# **ما قبل وجود المشروع:**

هو مشروع مقدم من Ahmad Fijr ، بعدما واجه صعوبة في تعلم مكتبات بايثون لبناء الواجهات الرسومية من ( Pyqt5 – Kivy – Tkinter ) وفي الواقع كانت مكتبة Tkinter هي اسهل مكتبة من ناحية الاستخدام ولكن عيبها أن تصميمها قديم ورديء ولا يماثل التطبيقات الحديثة في التصميم ، وهذا عدا عن أن التصميم فيها بشكل كتابي وبدون برنامج للتصميم ، وفي المقابل نرى برنامج التصميم Qt Designer مع مكتبة Pyqt5 يلبي القسم التصميمي من برمجة الواجهات الرسومية بشكل جيد جدا خصوصا بإمكانية التصميم بلغة CSS ولكن بعد التصميم علينا تحويل التصميم لكود بايثون ، وبعدها يبدأ العمل مع البرمجة الخلفية ولم تكن تجربة إيجابية إطلاقا ؛

بعدما وجد أن بناء الواجهات الرسومية باستخدام لغة بايثون عملية صعبة وتستهلك الكثير من الوقت وجهد ، ولهذا حاول إيجاد حل لهذه المشكلة بحيث تكون عملية بناء الواجهات أسهل لمبرمجي بايثون وبنفس الوقت حديثة ، حيث أنه فضل أن يتعلم لغة C# على أن يكمل العمل مع الطرق العادية لبناء الواجهات الرسومية بلغة بايثون ، ولم يتعلم شيء غير أساسيات لغة C# على موقع Sololearn ، وبعدها أتطلع على بناء الواجهات الرسومية من خلال لغة C# وبرنامج Visual studio وكانت التجربة الرائعة ولا تقارن إطلاقا ب مكتبات بايثون لبناء الواجهات الرسومية ، ولكن كان العيب أن لغة C# من حيث العمليات التي تحدث في الخلفية معقدة نوعا ما ، والوظيفة الذي تستغرق في كتابتها بلغة C# 10 اسطر – نفس الوظيفة بالضبط تأخذ سطرين مع لغة بايثون ؛

والآن لدينا لغتان ، لغة بايثون التي هي ممتازة من حيث سهولة وسرعة كتابة الكود ، وسيئة مع الواجهات الرسومية – وفي المقابل لغة C# التي هي ممتازة من حيث سهولة وسرعة إنشاء وتصميم الواجهات الرسومية ، وسيئة مع العمليات الخلفية بسبب طريقة كتابة كود اللغة ؛ ومن هنا ولدت فكرة مشروع **BeBot Python**.

# **وصف المشروع:**

هو مشروع يهدف ل توفير وسيلة سهلة وفعالة تمكن مبرمجي Python من بناء برامج الواجهات الرسومية لسطح المكتب عمن خلال استفادة من خصائص Visual studio ولغة C# لصالح Python ، بحيث تعمل في الواجهة سي شارب وفي الخلفية بايثون .

ويمكن العكس أي أن يستفيد مبرمجي لغة C# و Dot Net عموما من كامل خصائص لغة Python لصالح مشاريعهم.

# **ما هي BeBotPython ؟**

هي خوارزمية وفكرة لتنظيم وترتيب العلاقات بين لغة C# وبايثون، وفق اسلوب كتابي للرمز البرمجي.

وهي تتيح بناء الواجهات الرسومية باستخدام visual studio ولغة C# لربط عناصر الواجهة الرسومية فيما بعضها وهذا يتطلب فقط معرفة اساسيات C# ومن ثم عمل حدث للعناصر مثل وجود زر وعند النقر على الزر تحدث الامور التي في الخلفية وهنا يأتي استخدام المكتبة المكتوبة بلغة سي شارب BeBotPythonLib التي تمثل واجهة تحكم لبيئة بايثون الافتراضية المصغرة والتي تتيح استخدام واستعمال اكواد بايثون عبر برنامج مصغر كالسكربت تماما يعمل في الخلفية .

# **كيف احصل البيئة المصغرة ل بايثون ؟**

من خلال برنامج BeBot Python small virtual environment generator أو BPSVEG ، الذي يتيح لك انشاء بيئة مصغرة افتراضية ل بايثون على شكل ملف تنفيذي ، بعد اختيار وتحديد المكتبات المرادة ، وكلما استخدمت مكتبات خارجية أكثر كلما زاد حجم البيئة الافتراضية ، وانشاء واجهة الربط بين اللغتين ، وهذه الملفات ستقوم بنقلها لمشروع visual studio.

وبشكل عام كل المكتبات القياسية الأساسية مضافة بشكل افتراضي ولا حاجة لإضافتها ، والحجم الافتراضي للبرنامج التنفيذي للبيئة المصغرة الافتراضية تقريبا 18 M ، وإن إضافة مكتبة قياسية لا تؤثر على الحجم البيئة المصغرة ولكن تؤثر على السرعة ، لذى نضيف فقط المكتبات التي نحتاج إليها.

# **ما فرق بين BeBotPython وIronPython ؟**

BeBotPython تختلف عن IronPython كل الاختلاف والبعد فهو يمثل اصدار بايثون 2.7 ، والذي يكون عبارة عن سي شارب ومكتباتها ولكن مع Syntax بايثون ، ولا تدعم إنشاء واجهات السهلة من visual studio بطريقة السحب والإفلات والتحكم السهل في الخصائص ، ولا تدعم المكتبات الحديثة مثل بايثون العادي ، والإكمال التلقائي سيء جدا معها.

أما BeBotPython تجعل الواجهة تدعم بيئة بايثون الموجودة اساسا في جهازك بغض النظر عن إصدار بايثون الموجود في جهازك عبر إنشاء بيئة مصغرة منه ، وبالتأكيد تدعم كافة إصدارات بايثون الموجودة من 3 فما فوق وحتى الإصدارات القادمة مدعومة والمكتبات بشكل تلقائي .

# **برامج تمت بناؤها بطريقة الدمج بين اللغتين :**



# **المميزات المتوفرة في الإصدار الحالي:**

هناك وضعان للدمج بين لغة بايثون و سي شارب ، الوضع الخطي و وضع المزامنة الآنية .

ملاحظة : المقصود في البرنامج الرئيسي هو برنامج الواجهة الرسومية المكتوب بلغة سي شارب.

## **مميزات الوضع الخطي:**

* يمكن استخدامه لربط الواجهات الرسومية المكتوبة بلغة بايثون مع واجهات سي شارب
* يمكن إظهار شاشة السكربت للمستخدم
* يمكن أن يعمل كبرنامج طرفي مع إظهار أو إخفاء شاشته
* يعمل في الخلفية حتى بعد إغلاق البرنامج الرئيسي
* غير مرتبط مع البرنامج الرئيسي
* تتم معه العمليات بشكل خطي ، أي إنه يعمل – يعالج البيانات – ثم يغلق.
* يمكن أن يشكل خيط جديد ، أو يجعل البرنامج الرئيسي ينتظر حتى ينتهي
* تبادل البيانات بين اللغتين يكون إما عبر الملفات النصية أوArguments

## **مميزات المزامنة الانية:**

* التزامن بين اللغتين
* يعمل مع عمل البرنامج الرئيسي ويغلق عند إغلاقه
* بث حي لمخرجات بيئة بايثون في فيجوال ستوديو
* وجود تتبع لأخطاء الكود بايثون في فيجوال ستوديو
* أسرع من الوضع الخطي
* تتبادل البيانات بين اللغتين عبر الدوال ويمكن عبر الملفات النصية أوArguments
* تتعامل بطريقة مشابه لتعامل وتفاعل المستخدم مع بايثون ولكن عن الطريق لغة سي شارب
* يمكن أن يشكل خيط جديد ، أو يجعل البرنامج الرئيسي ينتظر حتى ينتهي

## **مميزات عامة:**

* بناء برمجيات سطح المكتب بسهولة وسرعة بناء أكبر
* وجود مكتبة BeBotPythonLib.DLL التي تمثل واجهة ربط لغة Python مع C#
* لا تتطلب معرفة متعمقة في لغة C# فقط الأساسيات
* لا حاجة لوجود بايثون لدى المستخدم النهائي
* إمكانية التعديل والتحكم بسهولة بكامل خصائص كلا من لغة C#و Python
* استهلاك منطقي للموارد مع إمكانية للتخصيص توزيع المهام عبر اللغتين
* تعتمد فكرة البيئة المصغرة الافتراضية ل بايثون
* حجم أقل في البرنامج النهائي مقارنة مع مكتبة Pyqt5 في بايثون
* وجود برنامج PBSVEG منشئ البيئة الافتراضية المصغرة لبايثون

# **المميزات و تطويرات المشروع في الإصدارات القادمة:**

* تحسين سرعة البيئة المصغرة من خلال التحكم في imports
* دعم إنشاء الواجهات الرسومية على لأنظمة اللينكس والماك
* تحسين استيراد البيانات ، جعله أسرع وأكثر سلاسة وعملية
* دعم استيراد نصوص طويلة
* إضافة أحداث لمكتبة BeBotPythonLib مثل (عند الإغلاق – عند بدأ التشغيل – عند استيراد البيانات – عند حدوث خطأ – عند إرسال البيانات) وغيرها
* دعم إنشاء بيئة مغلقة المصدر بدون ملف Code.py
* دعم إظهار وإخفاء نافذة وضع المزامنة الآنية
* دعم تتبع للأخطاء في الوضع الخطي
* تقليل حجم المكتبة لحد أدنى أقل من 18 M ، عن طريق الاستغناء عن المكتبات الأساسية
* تحسين البيئة وجعلها لا تؤثر على واجهة سي شارب
* تحسين مفهوم المزامنة ، من خلال انتظار الدوال حتى تنتهي
* دعم عرض مخرجات Console التابع لواجهة سي شارب
* دعم الكامل لمشاريع WPF
* دعم مشاريع ASP.net
* دعم مشاريع العاب unity على فيجوال ستوديو
* جعل المشروع يدعم تطبيقات الاندرويد ، أو تطوير مشروع جديد لهذا الغرض

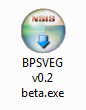
ويمكنكم أيضا اقتراح التطويرات أخرى لنقوم بها

# **فكرة عمل مشروع BeBot Python ببساطة:**

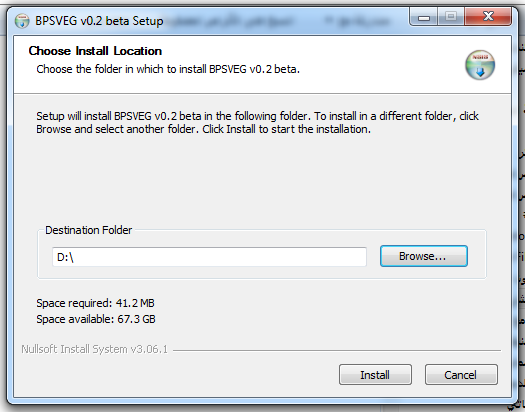
1. تفتح منشئ البيئة بايثون المصغرة الافتراضية **BPSVEG**
2. تختار المكتبات التي تريد إضافتها في البيئة المصغرة ل بايثون
3. تضغط على setup environment
4. سيقوم البرنامج بإنشاء بيئة مصغرة لبايثون تحتوي على المكتبات المضافة
5. تنقل البيئة والملفات ل مشروع windows form من نوع Dot Net
6. تبدأ ببناء برنامجك

# **كيف تقوم باستخدام مشروع BeBot Python:**

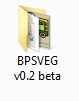
في البداية تقوم بتحميل برنامج BPSVEG وتثبيته



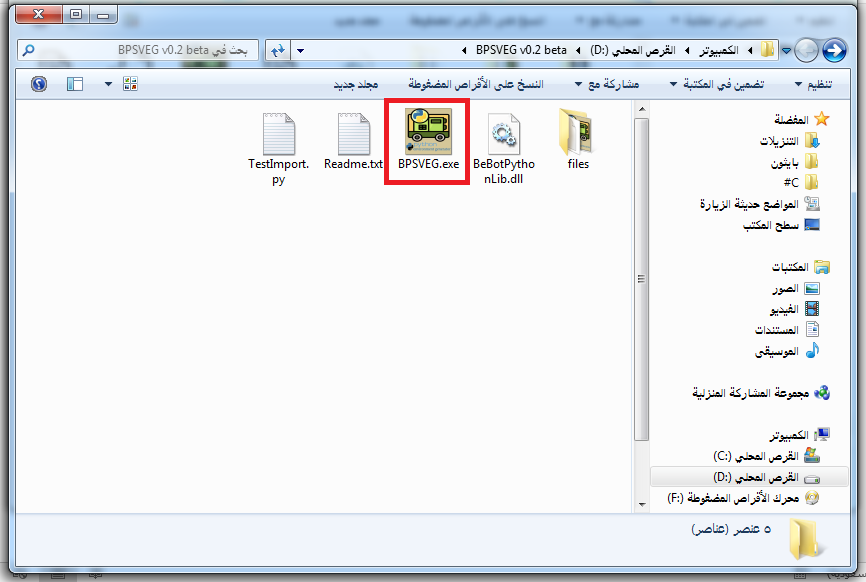
تختار مسار التثبيت



بعد تثبيت البرنامج سيظهر لك هذا المجلد



بعد الدخول عليه



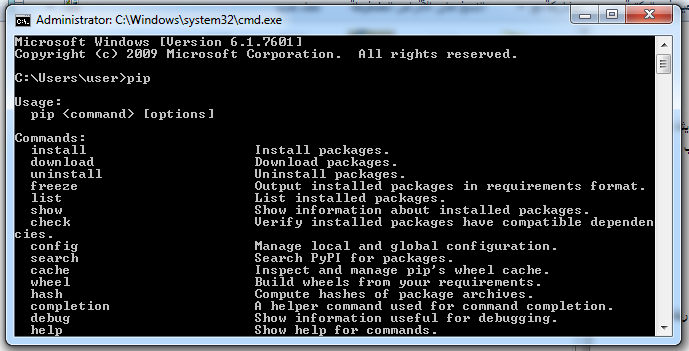
**قبل فتح البرنامج:**

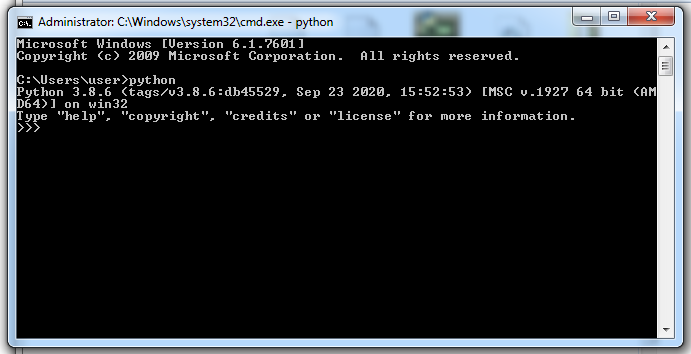
1. يجب أن تكون لغة بايثون مثبتة مسبقا في جهازك.

أي إصدار من 3 فما فوق

1. أن تكون إعدادات الPath مضبوطة.

يجب أن يعمل pipو python عندما تكتب في cmd





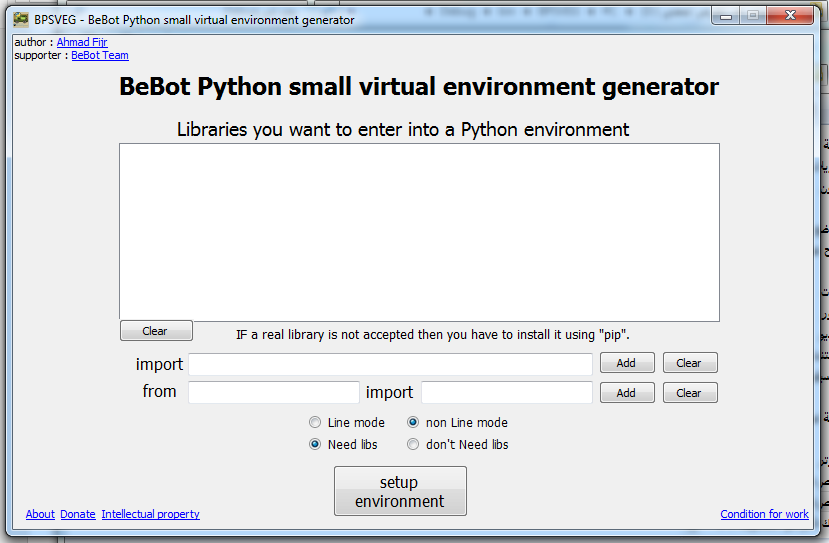
1. تقوم بكتابة هاذين الأمرين في cmd

pip install cryptography

pip install pyinstaller

والان بعدما قمت بهذه الأشياء اصبح بإمكانك تشغيل البرنامج كيفما شئت وبأي وقت كان.

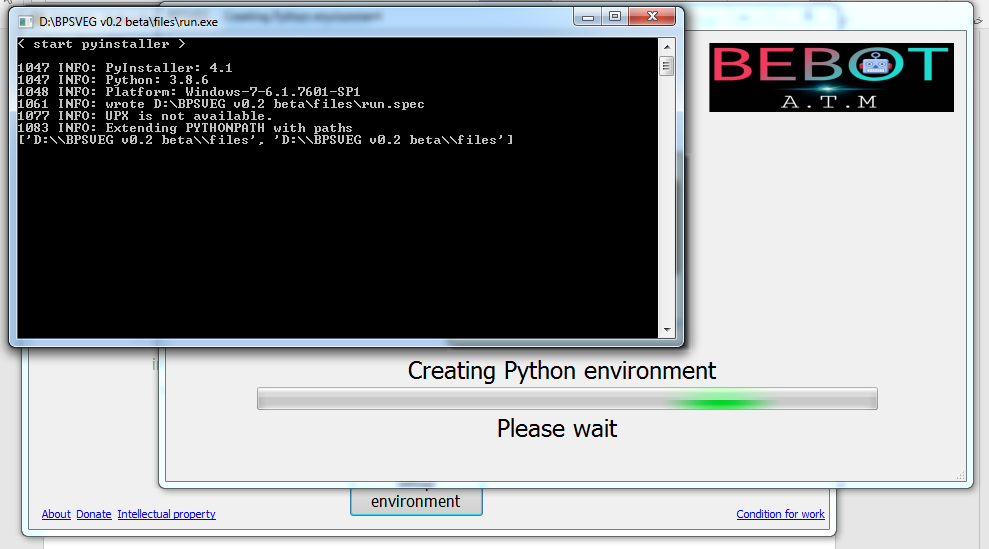
**نقوم بتشغيل البرنامج:**



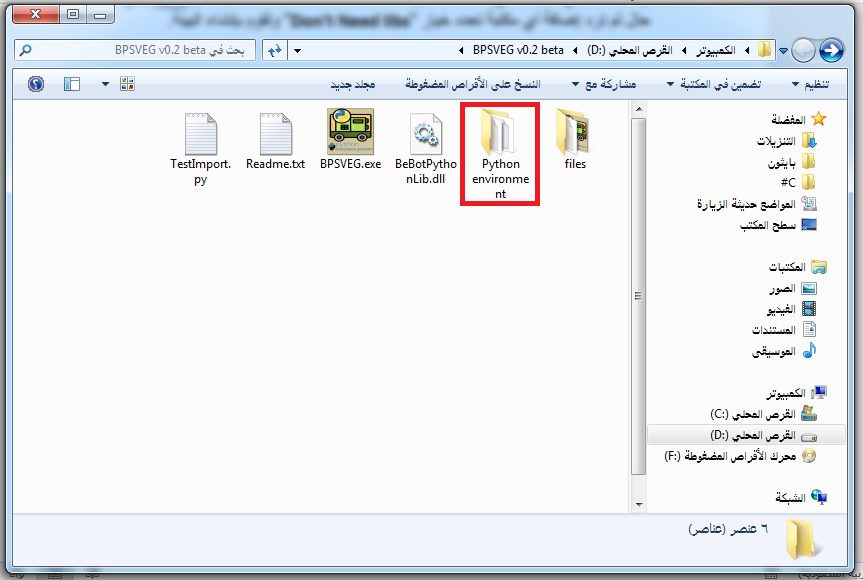
من هنا تقوم بإضافة المكتبات للبيئة المصغرة، وكلما زادت المكتبات كلما زاد حجم البيئة المصغرة ويختلف الحجم من مكتبة ل أخرى، بشكل عام كل المكتبات القياسية الأساسية موجودة مسبقا في البيئة المصغرة التي ستنشئ، ولا حاجة لإضافتها، وإن إضافتها لا تؤثر على الحجم ولكن قد تؤثر على السرعة، ويكون حجم البيئة المصغرة بدون إضافة مكتبات تقريبا 18 M، والمكتبات التي تضاف هي مكتبات الطرف الثالث المثبتة مسبقا باستخدام الأمر pip ، وفي حال لم ترد إضافة أي مكتبة تحدد خيار "**Don't Need libs**" وتقوم بإنشاء البيئة.

والآن سنقوم بإنشاء بيئة بدون إضافة مكتبات

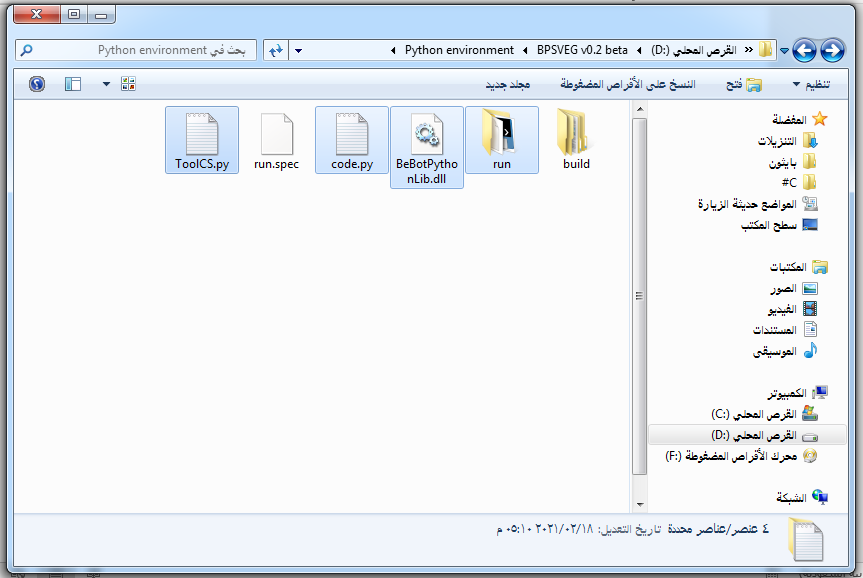
نحدد خيار "**Don't Need libs**" ثم نضغط على **setup environment"**"



بعدها ستظهر لنا البيئة المنشئة في مجلد البرنامج

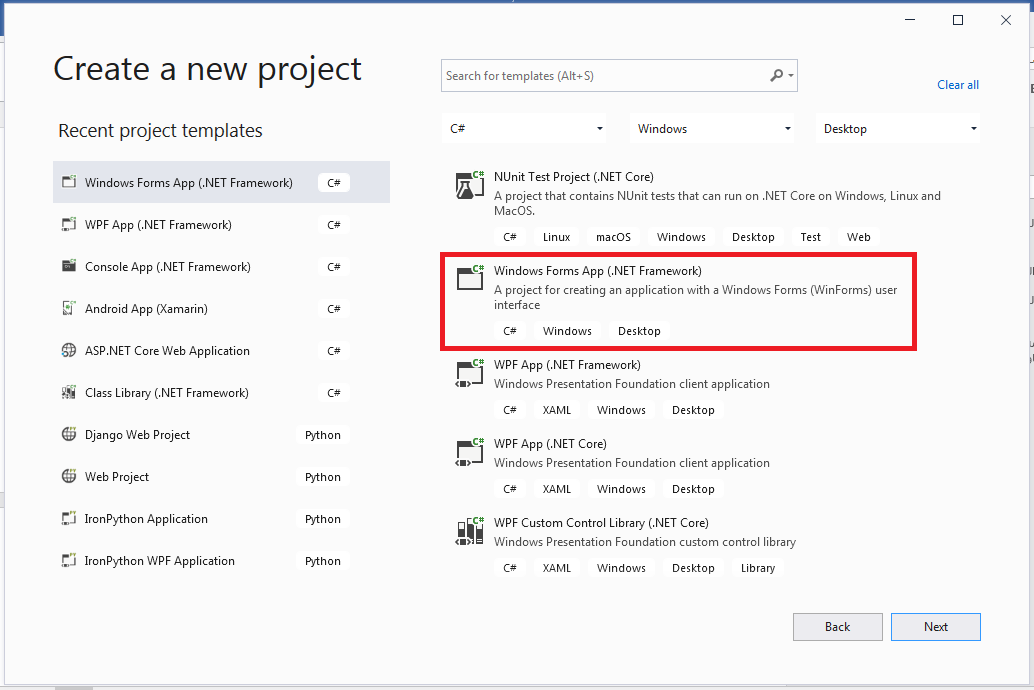


بعد الدخول عليه هذه هي الملفات المراد نقلها ل فيجوال ستوديو

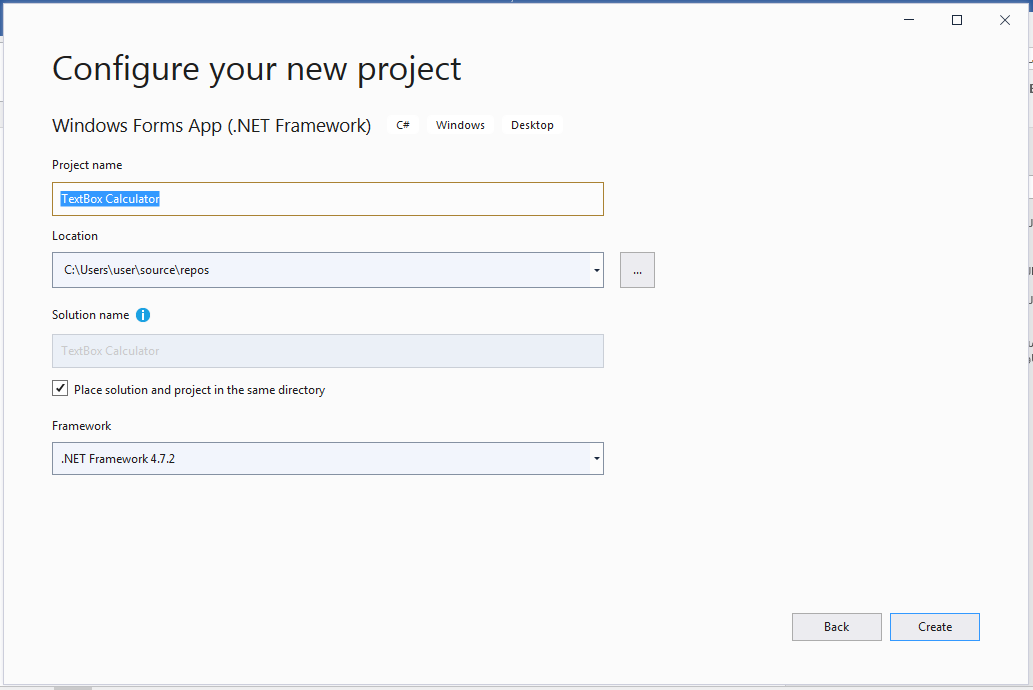


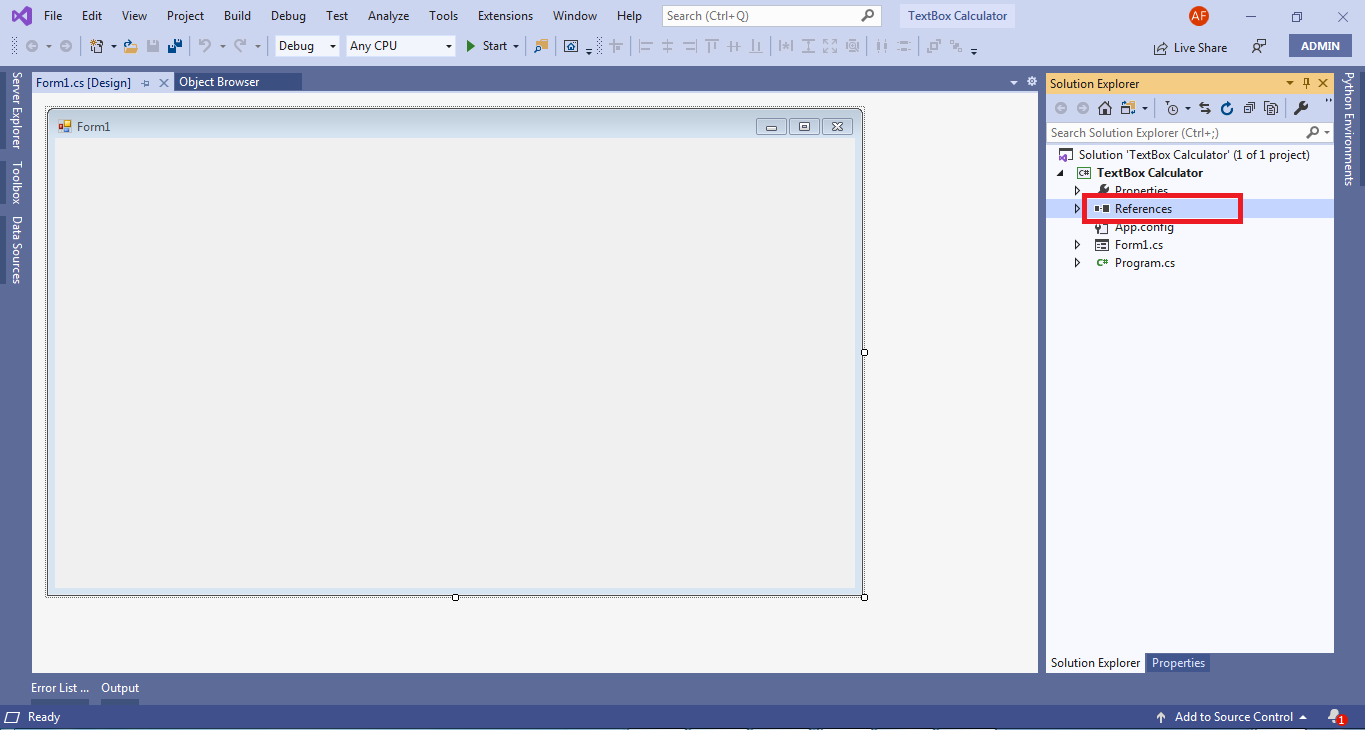
والآن تم إنشاء البيئة المصغرة بنجاح وهذه هي الملفات المراد نقلها ل فيجوال ستوديو

نقوم بفتح برنامج فيجوال ستوديو ونقوم بإنشاء مشروع windows form من نوع dot Net

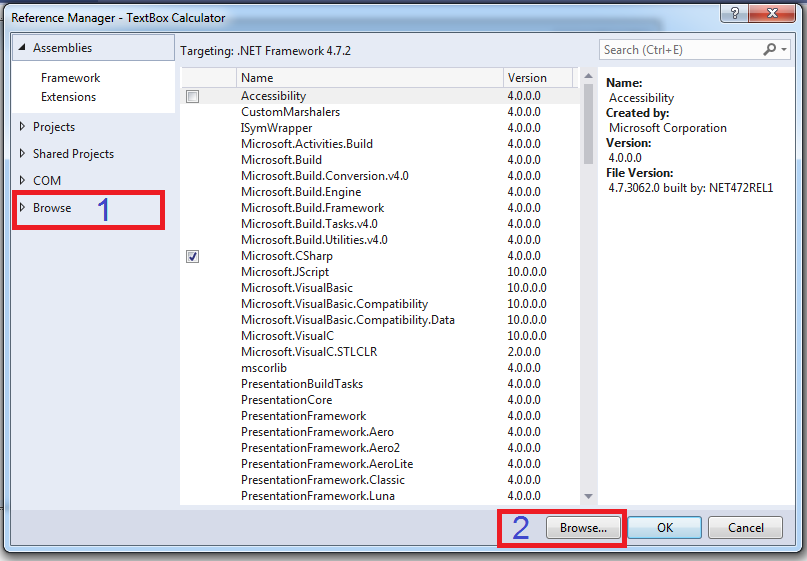


سنقوم بإنشاء مشروع آلة حاسبة بسيطة في هذا المثال

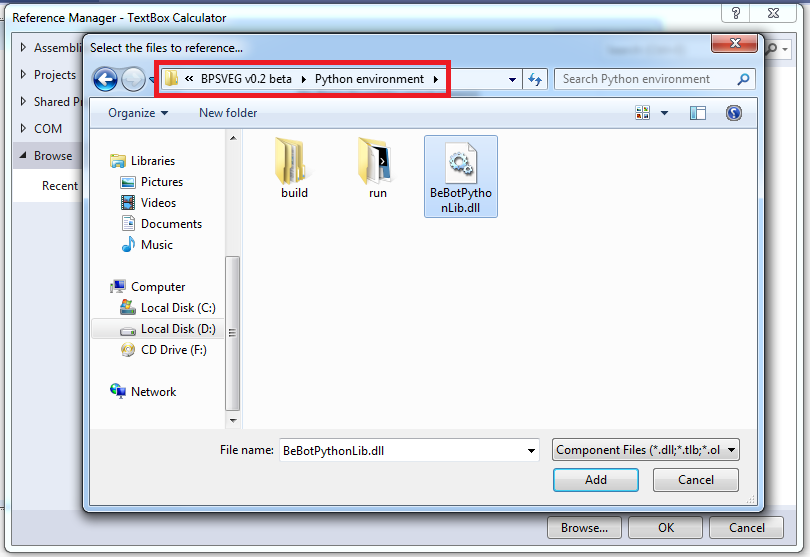




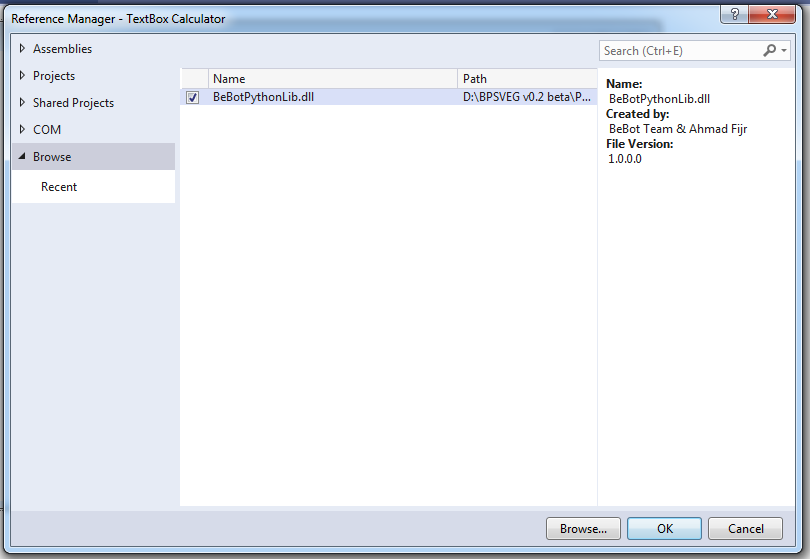
بعد إنشاء المشروع بنجاح، نضغط بالزر الأيمن للفأرة على References ثم نختار Add References



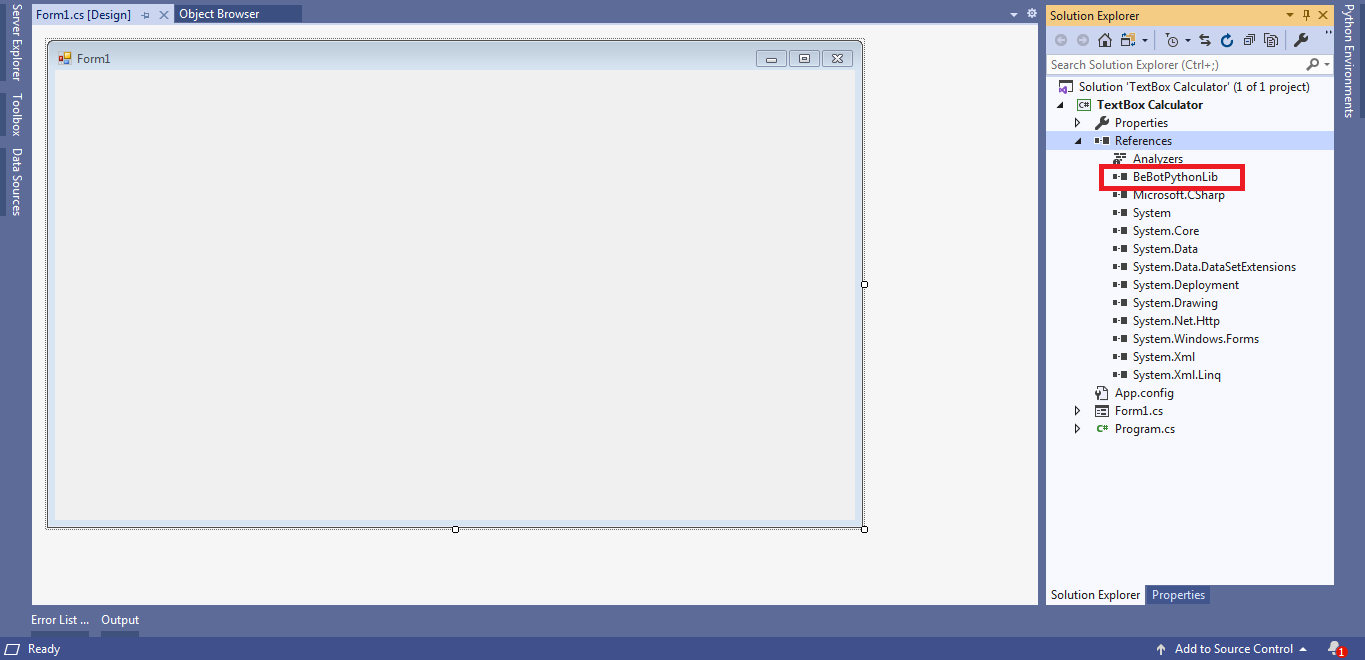
بعدها ستظهر هذه النافذة، نضغط على Browse كما في الصورة



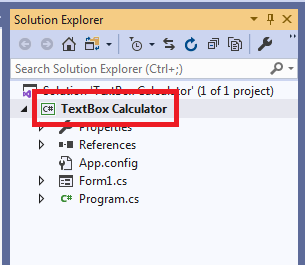
نذهب إلى مسار البيئة المصغرة التي أنشأناها ونختار ملف **BeBotPythonLib.dll** الذي يمثل مكتبة مكتوبة بلغة C# وهي عبارة عن API للتحكم في بيئة بايثون المصغرة، وهي التي تمكنك بالتحكم وتحديد كيف وأين ومتى يتم استخدام بايثون، من خلال الدوال التي توفرها، هي التي تجعل عملية الدمج بين اللغتين بسيطة وسهلة، وسيتم التطرق لاستخدامها لاحقا، ثم نضغط على Add.



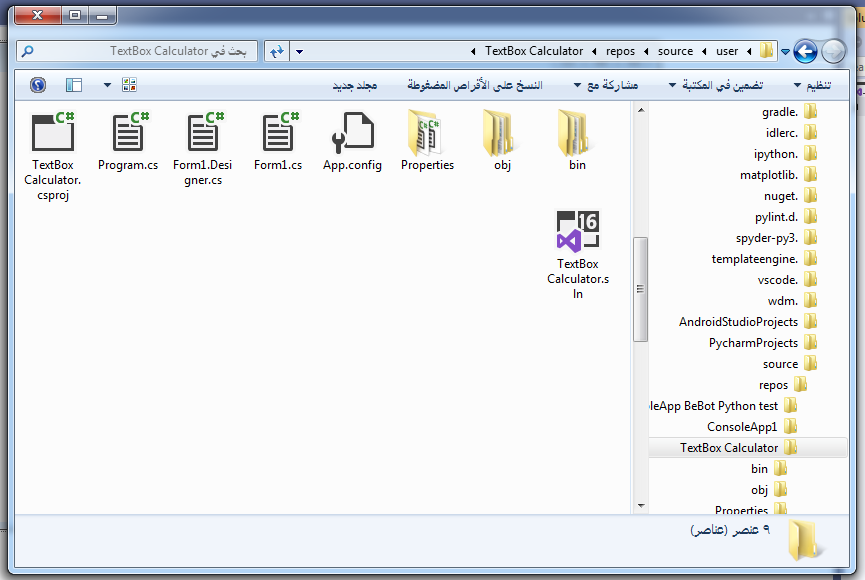
نضغط ok لإضافتها



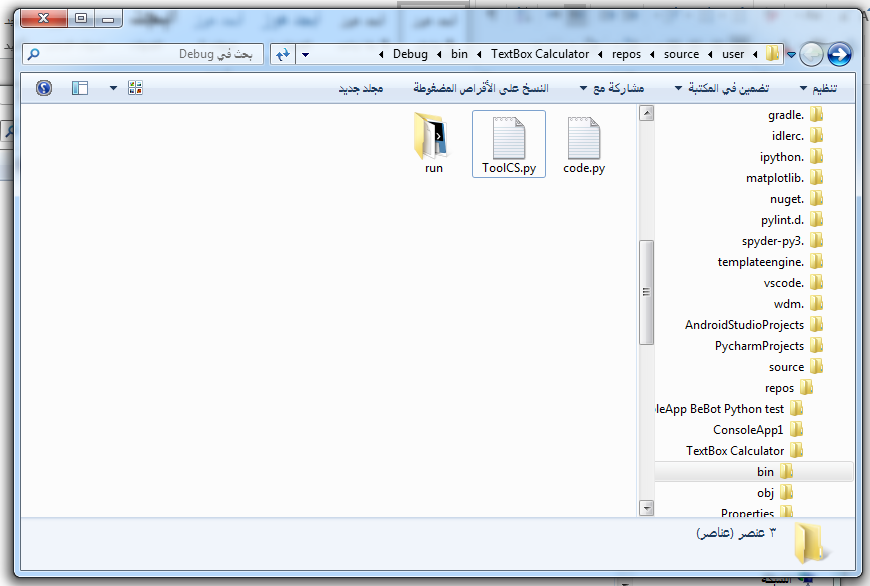
تمت إضافة المكتبة بنجاح، والآن سنقوم بإضافة ملفات البيئة المصغرة للمشروع



نضغط بالزر الأيمن من الفأرة على اسم المشروع كما في الصورة، ثم نختار Open Folder in Files Explorer



سيقوم بفتح مجلد المشروع نذهب ل bin ثم ل Debug ، وهو المسار الذي سيتم توليد فيه الملف التنفيذي لبرنامج سي شارب، ثم نقوم بنقل ملفات البيئة المصغرة لهذا المسار



الملفات المنقولة، بعد النقل لاحظ مجلد run والذي يحوي بيئة بايثون، ووجود ملفين بايثون

ملف ToolCS.py والذي يحوي على الدوال بايثون المفيدة في عملية الدمج

وملف Code.py وهو الملف الذي ستقوم بكتابة الرمز البرمجي ل بايثون الخاص بك

محتوى ملف ToolCS.py



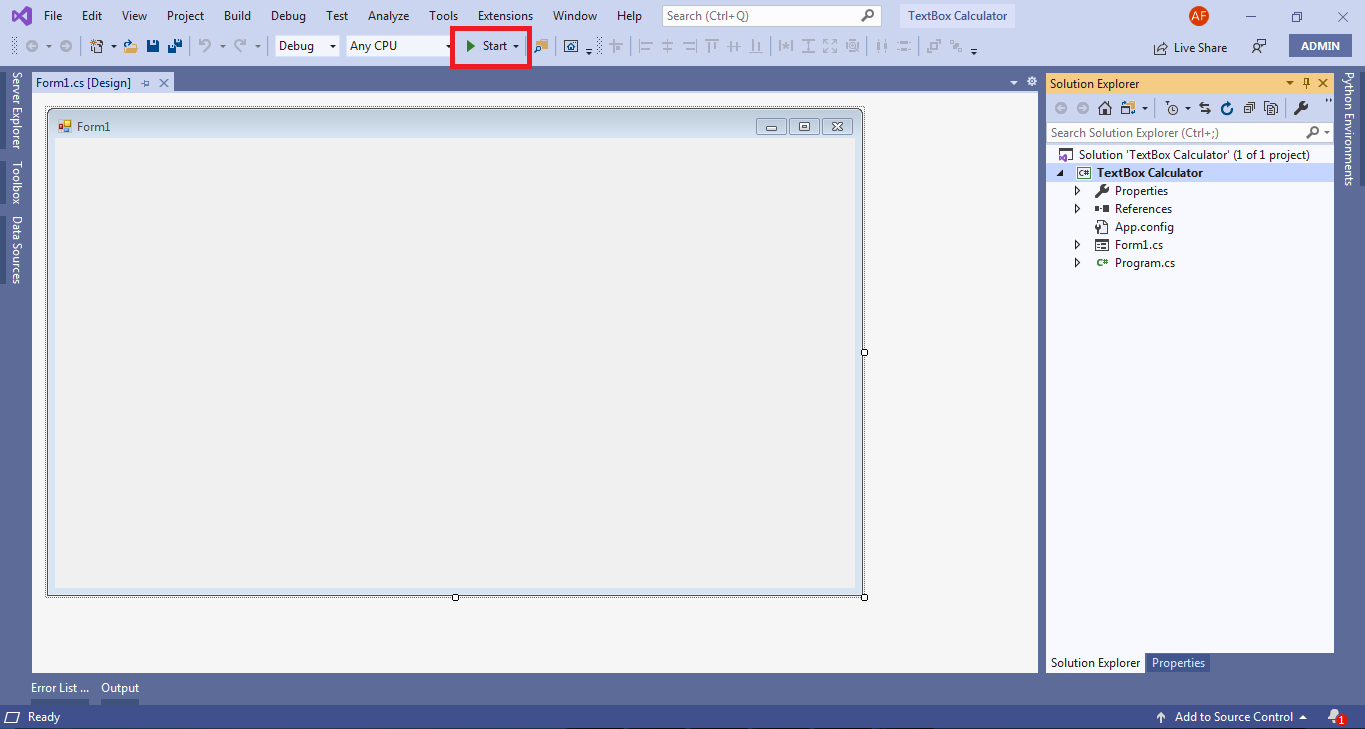
محتوى ملف Code.py



لاحظ أن دالة input تمت إعادة تعريفها لتتوافق مع الدمج ، وللعلم هناك دوال متقابلة في كل من بايثون وسي شارب سنقوم بذكرها لاحقا.

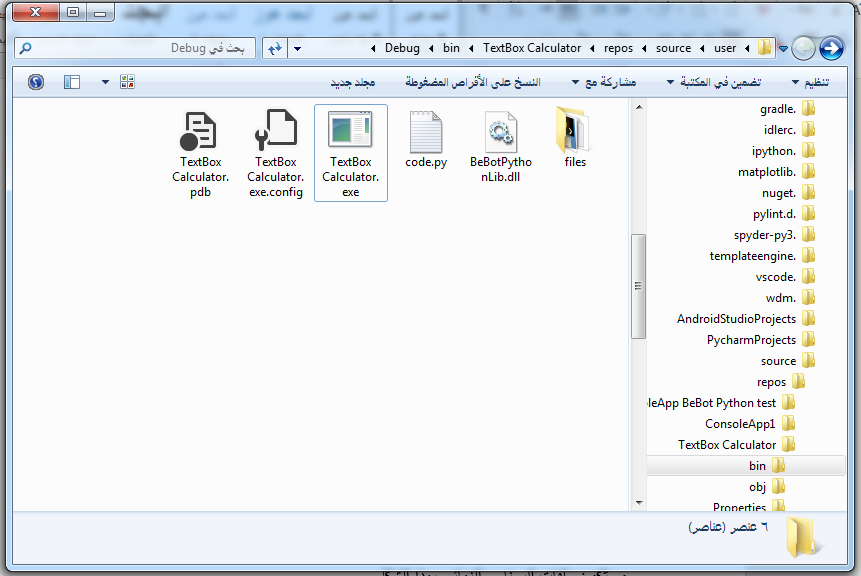
وبعدها نقوم بنقل ToolCS.py في داخل مجلد run ، وسنغير اسم المجلد ل files ، يمكنك وضع أي اسم تريد.

والآن سنقوم بالضغط على زر **start** في فيجوال ستوديو ونرى كيف سيكون شكل النهائي ل ملفات البرنامج



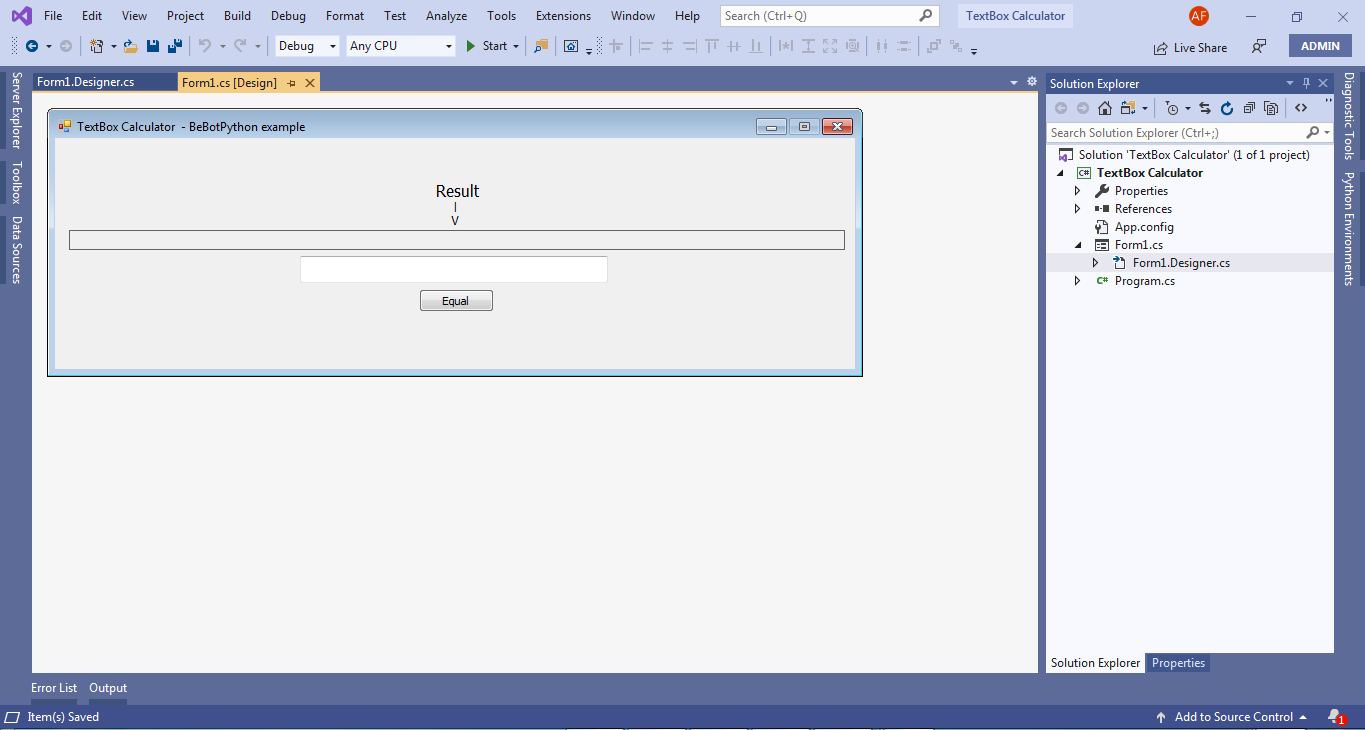
ستظهر لنا الواجهة widows form

وستكون ملفات البرنامج النهائي بهذا الشكل



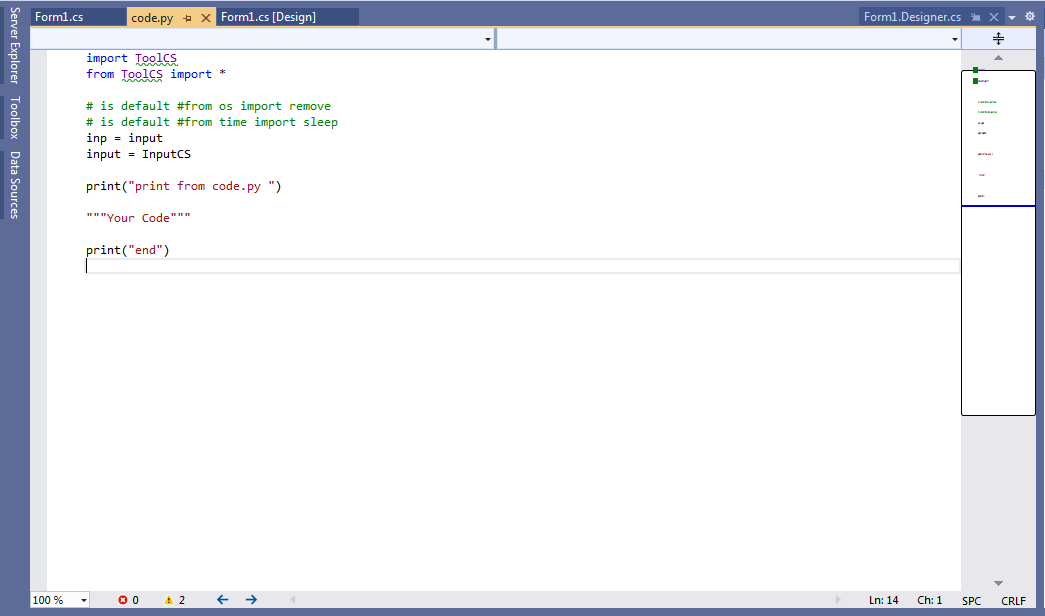
لاحظ أنه تم إنشاء مكتبة BeBotPythonLib وأيضا ملف تنفيذي بامتداد EXE وهذا الملف هو الذي يمثل البرنامج النهائي الذي نشغله.

والآن سنقوم بتصميم الآلة الحاسبة فقط تصميم دون كتابة أي رمز برمجي.

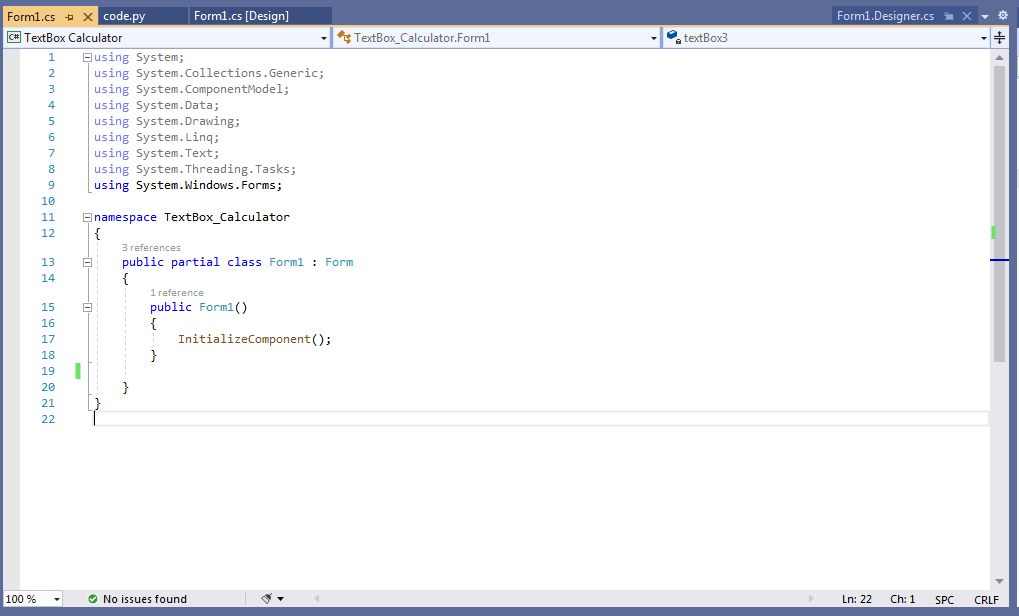


يصبح التصميم بهذا الشكل

وبعدها نفتح ملف code.py وبرمجة الخلفية للبرنامج

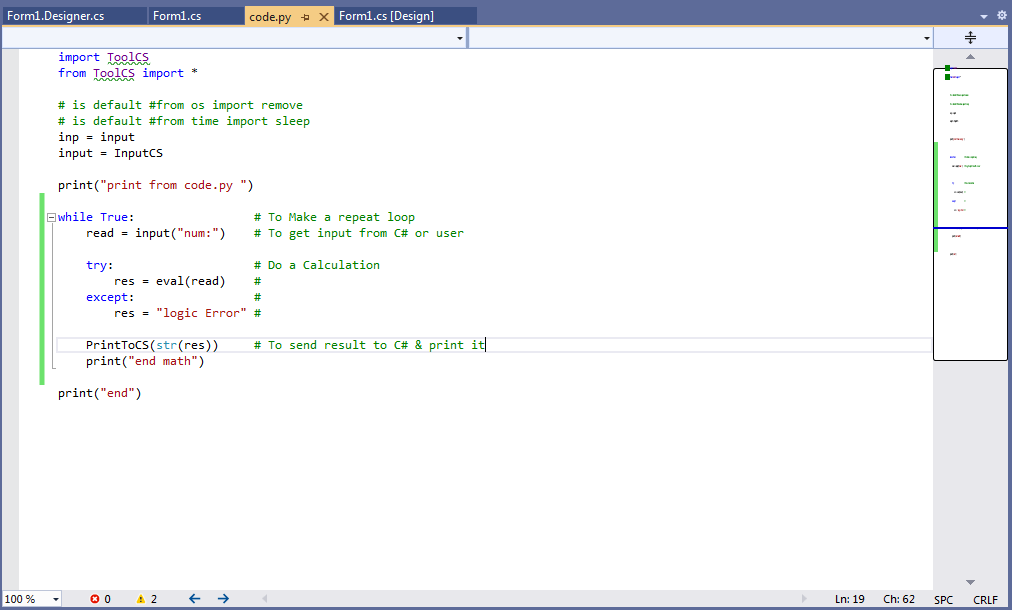


يمكنك فتح ملف code.py بأي محرر نصوص، ولكن قمنا بفتحه هنا لجعل كل شيء في داخل فيجوال ستوديو

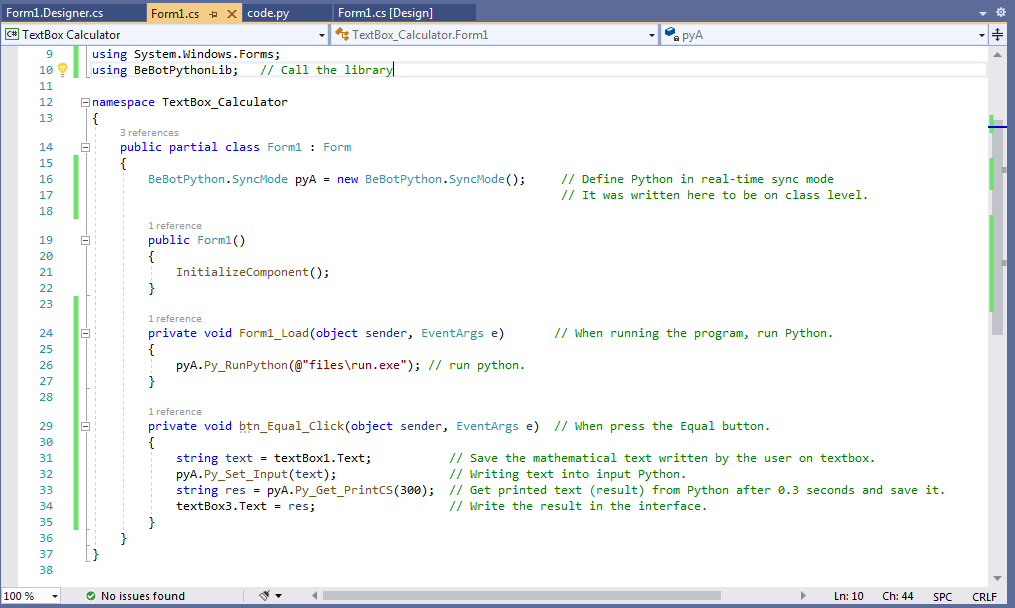


ومن هنا نبدأ بكتابة الكود البرمجي الذي يعمل في الخلفية يتم كتابهم في هاذين المكانين ، وللعلم سنستخدم في هذا المثال وضع المزامنة الآنية بين اللغتين.

يفضل أن نقوم بالبداية بكتابة كود بايثون

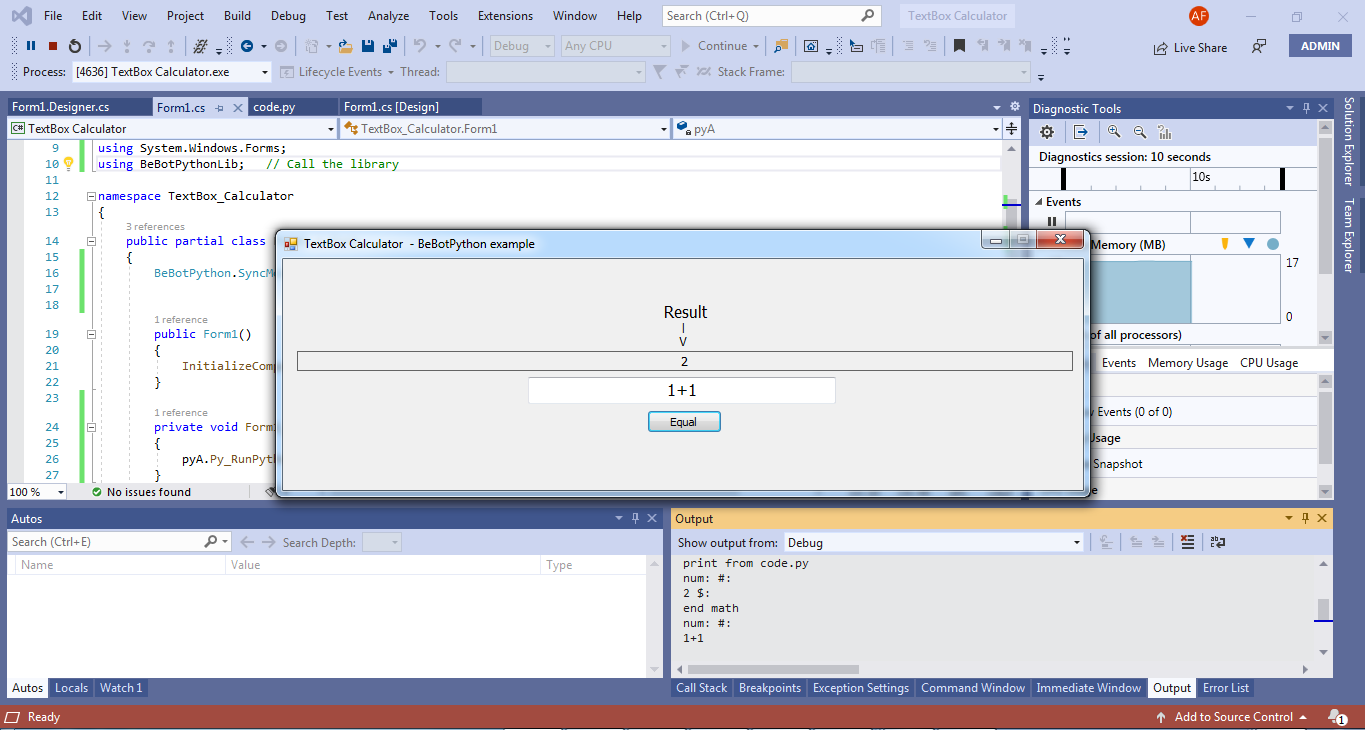


كود بايثون



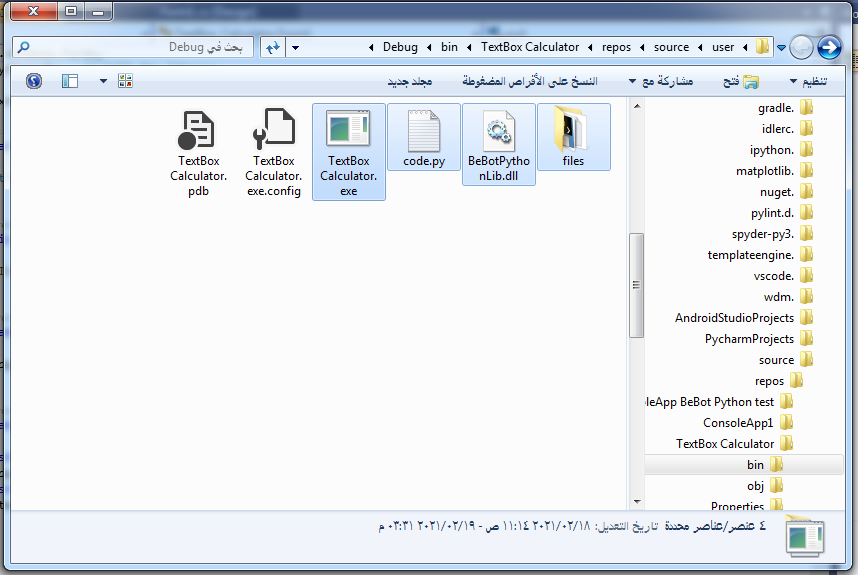
كود سي شارب

والآن سنقوم بالضغط على زر start في فيجوا ستوديو ونشغل البرنامج

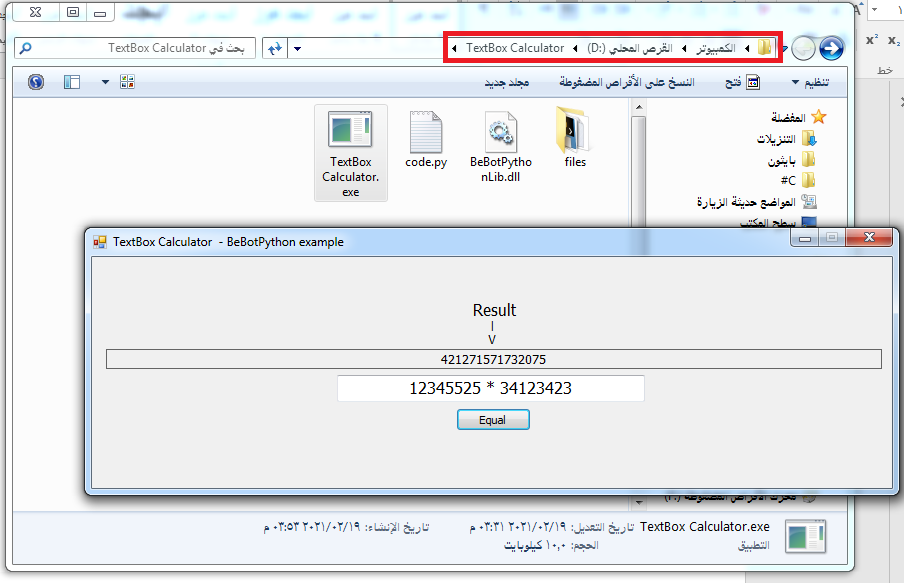


سيعمل البرنامج هكذا، ولاحظ أن البرنامج يعرض ما يحدث مع بايثون في شاشة في Output

ومبروك لقد أنهيت البرنامج بنجاح، وسنقوم بإعداد البرنامج النهائي



هذه الملفات التي تسلم للزبون النهائي بإمكانك انشاء مجلد ونسخ الملفات إليه



بعد إنشاء مجلد ونسخ الملفات إليه، هذه هي النتيجة النهائية ، وهذا المجلد الذي تسلمه للمستخدم النهائي.