثة في Typescript او ES6+ لا تدعمها جميع المتصفحات، لذلك هذا الملف يقوم بالنيابة عنك بتحويل أي ميزة جديدة إلى إصدارات أقدم يفهمها جميع المتصفحات لضمان عمل التطبيق لدى جميع المستخدمين.

8.3.3.4.ملف styles.css:

وفي هذا الملف يتم كتابة أي تنسيقات css يُراد منها ان تكون عامة وتُطبق بجميع أجزاء المشروع، فمثلاً لو أردنا ان نطبق تنسيقات على عنصر <p> فإنه عند كتابة تنسيقات لعنصر <p> في هذا الملف فإن هذه التنسيقات سوف تُطبق لكافة عناصر <p> الموجودة في المشروع، لذلك يجب الانتباه عند استخدام هذا الملف، ولعل من أشهر استخدامات هذا الملف هو استدعاء مكتبات وأطر عمل css (مثل bootstrap) في هذا الملف لكي تُصبح عامة ونستطيع الاستفادة من مميزاتها في جميع أجزاء المشروع.

مع العلم ان امتداد هذا الملف قد يختلف باختلاف نوع التنسيق الذي تم اختياره عند بداية انشاء مشروع Angular جديد، لذلك قد يكون css او scss او sass.

9.3.3.4.ملف test.ts:

وهو مشابه main.ts ولكن من ناحية testing حيث يعتبر نقطة الانطلاق لعملية testing التي يقوم بها Angular.

4.3.4.ملف editorconfig.:

هذا الملف يحتوي على بعض الإعدادات الافتراضية لمحرر الأكواد، منها على سبيل المثال جميع الملفات التي تنتهي بالامتداد *.ts يجب ان تحتوي على علامة تنصيص مفردة وليست مزدوجة، وفي الحقيقة قد لا تحتاج هذا الملف نهائياً.

5.3.4.ملف gitignore.:

وهذا الملف نُحدد فيه أي الملفات التي لا نُريد ان نرفعها في حال أردنا ان نرفع نسخة من مشروعنا وهو في مرحلة التطوير على موقع GitHub، ومن أشهر هذه المجلدات مجلد node_modules الذي أشرنا له سابقاً.

6.3.4.ملف angular.json:

وهذا الملف كان يُسمى سابقاً angular-cli.json ولكن مع الإصدارات الجديدة من angular تغير الاسم، وهذا الملف يحتوي على جميع اعدادات مشروع angular الخاص بك، وسوف نستعرض محتويات هذا الملف ومن ثم نقوم بالتعليق على الأجزاء المهمة، كالتالي:

ملف angular.json

```
1. {
2.   "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
3.   "version": 1,
4.   "newProjectRoot": "projects",
5.   "projects": {
6.     "my-project": {
7.       "projectType": "application",
8.       "schematics": {},
9.       "root": "",
10.      "sourceRoot": "src",
11.      "prefix": "app",
12.      "architect": {
13.        "build": {
14.          "builder": "@angular-devkit/build-angular:browser",
15.          "options": {
16.            "outputPath": "dist/my-project",
17.            "index": "src/index.html",
18.            "main": "src/main.ts",
19.            "polyfills": "src/polyfills.ts",
20.            "tsConfig": "tsconfig.app.json",
21.            "aot": true,
22.            "assets": [
23.              "src/favicon.ico",
24.              "src/assets"
25.            ],
26.            "styles": [
27.              "src/styles.css"
28.            ],
```

```
            "scripts": []
          },
          "configurations": {
            "production": {
              "fileReplacements": [{
                "replace": "src/environments/environment.ts",
                "with": "src/environments/environment.prod.ts"
              }],
              "optimization": true,
              "outputHashing": "all",
              "sourceMap": false,
              "extractCss": true,
              "namedChunks": false,
              "extractLicenses": true,
              "vendorChunk": false,
              "buildOptimizer": true,
              "budgets": [{
                  "type": "initial",
                  "maximumWarning": "2mb",
                  "maximumError": "5mb"
                },
                {
                  "type": "anyComponentStyle",
                  "maximumWarning": "6kb",
                  "maximumError": "10kb"
                }
              ]
            }
          }
        },
        "serve": {
          "builder": "@angular-devkit/build-angular:dev-server",
          "options": {
            "browserTarget": "my-project:build"
          },
          "configurations": {
            "production": {
              "browserTarget": "my-project:build:production"
            }
          }
        },
        "extract-i18n": {
          "builder": "@angular-devkit/build-angular:extract-i18n",
          "options": {
            "browserTarget": "my-project:build"
          }
        },
        "test": {
          "builder": "@angular-devkit/build-angular:karma",
          "options": {
            "main": "src/test.ts",
            "polyfills": "src/polyfills.ts",
            "tsConfig": "tsconfig.spec.json",
```

```
82.             "karmaConfig": "karma.conf.js",
83.             "assets": [
84.               "src/favicon.ico",
85.               "src/assets"
86.             ],
87.             "styles": [
88.               "src/styles.css"
89.             ],
90.             "scripts": []
91.           }
92.         },
93.         "lint": {
94.           "builder": "@angular-devkit/build-angular:tslint",
95.           "options": {
96.             "tsConfig": [
97.               "tsconfig.app.json",
98.               "tsconfig.spec.json",
99.               "e2e/tsconfig.json"
100.                ],
101.                "exclude": [
102.                  "**/node_modules/**"
103.                ]
104.              }
105.            },
106.            "e2e": {
107.              "builder": "@angular-devkit/build-angular:protractor",
108.              "options": {
109.                "protractorConfig": "e2e/protractor.conf.js",
110.                "devServerTarget": "my-project:serve"
111.              },
112.              "configurations": {
113.                "production": {
114.                  "devServerTarget": "my-project:serve:production"
115.                }
116.              }
117.            }
118.          }
119.        }
120.      },
121.      "defaultProject": "my-project"
122.    }
```

نلاحظ ان هذا الملف عبارة كائن – بدون اسم – يبدأ من السطر رقم 1 وينتهي بالسطر 122، وهو ايضاً يحتوي على كائنات فرعية، واو لها هو ما هو موجود في السطر 2 هو عبارة عن امتداد لملف معين موجود في node_modules يحتوي على جميع الأوامر المتاحة في هذا الملف بحيث لو قمنا بكتابة أي امر لا ينتمي لهذا الملف سوف تظهر لدينا رسالة خطأ.

وننتقل إلى السطر رقم 5 حيث نلاحظ وجود كائن فرعي باسم projects ويحتوي بداخله على كائن فرعي آخر باسم my-project والذي هو عبارة عن اسم المشروع الخاص بنا، حيث يبدأ من السطر 6 إلى السطر 119، ومن هذا المنطلق نستطيع ان نقول انه

قد يكون لدينا أكثر من مشروع على شكل كائنات ونستطيع تحديد الافتراضي منها في السطر 121، ولكن بما انه لا يوجد إلى مشروع واحد فقط فسوف نتكلم عن بعض اعدادات هذا المشروع.

حيث في السطر 10 تم تحديد ان يكون مجلد src هو الذي يحتوي هذا التطبيق وفي حال الرغبة على سبيل المثال تغيير اسم هذا المجلد إلى source فنحتاج تغييره هنا في هذا الملف إلى هذا الاسم الجديد لكيلا يحدث أي خطأ.

اما السطر 11 فنحدد أسماء prefix لي components فلو لاحظنا ان component الاب ذو الاسم root الاسم الخاص به مسبوق بكلمة app وهذا app هي القيمة التي اسندناها لي الخاصية prefix في هذا الملف، فلو مثلاً اردنا ان ننشئ component جديد وليكن اسمه home فأن angular سوف يُضيف app قبل هذا الاسم ليصبح اسم هذا component هو app-home وهكذا بقية components الأخرى، وفي حال اردنا تغييرها فقط نقوم بتغيير القيمة في هذا الملف ولتكن مثلاً selector فأن angular سوف يُضيف هذه القيمة لأي أسم component جديد وفي حال عدم الرغبة في إضافة أسماء فقط نسند قيمة فارغة للخاصية prefix، اما الفائدة من إضافة angular لهذا prefix في أسماء components هو لتمييزها عن غيرها من عناصر HTML الأخرى، فكما اشرنا سابقاً لا حظنا انه في ملف index قمنا بوضع component الأب على شكل عنصر <app-root></app-root> فلو نفترض انك اضفت component جديد واسميته section ولم تضع prefix له فأن اسم هذا component في حال أردت استخدامه في مشروعك سوف يكون <section></section> ومن المعروف ان section من عناصر HTML لذلك قد يحدث خطأ ما، لذلك يُفضل وضع prefix لأي اسم component.

وسطر 16 نحدد فيه مسار المشروع عندما نقوم ببنائه في شكله النهائي الذي نرفعه على server حيث سوف يتم انشاء مجلد باسم dist وبداخله مجلد بنفس اسم المشروع الخاص بك، وايضاً تستطيع التحكم به وتغييره بالاسم الذي تُريده.

وسطر 17 و18 و19 نحدد مسار ملفات index.html وmain.ts وPolyfills.ts التي أشرنا لها سابقاً ونستطيع تغيير هذه الأسماء ولكن لابد ان نغيرها في الإعدادات في هذا الملف.

وسطر 20 خاص باسم ومسار ملف tsconfig-app.ts وسوف نتكلم عنه بعد قليل.

اما سطر 21 خاص بطريقة angular في عمل compiler لملفات المشروع وهو ما يسمى نظام ivy وهذا النظام مدعوم بشكل كامل في الاصدار 9 وما فوق حيث يقوم بطريقة معينة لتسريع وبنفس الوقت الملفات الناتجة bundles تكون احجامها قليلة، وهي بشكل افتراضي مفعلة بالقيمة true وتستطيع ابطالها بوضع false ولكن لا يُفضل عمل ذلك إلا في اضيق الحدود مثلاً احتجت إلى مكتبة خارجية وهذه المكتبة ضرورية جداً في مشروعك ولكنها تتعارض مع هذا النظام (بسبب انه نظام جديد) لذلك قم بإسناد القيمة false، وسوف يقوم angular بتعطيل هذا النظام والرجوع إلى النظام القديم وهو webpack.

اما الاسطر 22 و23 و24 فيتم تحديد مسار ملف ايقونة التطبيق ومجلد assets.

اما 26 و27 فنحدد ملف styles.css الذي أشرنا له سابقاً حيث نقوم بتعريفه هنا على شكل مصفوفة ممكن ان نعرف فيها أكثر من ملف تنسيق.

اما السطر 29 فنحدد فيه ملفات الجافاسكربت من مكتبات خارجية نُريد تضمينها في مشروعنا مثل JQuery على سبيل المثال.

اما الاسطر 31 إلى 57 فعند طريقها نستطيع اجراء بعض الإعدادات اثناء التطوير على المشروع، مثلاً ما هو موجود في الاسطر 34 و35 حيث يقوم باستبدال ملف environment الخاص بالنسخة النهائية للتطبيق بملف environment الخاص بنسخة التطوير للتطبيق.

7.3.4.ملف karma.conf.ts:

وهذا الملف فيه بعض إعدادات مكتبة karma وهي المكتبة التي يستخدمها angular في عمل testing للمشاريع الخاصة به.

8.3.4.ملفي package-local.json وpackage.json:

ملف package-local.json هو باختصار ملف تفصيلي لملف package.json، لذلك سوف نكتفي بشرح الملف الأخير، لذلك في البداية سوف نستعرض هذا الملف ثم من بعدها سوف نشرح اهم اجزاءه، كالتالي:

```
{
  "name": "my-project",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "test": "ng test",
    "lint": "ng lint",
    "e2e": "ng e2e"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "~10.1.5",
    "@angular/common": "~10.1.5",
    "@angular/compiler": "~10.1.5",
    "@angular/core": "~10.1.5",
    "@angular/forms": "~10.1.5",
    "@angular/platform-browser": "~10.1.5",
    "@angular/platform-browser-dynamic": "~10.1.5",
    "@angular/router": "~10.1.5",
    "rxjs": "~6.6.0",
    "tslib": "^2.0.0",
    "zone.js": "~0.10.2"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.1001.5",
    "@angular/cli": "~10.1.5",
    "@angular/compiler-cli": "~10.1.5",
    "@types/node": "^12.11.1",
    "@types/jasmine": "~3.5.0",
    "@types/jasminewd2": "~2.0.3",
    "codelyzer": "^6.0.0",
    "jasmine-core": "~3.6.0",
    "jasmine-spec-reporter": "~5.0.0",
    "karma": "~5.0.0",
    "karma-chrome-launcher": "~3.1.0",
    "karma-coverage-istanbul-reporter": "~3.0.2",
```

```
    "karma-jasmine": "~4.0.0",
    "karma-jasmine-html-reporter": "^1.5.0",
    "protractor": "~7.0.0",
    "ts-node": "~8.3.0",
    "tslint": "~6.1.0",
    "typescript": "~4.0.2"
  }
}
```

في البداية لابد ان نبين ان هذا الملف ليس خاص بمشاريع angular وانما هو ملف لأي مشروع من مشاريع Javascript مثل node.js او vanilla javascript، وهو يعتبر كأنه ملصق تعريفي يحتوي على بعض المعلومات عن المشروع الخاص بك مثل الاسم والاصدار ووصف للمشروع وغيره، وبالنسبة لمشاريع Angular هنالك بعض الإضافات منها scripts وهي عبارة عن طريقة نختصر بها الأوامر الطويلة ببعض الأوامر المختصرة، فمثلاً لإنشاء component جديد بشرط ان لايتم انشاء مجلد له ولا يُنشئ ملف test، لابد ان نكتب الأمر التالي: ng generate component --flat --skip-tests ولنفرض اننا نستخدم هذا الأمر بكثرة فعندئذ نستطيع اختصار هذا الامر باسم مختصر نختاره نحن ونضعه في الكائن scripts في ملف package.json، وليكن هذا الأسم mkc، كالتالي:

```
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "test": "ng test",
    "lint": "ng lint",
    "e2e": "ng e2e",
    "mkc": "ng generate component --flat --skip-tests",
  },
```

وفي terminal بدلاً من أن نكتب الأمر الطويل السابق نكتفي بكتابة الأمر npm run mkc c1 حيث c1 يمثل اسم component الجديد.

اما dependencies فهذا الكائن يحتوي على جميع المكتبات سواء الداخلية او الخارجية المضمنة ضمن المشروع لديك، بحيث لو افترضنا اننا قمنا برفع المشروع الخاص بنا على GitHub او رفعناه على رابط معين ولشخص آخر لكي يطلع عليه، ففي هذه الحالة لقد أشرنا سابقاً ان ملف node_modules لا يتم رفعه لكبر حجمه ولكن الشخص الآخر الذي تم ارسال هذا المشروع له لا يستطيع تشغيله إلا بوجود هذا المجلد لذلك جاءت فكرة dependencies حيث أي مكتبة خارجية او داخلية من ضمن angular عند تثبيتها سوف يتم وضع اسمها والاصدارة الخاصة بها هنا في هذا الملف، وعندما تقوم بإرسال مشروعك لشخص آخر بمجرد ان يكتب في terminal الخاص به npm install (**بشرط ان يكون بنفس مجلد المشروع**) سوف يتم قراءة هذه المكتبات من هذا الملف وتحميلها جميعاً وتضمينها في مشروعك.

اما divDependencies فهي مشابهه لي dependencies والفرق الوحيد ان divDependencies يوجد بها المكتبات التي نحتاجها اثناء التطوير فقط اما dependencies فيوجد بها الملفات التي يحتاجها المشروع بنسخته النهائية، اما كيف نضيف المكتبات الخارجية سواء في divDependencies او dependencies، فنؤجل الكلام بها إلى ان نصل إلى الفصل الخاص بإضافة مكتبات خارجية إلى مشاريع Angular.

الفصل الخامس

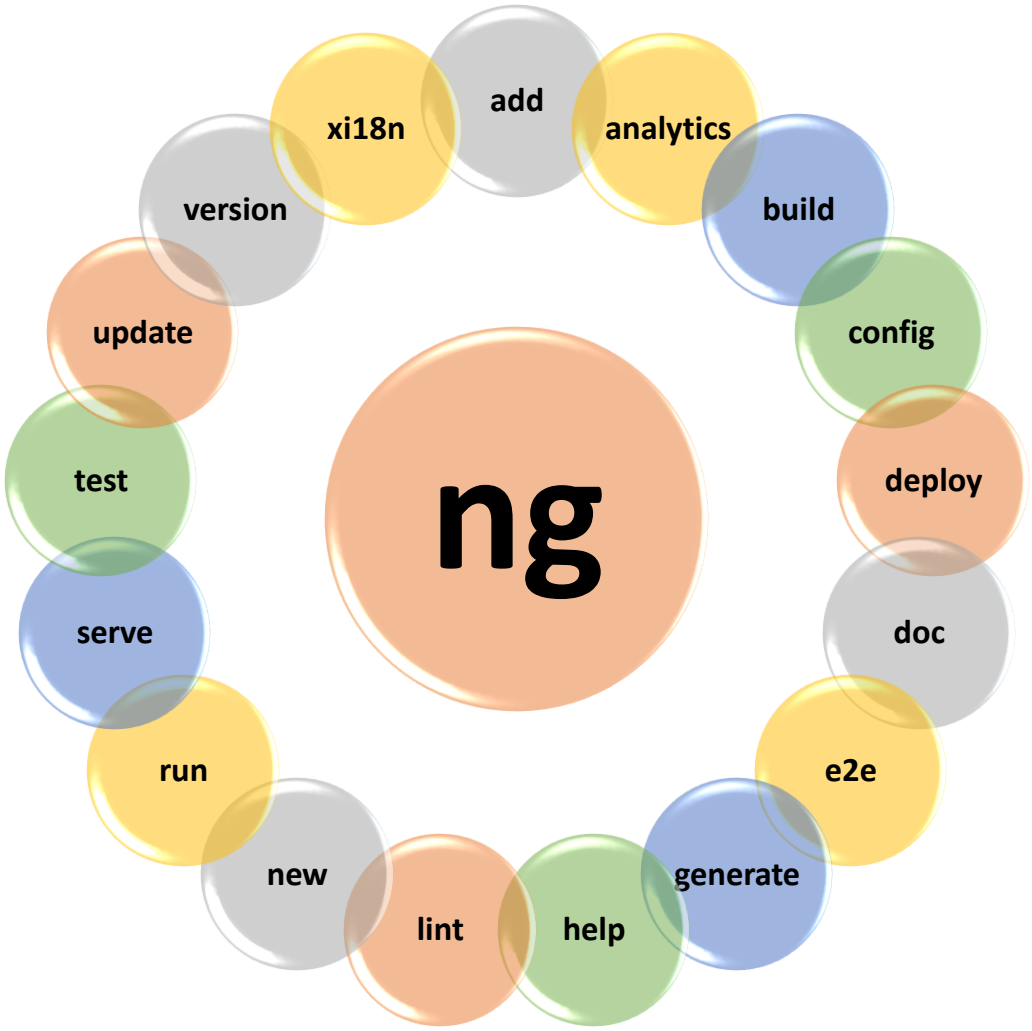
Angular CLI

1.5.مقدمة:

هي طريقة اتاحتها لنا Angular لتعامل مع المشاريع بطريقة سهلة وميسرة من خلال مجموعة من الأوامر التي نكتبها في terminal وتؤدي جميع المهام من إنشاء مشروع جديد إلى بناء هذا المشروع بنسخته النهائية.

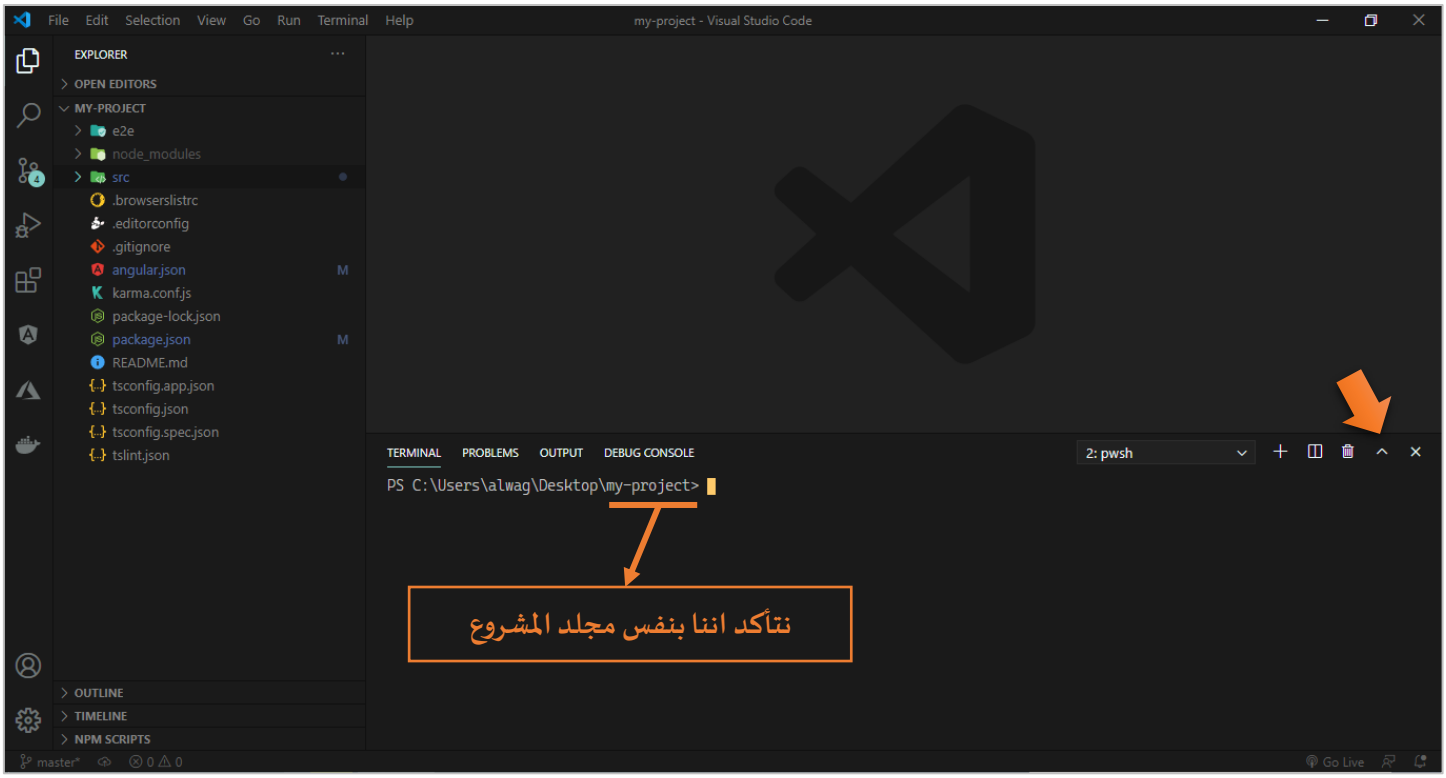
وقد تعلمنا سابقاً كيف نقوم بتحميل وتثبيت Angular CLI عندما تكلمنا عن انشاء مشروع Angular جديد، وايضاً تعاملنا مع امر من هذه الأوامر في السابق وهو الامر new الذي يُتيح لنا انشاء مشروع جديد، ولكن لم اقم بشرحه وذكر بعض البارامترات التي يمكن ان نمررها إليه، وهو ما سوف نفعله بإذن الله في هذا الفصل.

وسوف اعتمد في شرح هذه الأوامر على الموقع الرسمي Angular وهو https://angular.io/cli، لذلك عزيزي المتعلم يجب عليك ان تطلع على هذا الرابط في حال اردت الإلمام بجميع هذه الأوامر، مع انني لا انصح بمثل هذا الأمر لأنه من الصعوبة حفظها جميعاً مع البارامترات التي نمررها لها ولكن نتعلم الأساسيات وباقي الأوامر تستطيع تعلمها في حال احتجت لها حيث تقوم بالذهاب إلى الرابط والاطلاع على الأمر وكتابة في مشروعك، ويجب ان اشير ايضاً ان هنالك أوامر قد لا تحتاج لها نهائياً، ولا يمنع من معرفتها لوقت الحاجة، اما من ناحية الأوامر الرئيسية فيمكن حصرها، بالشكل التالي:

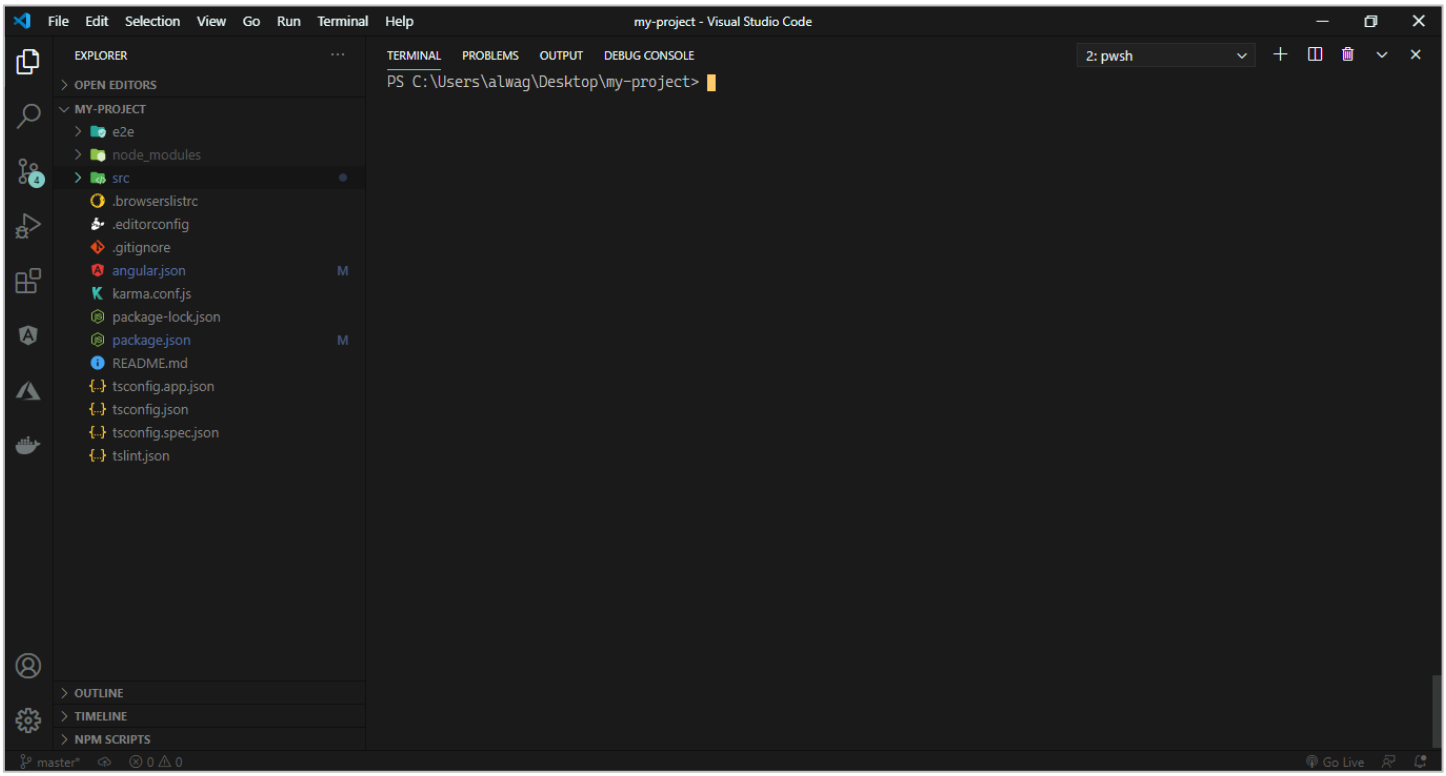


2.5.ng help:

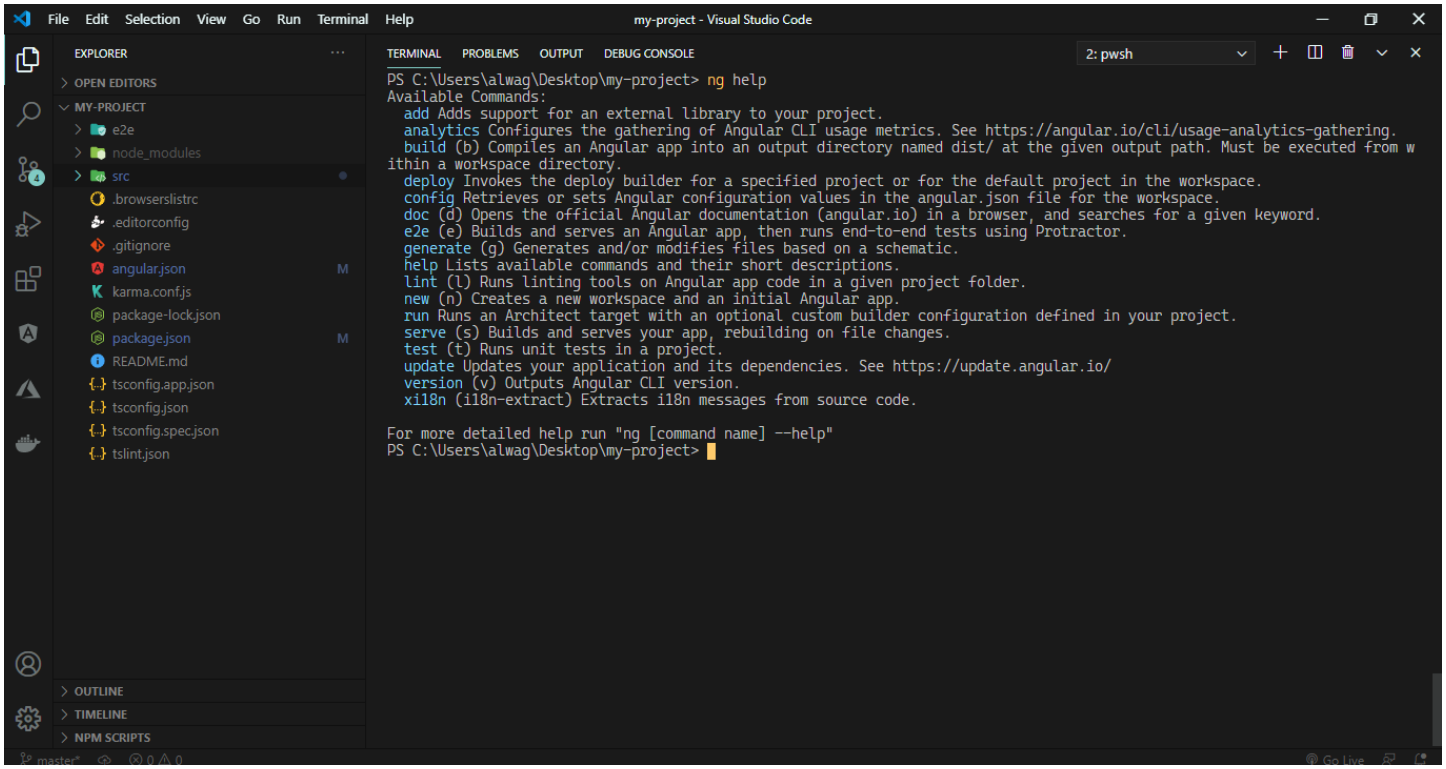
عن طريق هذا الأمر نستطيع الاطلاع على جميع الأوامر الرئيسية في CLI، لذلك سوف نذهب إلى نفس المشروع الذي قمنا بإنشائه في الفصل السابق وبعدها نفتح شاشة terminal كما تعلمنا سابقاً، كالتالي:



لابد ان نتأكد أننا بنفس مجلد المشروع، وبنفس الوقت لنقوم بالضغط على الزر المؤشر عليه بالسهم في الشكل السابق لكي نقوم بتكبير terminal، بحيث يصبح شكله كالتالي:



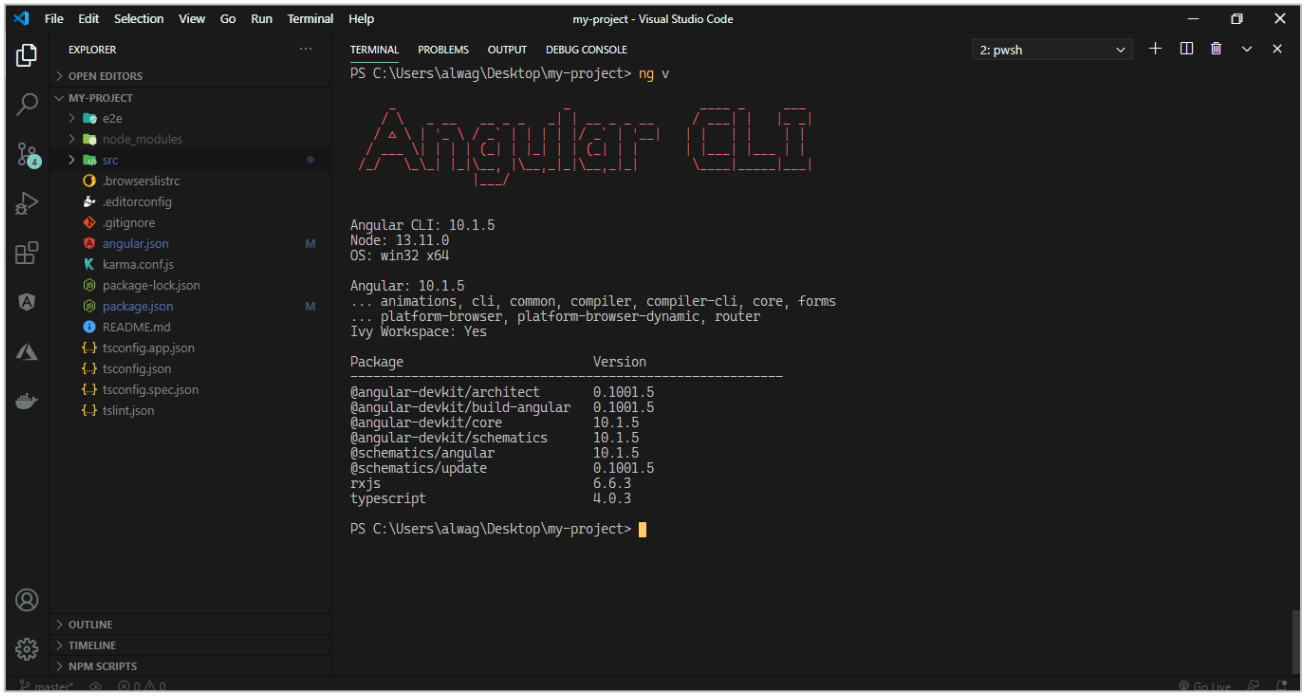
الآن لنقوم بكتابة الامر ng help في terminal ومن ثم نضغط زر Enter من لوحة المفاتيح وننتظر قليلاً إلى ان يتم عرض جميع الأوامر المتاحة مع تعريف لكل أمر واختصار للأمر أن وجد، كالتالي:



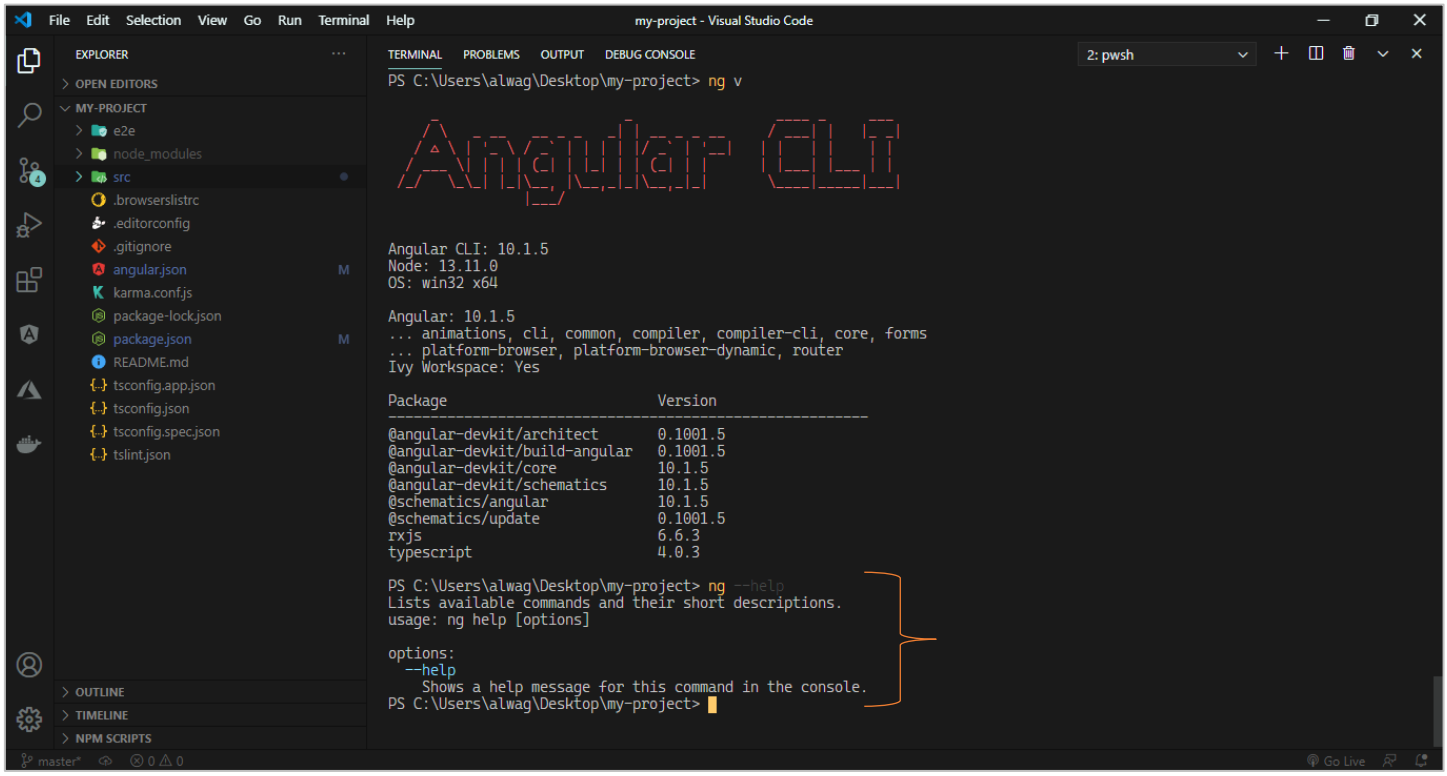
نلاحظ ظهرت لنا جميع الأوامر المتاحة مع الاختصار لبعض الأوامر كالأمر build والاختصار الخاص به هو b او الامر serve والاختصار الخاص به هو s، وايضاً هنالك شرح مبسط لهذه الأوامر.

ng version.3.5 :

عن طريق هذا الامر نستطيع معرفة رقم الاصدارة لإطار عمل Angular بالإضافة لجميع المكتبات المدمجة معها، واختصار هذا الامر هو v لذلك تستطيع كتابة الامر ng v او ng version، كالتالي:



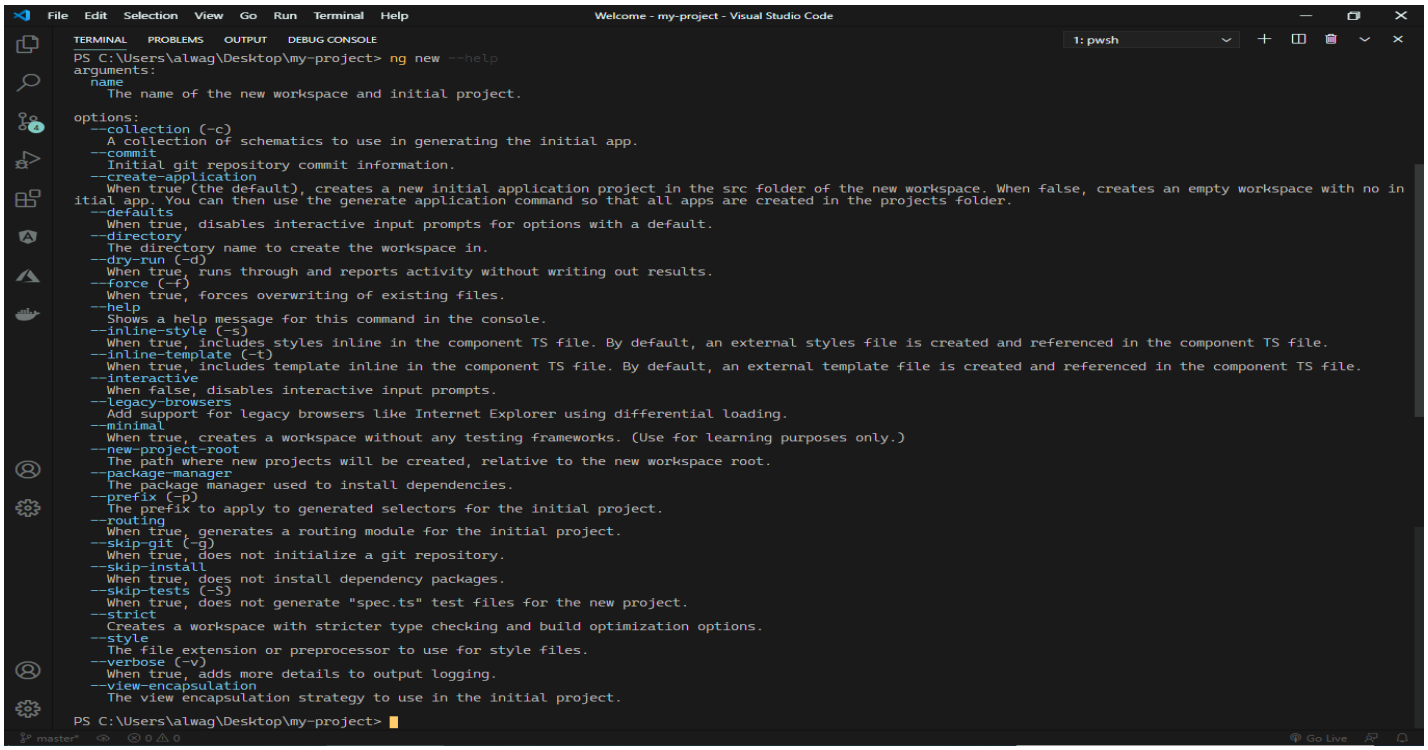
كما ان هذا الامر يأخذ option واحد ويسمى ايضاً flag وهو help-- (شرطتين ومن ثم اسم الخيار option) وهذا الخيار يعطينا معلومات أكثر عن هذا الأمر وماذا إذا كان هنالك options أخرى او بارامترات، كالتالي:



نلاحظ ان هذا الامر لا يحتوي إلا على option واحد فقط وهو help--

ng new.4.5 :

وهذا الامر تعاملنا معه في السابق حيث يقوم بإنشاء مشروع Angular جديد، لذلك سوف نقوم بكتابة هذا الامر ومن ثم option ذو الاسم help-- (هذا الخيار يعمل مع جميع الأوامر) ولنرى النتيجة، كالتالي:



نلاحظ ظهرت لنا مجموعة من options التي يمكن ان نستخدمها مع هذا الأمر وايضاً ظهرت لنا arguments او بصيغة أخرى بارامتر واحد يستقبله هذا الأمر وهو name والمقصود به اسم المشروع بحيث يكون بالشكل التالي ng new name حيث يتم استبدال الكلمة name باسم المشروع الخاص بك، وهو الذي قمنا بفعله سابقاً عندما انشأنا مشروع angular جديد حيث كتبنا الامر ng new my-project، هذا من جهة اما من جهة options التي ظهرت لنا في الشكل السابق سوف اشرح أهمها بالجدول التالي:

الأمر	الشرح	الصيغة
--create-application	القيمة الافتراضية له true ونستخدمه في عدة حالات منها ان يكون لدينا أكثر من مشروع لذلك نريد ان يقوم بإنشاء Workspace ولكن لا يُنشئ المشروع لذلك نضع قيمته false، ولأنشاء المشروع نستخدم الامر generate والذي سوف نتكلم عنه لاحقاً	ng new name **--create-application=false | true** (حيث العلامة | تعني OR)
--directory	هذا الخيار نتحكم فيه باسم المجلد الخاص بالمشروع والقيمة الافتراضية له هي اسم المشروع ويمكن تغييره بالاسم الذي نريده في حال الرغبة بذلك	ng new name --directory="any name"
--inline-style	في هذا الخيار إذا كانت قيمته true نلغي إضافة ملف تنسيقات css لأي component في المشروع ونكتفي بكتابة التنسيقات بنفس ملف class او يسمى ملف ts لي component والاختصار له هو -s	ng new name --inline-style=false | true ng new name -s=false | true
--inline-template	هذا الخيار مشابه للخيار السابق والفرق هنا يختص بملف template او ما يسمى ملف HTML والاختصار له -t	ng new name --inline-template=false | true ng new name -t=false | true
--minimal	هذا الخيار في حال كانت قيمته true فإنه سوف يتم انشاء المشروع بدون أي مكتبات او اطر عمل لي testing مثل karma	ng new name --minimal=false | true
--prefix	هذا الخيار لتحديد prefix لأسماء components التي اشرنا لها سابقاً	ng new name --prefix="any prefix"

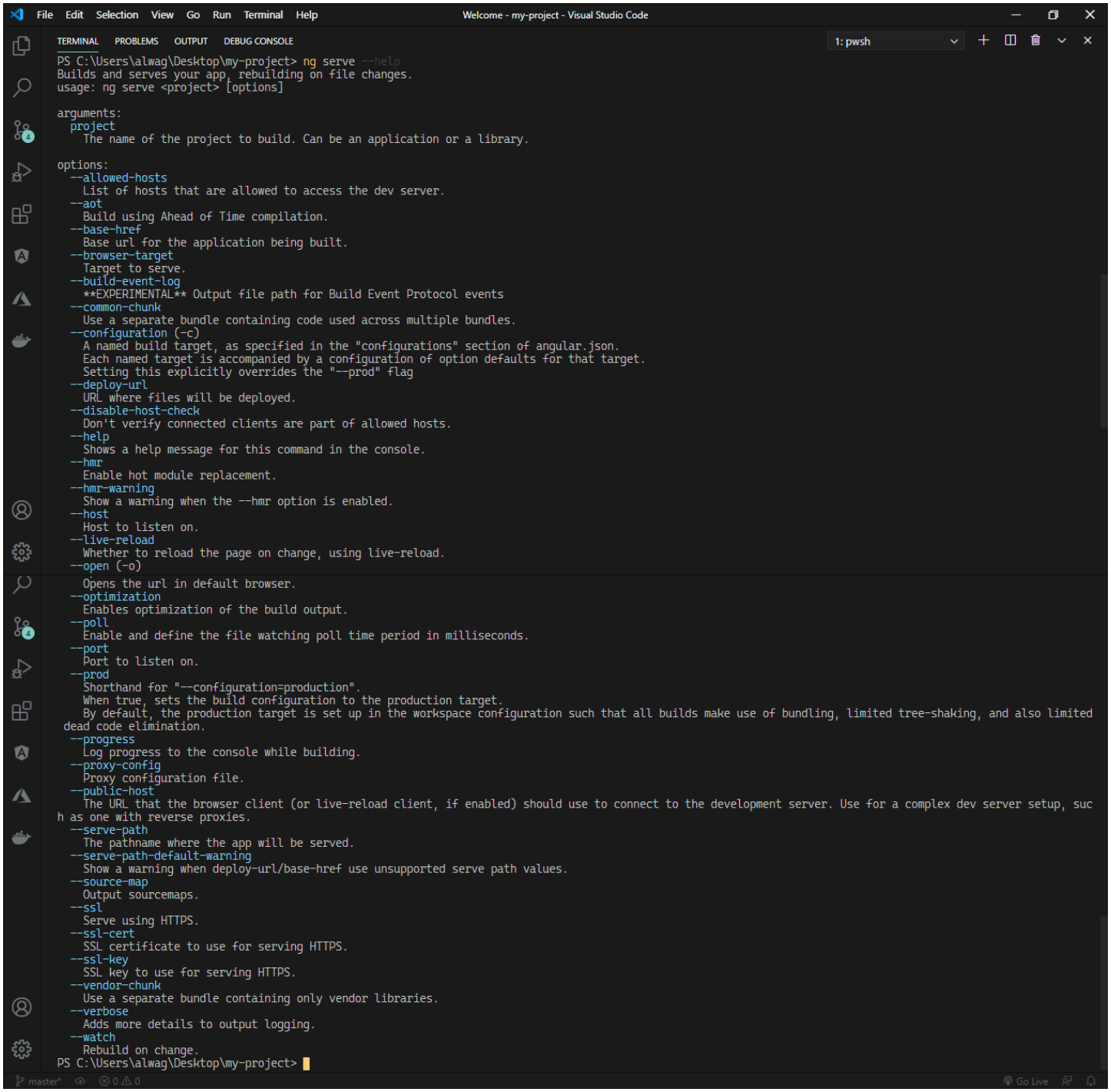
الأمر	الشرح	الصيغة
	بحيث ان لم يتم تغييرها فإن القيمة الافتراضية لها هي app	
--package-manager	هنا نحدد أي package manager نريد ان نستخدمه لتعامل مع dependencies وهي المكتبات الخارجية التي نقوم بتضمينها في مشروعنا سواء في مرحلة البناء او التطبيق النهائي، وهي تأخذ مجموعة من القيم npm-yarn-xnpm-pnpm-cnpm	ng new name --package-manager=npm | yarn | xnpm | pnpm | cnpm
--routing	هنا نحدد هل يتم إدراج ملف routing في المشروع وفي حال عدم كتابة هذا الخيار فسوف تأتيك رسالة تطالبك بتحديد هل يتم ادراج هذا الملف ام لا والاجابة (y) للأدراج و(n) لعدم الأدراج، كما فعلنا في الفصل السابق عندما اضفنا مشروع جديد	ng new name --routing=true | false
--dry-run	هذا الأمر يقوم بمحاكاة انشاء مشروع Angular جديد بجميع الملفات بحيث يعرضها جميعاً في terminal ولكن لن يُنشئها حقيقة، وانما يقوم فقط بمحاكاة انشاء هذه الملفات، وهذه الطريقة لها مجموعة من الفوائد فمثلاً قمت بإنشاء مشروع Angular جديد عن طريق الأمر new واضفت مجموعة من الخيارات ففي هذه الحالة تريد ان ترى كيف وماهي الملفات قبل ان يتم انشائها بشكل واقعي، واختصار الامر هو d-	ng new name --dry-run=true | false
--skip-git	هذا الخيار يسمح بإنشاء مجلد git في حال كان هذا المشروع سوف يتم رفعه على GitHub وتكون القيمة true في حال عدم السماح بإنشاء هذا المجلد والاختصار الخاص به هو g-	ng new name --skip-git=true | false

الأمر	الشرح	الصيغة
--skip-install	هذا الخيار يسمح او يمنع تحميل dependencies وهي المكتبات الإضافية في node_modules والقيمة الافتراضية له false	ng new name --skip-install=true | false
--skip-tests	هذا الخيار يسمح او يمنع إضافة ملف spec.ts او بصيغة أخرى ملف testing الخاص component بحيث يُنشئ فقط ملف css وملف html وملف ts، والفرق بينه وبين minimal ان هنا يتم إضافة جميع المكتبات وأطر العمل الخاصة بعمل testing للمشروع ولكن لا يضيف ملف spec.ts لأي component بشكل تلقائي عند إنشاء هذا component وانما في حال الرغبة بإضافة هذا الملف لأي component معين فنستطيع اضافته عن طريق الامر generate والتي سوف نتكلم عنها لاحقاً بإذن الله	ng new name --skip-tests=true | false
--strict	إذا كانت قيمته true سيتم تطبيق strict mode وهو وضع يجبر المطورين على تحديد نوع المتغير عند تعريفه	ng new name --strict=true | false
--style	نفس ما قيل في routing يُقال هنا والفرق ان هذا الخيار يحدد نوع التنسيق لملفات styles، والقيمة التي يتم اسنادها له هي css | scss | sass | less | style	ng new name --style=css | scss | sass | less | style
--view-encapsulation	وهذا الأمر يتعامل مع طريقة تعامل مشروعك مع DOM او ما يسمى تغليف هذا DOM وتأخذ مجموعة قيم Emulated|Native|None|ShadowDom ويُفضل ابقائها على الوضع الافتراضي وهو Emulated، مع العلم انه تم شرح هذه الجزئية بالتفصيل في كتاب Angular Components and Services	ng new name --view-encapsulation = Emulated | Native | None | ShadowDom

ng serve.5.5 :

هذا الامر يقوم بصيغة مبسطة بتشغيل التطبيق الخاص بك على المتصفح وليس بنائه بشكله النهائي، حيث يقوم بتجميع جميع ملفات المشروع في الذاكرة وتهيئتها لكي تعمل على المتصفح، ويتم استخدام هذا الامر لكي نقوم بتجربة التطبيق اثناء عملية التطوير، وبشكل افتراضي يكون host الخاص به هو localhost وport هو 4200، بمعنى انه يتم تشغيل التطبيق في المتصفح على الرابط التالي: http://localhost:4200/

اما الآن لنقوم بكتابة الأمر ng serve –help لنستعرض جميع arguments وoptions الخاصة بهذا الأمر، كالتالي:



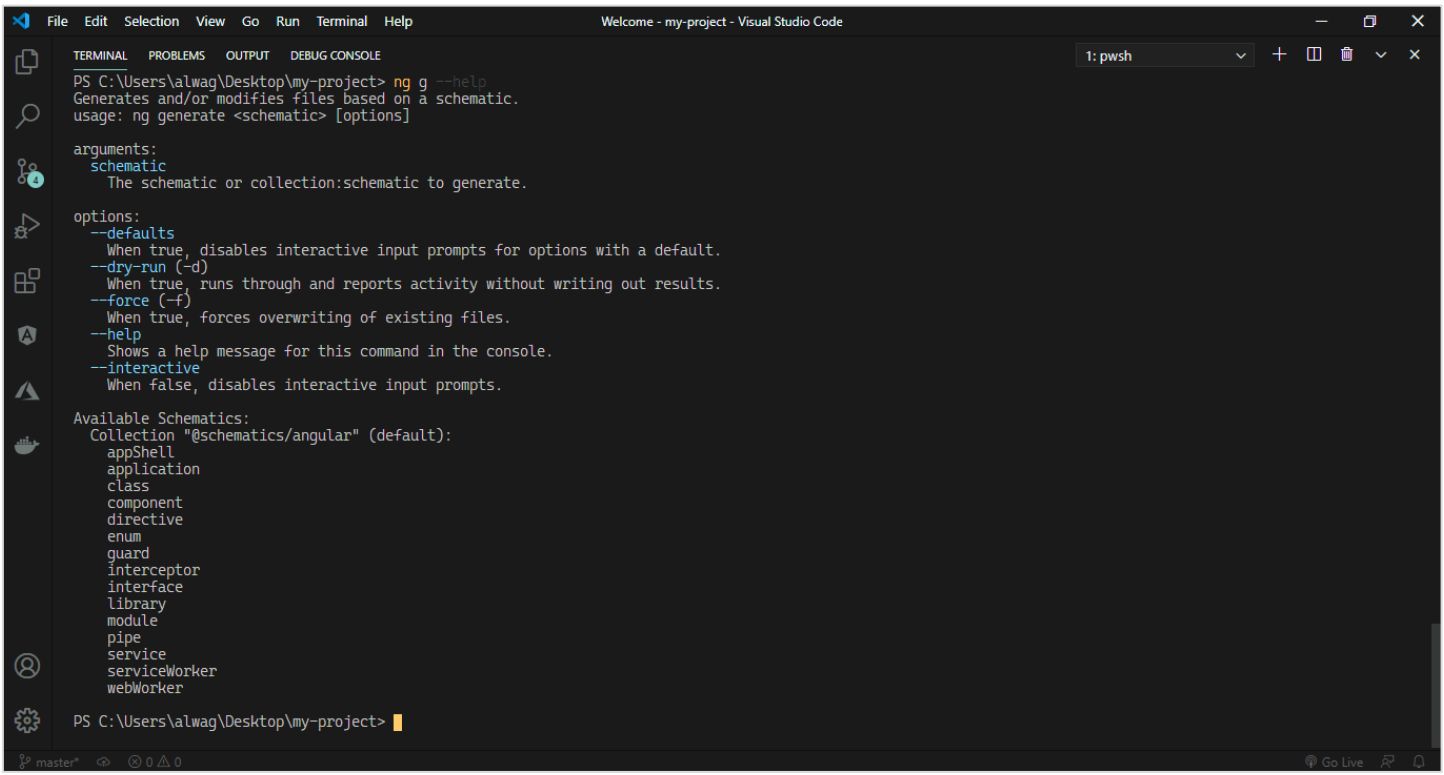
كما هو واضح من الشكل السابق الكم الهائل من options التي ممكن ان نستخدمها مع هذا الأمر وبطبيعة الحال لن يتم استعراضها جميعاً وانما سوف اشرح أهمها، اما من ناحية arguments فهو واحد فقط عبارة عن اسم المشروع مع العلم ان الافتراضي له هو اسم المشروع الخاص بك بمعنى لو لم تقم عزيزي المتعلم بكتابة اسم المشروع فلا ضير سوف يقوم بقراءة المشروع الافتراضي ويقوم بتشغيله، اما السبب في وجود هذا argument هو في حال وجود أكثر من مشروع فتستطيع عندها كتابة اسم المشروع المستهدف.

ونرجع مرة أخرى لي options الخاصة بهذا الأمر، والتي سوف نستعرض أهمها في هذا الجدول، كالتالي:

الأمر	الشرح	الصيغة
--open	بعدما عملنا serve للمشروع يقوم هذا الخيار بفتح المتصفح الافتراضي لديك، والاختصار الخاص به هو -o	ng serve --open ng s -o
--port	نستطيع عن طريق هذا الخيار تغيير port الافتراضي وهو 4200 ونستخدم هذا الامر في حال كان لدينا اكثر من مشروع ونريد ان نعمل لها serve	ng serve --port=8088
--host	اما هذا الخيار فنقوم بتغيير host الافتراضي وهو localhost إلى أي host آخر ومن فوائده على سبيل المثال هو استخدام public IP الخاص بشبكة LAN الخاص بك لكي نستطيع تشغيل وتجربة هذا التطبيق على جميع الأجهزة والتأكد من عملهم بدون أي مشاكل	ng serve --host=0.0.0.0
--watch	وهذا الخيار بشكل افتراضي تكون قيمته true وهو مراقبة أي تغييرات تحدث بالتطبيق وتطبيقها على النسخة التي تم عمل serving لها في الذاكرة.	ng serve --watch=true | false
--live-reload	هذا الخيار في حال تم الكشف التغييرات عن طريق الخيار watch-- فإنه يقوم بتحديث المتصفح تلقائياً لكي يقوم بتطبيق هذه التعديلات عليه، مع العلم ان القيمة الافتراضية له هي true	ng serve --live-reload=true | false
--prod	هذا الخيار مفيد في حال اردت ان تعرف شكل الملفات التي سوف تخرج في التطبيق بشكله النهائي بالطبع نستخدم هذا الخيار للاختبار والتجربة فقط	ng serve --prod=true | false
--aot	هذا الخيار قيمته الافتراضية في الأمر build والخاص بإخراج التطبيق بشكله النهائي هو true اما هنا قيمته هي الافتراضية false ونستخدمه هنا لمحاكاة شكل الملفات باستخدام هذه التقنية لمعرفة حجم وماهية الملفات الناتجة (لتجربة والاختبار فقط)	ng serve --aot=true | false

ng generate.6.5 :

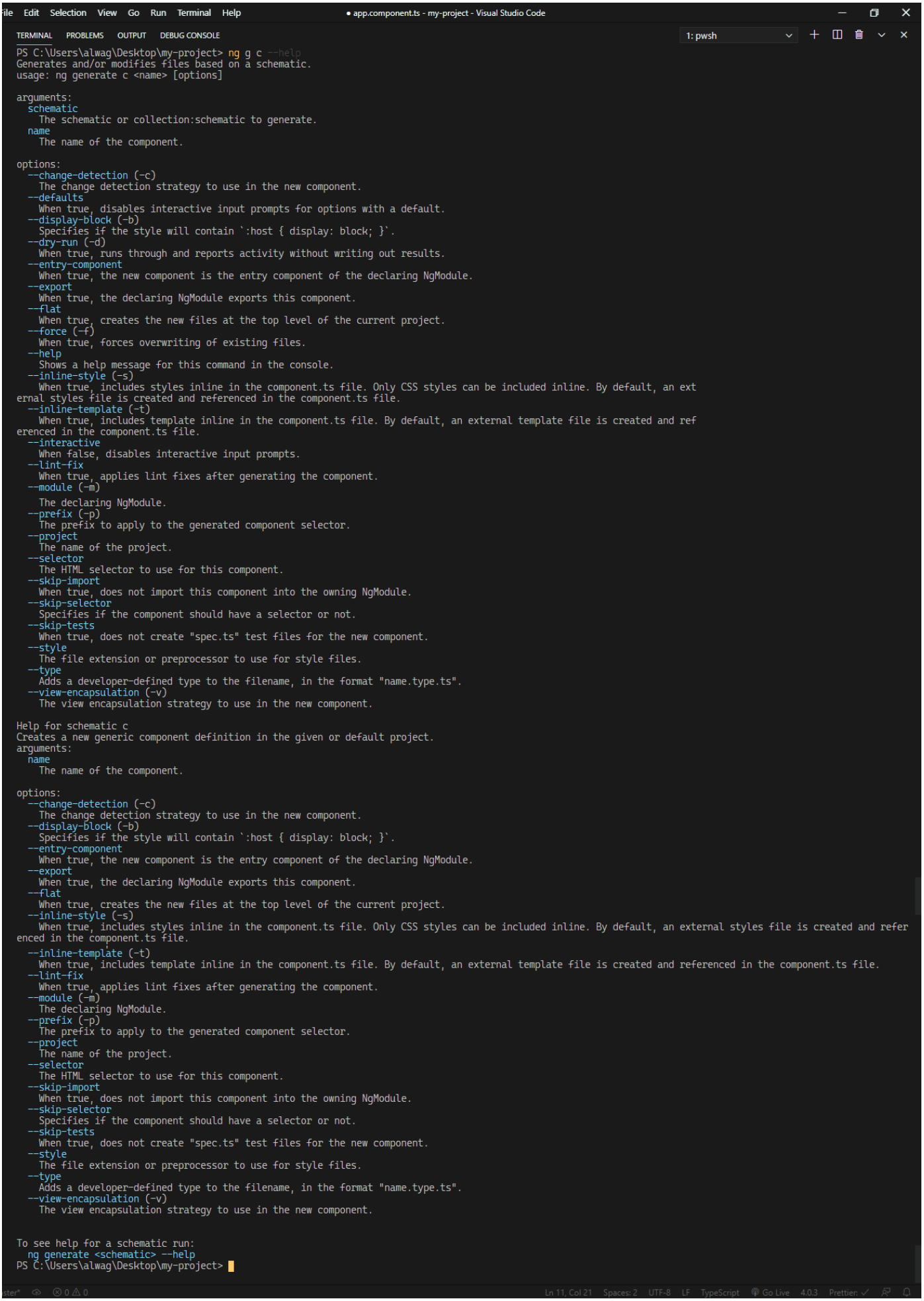
ولعل هذا الأمر من أكثر الأوامر استخداماً من قبل مطوري Angular حيث عن طريقه نقوم بالتحكم وإضافة أي ميزة إلى المشروع، مثل إضافة component او directive او ...الخ، وبطبيعة جميع هنا سوف نتعلم كيف نقوم بإنشاء وإضافة هذه الميزات إلى المشروع اما كيفية التعامل معها فسوف يتم التطرق لها في الكتب الأخرى من هذه السلسلة بإذن الله، واختصار هذا الامر هو (g)، وكما جرت العادة لنكتب الخيار help-- لنعرف جميع الخيارات وschematic، كالتالي:



في هذا الجزء لن نركز على الخيارات options وانما سوف نركز على schematics او بصيغة أخرى المميزات التي نُريد اضافتها إلى المشروع مع التطرق للخيارات options الخاصة بهذا schematic، كالتالي:

ng generate component .1.6.5:

هذا schematic نستطيع عن طريقة انشاء components المختلفة، والاختصار له c، اما argument فهو يأخذ argument واحد وهو اسم component الذي نُريد انشائه ويرمز له بالرمز name، بحيث يصبح الأمر ng generate component name او بالصيغة المختصرة ng g c name حيث name يتم استبداله باسم component الذي تُريد وضعه لهذا component ومن ناحية الخيارات فله مجموعة متعددة من الخيارات options نستعرضها عن طريق كتابة الخيار help-- بعد الأمر ng g c name، كالتالي:


File Edit Selection View Go Run Terminal Help
• app.component.ts - my-project - Visual Studio Code
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
1: pwsh
PS C:\Users\alwag\Desktop\my-project> ng g c --help
Generates and/or modifies files based on a schematic.
usage: ng generate c <name> [options]
arguments:
schematic
The schematic or collection:schematic to generate.
name
The name of the component.
options:
--change-detection (-c)
The change detection strategy to use in the new component.
--defaults
When true, disables interactive input prompts for options with a default.
--display-block (-b)
Specifies if the style will contain `:host { display: block; }`.
--dry-run (-d)
When true, runs through and reports activity without writing out results.
--entry-component
When true, the new component is the entry component of the declaring NgModule.
--export
When true, the declaring NgModule exports this component.
--flat
When true, creates the new files at the top level of the current project.
--force (-f)
When true, forces overwriting of existing files.
--help
Shows a help message for this command in the console.
--inline-style (-s)
When true, includes styles inline in the component.ts file. Only CSS styles can be included inline. By default, an external styles file is created and referenced in the component.ts file.
--inline-template (-t)
When true, includes template inline in the component.ts file. By default, an external template file is created and referenced in the component.ts file.
--interactive
When false, disables interactive input prompts.
--lint-fix
When true, applies lint fixes after generating the component.
--module (-m)
The declaring NgModule.
--prefix (-p)
The prefix to apply to the generated component selector.
--project
The name of the project.
--selector
The HTML selector to use for this component.
--skip-import
When true, does not import this component into the owning NgModule.
--skip-selector
Specifies if the component should have a selector or not.
--skip-tests
When true, does not create "spec.ts" test files for the new component.
--style
The file extension or preprocessor to use for style files.
--type
Adds a developer-defined type to the filename, in the format "name.type.ts".
--view-encapsulation (-v)
The view encapsulation strategy to use in the new component.
Help for schematic c
Creates a new generic component definition in the given or default project.
arguments:
name
The name of the component.
options:
--change-detection (-c)
The change detection strategy to use in the new component.
--display-block (-b)
Specifies if the style will contain `:host { display: block; }`.
--entry-component
When true, the new component is the entry component of the declaring NgModule.
--export
When true, the declaring NgModule exports this component.
--flat
When true, creates the new files at the top level of the current project.
--inline-style (-s)
When true, includes styles inline in the component.ts file. Only CSS styles can be included inline. By default, an external styles file is created and referenced in the component.ts file.
--inline-template (-t)
When true, includes template inline in the component.ts file. By default, an external template file is created and referenced in the component.ts file.
--lint-fix
When true, applies lint fixes after generating the component.
--module (-m)
The declaring NgModule.
--prefix (-p)
The prefix to apply to the generated component selector.
--project
The name of the project.
--selector
The HTML selector to use for this component.
--skip-import
When true, does not import this component into the owning NgModule.
--skip-selector
Specifies if the component should have a selector or not.
--skip-tests
When true, does not create "spec.ts" test files for the new component.
--style
The file extension or preprocessor to use for style files.
--type
Adds a developer-defined type to the filename, in the format "name.type.ts".
--view-encapsulation (-v)
The view encapsulation strategy to use in the new component.
To see help for a schematic run:
ng generate <schematic> --help
PS C:\Users\alwag\Desktop\my-project>
Ln 11, Col 21 Spaces: 2 UTF-8 LF TypeScript Go Live 4.0.3 Prettier:

ولأهمية هذا الأمر وكثيراً ما يستخدمه مطوري Angular فسوف أقوم بالتفصيل فيه حيث يحتوي على كمية كبيرة من الخيارات التي تسهل انجاز المهام من جهة ومن جهة أخرى تُلبي احتياجات المطور، مع العلم ان بعض هذه الخيارات مررنا عليها من خلال شرحنا للأوامر السابقة، كالتالي:

الأمر	الشرح	الصيغة
--change-detection	عن طريق هذا الخيار نستطيع تحديد نظام Change Detection الخاص بإطار عمل Angular لهذا component الذي نُريد انشائه وبشكل افتراضي يأخذ القيمة Default.	ng g c name --change-detection=OnPush | Default
--dry-run	تم الاشارة لها سابقاً مع الأمر new	
--entry-component	هنا نقوم بتسجيل component الذي نُريد انشائه في NgModule الرئيسي على انه Entry Component وغالباً يكون لي Component الديناميكية	ng g c name --entry-component=true | false
--export	هذا الخيار يسمح لنا بعمل export لي component الذي نريد انشائه، وغالباً هذا نستخدم هذا الخيار في حال كان لدينا اكثر من Modules فنستطيع عندها وضع مجموعة من components المتشابهة مع بعضها ويتم تعريفها في Module فرعي وهذا Module يتم تعريفه في Module الرئيسي وبنفس الوقت نقوم بعمل Export لهذه Component لكي نستطيع استخدامها مع components أخرى خارج أطار هذه المجموعة المتشابهة	ng g c name --export=true | false
--flat	يتم استخدام هذا الخيار option في حال أردنا ان يتم انشاء component بدون مجلد فقط يقوم بإنشاء ملفات component لأنه بشكل افتراضي عندما لم نكتب هذا الخيار فإن angular سوف يقوم بإنشاء مجلد بنفس اسم component ووضع الملفات بداخله	ng g c name --flat=true | false

الأمر	الشرح	الصيغة
--force	في حال كان لديك component سابق وأدرت ان تُنشأ component جديد بنفس الاسم فإن Angular لن يسمح لك بذلك ولكن عن طريق هذا الخيار تستطيع استبدال component الجديد بدلاً من component القديم بشرط ان يكونا بنفس المسار ونفس الاسم	ng g c name --force=true | false
--inline-style	تم التطرق لها سابقاً ولكن هنا على مستوى component وليس على مستوى كافة المشروع	
--inline-template	تم التطرق لها سابقاً ولكن هنا على مستوى component وليس على مستوى كافة المشروع	
--module	نستخدم هذا الامر في حال اردنا ان نقوم بتسجيل component الجديد في module معين، لأنه افتراضياً سوف يتم التسجيل في Root Module او ما يسمى ملف app.module.ts	ng g c name --module=module name
--prefix	تم التطرق لها سابقاً ولكن هنا على مستوى component وليس على مستوى كافة المشروع	
--lint-fix	لتعديل كافة الأخطاء التي تنتج من tslint مع العلم انه قد لا يقوم بتعديل كافة الأخطاء وانما بعضها قد يحتاج إلى تعديلها يدوياً	ng g c name --lint-fix
--selector	بشكل افتراضي يتم تسمية selector الخاص بأي component جديد بنفس اسم component مُضاف إليه app وفي حال اردنا ان نغير هذا selector بأي اسم نريده فنستخدم هذا الخيار، وبالطبع سوف يُضاف له app بشكل تلقائي مالم نُعدل هذه القيمة عن طريق الخيار prefix-- كما أشرنا سابقاً	ng g c name --selector=any name

الأمر	الشرح	الصيغة
--skip-import	هذا الخيار يمنه إضافة هذا component الجديد إلى أي module	ng g c name --skip-import=true | false
--skip-selector	يمنع من إضافة أي selector لي هذا component الجديد	ng g c name --skip-selector=true | false
--skip-tests	يمنع من إضافة ملف test او ما يسمى ملف spec لهذا component الجديد	ng g c name --skip-tests=true | false
--style	تم التطرق لها سابقاً ولكن هنا على مستوى component وليس على مستوى كافة المشروع	

وأخيراً يجب التنويه انه في حال أردنا ان ننشي مجلد وليكن مثلاً مجلد يحتوي بداخله مجموعة من components الفرعية، فنستطيع ذلك بكتابة اسم المجلد ومن ثم / اسم component وهكذا كلما أردنا ان ننشئ ملفات فرعية أخرى، فمثلاً لو أردنا ان ننشئ مجلد وليكن اسمه abc وبداخله component باسم xyz وآخر باسم asd، فنستطيع القيام بذلك عن طريق Angular CLI من خلال كتابة الأوامر التالية:

ng g c abc/xyz

ng g c abc/asd

ففي هذه الحالة سوف يبحث angular CLI عن المجلد الذي اسمه abc فإن لم يجده سيقوم بإنشائه ومن ثم انشاء مجلد فرعي باسم xyz وبداخله ملفات components وبنفس الطريقة مع asd.

2.6.5. ng generate application:

هذا الأمر نستخدمه في حال لم ننشأ المشروع من خلال الأمر new وذلك عن طريق إضافة الخيار --create-application والتي أشرت لها سابقاً فإذا جعلنا قيمتها false فعندئذً نضيف هذه المشروع من خلال هذه schematic وهي تأخذ argument وهو اسم المشروع والخيارات مشابهة لي الامر new.

3.6.5. ng generate (appShell – serviceWorker – webWorker):

جميع هذه الأوامر خاصة بالتعامل مع تقنيات Progressive Web Application PWA ولعلنا نفرد كتاب كامل يتكلم عن هذه التقنيات وأهميتها والأوامر الخاصة بها.

4.6.5. ng generate library:

نستطيع إضافة مكتبة وذلك بكتابة الامر السابق ومن ثم اسم المكتبة اما الخيارات فهي مشابهة لما قيل سابقاً، فقط هنا انها على مستوى مكتبة وليس component او على كامل المشروع.

5.6.5. ng generate interceptor:

وهذا الامر يقوم بإضافة ملف interceptor وهو عبارة عن ملف يساعدنا بعمل fetch للبيانات او الأخطاء عندما نتواصل مع server عن طريق مكتبة HttpClient، ولعلنا أن شاء الله سوف نُفرد كتاب لشرح هذه المكتبة وكيفية الاستفادة منها.

6.6.5. ng generate (class – enum – interface):

جميع هذه الملفات تتعلق بمفاهيم Object Oriented Programming OOP في لغة Typescript وجميعهم يستقبلون argument واحد وهو اسم هذه التقنيات، اما من ناحية الخيارات فتم التطرق إلى أغلبها سابقاً.

7.6.5. ng generate guard:

وهذا schematic يختص بالتعامل مع Angular Route Guard وقد تكلمت فيه بالتفصيل في كتاب Angular Routing and Modules، وهو يستقبل اسم هذا guard والخيارات تطرقنا لها سابقاً، إلا خيار واحد لم نتطرق له وهو --implements حيث ان guard يحتوي على أربع أنواع من الحماية وجميعها على هيئة interface، كالتالي:

```
(*) CanActivate
(*) CanActivateChild
(*) CanDeactivate
(*) CanLoad
```

لذلك نستطيع كتابة الخيار السابق ومن ثم نعطيه أحد القيم الموجودة في الشكل السابق، وفي حال لم نقم بكتابة هذا الخيار سوف يعطينا قائمة بها أربعة خيارات ونختار نوع الحماية الذي نُريده.

8.6.5. ng generate (pipe – directive):

هذين الأمرين مهمين ونستطيع عن طريقهما إضافة Pipes او Directives وكلا التقنيتين تم التطرق لهما بالتفصيل في كتاب Angular Pipes and Directives.

وكلاهما يأخذان نفس argument وهو الاسم والخيارات تم التطرق لها سابقاً ولكن هنا تكون على مستوى Directive او Pipe.

9.6.5. ng generate service:

وهذا الامر مهم وكثيراً ما يتم استخدامه وهو لإنشاء service وقد تم التكلم بالتفصيل عن services في كتاب Angular Components and Services، ومن ناحية argument فهي تأخذ الاسم والخيارات هي نفسها وتطرقنا لها سابقاً.

10.6.5. ng generate module:

أشرنا سابقاً انه يوجد على الأقل في أي مشرع module واحد وهو app.module.ts او ما يسمى Root Module وفيه يتم تعريف أي component موجود في المشروع او مكتبة خارجية او من ضمن إطار عمل Angular او ...الخ، ولكن في حال كان المشروع من متوسط إلى كبير وفي حال أردنا ان نستخدم تقنية Lazy Loading لكي نرفع من أداء التطبيق ففي هذه الحالة يُفضل تقسيم هذا Root Module إلى أجزاء ملفات او Modules صغيرة وربطها بي Root Module، او ربطها بملف route، مع العلم انه من الممكن ان تحتوي هذه modules الصغيرة هي الأخرى على modules اصغر منها بحسب كبر وتفرع المشروع الخاص بنا، ولا تقلق عزيزي المتعلم فقد تكلمت بشكل مفصل عن modules في كتاب Angular Routing and Modules تستطيع الاطلاع عليه لمعرفة جميع هذه التفاصيل وكيفية الاستفادة من هذه التقنيات، اما ما يهمنا الآن هو كيفية الاستفادة من Angular CLI لإنشاء هذه Modules الصغيرة وكيفية ربطها ببعضها البعض او ربطها بملفات routing.

وهذا الأمر يستقبل argument واحد وهو اسم هذا module الذي نُريد انشائه، اما الخيارات فقد تكلمنا عن أغلبها ما عدا هذه الخيارات التي حصرتها بهذا الجدول، كالتالي:

الصيغة	الشرح	الأمر
ng g m name -m=module path	في هذا الخيار نحدد هذا module نريد ربطه بأي module اكبر منه او بصيغة أخرى اعلى منه، فإذا أردنا ان ربطه بالRoot Module فنسند القيمة app لهذا الخيار وangular سوف يقوم بالباقي، اما إذا كان غير ذلك فلابد ان نحدد اسم هذا module ومساره بالكامل	--module
ng g m name --routing	في حال أردنا ان نضيف ملف routing لهذا module وسوف يتم تسجيل ملف routing تلقائياً بهذا module الجديد، فعندئذً نستخدم هذا الخيار	--routing
ng g m name –routing --route=route name	هذا الخيار من الخيارات الجديدة وله فائدة كبيرة في حال التعامل مع lazy loading حيث نمرر له اسم route (لك حرية اختيار الاسم الذي تُريده) وسوف يقوم تلقائياً بإعداد ملف Routing الرئيسي بحيث يتعامل مع هذا Module والموجودة فيه على انه Lazy Loading، ويجب التنويه انه لابد من كتابة الخيار routing– معه ولا يكتفي فيه فقط	--route

7.5. ng add:

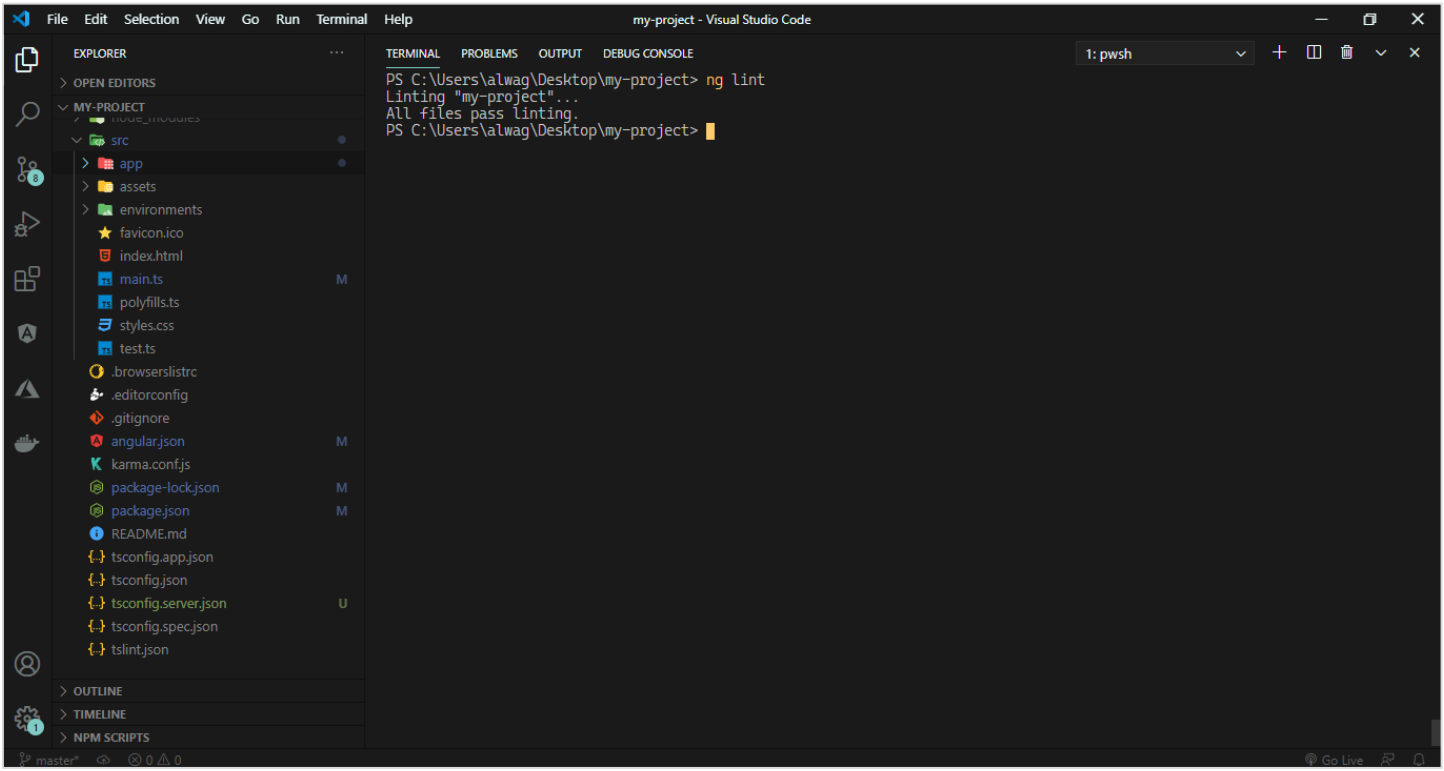
وهو من الأوامر التي تمت اضافتها في الإصدارات الحديثة من Angular CLI وهي طريقة لإضافة المكتبات الخارجية إلى المشروع، وهذا الأمر مشابهة إلى npm install، ولكن يجب التنويه ان ليس جميع المكتبات نستطيع اضافتها عن طريق هذا الأمر ولا غنى لنا عن npm، ومن امثلة المكتبات التي ممكن ان نضيفها عن طريق هذا الامر Angular Material او Angular CDK او Angular PWA او Angular Elements، ...الخ، وفي الحقيقة لم اجد موقع في شبكة الانترنت يحتوي على قائمة بجميع المكتبات المتاحة على هذا الأمر بعكس npm حيث بمجرد دخولك إلى الموقع الرسمي وبعد البحث عن أي مكتبة تُريدها ستجد طريقة تحميلها عن طريق npm.

وبالنسبة لصيغة فهي بسيطة نكتب الامر ng add ومن ثم نكتب اسم package المراد اضافته إلى المشروع.

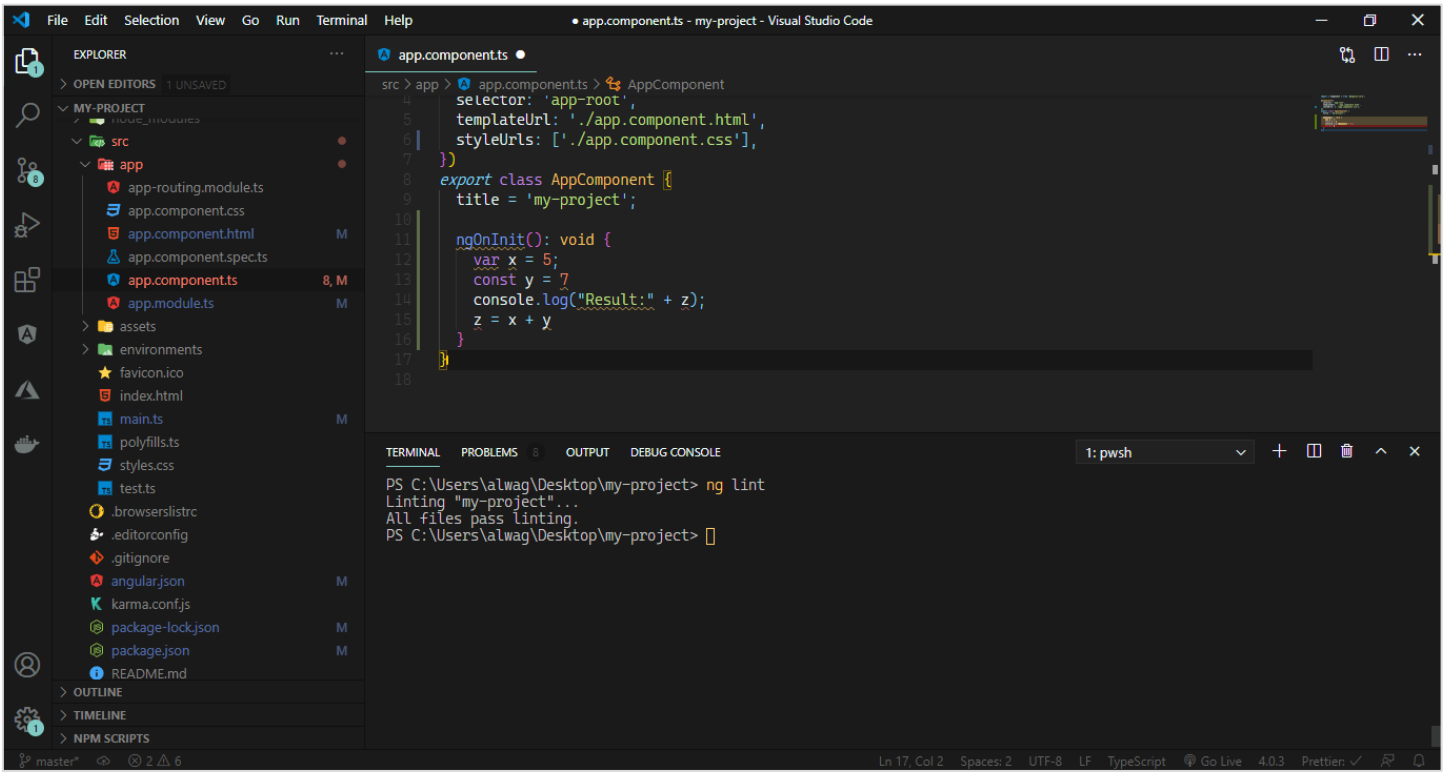
8.5. ng lint:

وقد مر علينا هذا الأمر ولكن كان بصورة خيار option باسم --lint-fix ومهمة هذا الامر هو قراءة المشروع كامل ومن ثم يظهر لنا أخطاء TSLint مثل ان يكون تم وضع علامة تنصيص مزدوجة بدلاً من علامتي التنصيص المفردة، او من ناحية تسمية المتغيرات السماح بأسلوب تسمية الباسكال، وغيره من القواعد وتستطيع عزيزي المتعلم الإطلاع على كافة هذه القواعد في ملف tslint.json.

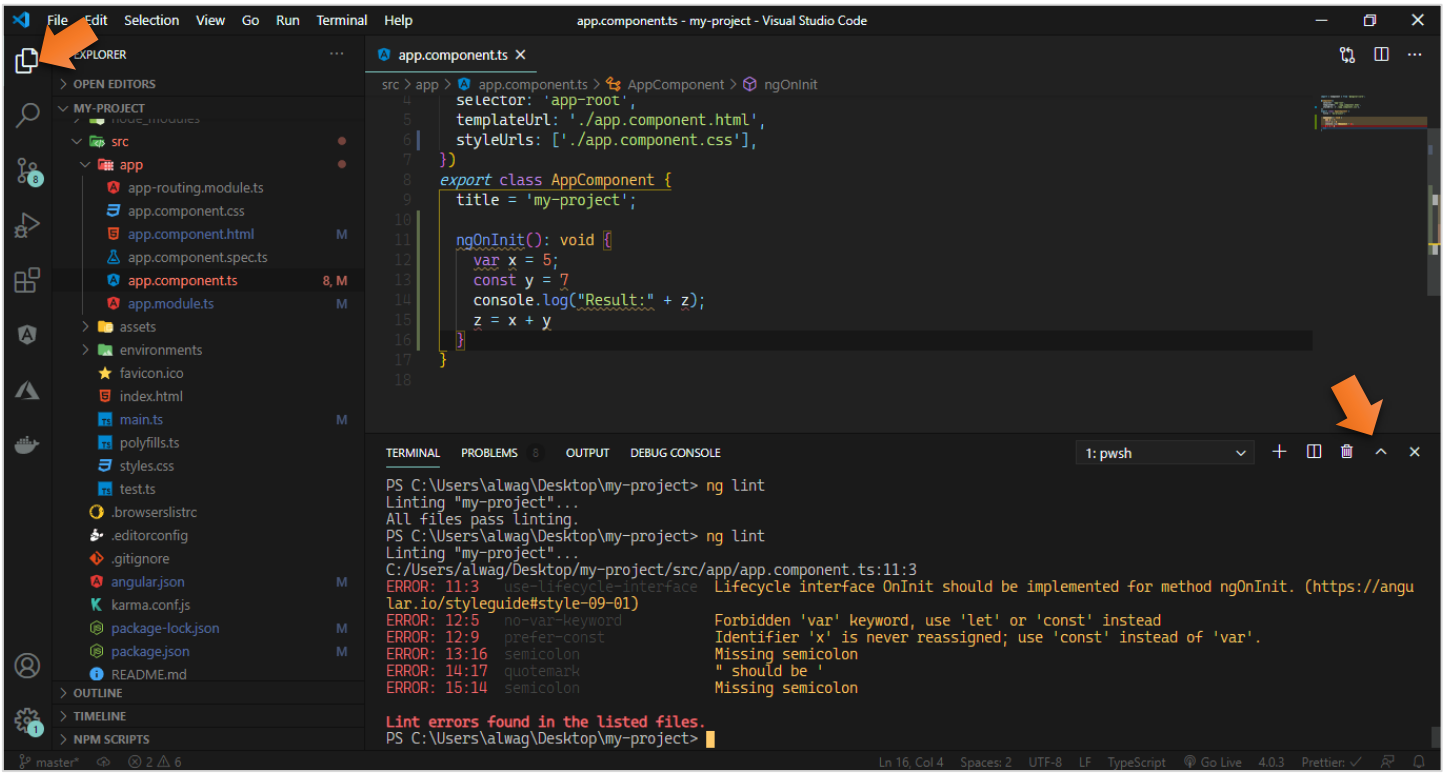
لذلك لنذهب إلى terminal ولنكتب الأمر ng lint ولنرى ما هي الأخطاء التي ممكن ان يكتشفها في المشروع، كالتالي:



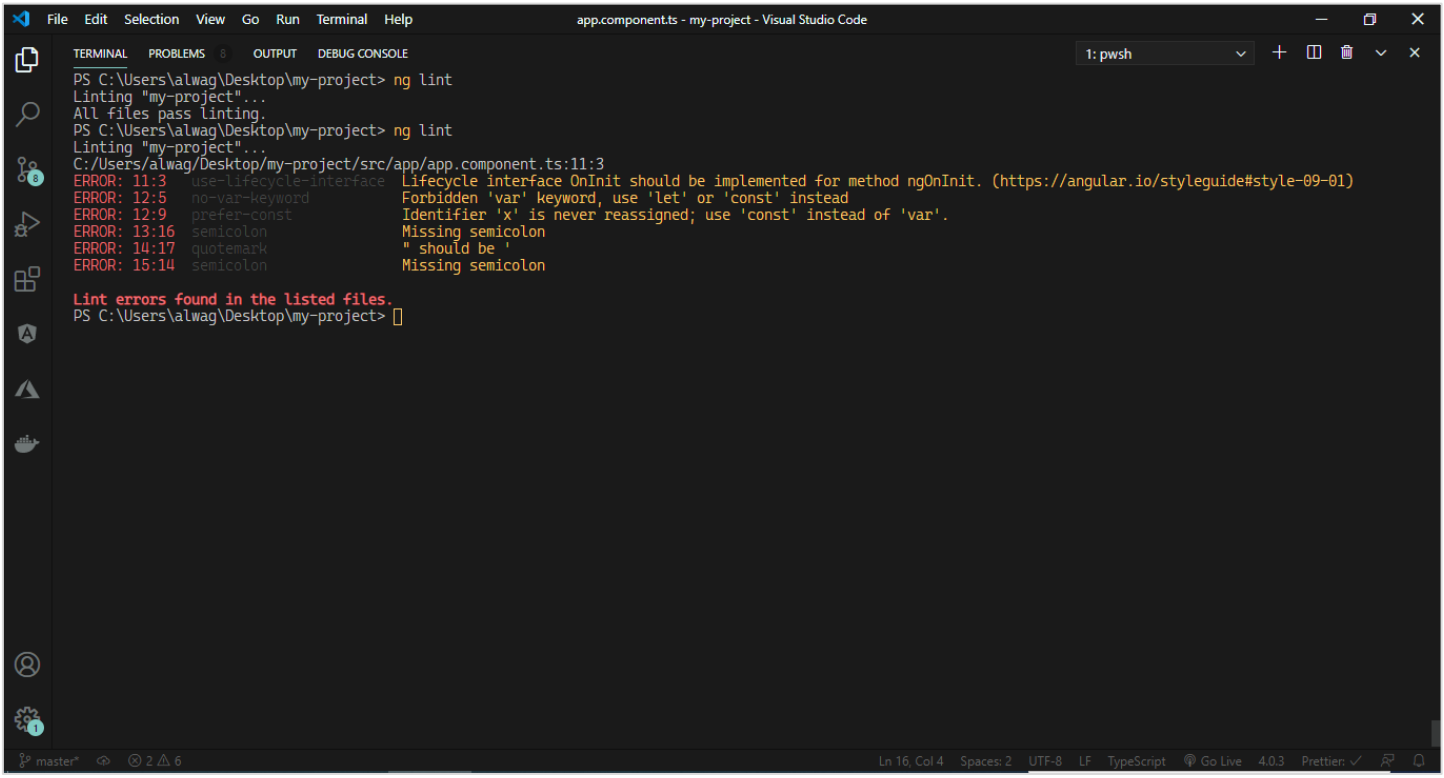
نلاحظ انه لم يكتشف أي خطأ وذلك بظهور الرسالة All files pass linting، ولكن لنقوم بتعمد كتابة بعض الأخطاء في ملف app.component.ts، كالتالي:



الآن لنقوم بكتابة الامر ng lint مرة أخرى في terminal، ولنرى النتيجة:

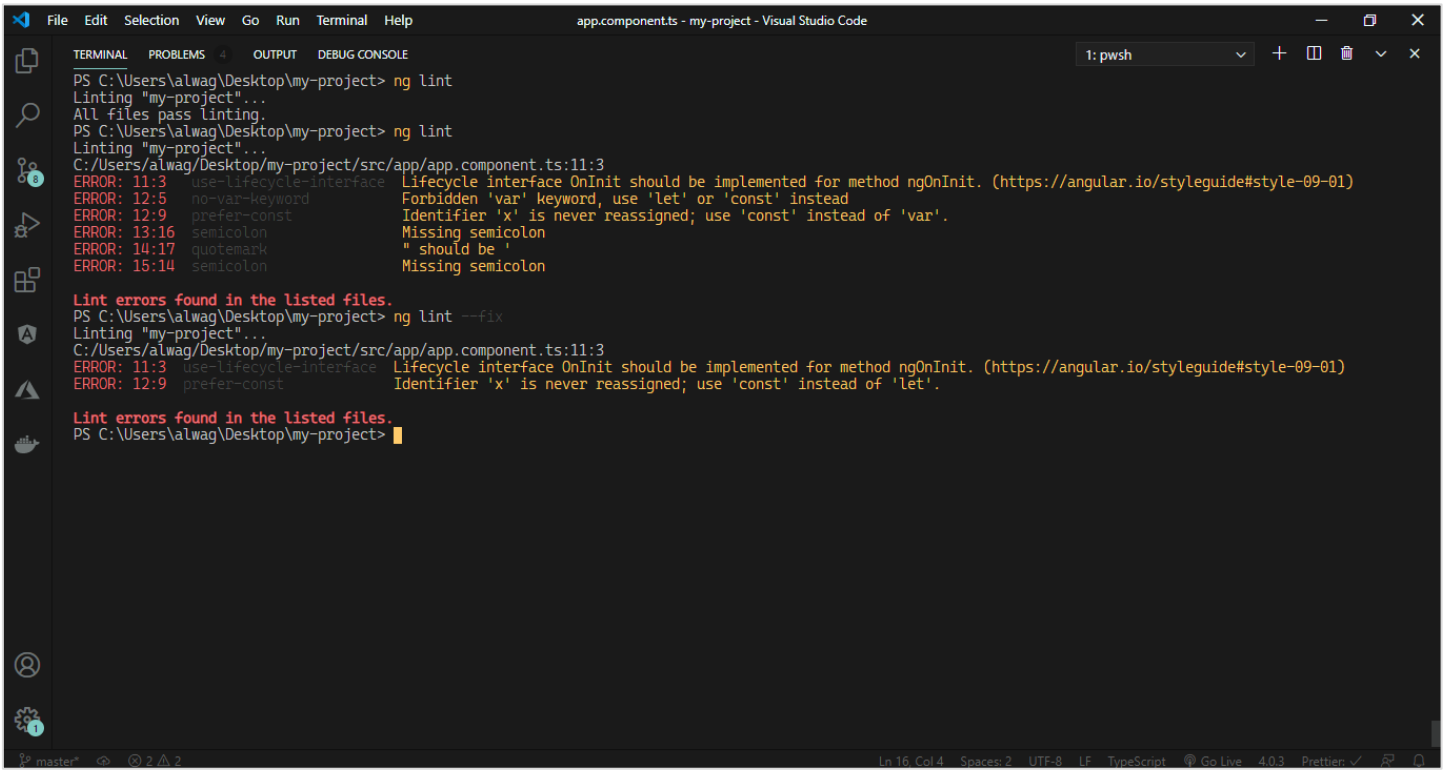


نلاحظ تم اكتشاف بعض الإخطاء ولنقم بتكبير terminal واغلاق شاشة Explorer الجانبية، كما تعلمنا سابقاً:



نلاحظ في البداية اسم الملف الذي به الأخطاء وهو app.component.ts ومن ثم يوجد قائمة بجميع الأخطاء المكتشفة، حيث الجزء باللون الأحمر يحدد رقم السطر ورقم العمود الموجود فيه الخطأ، اما اللون الرمادي فيظهر القاعدة التي بنى عليها ان هذا السكر فيه خطأ وهذه القاعدة موجودة في ملف tslint.json كما أشرت قبل قليل مع إمكانية التعديل على هذه القاعدة في حال انك ترا انها صحيحة، كأن تسمح بعلامة التنصيص المزدوجة، اما اللون الأصفر فيحتوي على شرح لهذا الخطأ.

الآن لنقوم بتصحيح هذه الأخطاء، عن طريق الخيار fix--، كالتالي:



نلاحظ تم تعديل أربع أخطاء من أصل ستة، لذلك تستطيع تكرار كتابة fix-- مرة أخرى او تقوم بتعديل هذا الخطأ يدوياً.

9.5. ng doc:

وهذا الأمر نمرر له أي شيء نريد ان نبحث عنه في الموقع الرسمي لي Angular وبالتحديد في Documentation وليس له خيارات options إلا خيار واحد وهو help، ولنفرض اننا نُريد ان نبحث عن pipe فنكتب الامر ng g doc pipe.

10.5. ng update:

وهذا الأمر تم إدراجه في الإصدارات الحديثة من Angular CLI حيث يسهل على المطورين تحديث مشاريعهم من الإصدار الحالي إلى الإصدار الجديد من إطار عمل Angular، وكما هو معلوم لأي مطور Angular انه كل ستة أشهر يتم اصدار نسخة من Angular، ولذلك ولتسهيل على المطورين تم إضافة هذا الأمر.

وهو يأخذ مجموعة من الخيارات، أهمها:

الصيغة	الشرح	الأمر
ng update --all	لتحديث جميع packages في ملف package.json	--all
ng update --next	لتحديث إلى آخر اصدار وان كان beta	--next
ng update @angular/cli @angular/core	وهنا يتم تحديد مكتبات معينة والتي بالغالب angular cli & angular core	--packages

11.5. ng build:

وهذا الأمر نستطيع عن طريقه بناء المشروع بشكله النهائي، ولكن نحتاج إلى الخيار prod-- مع هذا الأمر لكي نستفيد من تقليل حجم الملفات النهائية للمشروع وتطبيق aot بشكل افتراضي (وهو طريقة angular في عمل compiler لملفات المشروع حيث يقوم بقراءة ملفات المشروع وفي حال وجود أي خطأ سوف يتوقف ولن يقوم ببنائه).

وبعد الانتهاء من بناء المشروع سوف يتم انشاء مجلد جديد باسم dist وهذا المجلد يحتوي على ملفات المشروع النهائية التي نقوم برفعها على أي server او cloud نُريدها.

الفصل السادس

Add Third Party Libraries to the Project

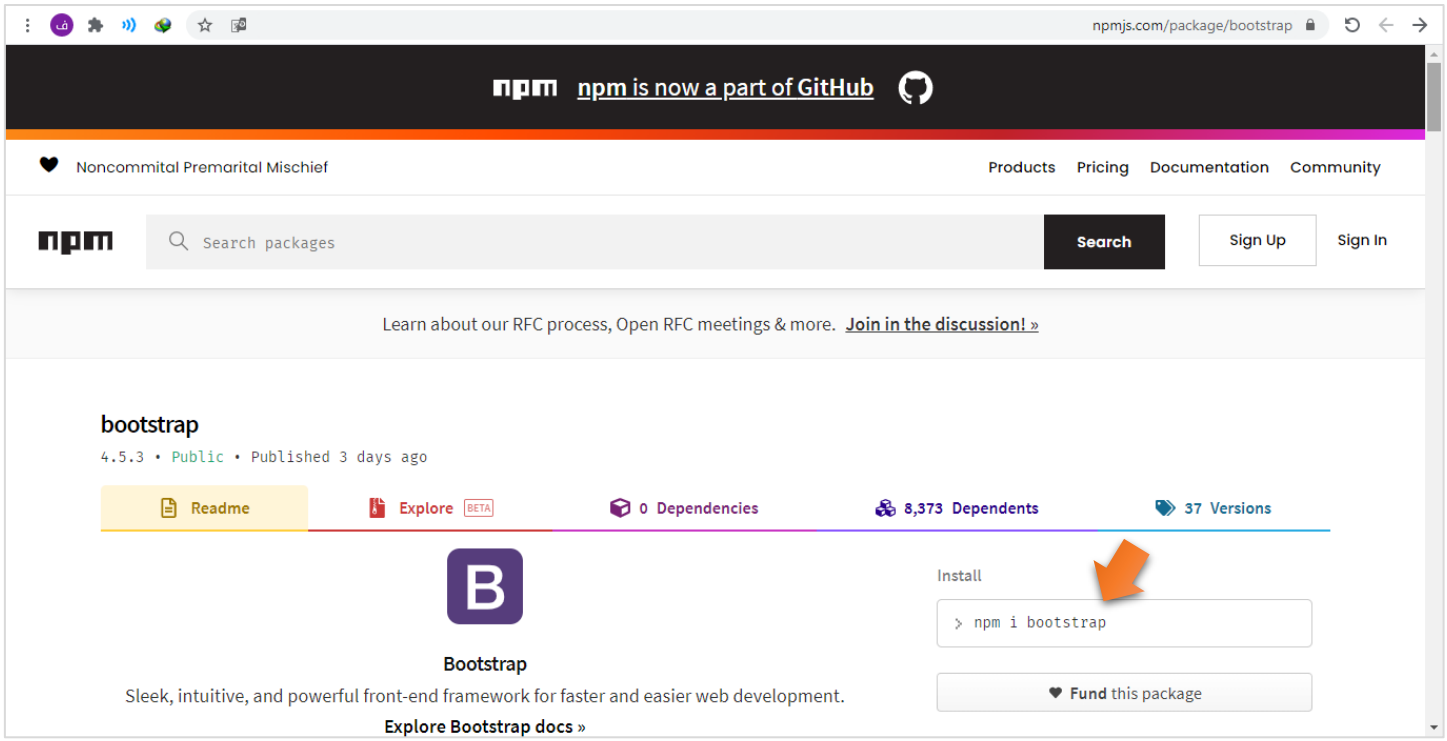
1.6. مقدمة:

نستطيع ان نقول ان هذا الفصل من اقل فصول الكتاب وربما من اقل فصول جميع الكتب في هذه السلسلة، والسبب ليس لأنه غير مهم وانما ليس له قواعد ثابته من جهة وثانياً تم التكلم عنه بصورة غير مباشرة أكثر من مرة في ثنايا هذا الكتاب.

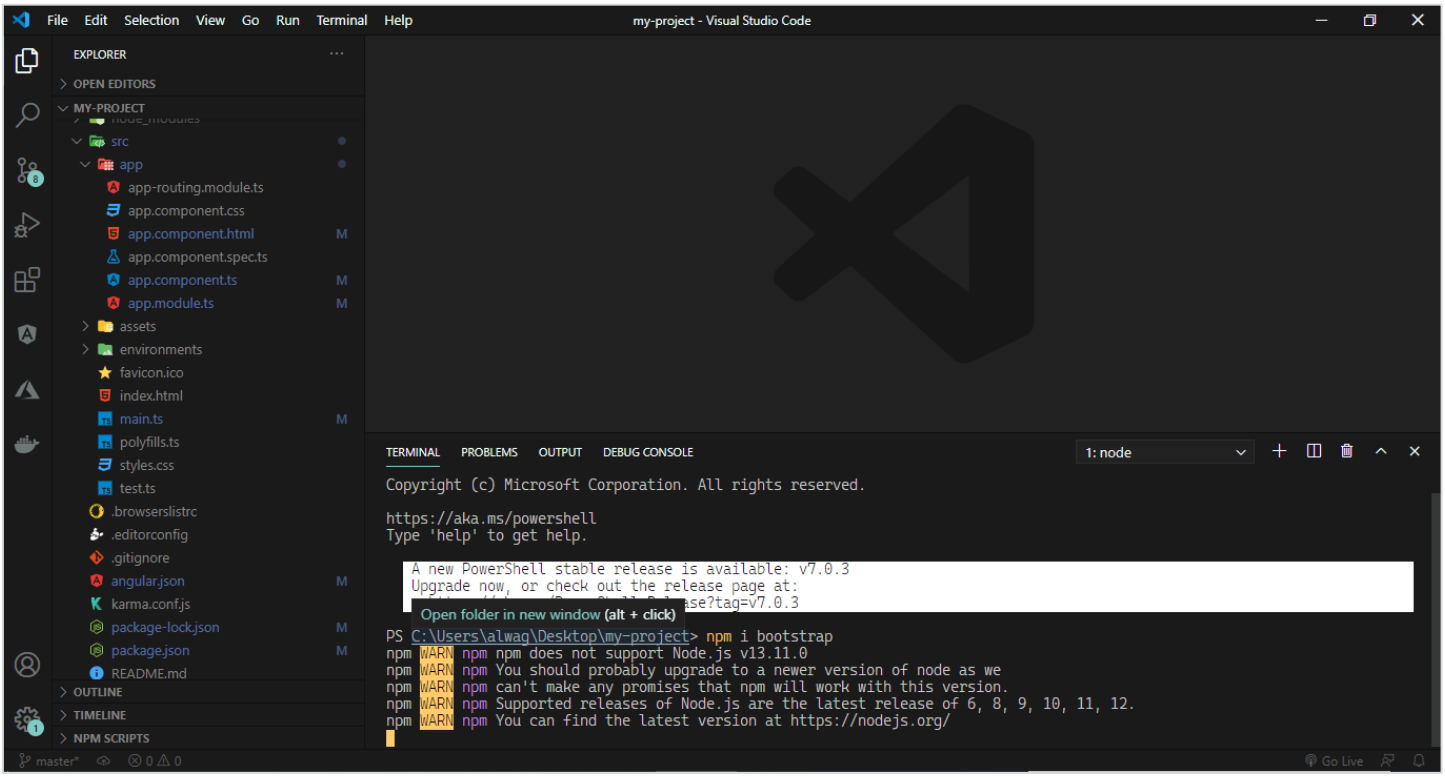
وبما ان المكتبات كثيرة ومتعددة، فسوف أُعطي مثال على مكتبتين الأولى خاصة بمكتبات css وهي Bootstrap والثانية هي مكتبة YouTube Plyer.

2.6. إضافة مكتبة Bootstrap:

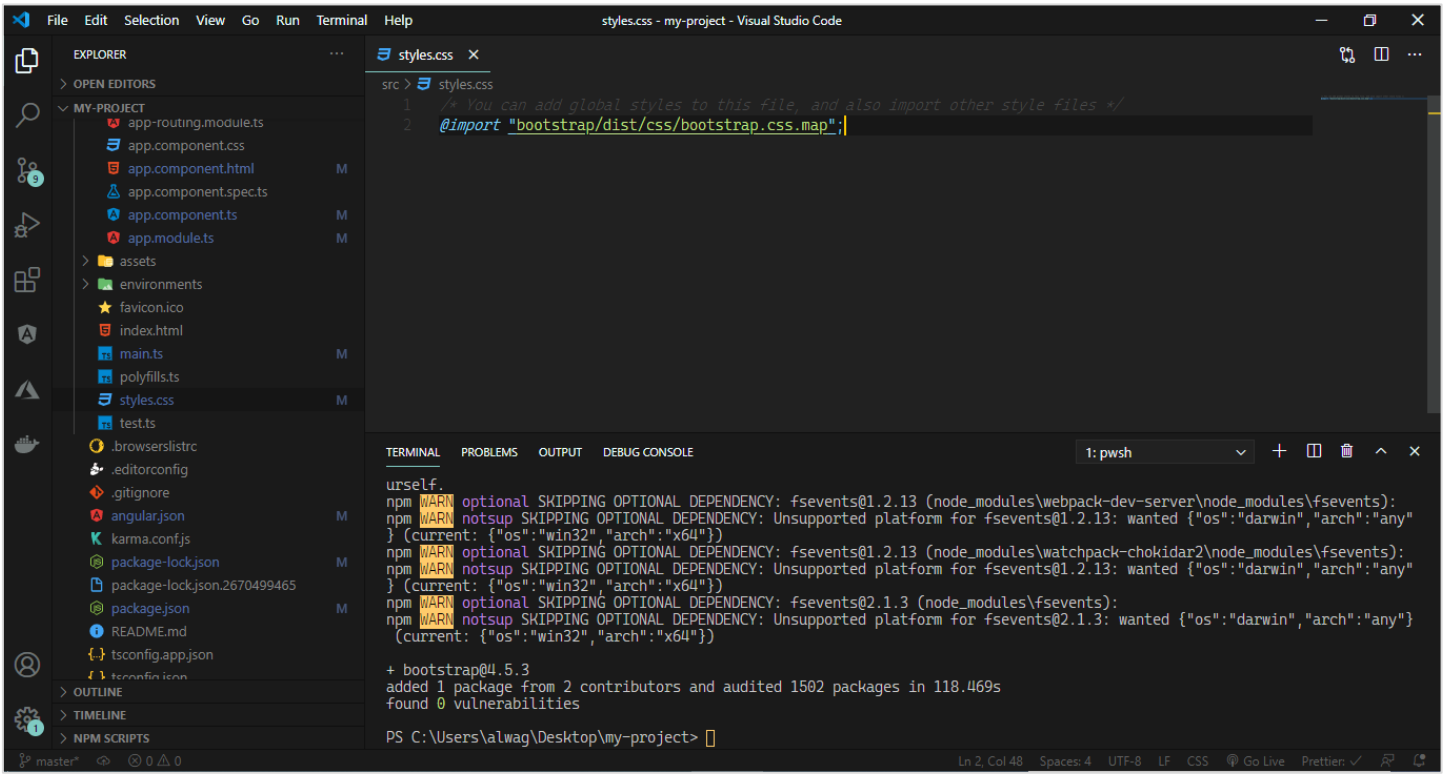
لإضافة مكتبة Bootstrap نذهب إلى موقع https://www.npmjs.com/ ومن ثم نبحث عن هذه المكتبة، كالتالي:



بعدها نقوم بنسخ الأمر npm I bootstrap كما هو موضح في الصورة السابقة، ومن ثم نقوم بلصق هذا الامر في terminal مع التأكد اننا في نفس مجلد المشروع (مهمة هذه الخطوة وإلا لن تعمل)، كالتالي:



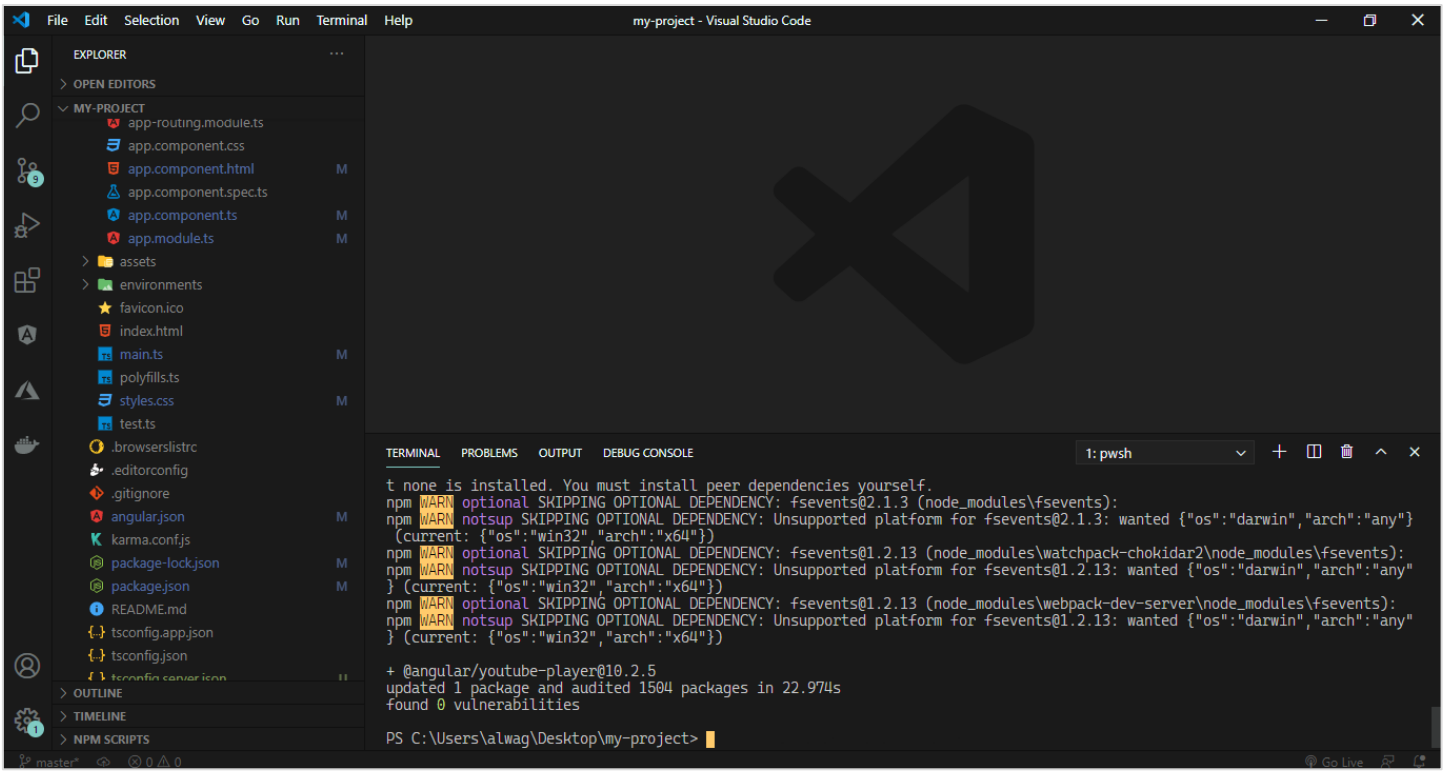
بعد الانتهاء من التحميل يتبقى لنا آخر خطوة وهي استدعاء مكتبة css الخاصة بهذه المكتبة، وفي حال كنا نريد جميع ملفات المشروع الاستفادة من هذه المكتبة فلابد ان نستدعيها في ملف styles.css اما في حال أردنا ان تكون خاصة بي component محدد فنستدعيها بملف style الخاص بهذا component، كالتالي:



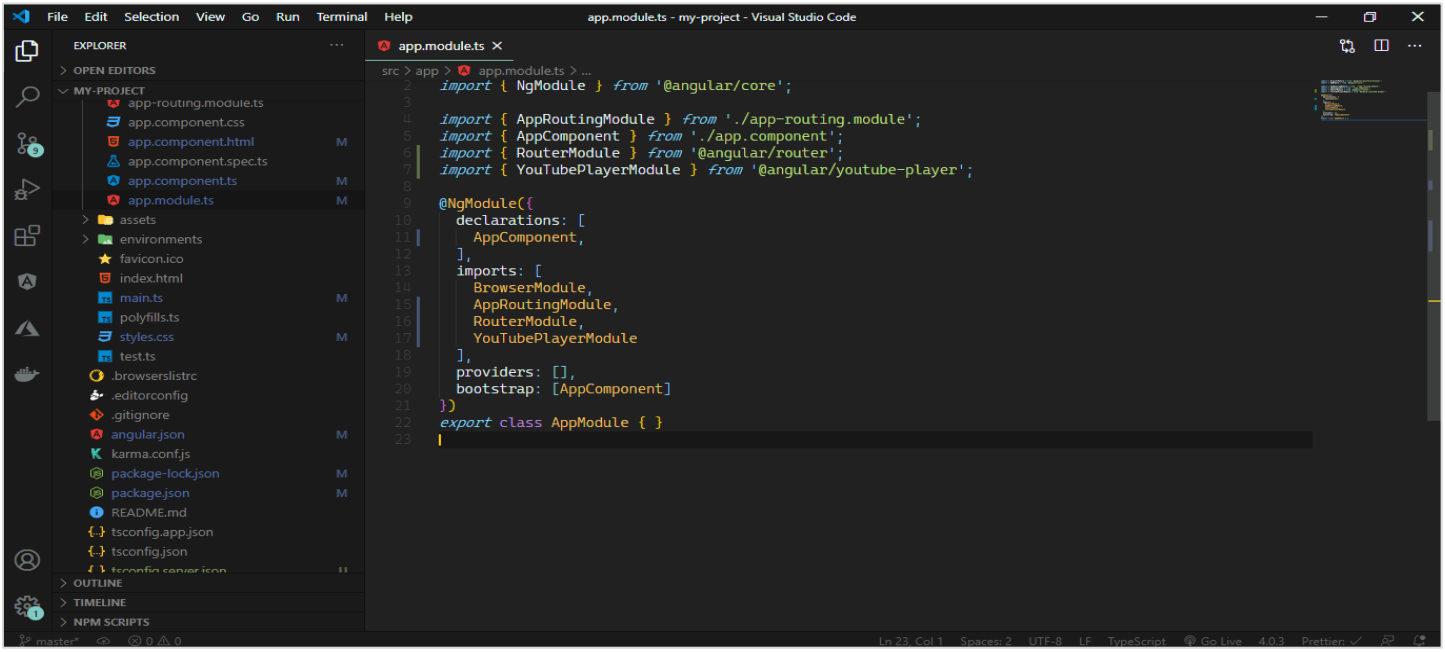
وفي حال تساؤلك عزيزي المتعلم من اين اتيت بهذا المسار، فالإجابة هي انني بحثت عنه يدوياً في مجلد node_module ومن ثم في مجلد bootstrap وهكذا إلى ان وصلت إلى الملف المنشود.

3.6. إضافة مكتبة @angular/youtube-player:

هذه المكتبة Typescript، لذلك هي تختلف من ناحية الإعداد اما التحميل والتثبيت في المشروع فالأمر واحد وذلك عن طريق الأمر npm i @angular/youtube-player، كالتالي:



ولإعداد هذه المكتبة لكي نستطيع استخدامها في مشروعنا نحتاج أن نستدعي Module الخاص بها في Root Module او أي Module آخر في المشروع لديك بحسب احتياجك، كالتالي:



مع العلم ان اسم Module الخاص بالمكتبة أخذته من الموقع الخاص بها، وهكذا بقية المكتبات لابد ان تعتمد على موقعها الرسمي لكي تعرف كيف تضيفها إلى مشروعك بدون مشاكل.

المراجع

References

References:

1. Freeman, **Adam, Pro Angular 9**, Apress, 2020.

2. Agarwal, Uttam, **Hands-On Full Stack Development with Angular 5 and Firebase**, Packt, 2018.

3. Delaney, Jeff, **The Angular Firebase Survival Guide**, leanpub, 2018.

4. Saha, Depasis, **Angular 7.0 for Beginners**, 2018.

5. Vijay, Santhosh, **Material for Angular Developer Course**, Leanpub, 2019.

6. Hussain, Asim, **Angular From Theory to Practice**, 2017.

7. D. Booth, Joseph, **Angular Succinctly**, Syncfusion, 2019.

8. Squad, Ninja, **Become a ninja with Angular**, 2019.

9. Steyer, Manfred, **Enterprise Angular**, Leanpub, 2020.

10. Clow, Mark, **Angular 5 Projects**, Apress 2018.

11. Nate Murray, Felipe Coury, Ari Lerner, and Carlos Taborda, **ng-book The Complete Guide to Angular**, Fullstack.io, 2018.

12. Bouchefra, Ahmed, **Practical Angular: Build your first web apps with Angular 8**, Leanpub, 2019.

13. Hajian, Majid, **Progressive Web Apps with Angular**, Apress, 2019.

14. Savkin, Victor, **Angular Router**, Leanpub, 2018.