



# الاسم و البرمجة

عمر سليم

I think soon all architects will need to become programmers."

اعتقد انه قريبا كل المعماريين سوف يحتاجون أن يكونوا مبرمجين

Marty Doscher of Synthesis and formerly of Morphosis, circa 2011

### تعريف بالكاتب:

عمر سليم:

- مدير لمشاريع نمذجة معلومات البناء بخبرة أكثر من 19 سنوات.
- مساعد باحث سابق بجامعة قطر.
- مؤسس مجلة BIMarabia وكذلك محرر لقاموس البيم النسخة العربية .BIM Dictionary
- قام بالاشتراك في تجهيز الأنظمة للعديد من المشاريع الكبيرة مع شركات مثل .(Saudi Diyar (Egypt (UCC (Qatar (EHAF (Qatar
- قام بالعمل في جزئية الدعم في مجال نمذجة معلومات البناء وكذلك في مجال التنسيق ومجال تطوير المحتوى للعديد من الفرق العاملة بتكنولوجيا البيم.
- يؤمن بأهمية البيم وأهمية استخدامه بديلاً عن الطرق التقليدية المتبعة وأنه ليس مجرد أداة استعراضية ثلاثة الأبعاد.
- قام بالعمل مع العديد من الاستشاريين في الهندسة المعمارية والإنسانية بهدف تطوير معايير تنسيق للمشاريع لتقليل نسب الخطأ ومشاكل التقاطعات.
- يستطيع العمل جيداً في فريق والعمل مع كافة المتخصصين سواء مقاولين ومهندسين أو ملاك أو مصممين لضمان ظهور ونجاح فكرة المشروع وتنفيذها بشكل صحيح.
- متخصص في إدارة الكاد وإدارة البيم وكذلك في النمذجة الثلاثية الأبعاد وأيضاً التدريب وبالطبع العمل في مشاريع البيم مع الفرق والتخصصات المختلفة.
- شارك في العديد من الأبحاث العلمية.



videos

[https://www.youtube.com/channel/UCZYaOLTtPmOQX1fgtDFW52Q?sub\\_confirmation=1](https://www.youtube.com/channel/UCZYaOLTtPmOQX1fgtDFW52Q?sub_confirmation=1)

بيم ارابيا

<http://bimarabia.com/>

<https://www.facebook.com/OMRSELM>

<https://www.linkedin.com/in/omarslm/>

Wordpress: <https://bimarabia.com/OmarSelim/> ;

Instagram: [https://www.instagram.com/omar\\_selim/](https://www.instagram.com/omar_selim/)

في حالة وجود اي نقص او معلومة لم تجدها او غير واضحة فبرجاء اخباري <https://www.linkedin.com/in/omarslm>

2	تعريف بالكاتب:
5	مقدمة عن البرمجة
13	لغات البرمجة لأشهر البرامج الهندسية
16	عن لغات البرمجة
17	Visual Basic .Net
18	c#
21	DYNAMO
34	تغيير اسماء العرض بواسطة الدينامو 29
42	Python
44	Grasshopper 3d
49	GenerativeComponents
50	AUTOLISP
93	Ruby
95	Matlab
97	JavaScript
98	MAXScript
100	GDL
102	Xamarin
103	BimQL
104	لغاة البرمجة رينج Ring programming language
107	مصادر لتعلم البرمجة

## مقدمة عن البرمجة

تعريف البرمجة :

هي إعطاء أوامر للحاسوب لأخباره كيف يتعامل مع حدث معين أو تنفيذ سلسلة من الأوامر المطلوبة أو عمل الأوامر المتكررة

الاوتوكاد و الريفيت و الاركيداد ما هم الا برامج

ليس مطلوباً منك في هذه المرحلة أن تبرمج برنامج كامل تنافس به الريفيت والاركيداد، ربما لاحقاً تعلو بك همتك

بل كتابة برامج قصيرة سريعة تساعدك في عملك

ما الذي سيعود على كمهندس معماري او انشائي و ميكانيكا او كهرباء من تعلم البرمجة ؟

أنت أدرى الناس بعملك ، أكثر من أي مبرمج آخر ، قد تكون لديك عمل متكرر سيأخذ منك أيام ، بکود بسيط يمكنك كتابة أسطر قليلة و تنهي العمل في دقائق

أمثلة بسيطة واقعية حدثت معي:

- لديك قطعة أرض كبيرة و تريد وضع (Y,X) على كل نقطة بالمشروع ، بالطريقة اليدوية سيأخذ هذا أيام لكن يمكنك كتابة كود بسيط او التعديل على برنامج موجود يمكنك الانتهاء منه في دقائق
- مشروع ضخم على الاوتوكاد و مطلوب تحويل وحدة قياس من الامريكي للبريطاني ، فإن تحويل رقم رقم سيأخذ وقت طويلاً ، استعنت بكود autolisp لتحويل البوصة لستنتمتر وقمت بالتعديل عليه وأنهيت التحويل سريعاً للتفرغ للتصميم
- قد تكون لديك فكرة معينة لتطوير البرنامج أو تشعر بقصور في البرنامج ، يمكنك سد النقص بنفسك و عمل أداة تؤدي ما تريده

فوائد البرمجة :

- حين تفكك كمبرمج تختلف كثيراً عن التفكير كمستخدم عادي و ستعلم البرامج أسرع
- تقليل زمن الرسم و التصميم من ساعات إلى دقائق بل إلى ثوانٍ في بعض الأحيان
- توفير أوقات كبيرة للرسامين و المهندسين الموجودين لتمكنهم من أداء أعمال او تصميمات أخرى لصالح الشركة
- توفير التكلفة الباهظة لشراء برامج رسم وتصميم أجنبية
- تحقيق الدقة المتناهية في تنفيذ الرسومات و التصميمات حيث يتم تنفيذها عن طريق البرمجة وليس بالرسم العادي في الاوتوكاد

- تخصيص البرنامج كما تحب، كم مرة فكرت لو كان البرنامج يفكر بطريقة أخرى كم سيكون أسهل ، مثلاً إدراج العمود في الريفيت من أي ركن وليس من المنتصف ،
- قد يكون البرنامج لديك قاصر في نقطة معينة ، ببساطة يمكنك عمل برنامج صغير يؤدي الوظيفة التي تريدها

معنى اعمل بنكاء و ليس بار هاق ، البرمجة ستسهل عليك انجاز العمل

و ليس المطلوب منك برنامج منافس ، بل أداة مساعدة مخصصة لك

و انت فاهم مجال عملك سواء عمارة او انشاء اكثراً من المبرمج خريج حاسبات و معلومات الذي لا يعرف ما يبرمجه

هناك مبرمجين عرب عملوا ببرامج في الريفيت و غيره من البرامج

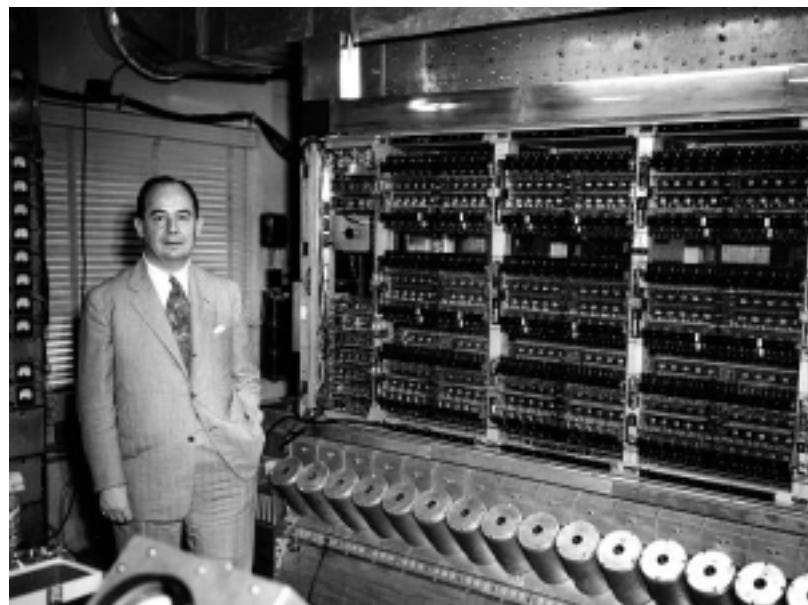
ايضا البرمجة ستطرد عنك الملل ، بدل من قضاء ساعات في تكرار بعض أوامر ، ستقوم خلال دقائق بالتفكير و تنشيط عقلك في ايجاد فكرة او ابداع كود يقوم بالمهمة و تنتهي من العمل مبكرا .

و ندعونا نبدا بتعريف الكمبيوتر ذلك الجهاز السحري

الكمبيوتر هو واحد من أكثر الهدايا الرائعة من العلم. وقد تم تطوير هذا الجهاز في الأصل من قبل تشارلز باياج.

وقد نهضت معظم البلدان سريعاً بسبب الحوسية. سنتحدث في هذا العدد عن تاريخ الكمبيوتر و البرمجة بصورة مختصرة و ذلك لتدعيم المعنى لـ

.Revit API



ما هو تاريخ البرمجة (مقاله للمهندس : مصطفى خليل)

تاريخ الـ .NET منذ بداية صناعة الحاسوب :

في بداية مرحلة صناعة الحاسوب قام العلماء بتصميم من النوع الكبير و كانت نتيجة التصميم جهاز ضخم يحتوى على خمس طوابق تقريباً . وكانت فكرة إنشاء جهاز حاسب آلي قد نتجت إثر ضخامة المعلومات و للتطوير من طريقة الكتابة على الورق ومن ثم توفير الوقت الكبير للعمليات الحسابية الأربع . وكانت العمليات الأساسية للحاسوب تبرمج عن طريق اللغة الثنائية 0,1 لأن جهاز الحاسوب لم يكن يفهم سوى هذه اللغة . ثم ظهرت لغة التجميع المعروفة بالأسمى Assembly . وكانت IBM من الشركات الرائدة في هذا المجال بحيث كانت تصمم جهاز حاسب آلي و تضع فيه نظام تشغيل صغير Operating System (OS) والتي أدت فيما بعد إلى تطور مصطلح نظام التشغيل O.S والجهاز إلى تطويره . ومعنى نظام التشغيل : هو النظام الذي يستخدم أجهزة Hardware الموجودة لدى الحاسوب ويتحكم بها بحيث يقوم بإستغلالها للحصول على أكبر قدر ممكن من الكفاءة .

و في تلك الفترة الزمنية ظهر العالم Tim Paterson حيث تمكّن من تطوير وتصميم نظام تشغيل خاص به عُرف باسم DOS . مما أدى إلى تبني شركة IBM نظامه بالكامل وعملت على تشجيعه . كان النظام يدعى DOS من الجملة المختصرة Operating System و معناها نظام تشغيل الفرق .

وكانت الجمل و عدد سطور أي برنامج طويل جداً و كان العمل صعباً حيث ان المبرمج كان يتعامل مع الذاكرة مباشرةً من خلال وضع وجمع وحذف .. الخ على سبيل المثال لطباعة كلمة على المبرمج أن يكتب ما بين 15 الى 20 سطر لتنفيذ ذلك فكان يتوجب على المبرمج من إيجاد طريقة جديدة لتوفير الوقت والتقليل من حجم البرنامج . فقام العلماء بابتكار طريقة جديدة وهي طريقة الدوال Procedure حيث قاموا بوضع أكثر من Instruction في مكان واحد فقط ، ومن هنا نشأ مفهوم المكتبات التي تضم أكثر من دالة وتطورت البرمجة فعرفت باسم Structured Programming و معناها البرمجة التركيبيّة . ومنها طورت عدة لغات مثل COBOL ولغة C الشهيرة و FORTRAN والتي عرفت بالجيل الثالث لغات البرمجة .

كانت شركة أبيل ماكتوش قد عملت على تطوير نظام فريد من نوعه فقد كان يستخدم الصور للدلالة على الأعمال أو ما يعرف بـ (GUI – Graphical User Interface) ، فقادت ببرمجة شكل الملف وشكل المجلد ... الخ ، وذلك بدل الشاشة السوداء التي كانت معروفة في ذلك الوقت والمسماة Command Prompt . و من هذا النظام تم التعرف على تقنية الـ Interface وهو يعني ان يتعامل المستخدم بشكل مباشر مع الكمبيوتر من خلال رسومات .

و في تلك الائتمان من التفاعل العلمي لتطوير الحاسوب ظهرت شركة SUN و انتجت نظام تشغيل جديد يدعى Linux ، وكانت تعتمد على النظام المفتوح Open Source و هو ما يمكّن المبرمجين من تطوير نظام التشغيل تبعاً لمتطلباتهم و دون التقيد بنظام ثابت . و من هنا بدء التسابق لإنتاج النظام التشغيل الكامل بواسطة كبريات الشركات مثل Microsoft و شركة Sun و شركة Apple Macintosh و شركات أخرى .

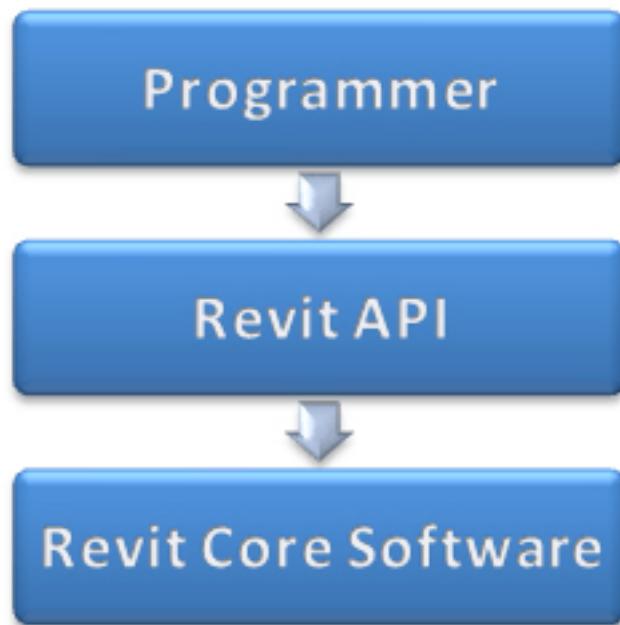
شعرت شركة مايكروسوفت بعجزها و بتحبيها عن السوق فأبتكرت فكرة التطبيقات Application و هو تحويل الواجهة السوداء التي كانت في نظام DOS إلى إشكال و رسومات و كانت أول بداية لظهور برنامج Windows وكان وقتها تطبيق و ليس نظام Windows بمعنى مشابه ببرنامج ريفيت هو تطبيق على نظام Windows

ما أدى إلى الحاجة لتوحيد طريقة البرمجة لدى المبرمجين لكي يكون الكود ديناميكي وسلس وله قواعد وشروط ، ولكن لا يخرج كل مصمم بأفكار بعيدة عن الهدف المقصود ظهر مفهوم Object Oriented Programming OPP – و هو بمعنى مشابه - تعدد اللغات و النحو ثابت .

إذن فما هي العلاقة بين برمجة أحد برامج البيم و ل يكن الريفيت مثلاً و لغة البرمجة:

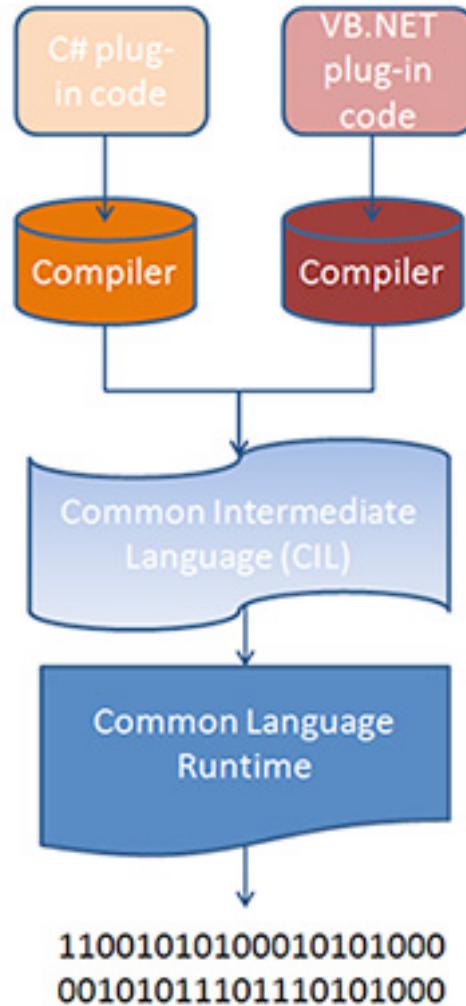
API هى اختصار Application Programming Interface. أي الواجهة البرمجية التطبيقية بمعنى انك يمكنك ان تصدر بعض الاوامر لتطبيق ما ليقوم بعمل شيء معين و لذا فهو تتيح للآخرين التواصل مع التطبيق "Revit" من خلال أي آلية أخرى. على سبيل المثال حينما تقوم برسم حوائط متصلة يتطلب من المستخدم خلق غرفة عن طريق "Rooms" ولكن ممكن ب API ان تأمر الريفيت ان يخلق غرفة بشكل تلقائي بعد الانتهاء من تكوين فراغ...

ولكي تأمر الريفيت بالقيام بذلك الوظيفة يتطلب التخاطب مع الريفيت بلغة ما... و ذكرت لغة ما لأن كما ذكرت سابقاً نظام الـ OOP يتتيح التخاطب مع الآلة بأي لغة. في الحقيقة فإن تلك اللغات هي ترجمة للـ Binary Code لكي يكون من السهل على المبرمج فهم ما يكتبه و ما ستقوم به الآلة.



و بعد كتابة البرنامج يتم تحويل تلك اللغة إلى لغة وسيطة (CIL – Common Intermediate Language) و هذا التحويل يسمى Compiling. وهناك برامج عديدة تقوم بذلك الوظيفة و أشهرها برنامج Visual Studio و NetBeans و Eclipse حيث يكون المنتج في صورة DLL او EXE و بعد ذلك يتم تحويلها للغة Binary و تسمى تلك المرحلة CLR – Common Language RunTime. وهي لغة الآلة الأساسية يتم تكوينها داخل الـ Revit أثناء تشغيل البرنامج.

ولكن لكي تستطيع ان تكتب برنامج يفهمه الـ Revit كان لزاماً على شركة Autodesk اتاحة إمكانية التدخل في إطار محدد لبرنامج Revit و تدعيمه بتطبيقات من قبل مستخدميها. و بدون ذلك لكان من المستحيل كتابة اي برنامج تطبيقي للريفيت. و اقصد بطار برنامج محدد لبرنامج Revit انه متاح لك بعمل وظائف تطبيقية معينة على سبيل المثال لا يمكن عن طريق API تخليق Ceiling و لكن يمكن نمذجة حوائط. بإختصار شديد Revit API هو واجهة برمجية لجوهر الرئيسي لبرنامج Revit



## و قبل ان نبدأ دعونا نفرق بين Macros vs. Add-Ins

الطريقتين تستعملان في برمجة برامج مساعدة و الكود بينهما متشابه جدا

تطوير المايكرو اسهل و ابسط , لكنك محدود في أدوات التطوير و مشاركتها مع الآخرين

تطلب منك معرفة اضافية و إعدادات **Add-Ins**



## Macros

يتم تنصيبها [SharpDevelop](#)، بيئة عمل حرة و مفتوحة المصدر تسمى Integrated Development Environment, or IDE مع الريفيت

تنصل بالريفيت مباشرة ، لذلك عملية إنشاء الماكرو بسيطة نسبياً لكن هناك مبرمجين يفضلون Microsoft Visual Studio. ولكن لتطوير وحدات الماكرو يجب استخدام [SharpDevelop](#).

مكان تخزين الماكرو `\C:\ProgramData\Autodesk\Revit\Macros`

يمكنك مشاركة وحدات الماكرو مع الآخرين سواء من خلال إرسالهم الملف RVT (إذا كان الماكرو مدمج به ) أو إرسالهم شفرة المصدر (ولكن هذا سوف يستغرق قليلاً من الجهد من جانبهم لإنشاء ماكرو على الريفيت لديهم)

## Add-Ins

يتم تطوير الوظائف الإضافية [Add-Ins](#) خارجياً للريفيت، عادة باستخدام [Visual Studio](#) ، النسخة المجانية تسمى فيسوال ستوديو إكسبريس [Visual Studio Express](#) ، أو أي IDE أخرى.

كود [Add-Ins](#) يكون مدمجاً ك `.dll`

و يكون في `\C:\Users\omar\AppData\Roaming\Autodesk\revit\Addins`

# لغة برمجة مرئية visual programming language

هي أية لغة برمجة تسمح للمستخدمين بإنشاء برامج عن طريق التلاعب بعناصر برنامج بيانيًا بدلاً من تحديدها حرفياً، أي أنها تسمح بالبرمجة بواسطة التعبيرات البصرية والترتيبيات المكانية للنصوص والرسوم البيانية والرموز. على سبيل المثال، العديد من لغات البرمجة المرئية المعروفة باسم تدفق البيانات *dataflow* أو البرمجة البيانية تقوم على فكرة "الصناديق والأسماء"، حيث يتم التعامل مع المربعات أو غيرها من كائنات الشاشة على أنها كيانات متصلة بواسطة أسماء أو خطوط أو أقواس والتي تمثل العلاقات.

## لماذا البرمجة البصرية؟

تعتبر البرمجة البصرية صورة من صور البرمجة ولكن بصورة مبسطة فالبرمجة التقليدية المعروفة تعتبر معقدة جداً بالنسبة للمعماريين والمصممين والفنانين مع إمكانياتها الهائلة إلا أنها لم تكن منتشرة ومستخدمة إلى أن ظهرت هذه التطبيقات منذ العام ٢٠٠٧ عندما ابتكر دافيد روتون تطبيق *Grasshopper 3D* والذي يعمل على برنامج *Rhinoceros* ليبدأ عصر *Parametric design* ولم تنتظر طويلاً حتى ظهر ما يعرف بالـ *MCG* في العام ٢٠١٥ وأخيراً ظهر تطبيق *Dynamo* لينضم إلى برنامج *Revit* وليلعنه بداية عصر جديد لنموذج المعلومات بعد اندماجها مع *3DsMax* ويبداً سباق التنافس بين الشركات المنتجة لهذه البرمجيات وبالتالي على المعماريين والمصممين ليتفوق من يملك المعلومة في صناعة البناء بعدهما كان التطور في صناعة البناء مرتبط أكثر بتطور خامات البناء أصبح اليوم مرتبط أكثر بتطور هذه البرمجيات والتي تتقن في الإستخدامات الجديدة لخامات البناء التقليدية ليخرج إلى النور تصاميم مبتكرة وفوق إمكانيات العقل البشري وحده.

## هل يحتاج إلى خبرة في مجال البرمجة؟

الشركات المنتجة لهذه البرمجيات تطمئن المستخدمين بأن هذه البرمجيات لا تحتاج إلى الخبرة العالية في مجال البرمجة إلا أن الإمام بقواعد المنطق البصري وعلى الرياضيات وأحياناً الفيزياء وعلم الأحياء أيضاً قد يكون مفيد عند تعلم هذه التطبيقات.

ومن المتوقع أن المنافسة لن تخفت حدتها في تطوير برمجيات صناعة البناء في السنوات القليلة المقبلة وإنما ستتوسع أكثر وأكثر وقد تمت ليكون الحاسوب الآلي هو المفك والمدير بدلاً من المهندس يوماً ما.

كيف أبدأ؟

البداية تكون من تحديد ما هو البرنامج الذي تحتاج للعمل عليه و تتعلم لغة البرمجة المناسبة له

البيم (نمذجة معلومات المبني هي تمثيل رقمي شامل للمبني التي تشمل الهندسة ثلاثية الأبعاد والبيانات المرتبطة بها) هو ليس برنامج فلا توجد لغة برمجة له ، توجد فقط لغات برمجة للبرامج التي تطبق البيم

لذلك يجب تحديد اولا ما هو البرنامج الذي تريده برمجته ، سناقش الان أهم البرامج وقد توسيع قليلا و ذكرت ببرامج ليست تحت مظلة البيم لكن الواقع العملي اننا نستخدمها في الحياة العملية ، و سناقش أهم لغات البرمجة ثم نفرد حديثا لكل لغة

## لغات البرمجة لأشهر البرامج الهندسية

- الريفيت يمكنك تعلم C# او الفيجوال بيسك VB او البايثون ، اذا كنت لا تعرف ايهما فتعلم ال C# لأن لها أمثلة و تطبيقات كثيرة
- تتوفر دروس تعليمية و أمثلة ل C# أكثر من VB
- الاسكتش اب ruby لديك SKETCHUP
- البلندر Blender و المايا maya يمكن عمل برامج لهم بلغة بايثون python و هي لغة سهلة مفتوحة المصدر
- الاركيcad archicad يمكن بناء العناصر به من خلال لغة برمجة GDL Script و هي اختصار Geometric Descriptive Language
- لبرمجة tekla عليك تعلم C# هذه الصفحة مفيدة <https://www.tekla.com/openAPI>
- كما يمكنك تعلم لغة برمجة مرئية visual programming language حيث لن تضطر حتى لكتابة كود ، يكفي أن تستوعب فكرة البرمجة و منطقها
- مثل ال DYNAMO للريفيت و Grasshopper للرينيو GenerativeComponents لبرمجة bentley
- الاتوكاد يمكنك تعلم الاتوليس (لغة ذكاء صناعي ) او vba (مشتقة من الفيجوال بيسك )
- لبرمجة الليسبات للاوتوكاد AUTOLISP مقدمة لتعلم ال
- محرك العاب قوي و شهير و بيئة التطوير المتكاملة (IDE) حيث يعمل البرنامج على عدة أنظمة تشغيل مثل ويندوز أو ماك OS ، وبالإمكان نشر الألعاب و التطبيقات على الأجهزة المختلفة مثل أجهزة الكمبيوتر ، xbox ، ipad ، سهل الاستعمال و يتم البرمجة بلغة ( JAVASCRIPT & C# & POO )



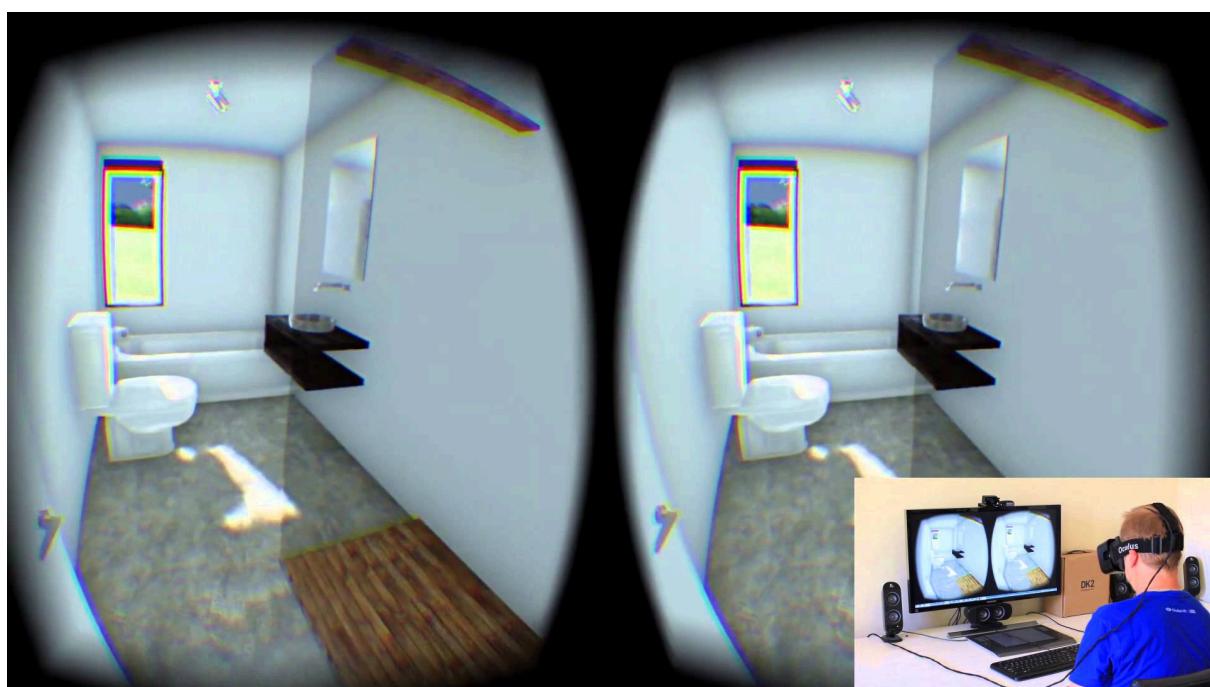
<http://www.unity3d.com>

- لبرمجة الواقع الافتراضي تحتاج لتعلم لغة Processing أو C# OR C++ أو استعمال UNITY

يقول بالمر لاكي، مؤسس أوكلوس:

"يملك الواقع الافتراضي القدرة على جعلك تقوم بأي شيء، أي شيء قد تتخيل أنك تقوم به في العالم الواقعي، إضافة إلى مجموعة من التجارب غير الممكنة في العالم الحقيقي."

"يمكنك أن تأخذ أشخاصاً من مختلف أصقاع الأرض وتضعهم معاً في غرفة افتراضية واحدة، حين تصل إلى ذلك بالشكل المطلوب ستتخلص من الحاجة للسفر وحرق أطنان من وقود الطائرات لتدور حول العالم. ستتخلص من الحاجة لاجتماعات الكبرى حيث تتفق الكثير من الموارد لمجرد أن يجعل الناس يلتقطون ويتحدثون في غرفة واحدة"



• فكر في برمجة المحمول ، يمكنك وضع اعلانات والربح منها بسهولة ، جرب Xamarin

## عن لغات البرمجة

لغات البرمجة (Programming Languages)، وهي مجموعة من اللغات الإلكترونية التي تستخدم مع الأنظمة، والتطبيقات الرقمية التي تعمل في الأجهزة الإلكترونية الحاسوبية والذكية، وتهدف إلى بناء نظام، أو تطبيق، أو صفحة إنترنت، وتعرف لغات البرمجة أيضاً بأنها عبارة عن عمليات إلكترونية تهدف إلى تجميع وتركيب مجموعة من البيانات معاً من أجل تصليح شيء في نظام إلكتروني، أو للعمل على بناء نظام جديد يستخدم في تطبيق ما، أو موقع على شبكة الإنترنت.

مقارنة سريعة بين أهم لغات البرمجة حتى تسهل عليك الاختيار فهناك لغات برمجة لا تعمل إلا على نظام تشغيل معين و هناك لغات أخرى تعمل على جميع أنظمة التشغيل



## Visual Basic .Net

لغة سهلة و بسيطة واحداً من لغات الجيل الثالث من لغات البرمجة، وتوصف بأنّها بيئّة التطوير المتكاملة IDE من مايكروسوفت، وقد صدرت لأول مرة في يوم 1 من الشهر الثامن لعام 1991 ميلادياً، وتعدّ من اللغات التي يسهل استخدامها وتعلمها، ويستطيع المبرمج الذي يجيد العمل على البرمجيّة استخدام مكوناتها لإنشاء تطبيق بسرعة، ويمكن من خلالها المرور عن عقبة الأخطاء التي يمكن أن تحدث أثناء وضع الأكواد، ويطرح لك العديد من الأمثلة ليسهل عليك عملية الكتابة.

### عيوبها:

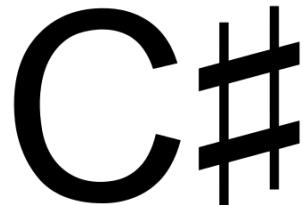
- إن سهولة هذه اللغة قد تسبّبت في جعل إمكانياتها محدودة جداً جداً، فمايكروسوفت قد تخلت عن القوة مقابل السهولة.
- عدم دعمها لكل أوجه البرمجة الكائنية.
- مترجم اللغة نفسه به شوائب.
- عدم مجانية بعض برامجها.

## C#

سي# (بالإنجليزية: C#) (تلفظ سي شارب) هي لغة برمجة متعددة الأنماط تتمتع بكونها سكونية التمثيل وأمرية وتعريفية ووظيفية وإجرائية وعمومية وشبيهة المنحى (غرضية الترجمة) (باستخدام الصفوف) كما تخضع لمبادئ البرمجة التركيبية المنحى.

قامت مايكروسوفت بتطوير هذه اللغة في إطار عملها على تطوير دوت نت وتمت الموافقة على تعبيرها من منظمة Ecma (المعيار Ecma-334) والمنظمة الدولية للمعايير (المعيار ISO/IEC 23270:2006). إن سي# إحدى لغات البرمجة المصممة للعمل على البنية التحتية المشتركة للغات البرمجة CLI. صُممَت لغة سي# لتكون لغة بسيطة وحديثة وعامة للأغراض وشبيهة المنحى . وقد قاد أندرس هيلسبرغ فريق تطويرها.

- أكثر الأكواد لبرمجة الريفييت مكتوبة به
- أكثر مبرمجين الريفييت يبرمجون بها
- كتابة الكود بها أسرع



مميزاتها:

- لغة قوية وفي نفس الوقت سهلة التعلم وبرامجها سريعة التطوير.
- تدعم البرمجة الكائنية.
- تعتمد على مكتبات إطار الـ.NET. مما يسهل عملية كتابة البرامج المعقدة دون مصادر خارجية.
- إستعمالها في مجال الويب.

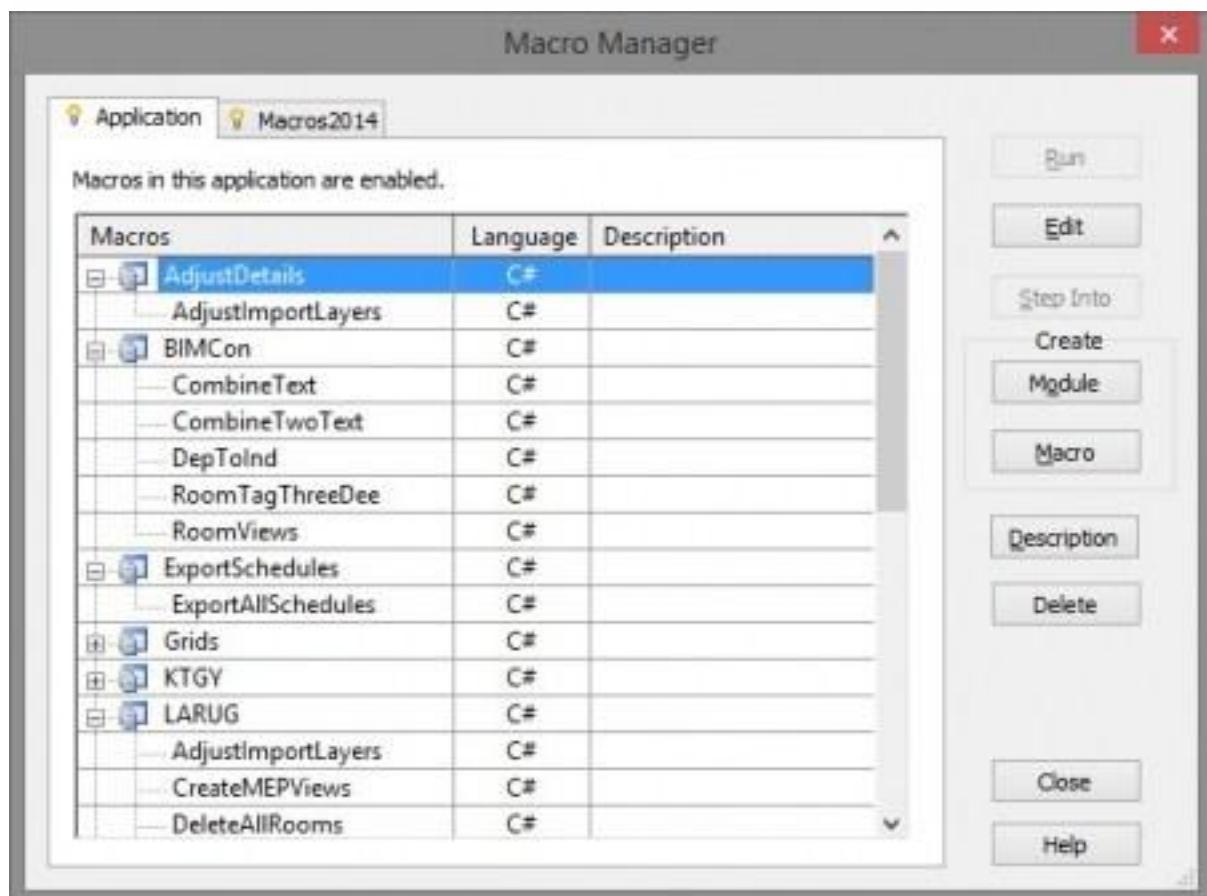
عيوبها:

- لا تعمل إلا على بيئة الويندوز، على الرغم من ظهور مشاريع كبيرة مونو ولكن تبقى حلولا غير رسمية.
- يعتبرها الكثيرون مجرد تقليد للغة الجافا، وجاءت مايكروسوفت بها للخروج من ورطتها مع شركة صن لا أكثر.

برمج مايكرو بالريفييت

من manage

micro



يمكنك برمجة المايكرو بسهولة خاصة لو تعرف #C#

مثلا لعمل اختيار قاعدة IF

```
(if (level.Elevation == 0
```

```
}
```

```
(do something if level is at 0 (true //
```

```
{
```

```
else
```

```
}
```

```
(do something if level isn't at 0 (false //
```

```
{
```

مراجع تساعدك

/Revit Coaster (my blog) : <http://revitcoaster.blogspot.com>

/The Building Coder : <http://thebuildingcoder.typepad.com>

/AEC DevBlog : <http://adndevblog.typepad.com/aec>

/Boost Your BIM : <http://boostyourbim.wordpress.com>

/SpiderInNet : <http://spiderinnet.typepad.com/blog>

Autodesk Developer Network :

<http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2484975>

Autodesk Discussions : <http://forums.autodesk.com/t5/Revit-API/bd-p/160>

/RevitForum.org : <http://www.revitforum.org/third-party-add-ins-api-r-d>

AUGI Forums : <http://forums.augi.com/forumdisplay.php?218-Revit-API>

## DYNAMO



# Dynamo

الدينامو لغة برمجة مرئية مفتوحة المصدر تُتيح للمستخدمين بناء إجراءات التشغيل الآلي للريفيت دون الحاجة لتعلم API ريفيت . وهذا يفتح فرصاً عديدة للمستخدمين من ريفيت لتخصيص سير العمل الخاصة بهم بسهولة.  
لغة البرمجة ( Programming language ) هي عبارة عن مجموعة من الأوامر، تكتب وفق مجموعة من القواعد تحدد بواسطة لغة البرمجة، ومن ثم تمر هذه الأوامر بعدة مراحل إلى أن تنفذ على جهاز الحاسوب.

لغات البرمجة المرئية هي لغات حديثة توفر إمكانية إنشاء نوافذ خاصة بالبرنامج بسهولة بدون الحاجة لكتابة الشيفرة الخاصة بالمكونات الخاصة بالنواوفذ. يتيح للمصممين بناء العلاقات البرامجية باستخدام واجهات المستخدم الرسومية . بدلاً من كتابة " كود " من نقطة الصفر

مفتوحة المصدر : هي البرمجيات التي يمكن الإطلاع والتعديل على شفرتها البرمجية وهي أكثر مرونة للمستخدم من البرامج الأخرى التي لا تتيح مرونة للمستخدم ولا يستطيع التعديل عليها والتي يسميها البعض ب البرمجيات الاحتكارية.

سؤال : ما الذي يستطيع الدينامو فعله ؟؟  
تصحيح السؤال اريد ان افعل كذا كيف يمكنني فعله .

يمكن تحميل الدينامو مجاناً من هنا  
<http://dynamobim.com/download>

ماذا يحدث عند تلاقي عملاقين لاجل هدف موحد ؟؟  
نمذجة معلومات البناء

و البرمجة هي إعطاء أوامر للحاسوب لأخباره كيف يتعامل مع حدث معين أو تنفيذ سلسلة من الأوامر المطلوبة أو عمل الأوامر المتكررة

لغة البرمجة ( Programming language ) هي عبارة عن مجموعة من الأوامر، تكتب وفق مجموعة من القواعد تحدد بواسطة لغة البرمجة، ومن ثم تمر هذه الأوامر بعدة مراحل إلى أن تنفذ على جهاز الحاسوب.  
الدينامو لغة برمجة مرئية visual programming language ( حيث لن تضطر حتى لكتابة كود ، يكفي أن تستوعب فكرة البرمجة و منطقها) مفتوحة المصدر تُتيح للمستخدمين بناء إجراءات التشغيل الآلي للريفيت دون الحاجة لتعلم API ريفيت . وهذا يفتح فرصاً عديدة للمستخدمين من ريفيت لتخصيص سير العمل الخاصة بهم بسهولة.

لغات البرمجة المرئية هي لغات حديثة توفر إمكانية إنشاء نوافذ خاصة بالبرنامج بسهولة بدون الحاجة لكتابة الشيفرة الخاصة بالمكونات الخاصة بالنواوفذ. يتيح للمصممين بناء العلاقات البرامجية باستخدام واجهات المستخدم الرسومية . بدلاً من كتابة " كود " من نقطة الصفر

مفتوحة المصدر : هي البرمجيات التي يمكن الإطلاع والتعديل على شفرتها البرمجية وهي أكثر مرونة للمستخدم من البرامج الأخرى التي لا تتيح مرونة للمستخدم ولا يستطيع التعديل عليها والتي يسميها البعض ب البرمجيات الاحتكارية.

لا شك ان الناتج قوي لان القاعدة الشهيرة "الكل اكبر من مجموع الأجزاء " و يضاف الى هذا السهولة الكبيرة في التعامل و عدم الحاجة لحفظ أكواد معقدة ، فيمكنك عمل برنامج قوي بسهولة كبيرة و دون كتابة أي كود

ما الذي سيعود علي كمهندس معماري او انسائي و ميكانيكا او كهرباء من تعلم البرمجة ؟

أنت أدرى الناس بعملك ، أكثر من أي مبرمج آخر ، قد تكون لديك عمل متكرر ستأخذ منه أيام ، بکود بسيط يمكنك كتابة أسطر قليلة و تنهي العمل في دقائق.

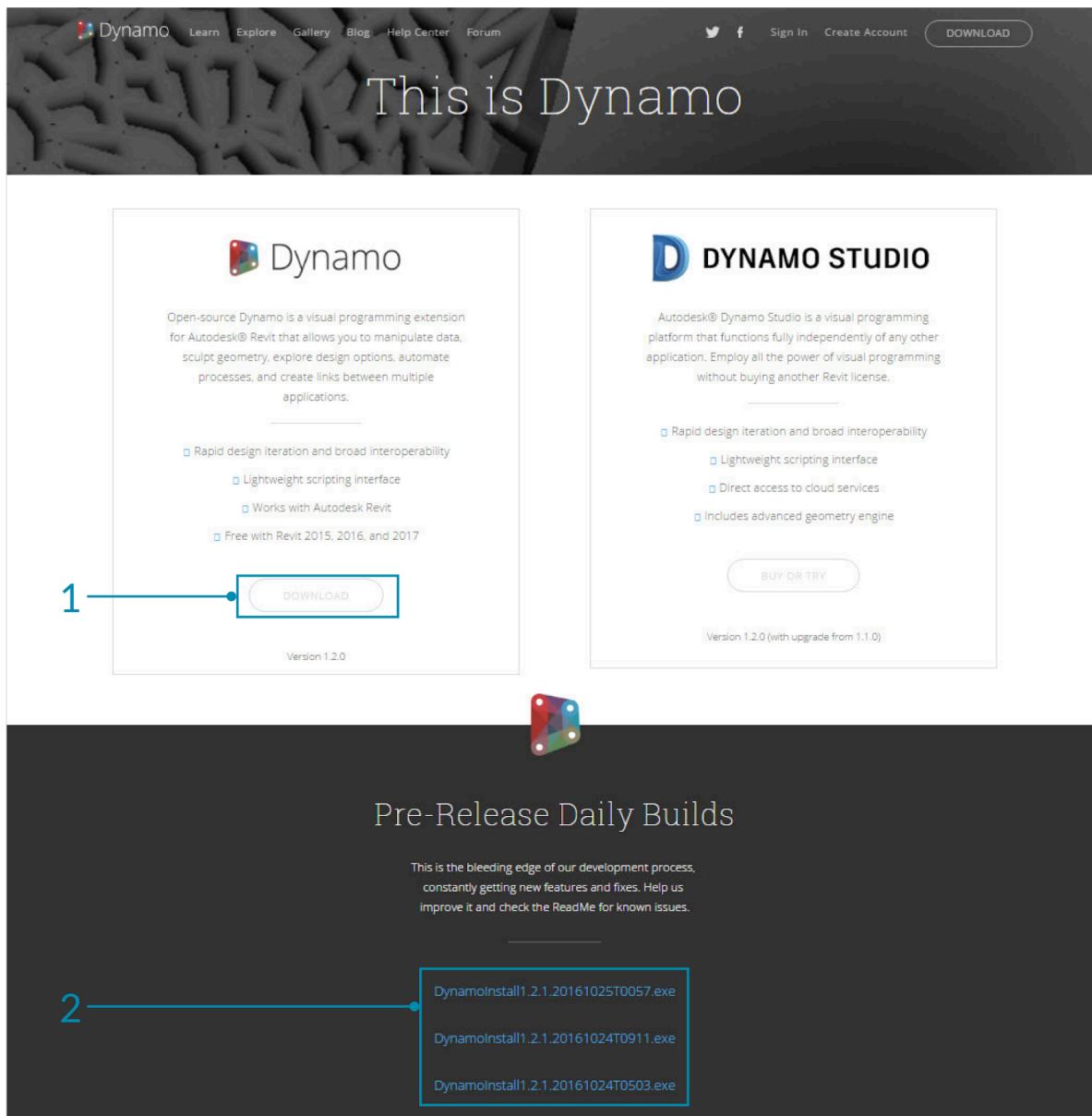
".I think soon all architects will need to become programmers"

اعتقد انه قريبا كل المعماريين سيحتاجون أن يكونوا مبرمجين

Marty Doscher of Synthesis and formerly of Morphosis, circa 2011

سؤال : ما الذي يستطيع الدينامو فعله ؟؟  
تصحيح السؤال اريد ان افعل كذا كيف يمكنني فعله .

يمكن تحميل الدينامو مجانا من هنا  
<http://dynamobim.com/download>



أمثلة بسيطة واقعية حدثت معى:

- لديك قطعة أرض كبيرة و تريد وضع (Y,X) على كل نقطة بالمشروع ، بالطريقة اليدوية سيرأخذ هذا أيام لكن يمكنك كتابة كود بسيط او التعديل على برنامج موجود يمكنك الانتهاء منه في دقائق
- مشروع ضخم على الاتوكاد و مطلوب تحويل وحدة قياس من الامريكي للبريطاني ، فإن تحويل رقم رقم سيرأخذ وقت طويـل ، أستعنت بـكود autolisp لـتحويل البـوصـة لـسـنتـيمـتر وقمت بالـتعديل عـلـيـه و أنهـيـت التـحـول سـرـيـعا لـلتـصـمـيم
- قد تكون لديك فكرة معينة لـتطوير البرنامج أو تـشـعـر بـقـصـور فـيـ البرـنـامـج ، يمكنـك سـدـ النـصـنـ بـنـفـسـك و عملـاـدـة تـؤـديـ ما تـرـيدـه

فوائد البرمجة :

- حين تفكك كمبيوتر مختلف كثيراً عن التفكير المستخدم عادي و ستتعلم البرامج أسرع
- تقليل زمن الرسم والتصميم من ساعات إلى دقائق بل إلى ثوانٍ في بعض الأحيان
- توفير أوقات كبيرة للرسامين والمهندسين الموجودين لتمكنهم من أداء أعمال أو تصميمات أخرى لصالح الشركة
- توفير التكلفة الباهظة لشراء برامج رسم وتصميم أجنبية
- تحقيق الدقة المتناهية في تنفيذ الرسومات والتصميمات حيث يتم تنفيذها عن طريق البرمجة وليس بالرسم العادي في الأوتوكاد
- تخصيص البرنامج كما تحب، كم مرة فكرت لو كان البرنامج يفكر بطريقة أخرى كم سيكون أسهل ، مثلاً إدراج العمود في الريفيت من أي ركن وليس من المنتصف ،
- قد يكون البرنامج لديك قاصر في نقطة معينة ، ببساطة يمكنك عمل برنامج صغير يؤدي الوظيفة التي تريدها

معنى أعمل بذكاء و ليس بارهاق ، البرمجة ستسهل عليك انجاز العمل

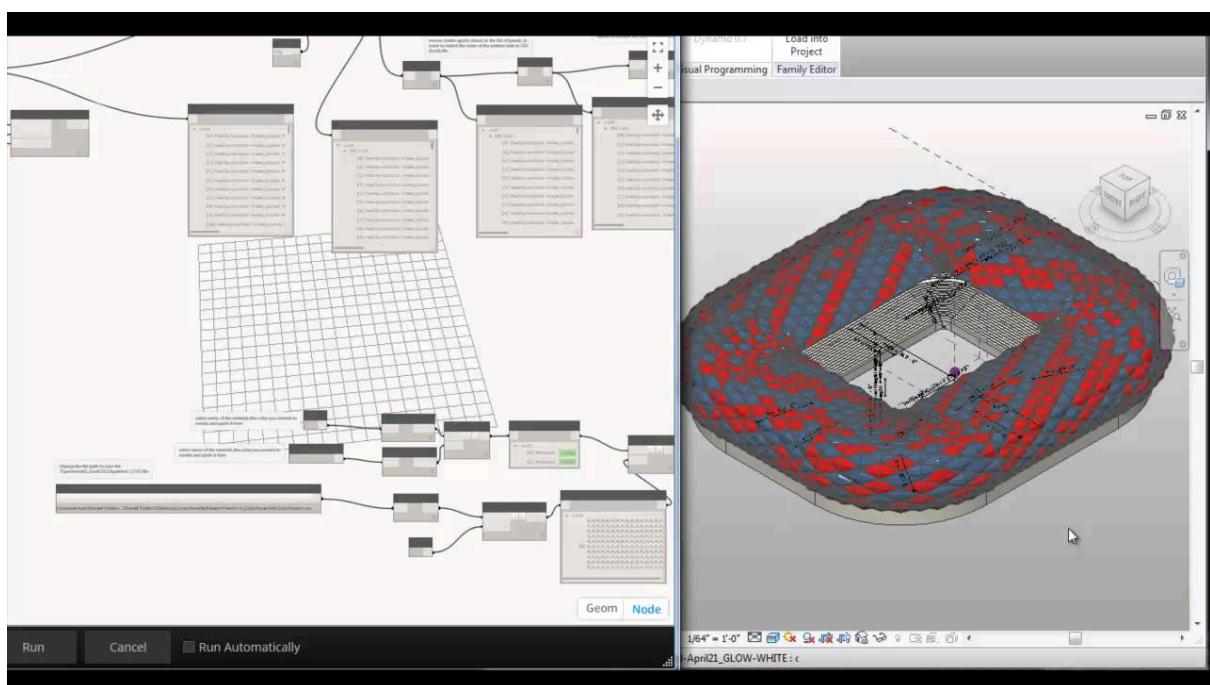
و ليس المطلوب مثل برنامج منافس ، بل أداة مساعدة مخصصة لك

و انت فاهم مجال عملك سواء عمارة او انشاء اكثرا من المبرمج خريج حاسوبات ومعلومات الذي لا يعرف ما يبرمجه

هناك مبرمجين عرب عملوا برامج في الريفيت و غيره من البرامج

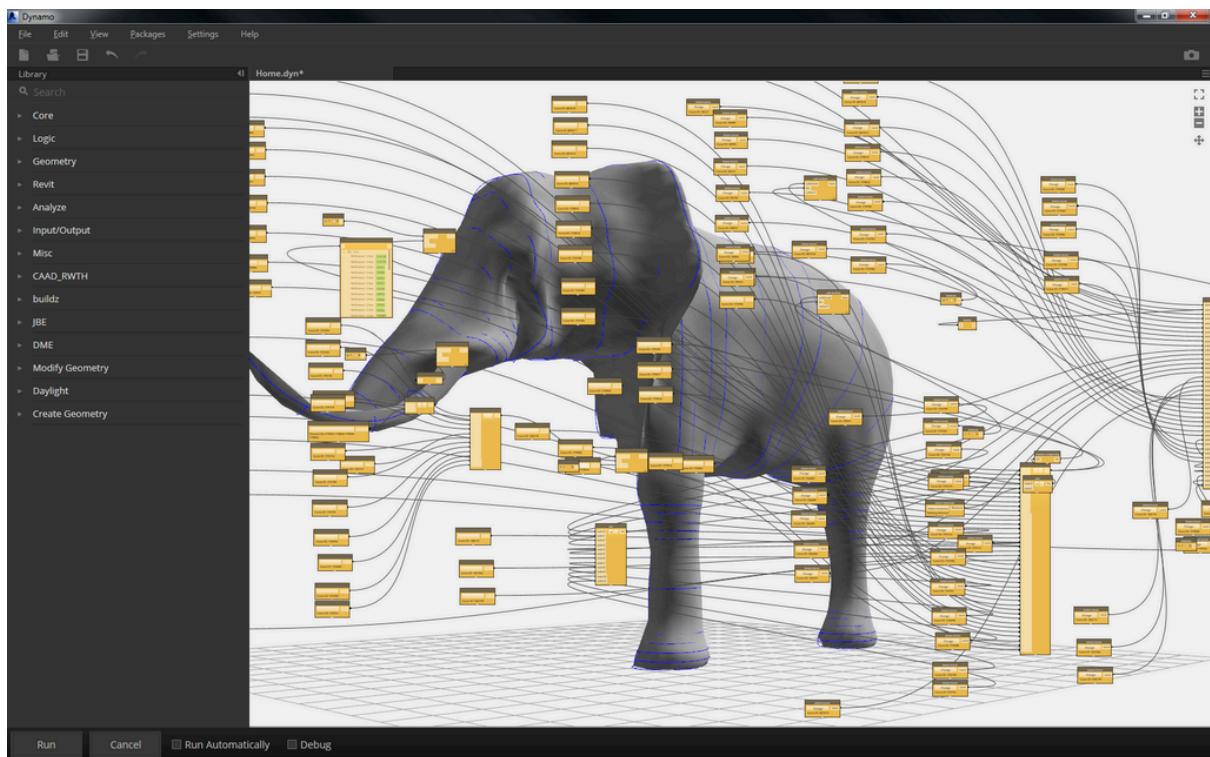
ايضا البرمجة ستطرد عنك الملل ، بدل من قضاء ساعات في تكرار بعض أوامر ، ستقوم خلال دقائق بالتفكير و تنشيط عقلك في ايجاد فكرة او ابداع كود يقوم بالمهمة و تنتهي من العمل مبكرا .

يمكنك الدخول من التحكم في الريفيت و تصميم أشكال معقدة بسهولة و تنفيذ العمليات المتكررة



هناك مشاريع فرعية منه مثل  
يمكنك التحكم بسهولة في شكل التسلیح <https://github.com/tt-acm/DynamoForRebar>

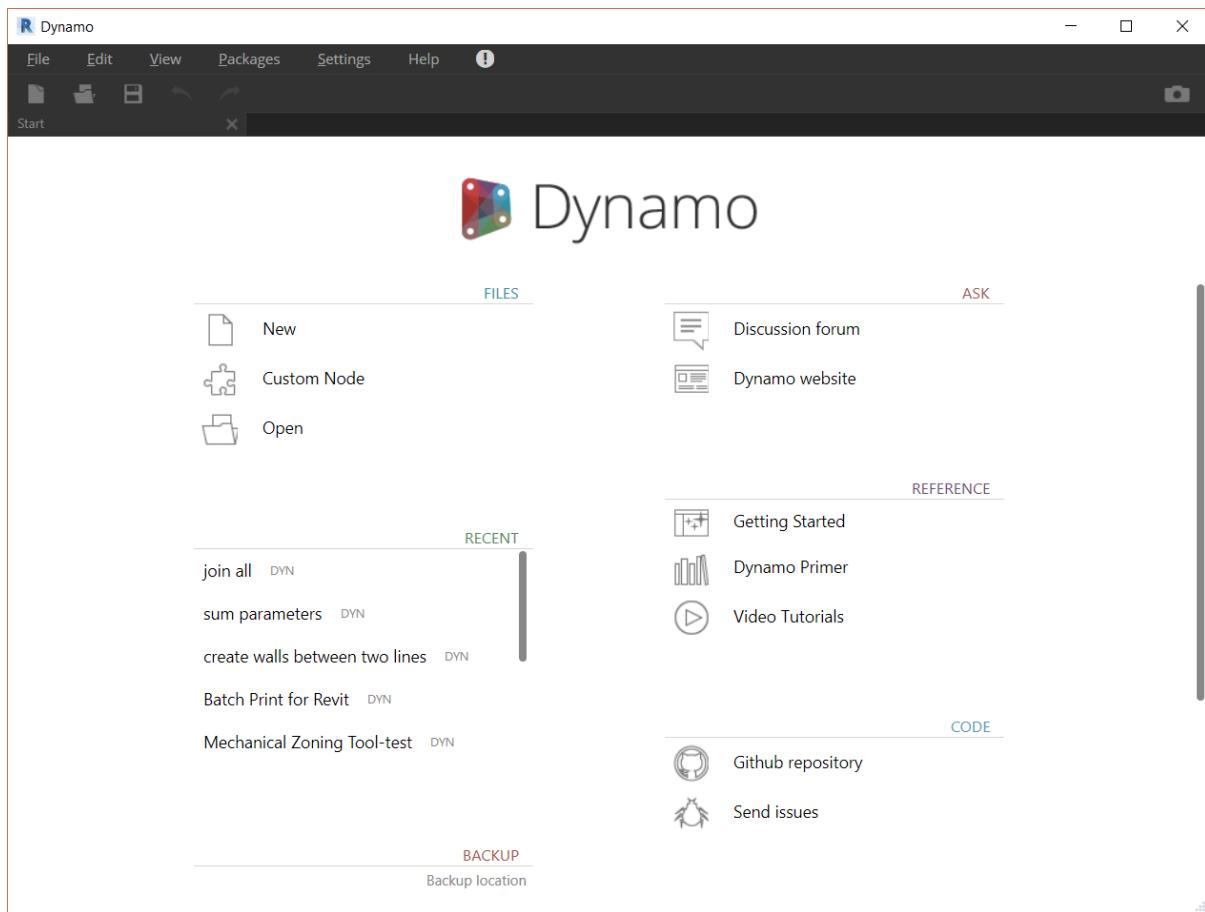
ب تكون شكل من متغيرات بحيث لما تتغير يتغير معك المجسم



عند تنصيب الدينامو نجد يظهر في ADDINS الريفيت

من بداية الاصدار ريفيت 2017 تظهر الدينامو في MANAGE بدون تنصيب

واجهة البرنامج

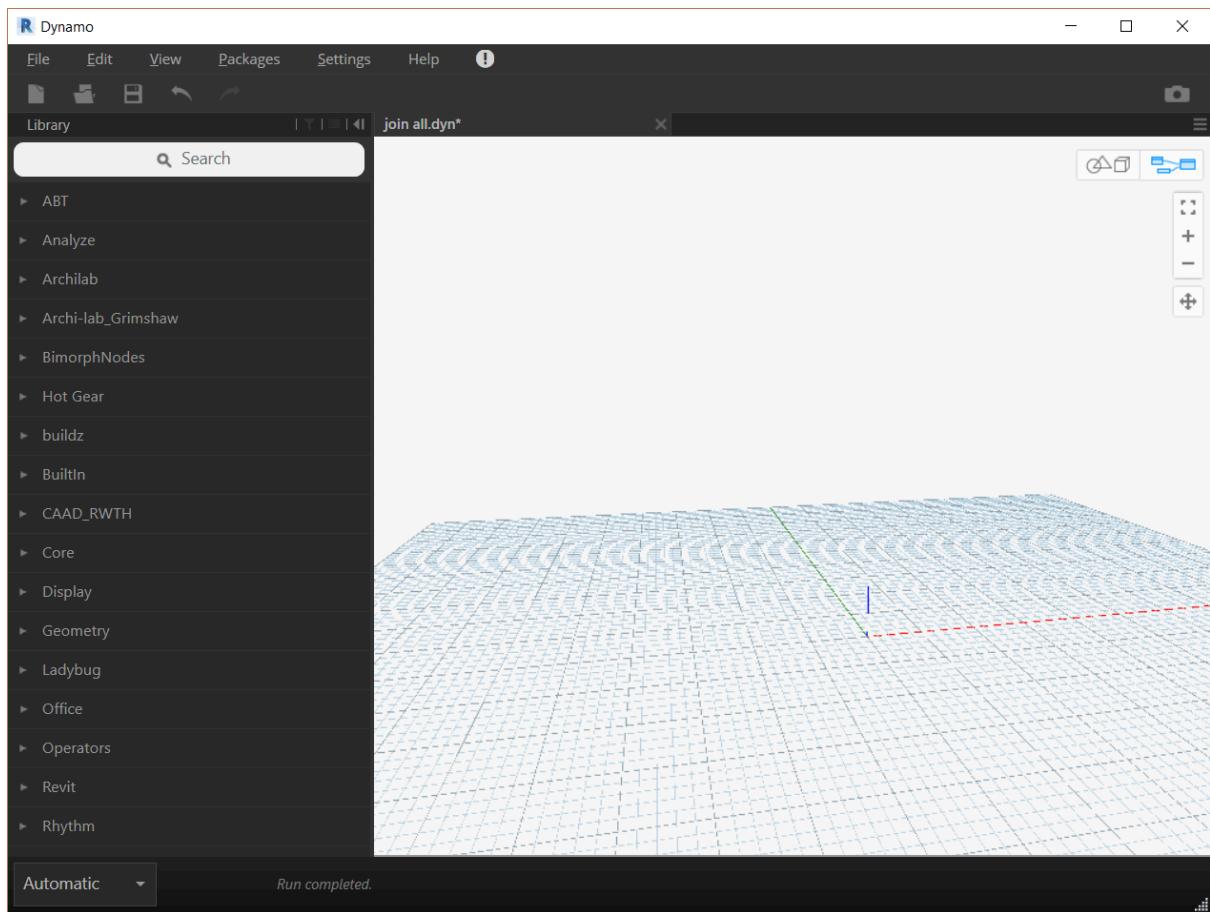


ملف جديد NEW •

فتح ملف OPEN •

آخر ملفات تم فتحها RECENT •

نختار ملف جديد لظهور لنا الملف التالي



القواعد العلوية

نجد بها أوامر كفتح و عمل ملف جديد File .1

أوامر مثل تراجع و نسخ و لصق و تحديد الكل Edit .2

تقريب و تبعيد View .3

حزم جاهزة يمكن تحميلها Packages .4

إعدادات البرنامج Settings .5

مساعدة Help .6

تحتل مساحة العمل البيضاء المساحة الكبرى من الواجهة و نضع فيها مكونات البرمجة

و نجد في أعلى مساحة العمل

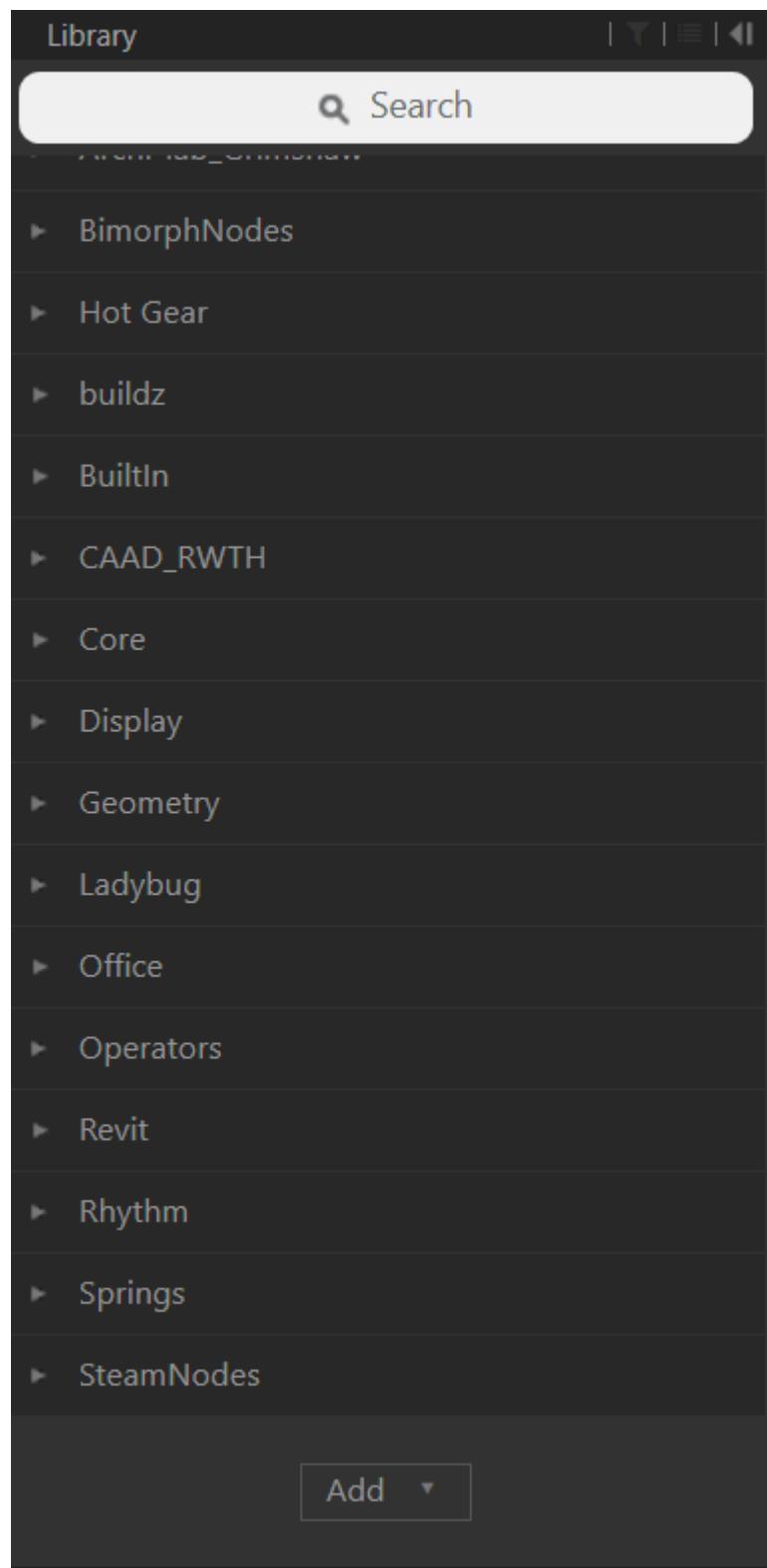


يكون التركيز على العناصر المكونة للبرمجة • **ENABLE GRAPH VIEW NAVIGATION**

التركيز على العناصر المجسمة • **ENABLE BACKGROUND 3D PREVIEW**

3D Preview	Graph Preview	Mouse Action
لا شيء	تحديد	الزر اليسار
اختيارات ال ZOOM	قائمة ال Context	الزر الأيمن
Pan	Pan	الضغط على البكرة
تقرير و تبعيد	تقرير و تبعيد	Scroll
لا شيء	إنشاء Code Block	الضغط على الزر اليسار مرتين

و في الجهة الشمال نجد مكتبة ال NODE التي تكون شفرة الكود



أهم المكتبات الأساسية

Analyze .1

Built-in Functions .2

Core .3

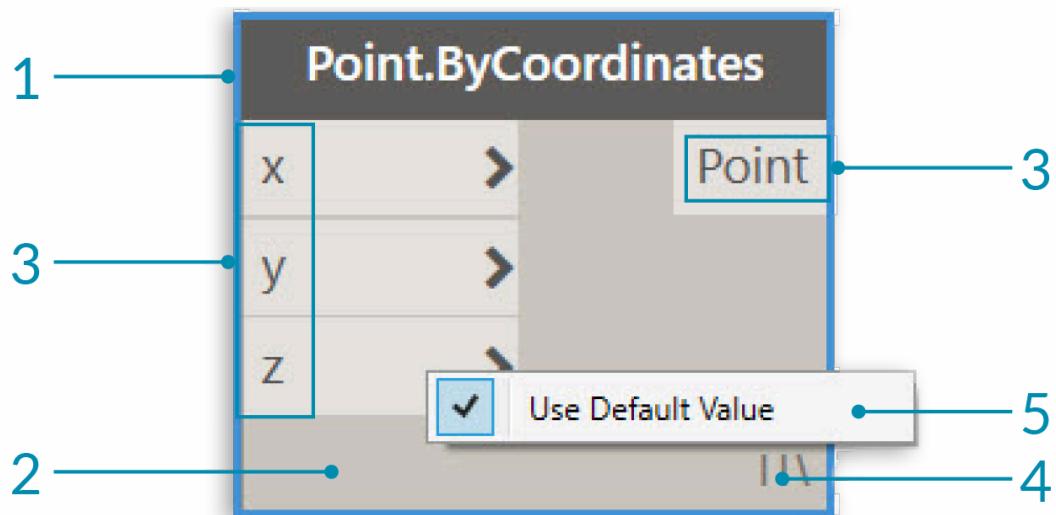
Geometry .4

Migration .5

Office .6

Operators .7

Nodes هي العناصر التي تضع وترتبط بينها ب Wires



.1 - اسم العقدة Name

.2 - جسم العقدة بالضغط على الزر اليمين تظهر لك قائمة خيارات Main

- يمكن من هنا توصيل العقدة بأي عقدة اخرى (Ports (In and Out .3
- تظهر لك النتائج Lacing Icon .4
- بعض العقد تحتوي قيم افتراضية كمدخلات يمكن الضغط بالزر الأيمن و تفعيل هذا الـ Default Value .5
- الاختيار

Wires يربط بين nodes

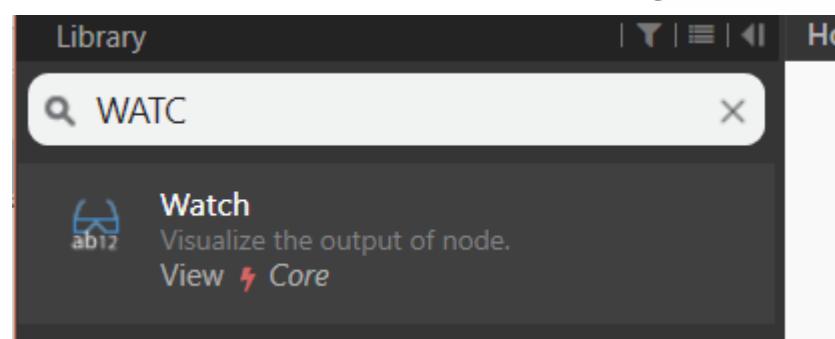
في أسفل البرنامج تجد اختيارين



AUTOMATIC تلقائي بمجرد كتابة أي جزء من الكود يتم تنفيذه  
MANUAL يدوى لا يعمل الا البرنامج الا بعد ضغط زر تشغيل

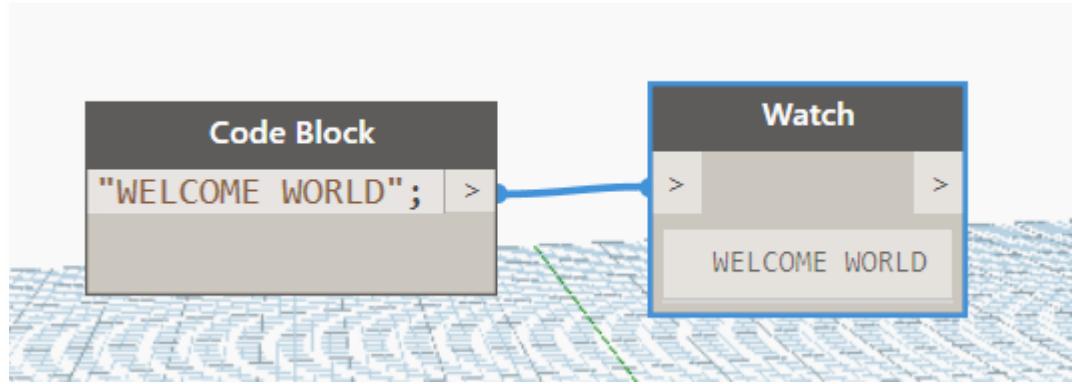
ونبدأ بعمل برنامج بسيط و هي كتابة HELLO WORLD و هي عادة في كل البرامج ان يكون اول برنامج كتابتها

نبدأ بعمل بحث عن



- CODE BLOCK كتابة نصوص او ارقام او كود برمجي
- WATCH لرؤية النتيجة

نضغط عليه مرتين لظهور في مساحة العمل ثم نقوم بالربط بينهم وذلك بالسحب من مخرج CODE BLOCK الى WATCH مدخل الـ WATCH



يمكن استبدال CODE BLOCK ب STRING لكتابة النصوص STRING



أمثلة لبعض العقد

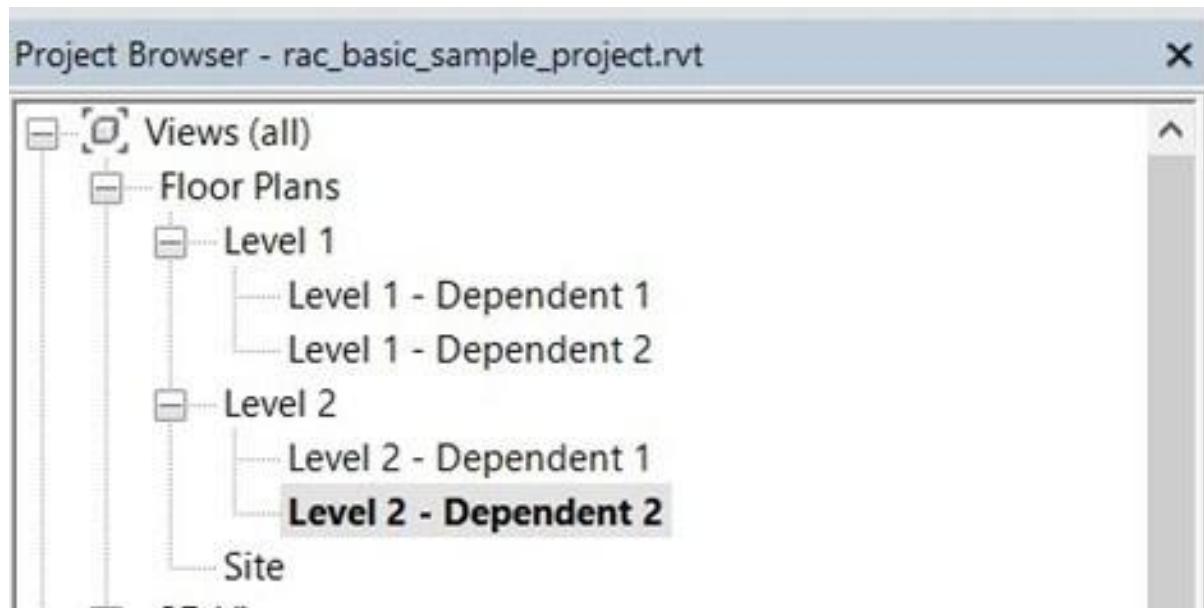


- .1 Boolean لا ام نعم
- .2 رقم رقابة كتابة
- .3 String نص كتابة
- .4 Slider Number متزلج رقم اصغر و اكبر تحددين
- .5 Integer Slider صحيحه الأرقام متزلج
- .6 Directory Path مجلد مسار تحديد
- .7 File Path الملف مسار تحديد

## تغيير اسماء العرض بواسطه الدينامو 29

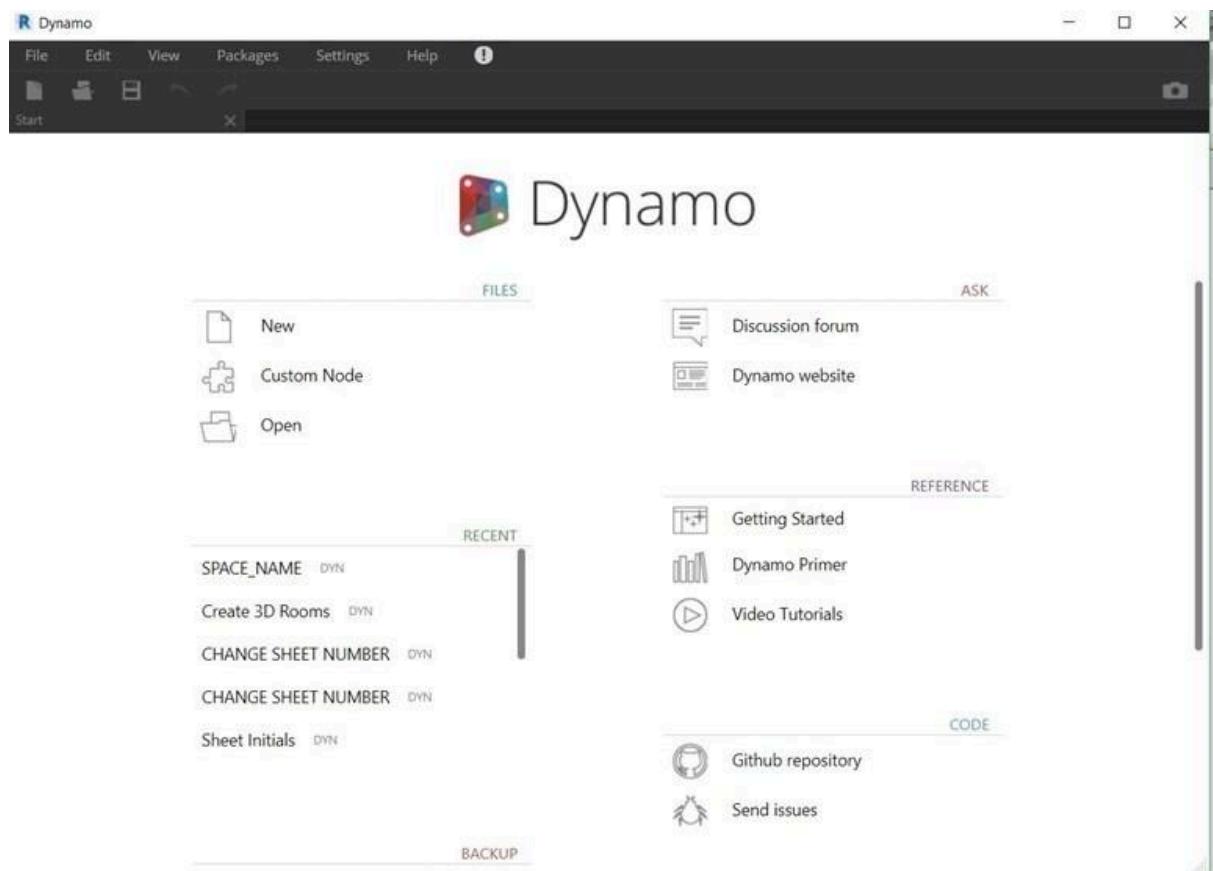
- الدينامو بيئة برمجية مرئية و سهلة يمكنك من داخلها برمجة الريفيت و نمذجة أشكال صعبة و تكرار الاوامر

مثلا هنا اريد تغيير اسماء ال views من Dependent الى Part



من اول الاصدار ريفيت 2017 اصبح الدينامو يوجد في

قبل هذا كان نثبته كملحق إضافي Addins



بعد فتح الدينامو نختار ملف جديد

يمكن أن نضغط بالزر الأيمن لتظهر لنا قائمة نختار منها ما نريد او من الشاشة التي على اليسار نبحث عنه

نبحث عن element type لنختار العناصر التي نرغب في التعامل معها

ثم all element of type لتحديد كل العناصر التي سنتعامل معها

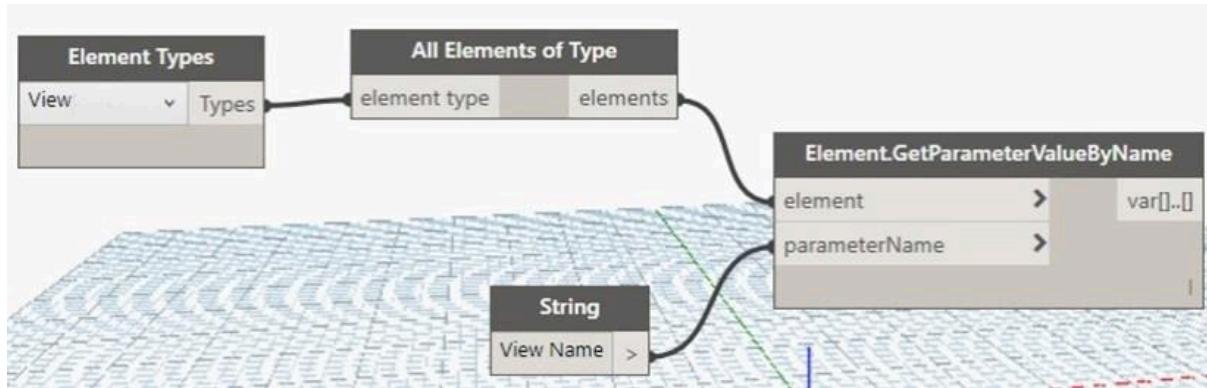
نقوم بالتوصيل بينها وذلك بالسحب من مخرج element type الى مدخل all element of type



لتحديد متغير معين نضيف Element.GetParameterValueByName

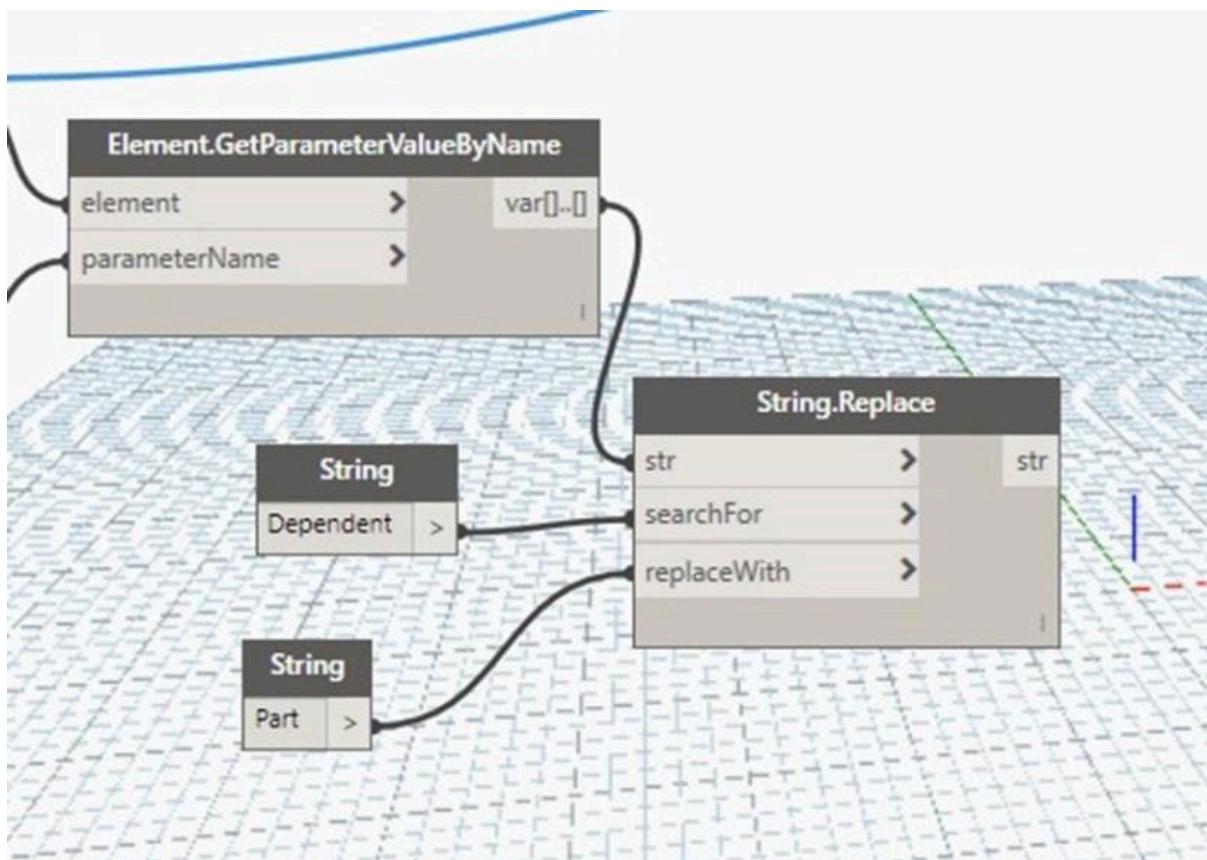
يحتاج الى تحديد العنصر و تحديد اسم البارمتر

نضيف string و نكتب فيه اسم لبارمتر



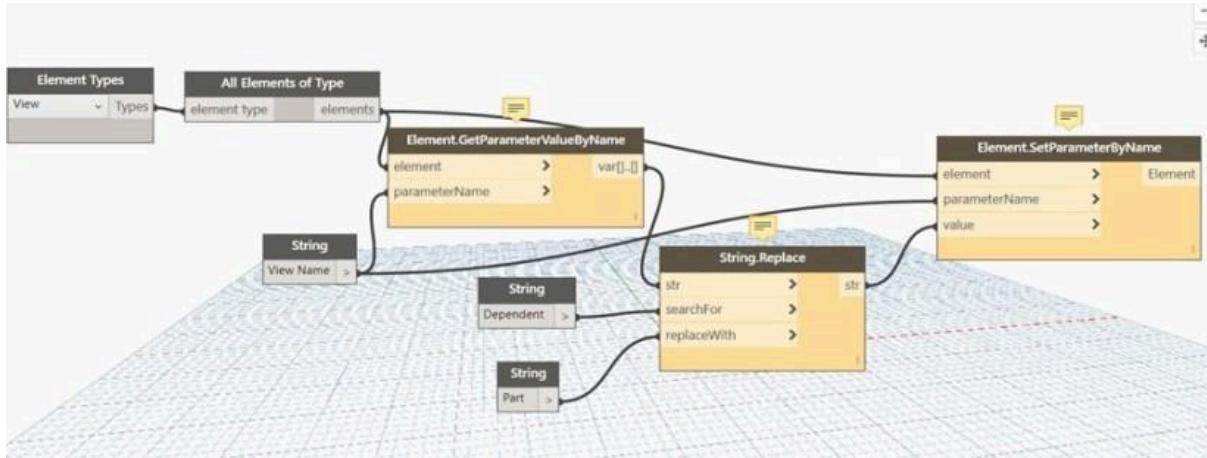
نضيف String.Replace لتبدل كلمة بأخرى

يحتاج كلمتين لنبحث عن كلمة و نبدلها بأخرى

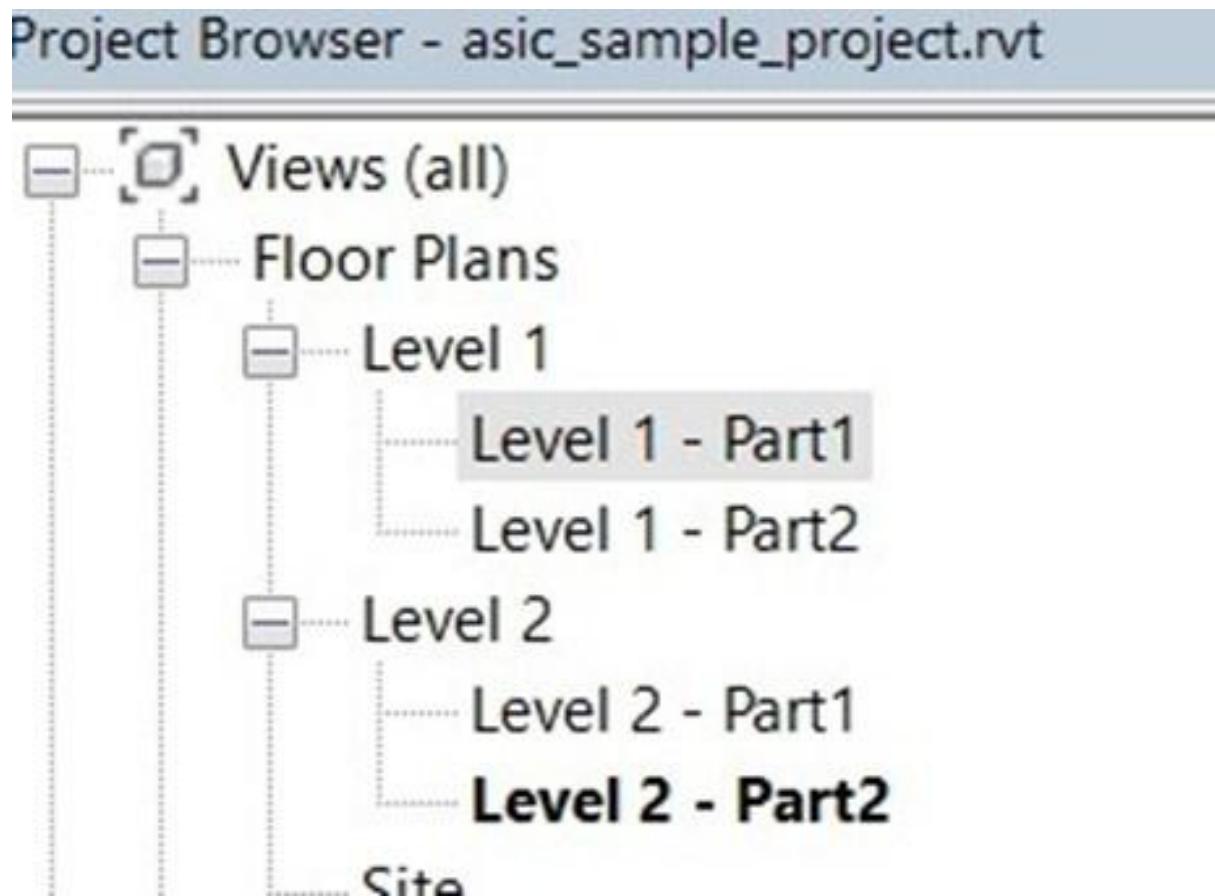


لتحديد قيمة البارمتر نضيف Element.SetParameterByName

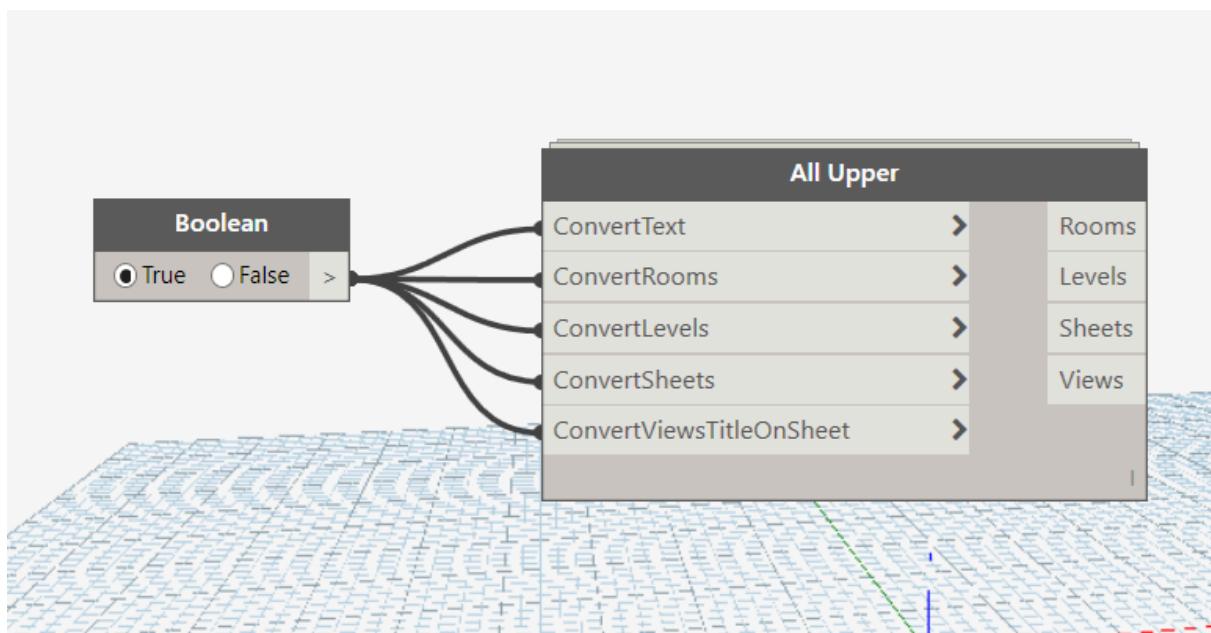
يحتاج لتحديد العناصر و تحديد اسم البرمتر و القيمة الجديدة



عند تشغيل البرنامج من run باسفل البرنامج يتغير الاسماء كما في الصورة



جعل كل الكتابات upper



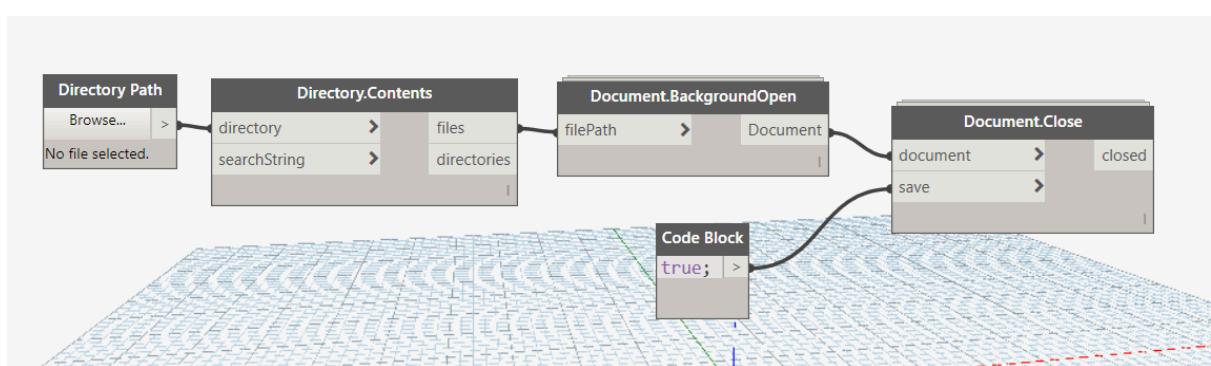
تحميل الملف

<https://drive.google.com/open?id=0Bw2zbRO83Gg4TFV5c3YxdjhkbFk>

مشاهدة الفيديو

[https://www.youtube.com/watch?v=LmDa7KZ2C88&list=PLNMim060\\_nULjpgFcq3xU9W7XGaq5X1cs&index=36](https://www.youtube.com/watch?v=LmDa7KZ2C88&list=PLNMim060_nULjpgFcq3xU9W7XGaq5X1cs&index=36)

تحديث كل الفاميلي و المشاريع للإصدار الحالي



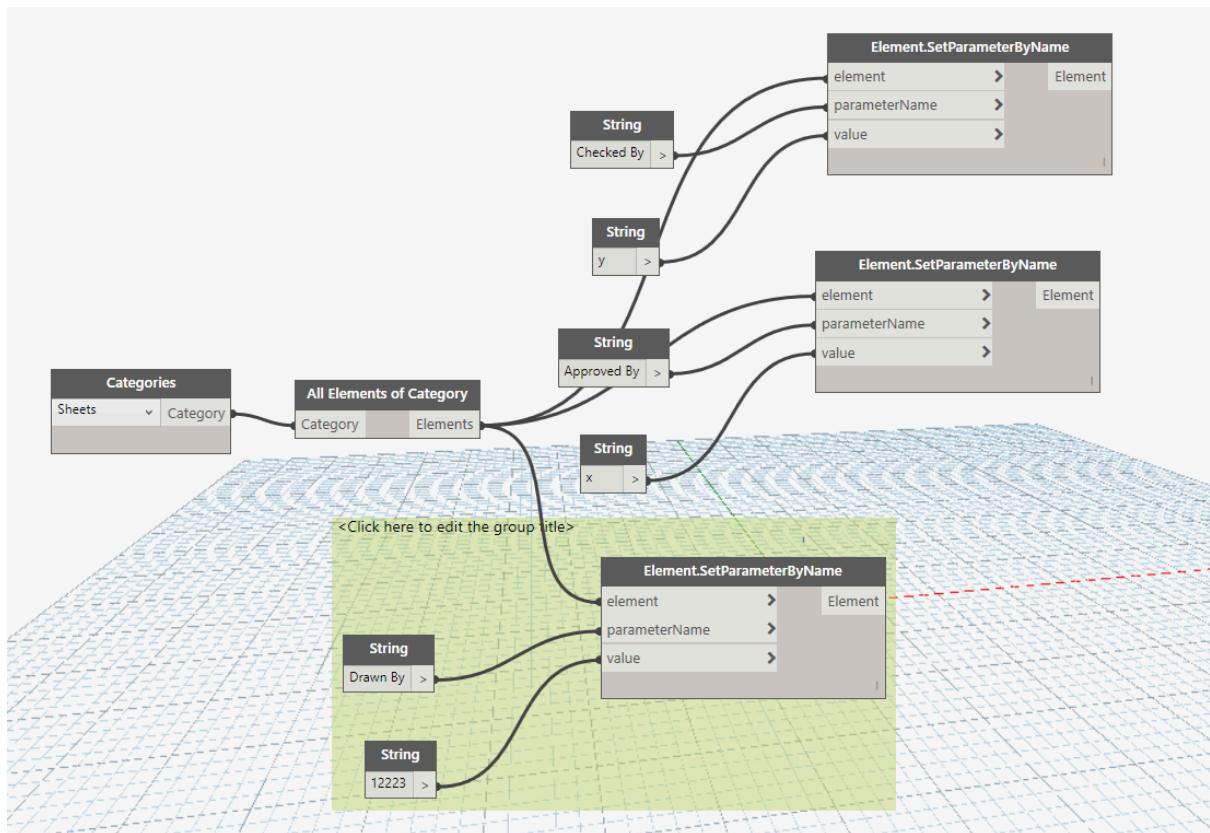
تحميل الملف

<https://drive.google.com/file/d/0B3oLhdhnXmutSIJncXZfbUxzYzA/view?usp=sharing>

مشاهدة الفيديو

[https://www.youtube.com/watch?v=xAl4-CmU0Ug&index=47&list=PLNMim060\\_nULjpgFcq3xU9W7XGaq5X1cs](https://www.youtube.com/watch?v=xAl4-CmU0Ug&index=47&list=PLNMim060_nULjpgFcq3xU9W7XGaq5X1cs)

-----  
تغير قيمة البارامتر



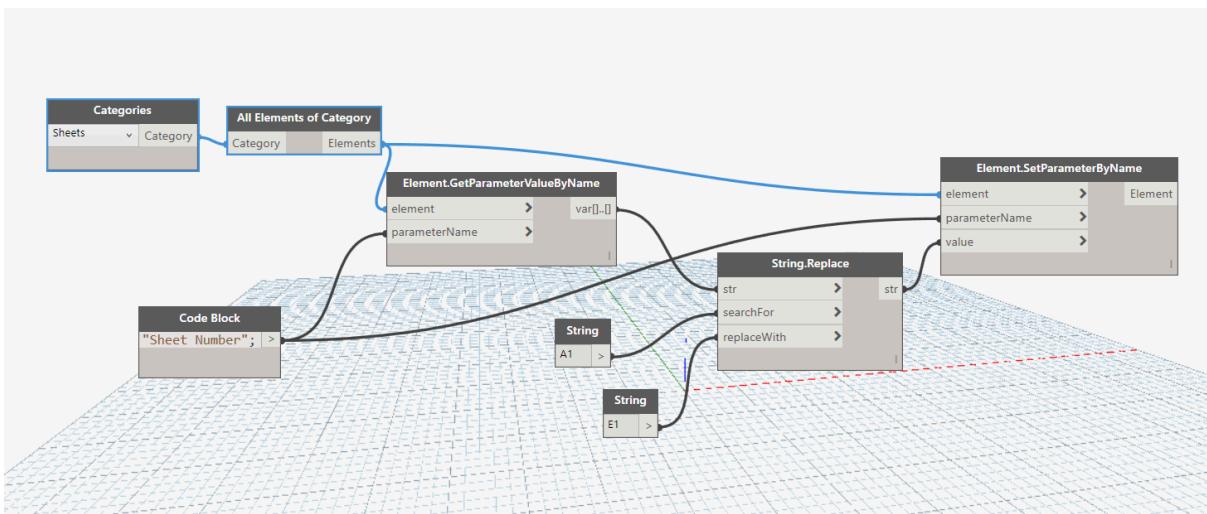
تحميل الملف

<https://drive.google.com/open?id=0Bw2zbRO83Gg4TI80WXMyZVJ2T28>

مشاهدة الفيديو

[https://www.youtube.com/watch?v=Kx6yy6pNYUU&list=PLNMim060\\_nULjpgFcq3xU9W7XGaq5X1cs&index=34](https://www.youtube.com/watch?v=Kx6yy6pNYUU&list=PLNMim060_nULjpgFcq3xU9W7XGaq5X1cs&index=34)

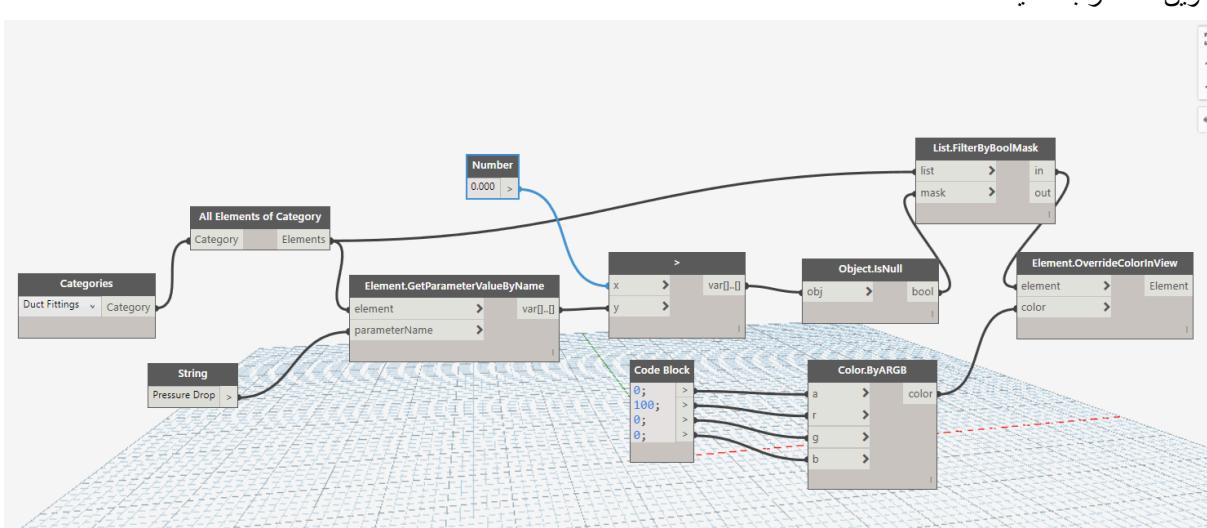
-----  
تغير رقم ال sheet



تحميل الملف

<https://drive.google.com/open?id=0Bw2zbRO83Gg4RERwUE84WDNtVFE>

مشاهدة الفيديو



تحميل الملف

<https://drive.google.com/open?id=0Bw2zbRO83Gg4Y1FGbzMxWml4VXM>

مشاهدة الفيديو



## Python



هي لغة عالية المستوى تنفذ أيضاً من جهة الخادم، تستخدم في تطوير الموقع الإلكتروني وتطبيقات الهاتف المحمول.

تعتبر لغة سهلة إلى حد كبير للمبتدئين، وذلك نظراً لقدرة قراءة تعليماتها بسهولة وكتابتها بإنجاز، أي أن المبرمج يستطيع أن يعبر عن فكرة أو تنفيذ فعل ما بتعليمات بأسطورة برمجية أقل من اللغات الأخرى.

هناك العديد من التطبيقات المصممة باستخدام هذه اللغة مثل: **Instagram** و **Rdio** و **Pintrest** من خلال الإطار المرتبط بشبكة الإنترنت. أيضاً مستخدمة من قبل: **NASA** و **Yahoo** و **Google**.

تم تطوير لغة البايثون سنة 1990 على يد المطور الهولندي **Guido Van Rossum**، كان تطوير اللغة أساساً للمساعدة في تطوير نظام تشغيل باسم **Amoeba**.

اكتسبت بايثون مع الوقت الكثير من المميزات، وصارت تعمل بكفاءة في عدة مجالات، كتطبيقات سطح المكتب ولغة لتوير موقع الويب كذلك يمكنها التعامل مع المكتبات البرمجية لعدة لغات كالجافا مثلاً.

مميزاتها:

– لغة بسيطة جداً من ناحية القواعد الكتابية ومع هذا لا تخلى عن القوة مثل الفيوجوال بيسك.

– تدعم المنحى الكاثني في البرمجة.

– تصميم الواجهات الرسومية من خلال عدة مكتبات برمجية متاحة ومجانية.

– كونها مفتوحة المصدر يجعلها تتطور بوتيرة كبيرة جداً.

- مجتمع اللغة يكبر يوماً بعد يوم ودعمها أكبر من أي وقت مضى وكذلك مصادر تعلمها وفيرة خصوصاً باللغة العربية.

- برامج البايثون تعمل على أي منصة لأنها لغة مفسرة وليس مترجمة.

عيوبها:

- أكثر ما يعيب البايثون هو بطيئها النسبي مقارنة بلغات أخرى.

- قوية لكن حالياً لا ترقى لمستوى لغات عريقة كالجافا.

[/https://www.python.org](https://www.python.org)

و في حالة رغبتك في برمجة الريفيت بلغة البايثون يمكنك هذا باستخدام RevitPythonShell

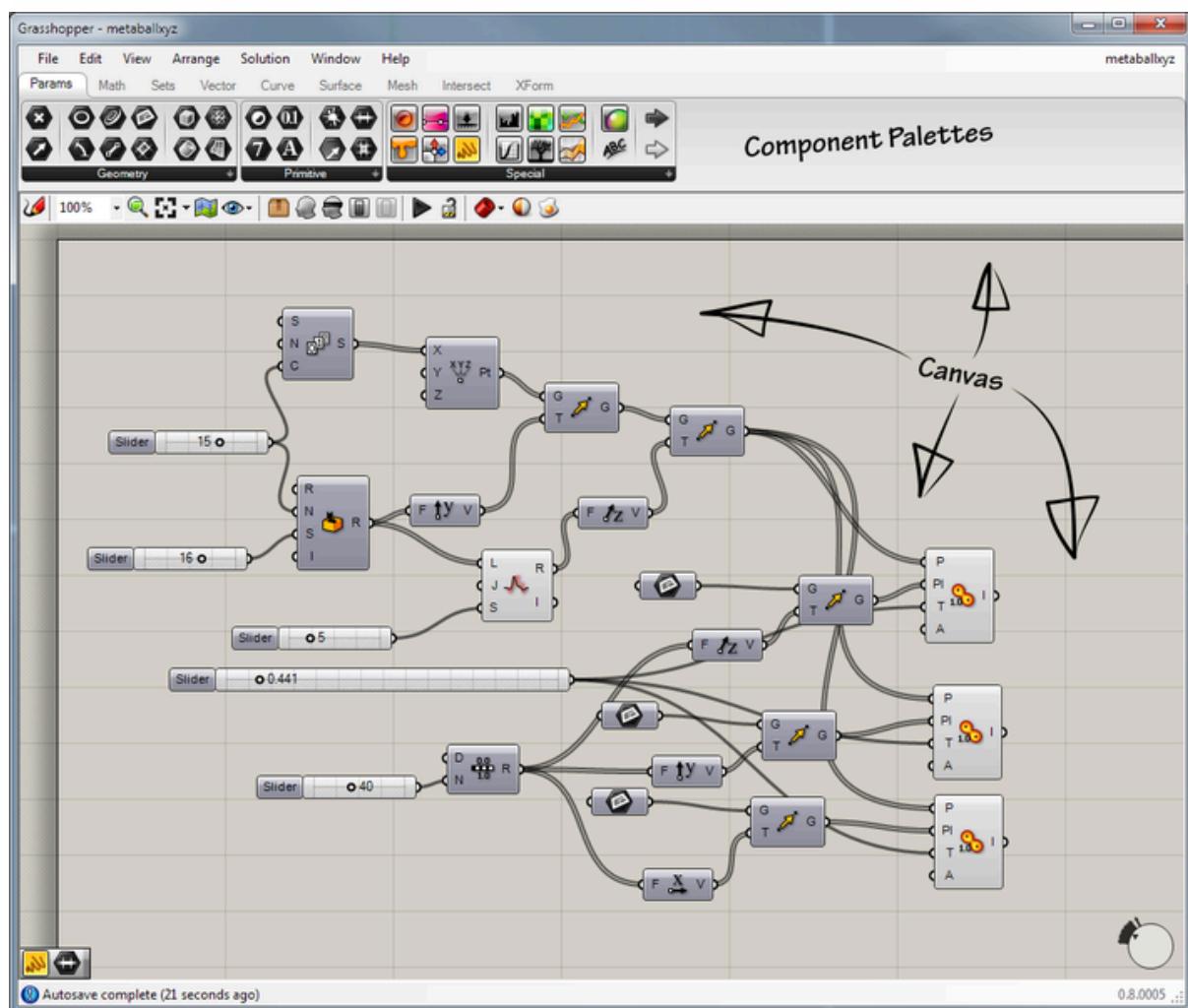
اداة مساعدة للبرمجة بلغة البايثون داخل Autodesk Revit and Vasari plugins . و كتابة Autodesk Revit and Vasari لهما ، و تتيح لك الوصول NET framework and the Revit API. الكامل الى

<https://github.com/architecture-building-systems/revitpythonshell>

[https://www.youtube.com/watch?v=N70cmZusNYg&list=PLNMim060\\_nUKT5xMU9yDahSuMjymHGmFo&pp=gAQBiAQB](https://www.youtube.com/watch?v=N70cmZusNYg&list=PLNMim060_nUKT5xMU9yDahSuMjymHGmFo&pp=gAQBiAQB)

## Grasshopper 3d

البرمجة عن طريق ادراج عناصر و الرابط بينهم فينتج لنا مجسمات في برنامج  
الذى تشبه بيئته الكاد **Rhinoceros 3d**

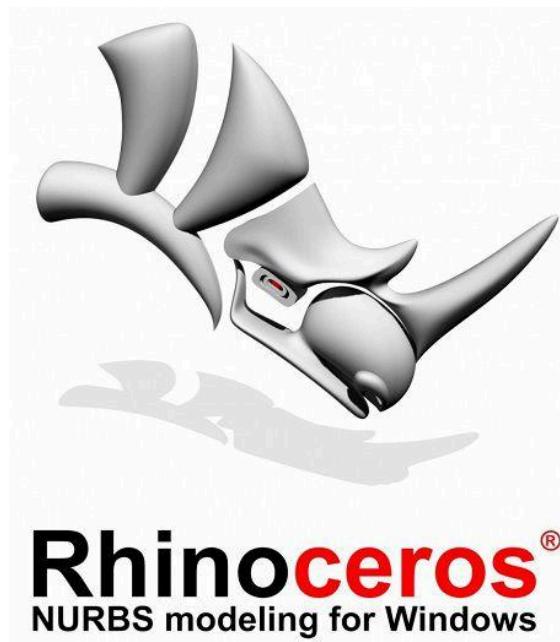


## Rhinoceros

برنامجه رسم ثلائي الأبعاد بيستخدم فى تصميم المنتجات المعقدة، و يتميز البرنامج بسهولة الرسم وخاصة فى رسم الاسطح surfaces

برنامجه لتصميم 3D والمنذجة. يستخدم البرنامج في تصميم CAD، والهندسة المعمارية، التصميم الجرافيكي. يحتوي على العديد من التطبيقات والأدوات الازمة لنموذج الكائنات مختلفة الحجم أو التعقيد. البرنامج يسمح لك للعمل مع كائنات NURBS، وإجراء تحليل للعمل وتحرير وتحويل المشاريع في أشكال مختلفة. تمكن أيضا تمديد ملامح البرنامج من خلال ربط العديد .plugins يسمح لك لتبادل البيانات مع مختلف برامج الهندسة والتصميم.

الأداة التي تسمح لك بالعمل مع أنواع مختلفة من العناصر الافتراضية : المجسمات والأسطح و منحنيات NURBS 'النورب' وكل هذه الأشياء من خلال نموذج رياضي يعطي نتائج واقعية جدا.



الخصائص الرئيسية:

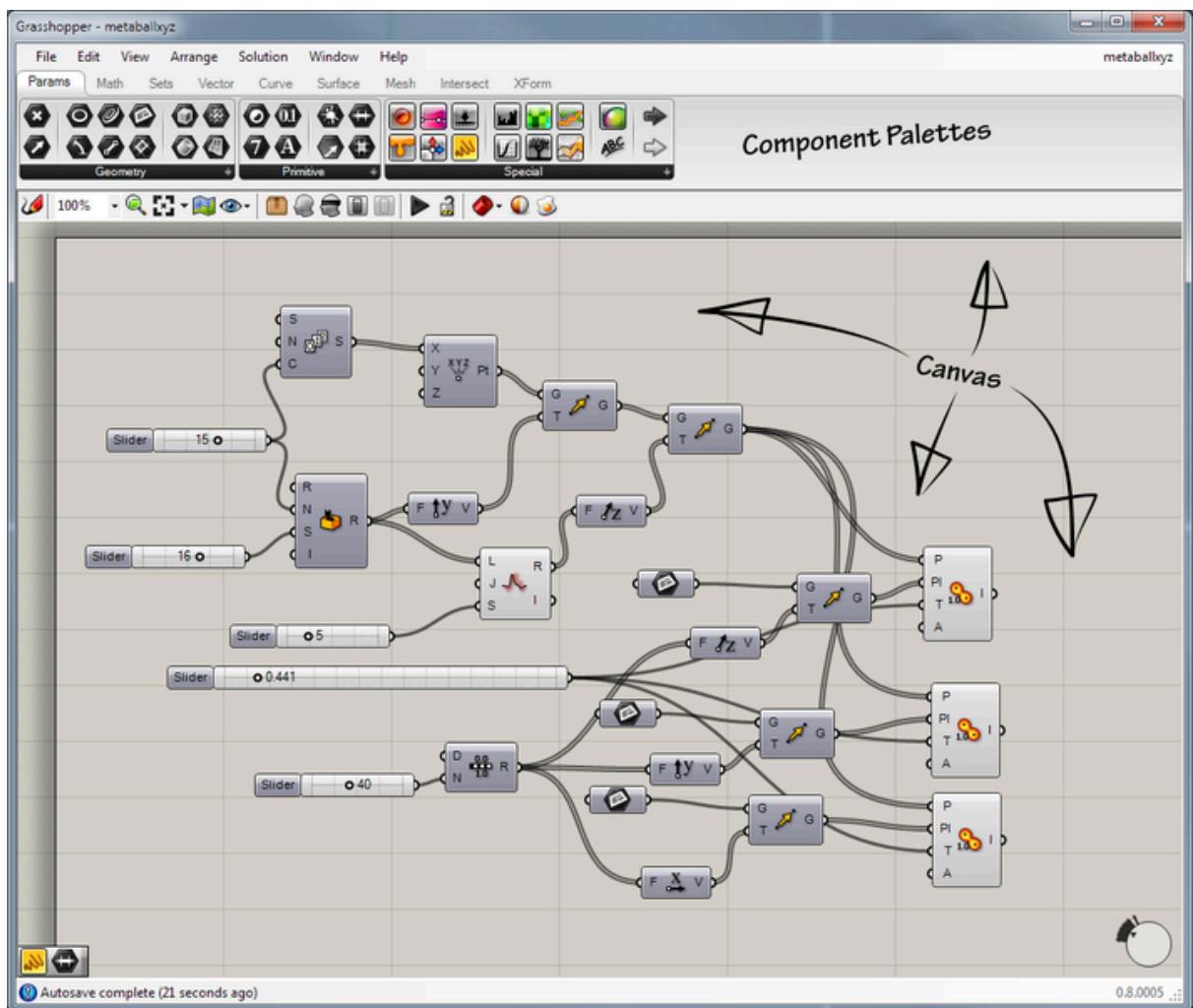
- تصميم المشروع من تعقيد عالية

- العديد من الأدوات والآثار لالنماذجة
- يدعم صيغ مختلفة
- يدعم اللغة الخاصة من الكتابات
- اتصال من مختلف plugins
- برنامج جراسهوبير (Grasshopper)

[www.grasshopper3d.com](http://www.grasshopper3d.com)

هو إضافة على برنامج راينو (Rhinoceros) ، مخصص للبرمجة modeling ويعمل خوارزميات وبرمجيات بصرية، بحيث لا يحتاج المستخدم إلى معرفة مسبقة بلغات برمجة الحاسوب، ويسهل للمصممين عمل تصاميم توليدية متنوعة.

يتم تصميم وعمل الخوارزميات باستخدام عناصر Components حيث يتم ربطها في مساحة الرسم الخاصة بالبرنامج بطرق مختلفة حتى تكون لنا أشكال متعددة.



شرح الجرسهوب بالعربي (Arabic) Grasshopper for Rhino (Arabic)

[https://www.youtube.com/playlist?list=PLNMim060\\_nUKsuVDh13xxMvX4Uuc6ReeC](https://www.youtube.com/playlist?list=PLNMim060_nUKsuVDh13xxMvX4Uuc6ReeC)

[www.grasshopper3d.com](http://www.grasshopper3d.com)

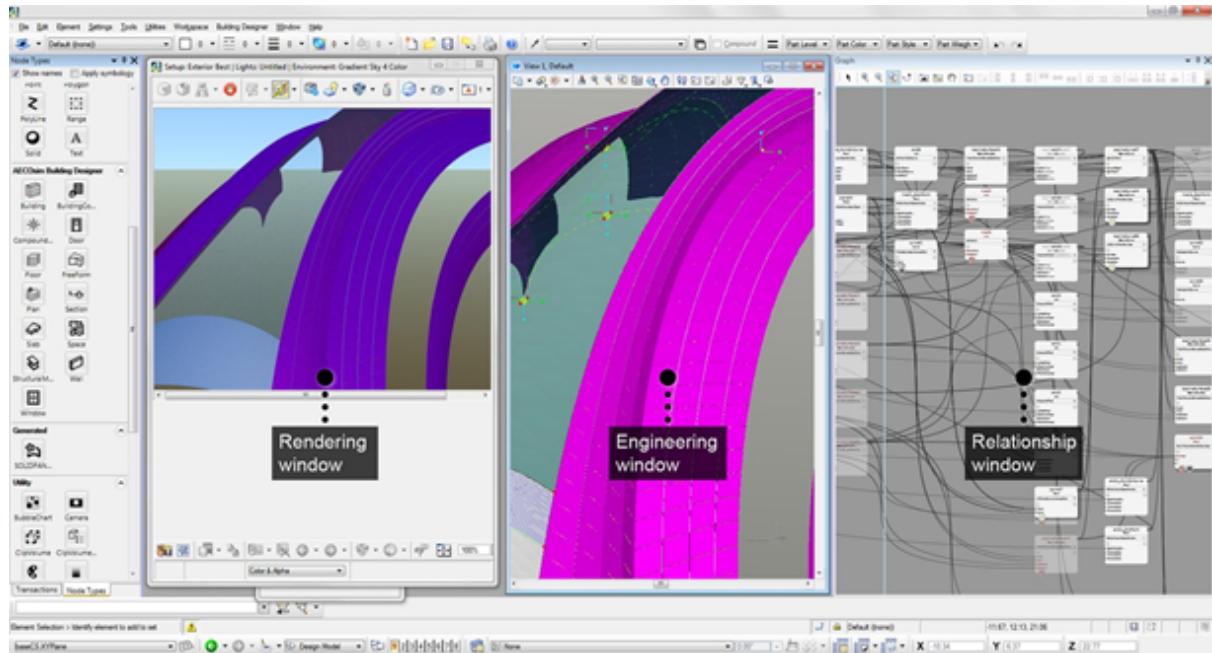


## GenerativeComponents

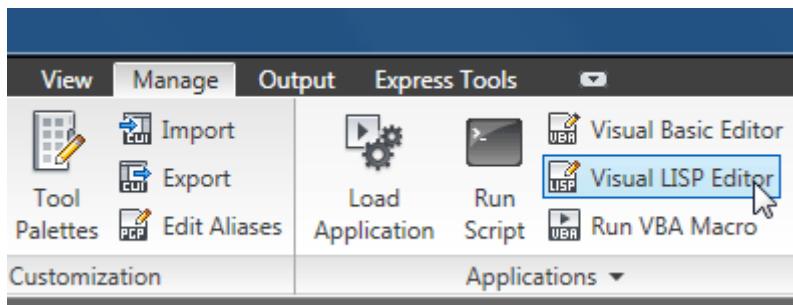
استكشاف بسرعة مجموعة واسعة من البدائل "ماذا لو" حتى لأكثر المباني تعقيدا مع GenerativeComponents

يمكنك استكشاف المزيد من الاحتمالات في وقت أقل وكفاءة أكثر لإنشاء وإدارة عملية البناء . من خلال أتمتة عملية التصميم، يمكنك تسريع لتكرار التصميم الخاص بك. متابعة تصاميم أكثر حرية في الشكل، حتى تتمكن من تصميم واستكشاف أشكال لا يمكن تصورها.

يعمل مع برامج bentley



## AUTOLISP



هي احدي لغات البرمجة الخاصة بشركة Autodesk ، مثلها مثل ObjectARX ، VBA و هي تعطي قوة للاوتوكاد ، و تمكنك من الرسم بسرعة ، و اختصار الخطوات و توفير الوقت .

كل ما يمكنك فعله في الاوتوكاد ، يمكنك بكل سهولة أن تفعله بالاتوليسب مزيداً لانتاجك .  
Lisp is a programmable programming language.

و هي مشتقة من لغة lisp و هي احد لغات الذكاء الصناعي و التي هي من انتاج جون ماكرثي في الخمسينيات ، و كان اول ظهور لل AUTOLISP في النسخة 2.18 في شهر يناير 1986 .  
و هي اختصار

LISP Processing.

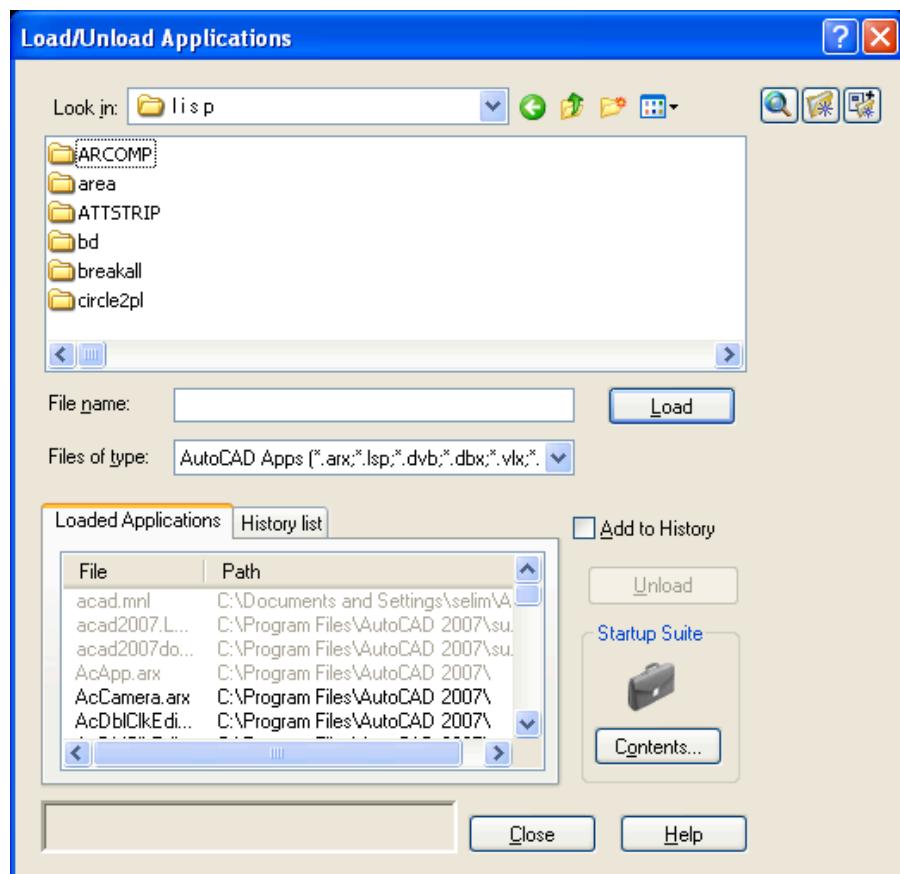
Or LISP Programming  
"Lots of Stupid Parentheses".  
و يقال انها اختصار

و هي لغة interpreted. بمعنى انها عكس الفيجوال بيسك لا يتم compiled لها .

لتحميل الليسب

من قائمة TOOLS أختـر

LOAD APPLICATION



اختر الليسب ثم LOAD  
او سحب الملف من Windows Explorer الى مساحة العمل بالاتوكاد

و لجعل الليسب يحمل تلقائيا مع الاتوكاد في كل مره يعمل : أختر

CONTENTS



او اكتب aupload في سطر الاوامر

Command: appload

ولكتابه اوامر تعمل عند بداية التشغيل نضعها في ملف acad.lsp او acaddoc.lsp و هو ملف يبحث عنه الاتوكاد عند بداية التشغيل في مسارات support

### قواعد لغة autolisp

- (1) لا تفرق لغة الاتوليسب بين الحروف الكبيرة والصغيرة .
- (2) تكتب لغة الاتوليسب في اقواس و لا بد ان يكون عدد الاقواس اليمنى مساوى لعدد الاقواس اليسرى و يتم التعامل مع الاقواس الداخلية اولا .

Command: (\* (+ 3 4) (+ 2 5))

49

في حالة ترك قوس مفتوح فأن البرنامج يخبرك ان هناك اقواس ناقصة

Command: (\* (+ 3 4) (+ 2 5))

( >

(3) يتم كتابة الرمز او الدالة اولا

Command: (+ 4 2)

6

(4) المتغير يمكن ان يكون اي حرف او رقم ما عدا

- () ' ; " \

و يجب ان يبدا المتغير بحرف و ليس رقم و لا يحتوي على مسافة

أمثلة على اسماء المتغيرات

AMRSELIM

WHOAMI

ENGWEB13

تعريف المتغير هو مساحة في الذاكرة ممحوza لقيمة متغيرة

مثلا س=ع 2+

فعندما تكون ع = 3 اذن س = 5 و عندما تكون ع = 10 تكون قيمة س = 12

(5) يجب ان تكون هناك مسافة بين الدالة و المتغير

Command: (+ 4 2)

6

عند اهمال المسافة بين الدالة و المتغير تظهر لا رساله خطأ BAD FUNCTION

Command: (+4 2)

4

6) عند كتابة تعليق عام نكتب قبله علامة ....

## وصف لل kod و مقدمة لمقطع في الليسب توضيح لجزء صغير

Command: (+ 4 2) ;comment

6

و الامتدادات لهذا الملف قد تكون:

**Lsp** المف العادي و يمكن تحريره بالتفكير العادي بالويندوز  
**mnl.\***

و اذا كان لدينا ثلاثة ملفات بنفس الاسم بالامتدادات السابقة فان الاتوكاد يحمل vlx ثم fas ثم lisp ملف ليسب محول اللغة الاله

و لتشفيه ملف الليسب (و ان كنت لا انصح بهذا . لجعل المنفعة عامة )

- من قائمة tools إفتح visual lisp editor

- في شاشة الـ console اكتب الأمر التالي:

(vlisp-compile '1sa "d:/try.lsp" "d:/try.fas")  
حيث المسار الأول مسار الملف المرغوب تشفيره والثاني مسار الملف المشفر

و يبعدا عن الترجمة توحد طريقة سهلة لعمل ليس

صمم لبس الاتو كاد بنفسك

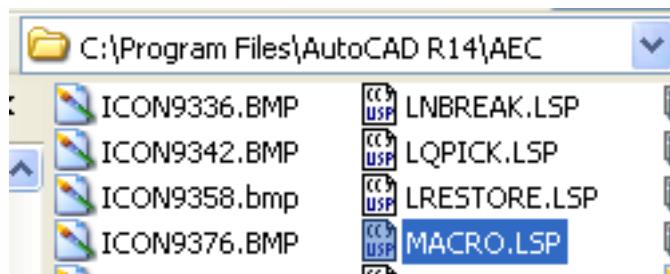
تسجيل ما تفعلة على الأتوCAD و تسجيلة في مایکر و لیس

مثلاً أرسم مستطيل اعمل له دوران و غير الليز

يتم تسحيلة و تصديره للسوق

الد نامح بـ محة George Omura

<http://www.nvacad.com/MacrolISP.htm>



### MacroLISP

:Download page

<http://www.omura.com/autocad/aecdownloads.htm>

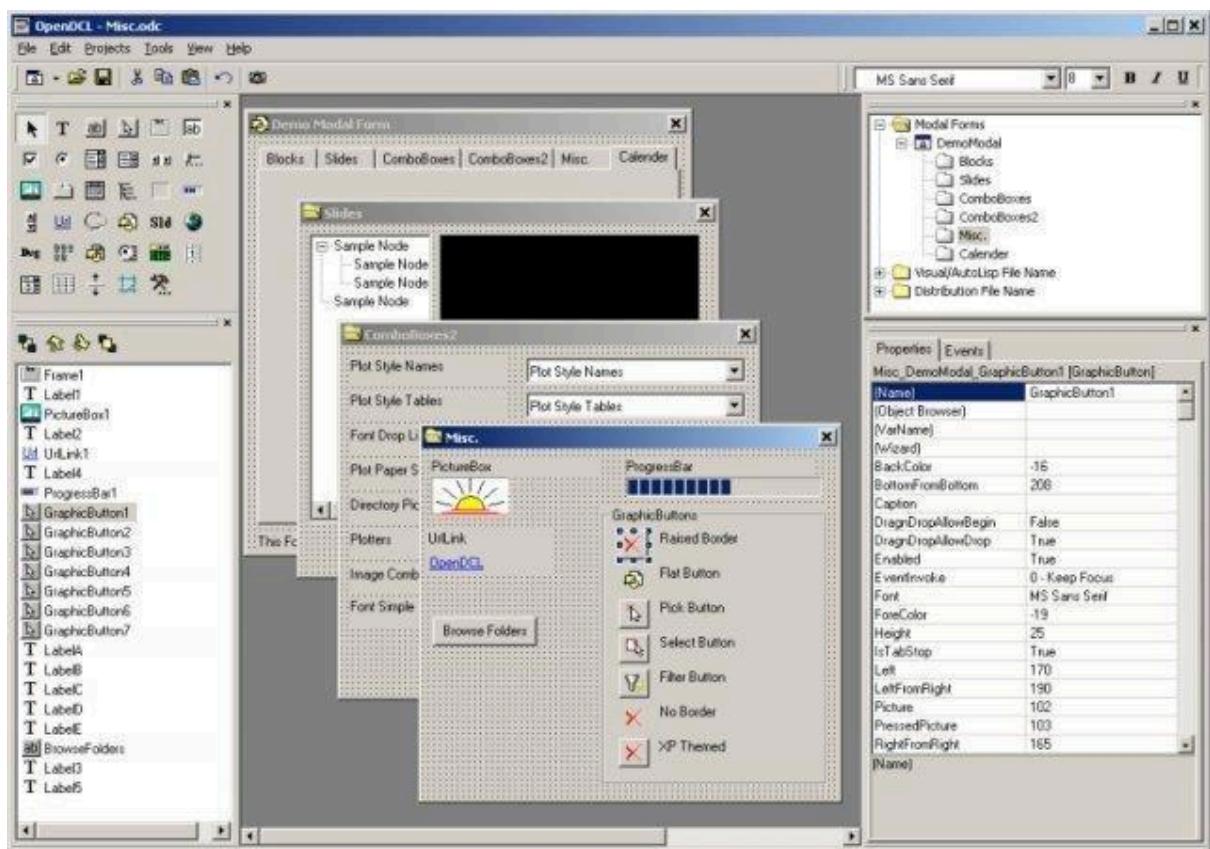
:Download and install file

<http://www.omura.com/autocad/aec43.EXE>

و هناك طريقة تسهل العمل : **OpenDCL** هو للمبرمجين أو توسيب الذين يريدون استبدال لغة البرمجة بالأوتوكالد محددة

**DCL** لغة واجهة المستخدم مع مجموعة غنية من عناصر واجهة المستخدم بالويندوز الحديثة.

<http://sourceforge.net/projects/opendcl/?source=directory>





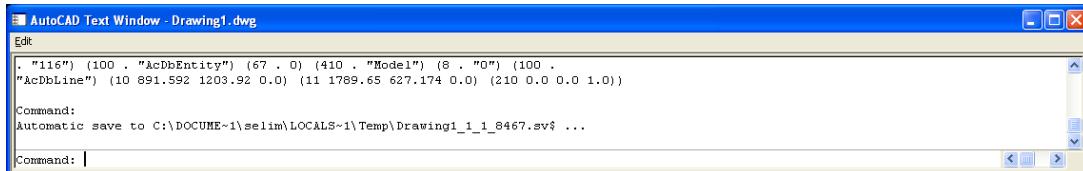


# اسیم و البرمجة

A word cloud visualization where the size of each programming language name corresponds to its popularity or prevalence. The most prominent word is "Java", which is rendered in a large, bold, dark green font. Surrounding "Java" are numerous other programming language names, each with a unique color. To the left of "Java", the word "JavaScript" is also quite large. Other notable words include "Python", "C++", "Haskell", "Ruby", "PHP", "Pascal", "Swift", "Mathematica", "Lisp", "Groovy", "Tcl", "SAS", "cT", "Max/MSP", "Dart", "Scala", "Haskell", "Cobol", "Erlang", "C#", "D", "Scal", "F#", "ML", "Assembly", "Scratch", "OpenEdge ABL", "ABAP", "Visual Basic", "Transact-SQL", "Clarion", "FoxPro", "Perl", "Logo", "Prolog", "PostScript", "Scheme", "R", "C++", "Haskell", "Go", "Fortran", "ActionScript", "Forth", "PL/SQL", "MATLAB", "C", "Lua", "TCL", "Delphi", "PL/I", and "Objective-C". The colors range from dark brown to bright yellow-green, creating a dense and colorful composition.

عمر سلیم

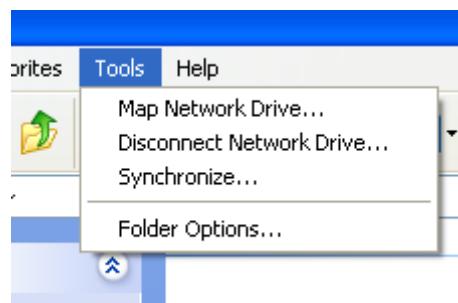
1) يمكن كتابة الاوامر مباشرة في سطر الاوامر الخاص بالاتوكاد command line و يفهم الاتوكاد ان اي امر يبدأ بالاقواس () هو يتبع لغة الالوليسب



2) او كتابة الاوامر في ملف txt مع تغيير الامتداد من TXT الى lisp و هو المعتمد ليسهل تبادلها و التعديل بها .

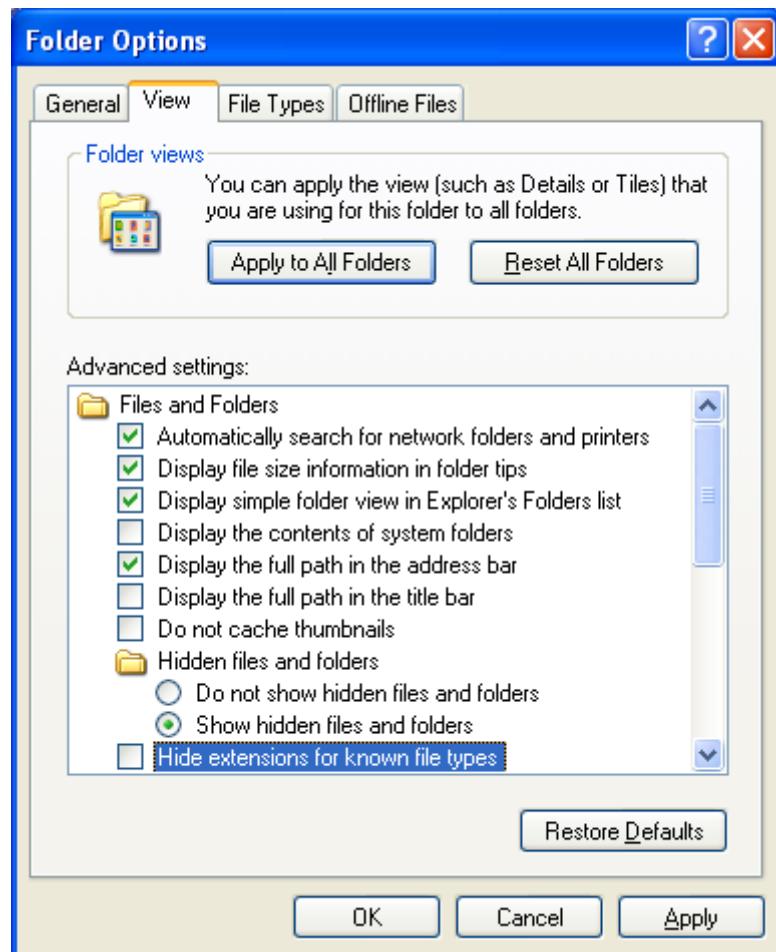


و اذا كان الامتداد غير ظاهر نذهب الى قائمة tools ثم الى folder option



ثم نختار view

ثم نقوم بازالة علامة "صح" من امام السطر  
Hide extensions for known file types



#### و الصيغة العامة

( اسما الدالة(نكتب هنا المتغيرات الخاصة بالدالة او يمكن تركها فارغه: c: )

نكتب هنا الدوال و الاوامر

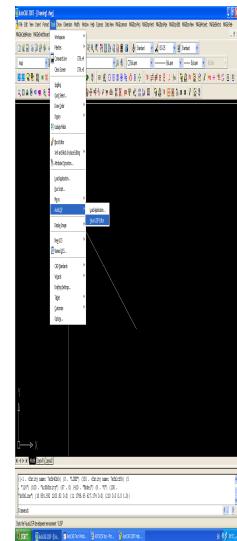
)

و يفضل ان يكون اسم الدالة هو نفس اسم الملف لتسهيل الامر و ليس شرطا .

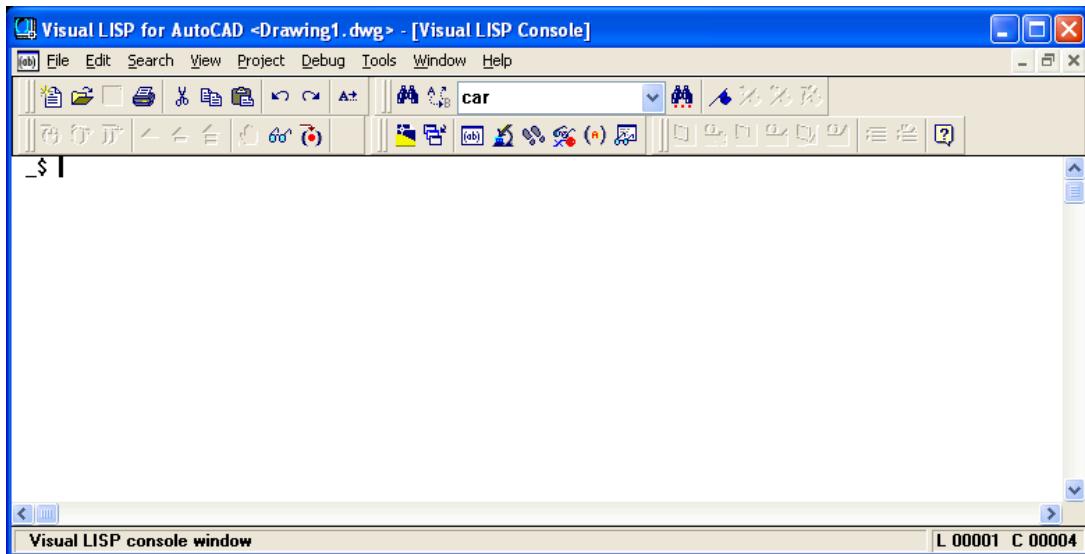
مثال بسيط نقوم فيه برسم مستطيل ( سيتم شرح المثال لاحقا )

```
(defun c:REC1()
  (setq a (getpoint))
  (setq b (getpoint))
  (command "rectangle" a b)
)
```

3) الدخول الى بيئه برمجة visual lisp editor  
و تعطي قوه للتعديل و معالجه الاخطاء  
من قائمه Tools اختر Autolisp Editor ثم



## نظهر أمامك شاشة VISUAL LISPFOR AUTOCAD



### الدوال الحسابية:

الامثلة التالية لشرح لغه الاتوليسب و يمكن كتابتها في سطر الاوامر مباشرة .

#### • الجمع +

المثال التالي نجمع فيه الرقم الاول على الرقم الثاني .  
مع العلم اننا (عكس اكثرا لغات البرمجة) نكتب علامة الجمع او لا

```
Command: (+ 1 2)
3
```

#### • الطرح -

```
Command: (- 4 1)
3
```

Command: (- 4 4)  
0

طرح رقم سالب من رقم موجب

Command: (- 4 -4)  
8

عند طرح ثلاثة ارقام يتم طرح الرقم الثاني من الاول ثم طرح الثالث من الناتج

Command: (- 9 4 2)  
3

- الضرب \*  
في هذا المثال نضرب الرقم الاول في الرقم الثاني .

Command: (\* 4 10)  
40

في المثال التالي نضرب ثلاثة ارقام في بعض .

Command: (\* 3 7 100)  
2100

ضرب رقم موجب في رقم سالب

Command: (\* 2 -3)  
-6

- القسمة /  
قسمة رقم على اخر

Command: (/ 21 3)  
7

و عند قسمة ثلاثة ارقام يتم قسمة الاول على الثاني , ثم يتم قسمه ناتج القسمه على الرقم الثالث

Command: (/ 8 4 2)  
1

مثال اخر

Command: (/ 22 3)  
7

في المثال السابق نجد ان لغة الاتوليس تجاهلت الكسور حيث ان القسمة كانت بين رقمين حقيقيين integer و عند القسمه على كسر فإنه يتعامل على انه double و يظهر لنا النتيجة بالكسور

Command: (/ 21.2 3)  
7.06667

و عند قسمه رقم صغير حقيقي على اخر حقيقي اكبر منه فأن الناتج يكون كسر و البرنامج سيتجاهل الكسور و تكون النتيجة صفر

Command: (/ 2 4)  
0

بينما عند قسمه كسر يتعامل البرنامج على مستوى الكسور

Command: (/ 2.1 4)  
0.525

و يجب الانتباه الى اهمية كتابه الصفر اذا كان العدد اقل من الواحد , فعند عدم كتابته تظهر لنا رسالة الخطأ التالية

Command: (\* .2 5)  
; Error: misplaced dot on input

و اهمية ترك مسافة بين الرقم والدالة و الا فسفرها على انها كلمه واحد و لن يفهمها

Command: (+2 5)  
; Error: no function definition: \*2

المثال التالي يشرح اهمية الاقواس و ترتيب العمليات . حيث ان البرنامج يقوم بحل ما داخل الاقواس اولا (و هو ما يسمى التعشيش) فيجمع 5 مع 6 ثم يضربها في 3 ليخرج لنا 11

Command: (\* 3(+ 5 6))  
33

- دالة اضافة واحد  
و هي دالة مستقله تضيف رقم واحد الى العدد

Command: (1+ 6)  
7

و عند كتابه رقم آخر غير الواحد تظهر لي رسالة تقول انه لا توجد دالة معرفة بهذا الاسم +2

Command: (2+ 6)  
; Error: no function definition: 2+

- دالة طرح واحد

Command: (1- 6)  
5

• دالة gcd  
تقوم هذه الدالة باقتراح انساب رقم يقبل عليه الرقمين القسمة عليه

Command: (gcd 100 15)  
5

• الدالة zerop  
لتتأكد هل الناتج صفر ام لا .  
في حالة الصفر يكون الناتج t  
و في حالة غير هذا يكون الناتج nil

Command: (zerop 0)  
T

Command: (zerop 9)

Nil

- الدالة abs و تعطي القيمة المطلقة للرقم (نقلب السالب لموجب)

Command: (abs -4)

4

Command: (abs 4)

4

- الدالة exp و تعطي القيمة الاسية

Command: (exp 1)

2.71828

Command: (exp 4)

54.5982

و عند اعطاء البرنامج عدد من المتغيرات يخالف المتوقع تظهر لنا رسالة الخطأ التالية

Command: (exp 5 2)

; Error: too many arguments

- الدالة EXPT و تعطي حاصل ضرب الرقم الاول في نفسه عدد من المرات تساوي الرقم الثاني

Command: (expt 2 3)

8

- الدالة ~ و تعطينا المتمم للعدد

.The bitwise NOT (1's complement) of the argument

Command: (~ 4)

-5

Command: (~ -4)

3

- الدالة MAX و هي تعطينا الرقم الاعلى

Command: (MAX 950 314)

950

- الدالة MIN و هي تعطينا الرقم الاصغر

Command: (MIN 3000 1000)

1000

• الدالة REM  
ايجاد REMINDER(المذكر) للرقم  
المتبقي من القسمة

Command: (REM 3 4)

3

Command: (REM 4 4)

0

المتبقي من قسمة 6 على 4

Command: (REM 6 4)

2

Command: (rem 42 16)

10

• الدالة FIX  
تقريب لاقرب عدد صحيح و الغاء الكسور

Command: (FIX 3.2)

3

Command: (FIX 3.7)

3

• الدالة FLOAT

تحول العدد الى كسر

Command: (FLOAT 3.2)

3.2

Command: (FLOAT 3)

3.0

و ربما يكون السؤال : ما هي الفائدة ؟

هل تذكر عندما نقسم عدد حقيقي على اخر حقيقي و كان البرنامج يتتجاهل الكسور ؟؟

Command: (/ 22 3)

7

تأمل المثال التالي عند استخدام الدالة FLOAT

Command: (/ (FLOAT 22) 3)

7.33333

• الدالة LOG  
ايجاد اللوغاريتم

Command: (LOG 1)  
0.0

Command: (LOG 10)  
2.30259

Command: (LOG 2)  
0.693147

• الدالة SQRT  
و تعطينا الجذر التربيعي

Command: (SQRT 144)  
12.0

و عند اعطائه رقم سالب , لا يتعرف عليه , و يعطينا رسالة خطأ.

Command: (sqrt -4)  
; error: function undefined for argument: -4

• دالة sin  
و تعطينا جا الزاوية  
و يتم ادخال الزاوية القطرية و ليست الدائرية

Command: (sin 165)  
0.997797

• دالة cos  
و تعطينا جتا الزاوية  
و يتم ادخال الزاوية القطرية و ليست الدائرية

Command: (cos 165)  
-0.0663369

• دالة atan  
و تعطينا ظل الزاوية  
و في حالة اعطائها زاويتين فأنها تعطي ظل (حاصل قسمة الاولى على الثانية)

Command: (atan 1)  
0.785398

Command: (atan 1 2)  
0.463648

• دالة angle  
و تستخدم لايجاد الزاوية

Command: (angle '(1.0 1.0) '(1.0 4.0))  
1.5708

Command: (angle '(5.0 1.33) '(2.4 1.33))  
3.14159

او يمكن كتابتها بطريقة اخرى بالاستعانة بـ دالة getpoint

Command: (setq a (getpoint))

```
Command: (setq b (getpoint))
Command: (angle c b)
0.724992
```

• دالة minusp  
هل الرقم المعطى سالب ام لا  
اذا كان سالب تعطي t  
اذا كان موجب تعطي nil

```
Command: (minusp -1)
T
```

```
Command: (minusp 13)
nil
```

• دالة inters  
و تعطي نقطة تلاقي اربع نقط

```
Command: (inters a b c d)
(-443.205 399.475 0.0)
```

دوال المقارنة  
و هي تعطينا t اذا كانت النتيجة صحيحة  
و nil اذا كانت النتيجة خاطئة

• يساوي =

```
Command: (= 3 3.0)
T
```

```
Command: (= 314 950)
nil
```

```
Command: (= "omar" "omar")
T
```

```
Command: (= "omar" "bush")
nil
```

عند اعطائه ثلاثة قيم فيشرط تساوي الثلاث قيم

```
Command: (= "AHMED" "AHMED" "SHARON")
nil
```

• علامة لا يساوي /=

```
Command: (/= "omar" "omar")
nil
```

```
Command: (/= "omar" "us")
T
```

Command: (< 10 20)

T

Command: (< 10 20 30)

T

Command: (< 3 4 5 5)

nil

اي الحرفين او لا

Command: (< "e" "t")

T

● اصغر من او يساوى

Command: (<= 5 8)

T

• الدالة eq

السلسلتين متساوين ام لا

اذا كانا متساوين يعطينا t

غير متساوين nil

Command: (setq a '(1 2) c '(2 3))

(2 3)

Command: (eq a c)

nil

Command: (setq a b)

(1 2)

Command: (eq a b)

T

• الدالة equal

مقارنة بين سلسلتين

(setq f1 '(a b c))

(setq f2 '(a b c))

(setq f3 f2)

(setq a 1.123456)

(setq b 1.123457)

Command: (equal f1 f3)

T

Command: (equal f3 f2)

T

Command: (equal a b)

nil

الفرق بين a b هو 0.000001 فقارن بين الفرق بين b و a 0.000001

Command: (equal a b 0.000001)

T

### الدوال النصية

• دالة SUBSTR

و هي لاظهار عدد معين من الاحرف من سلسله نصية

Command: (SUBSTR "MY NAME AMR SELIM" 9 3)

"AMR"

و تتكون من :

''' الجملة النصية.

9 بداية اقتصاص الحروف.

3 عدد الحروف المقصصة .

السلالس الرقمية

مجموعة من القيم المتتالية

و يستخدم امر لتحدد

Command: (setq x1(list 1 2 3 4))

(1 2 3 4)

• الدالة strcat

و هي تضم جملتين في جملة واحدة

(strcat " no god but allah " " Mohamed messenger of allah ")

"no god but allah Mohamed messenger of allah"

• الدالة car

عرض اول رقم في السلسلة

Command: (car x1)

1

و يمكن استخدامه دون وجود سلسلة مكونه بأمر list

يمكن هذا بكتابه السلسله مباشرة مع سبقها بعلامة '

Command: (car '(3 4 5))

3

Command: (car '((3 4) 5))

(3 4)

Command: (car '())

nil

• الدالة cdr

عرض ثاني رقم في السلسلة

Command: (cadr x1)

2

• الدالة caddr

عرض ثالث رقم في السلسلة

Command: (caddr x1)

3

• الدالة cdr

عرض السلسلة بدون الرقم الاول

Command: (cdr x1(

(2 3 4)

• الدالة nth

عرض عنصر معين من السلسلة الرقمية مع العلم انه يبدأ بترقيم اول عنصر بصفر.

Command: (nth 4 '(2 5 4 7 9 6))

9

• الدالة cons

اضافة عنصر لاول القائمة

Command: (cons 'a '(b c d))

(A B C D)

• الدالة apply

و هي تقوم بتنفيذ دالة معينة على السلسلة الحرفية

Command: (apply 'strcat '("a" "b" "c"))

24

Command: (apply 'strcat '("a" "b" "c"))

"abc"

• دالة APEND

و هي تضم السلسل الرقمية بعضها الى بعض في سلسلة واحدة

Command: (append '(a b c) '(d e f))

(A B C D E F)

Command: (append '((a) (b) (c)) '((d) (e) (f)))

((A) (B) (C) (D) (E) (F))

Command: (append '(egypt sudan) '(iraq syria) '(lebnan philsten) )

(EGYPT SUDAN IRAQ SYRIA LEBNAN PHILSTEN)

• دالة reverse

و تقوم بعكس السلسلة الرقمية

Command: (setq a '(1 2 3 4 5 6))

(1 2 3 4 5 6)

Command: (reverse a)

(6 5 4 3 2 1)

#### ● دالة member

و هي تأخذ عنصر و اسم السلسلة الرقمية و تعطينا السلسلة الرقمية من اول هذا العنصر

Command: (setq a '(2 3 4 5 6))

(2 3 4 5 6)

Command: (member 4 a)

(4 5 6)

و في حالة اعطائنا عنصر غير موجود في السلسلة الرقمية تعطينا nil

Command: (member 1 a)

nil

#### ● دالة listp

هل العنصر المعطى سلسلة رقمية ام لا

اذا كان سلسلة رقمية t

و اذا كان غير هذا nil

Command: (listp a)

T

Command: (listp t)

nil

#### ● دالة strlen

و تعيد عدد حروف الجملة بما فيها المسافات والارقام والرموز

Command: (strlen "destory israil")

14

#### دالة التحويل

##### ● دالة atof

لتحويل النص الى رقم حقيقي float

عند تخزين رقم مثل 314 على هيئة string فان العمليات الحسابية لا تؤثر عليه و لا يتغابب معها .

Command: (atof "2.2")

2.2

##### ● دالة atoi

لتحويل النص الى رقم طبيعي

Command: (atoi "2.2")

2

##### ● دالة rtos

## لتحويل الرقم الى نص

```
Command: (rtos 2)  
"2"
```

- دالة cvunit التحويل من وحدة الى وحدة اخرى سواء وقت او وزن او طول او .....الخ

```
Command: (cvunit 1 "minute" "second")  
60.0  
Command: (cvunit 1 "gallon" "furlong")  
nil  
Command: (cvunit 1.0 "inch" "cm")  
2.54  
Command: (cvunit 1.0 "acre" "sq yard")  
4840.0  
Command: (cvunit '(1.0 2.5) "ft" "in")  
(12.0 30.0)  
Command: (cvunit '(1 2 3) "ft" "in")  
(12.0 24.0 36.0)
```

- الدالة type و تعطينا نوع المتغير

```
Command: (SETQ A 2)  
Command: (TYPE A)  
INT
```

```
Command: (SETQ A "AMRSELIM")  
Command: (TYPE A)  
STR
```

- الدالة princ اظهار رساله على سطر الاوامر

```
Command: (princ ("يا حي يا قيوم"))  
""يا حي يا قيوم
```

و يمكن اضافة \n لجعل باقي الكتابة على سطر جديد

```
Command: (princ "good\n friend")  
good  
friend
```

و يمكن اضافة \t لجعل باقي الكتابة بعد مسافة

```
Command: (princ "father come \t hide")  
father come      hide
```

• الدالة ALERT  
اظهار رساله للمستخدم

Command: (ALERT ("سبحان الله و بحمده"))

• الدالة COMMAND  
و هي لاعطاء اوامر الاتوكاد و هناك الكثير من اوامر الاتوكاد تستخدم بالفعل في autolisp كرسم خط او النسخ

Command: (COMMAND "COPY")  
Select objects:

لتحويل من ال layout الى model

Command: (command "tilemode" 0)

لتحويل من ال layout الى model

Command: (command "tilemode" 1)

لرسم خط

Command: (setq a (getpoint))  
Command: (setq b (getpoint))  
Command: (command "line" a b "")

لرسم قوس

Command: (setq a (getpoint))  
Command: (setq b (getpoint))  
Command: (setq c (getpoint))  
Command: (command "arc" a b c)

• الدالة LOAD  
لتحميل ملف

(LOAD "الملف الذي تريده تحميه" "رساله خطأ عند عدم وجود الملف")

علامة \ لها معنى خاص في لغة الاتوليسب لذلك عند كتابة مسار ملف تكتب مزدوجة \\ او نستعيض عنها بـ /

(LOAD "c:/amrselim/lisp" "I can't see anything , iam blind")

• الدالة DEFUN  
و هي اهم الدوال على الاطلاق و لا يخلو منها ليسب ، و هي لتعريف الدالة و اعطائها اسم للاستدعاء  
و هي اختصار DEFIN FUNCTION تعريف دالة

Command: (defun s (n)(+1 n))

حيث n هنا متغير  
و يوجد نوعين من المتغيرات  
متغير يستخدم بواسطة الدالة

و متغير محلی يستخدم في الداله فقط و يتم سبقه بعلامة /

Command: (defun s (/n)(.....))

متغيرين احدهما محلی

Command: (defun s ( o /m)(.....))

بدون متغير

Command: (defun s ( )(.....))

```
(defun C:HATCH ()  
  (alert "Using the BHATCH command!")  
  (princ "\n OLDHATCH to get to real HATCH command.\n")  
  (command "BHATCH")  
  (princ)  
)
```

(defun C:HATCH ()

السطر الاول : اعطينا اسم للدالة .. اسم الداله هنا هو HATCH و C: لاعلام البرنامج ان يعامل الدالة كأي  
أمر من أوامر الأتوCAD  
السطر الثاني : رسالة اعلام للمستخدم  
السطر الثالث : تظهر رسالة للمستخدم  
السطر الرابع : اعطاء امر للبرنامج

Command: (defun c:ze() (command "zoom" "e"))

في هذا المثال عرفنا دالة باسم ze و هي بمثابة اختصار لامر zoom, Extents و تقييد هذه الدالة و امثالها في حالة نسيان المسافة بين اختصاري z , e ,

- الدالة setq تحديد قيمة لمتغير ، و هي اختصار set-quote تعنى اقتبس و set اجعل

Command: (setq x 1)

1

و يمكن اختصار الثلاث جمل التالية

Command: (SETQ A 1)  
Command: (SETQ S 2)  
Command: (SETQ E 3)

بالشكل التالي

Command: (SETQ A 1  
S 2  
E 3)

او

Command: (SETQ A 1 S 2 E 3)

و يمكن نقل قيمة المتغير الى متغير اخر

Command: (setq a 123)

123

Command: (setq b 'a)

A

لاستدعائها نستخدم !

Command: !x

1

• الدالة getpoint

لتحديد نقطة و اعطاء البرنامج احداثيتها لاستخدامها لاحقا

Command: (setq a (getpoint))

(-640.261 146.069 0.0)

• الدالة getdist

و يستخدم لمعرفة المسافة بين نقطتين حيث نعطيه احدى النقطتين ثم نحدد الثانية بالماوس او ندخلها كتابة

Command: (getdist a)

2727.41

(setq dist (getdist '(1.0 3.5)))

و يمكننا اضافه سؤال للمستخدم

(setq dist (getdist '(1.0 3.5) "How far? "))

او يمكن ترك تحديد النقطتين للمستخدم

(setq dist (getdist))

• الدالة distance

و تستخدم لمعرفة المسافة بين نقطتين (الفرق بينها و بين getdist اتنا هنا نعطيها النقطتين مباشرة )

Command: (distance a b)

262.061

• الدالة getint

و تستخدم للحصول على رقم صحيح من المستخدم

(getint "\nEnter your age :")

12

و عند اعطاء اي قيمة لا تساوي رقم صحيح لا يقبلها و يعطينا رساله الخطأ التالية

Command: (getint)

frog

Requires an integer value.

لابد من رقم صحيح فلا يقبل اي رقم به علامة عشرية

Command: (getint)  
15.0  
Requires an integer value.

و يتم هنا تخزينها في متغير

Command: (setq num (getint "Enter a number:"))  
Enter a number: 25

• الدالة getreal  
و تستخدم للحصول على اي رقم صحيح او كسر

Command: (getreal)  
2.2

(setq val (getreal "Scale factor: "))

• الدالة getstring  
للحصول على نص من المستخدم و الحد الاقصى 132 حرفاً اذا كتبت اكثراً من هذا سيتم تجنبه

Command: (setq q (getstring "it's a good day"))  
it's a good day  
"friday"

و اذا كتب المستخدم علامة | فأن البرنامج يحولها الى \| لنتتمكن من استخدامها كمسار في دوال أخرى

Command: (setq s (getstring T "Enter filename: "))  
Enter filename: c:\my documents\vlisp\secrets  
"c:\\my documents\\vlisp\\secrets"

• الدالة getcorner  
و هي تعطيها نقطة ثم تأخذ نقطة اخرى من المستخدم عن طريق تحريك مؤشر الماوس مبتعداً عن النقطة على  
هيئه مستطيل (الفهم اكثراً طبق ثم انظر في الامثلة العملية , الطريقة الثانية لرسم مستطيل)

Command: (setq b (getcorner a))  
(5250.86 639.033 0.0)

• الدالة ascii  
اعطاء رقم الاسكى للحرف

Command: (ascii "b")  
98

و في حالة عطائه كلمة يعطينا الاسكى لابن حرف فقط

Command: (ascii "a")  
97  
Command: (ascii "amr")  
97

• الدالة chr  
اعطاء الحرف المقابل لرقم الاسكى

Command: (chr 97(

"a"

- الدالة atom و تعطى nil اذا كان سلسله و t اذا لم يكن سلسله

```
Command: (setq a '(b c d e))  
(B C D E)  
Command: (atom a)  
nil  
Command: (atom b)  
T  
Command: (atom '(s d f))  
nil
```

- الدالة eval تخبرني بالقيمة المسجلة في المتغير

```
Command: (setq q 3)  
Command: (eval q)  
3
```

```
Command: (eval (abs -10))  
10
```

- الدالة quote تجعل الحروف كبيرة

```
Command: (quote q)  
Q
```

```
Command: !'a  
A  
Command: (quote (a b))  
(A B)
```

- الدالة setvar اعطاء قيمة لمتغير الانواد مثل cmdecho الخاص بالاستجابات الخاصه بسطر الاوامر

```
(setvar "cmdecho" 1)  
و filedia عرض مربعات الحوار التي تسمح لك باختيار الملفات او جعل التعامل في سطر الاوامر وله قيمتين
```

- 0 لا تعرض صناديق الحوار
- 1 عرض صناديق الحوار

```
(setvar "filedia" 1)
```

و cmddia عرض مربعات الحوار لاوامر متنوعة مثل الطباعة او جعل التعامل في سطر الاوامر وله قيمتين

- 0 لا تعرض صناديق الحوار
- 1 عرض صناديق الحوار

```
(setvar "cmddia" 1)
```

و هو متغير يتحكم في المسافة بين الجذب snapunit

```
(setvar "snapunit" '(12 12))
```

متغير يتحكم في ظهور كل ملفات الاتوكاد في شريط المهام او ظهور واحد فقط Taskbar



Command: taskbar

Enter new value for Taskbar <1>: 0

- الدالة getvar الحصول على قيمة متغير الاتوكاد

```
(if (<(getvar "fillettrad") 1)
    (setvar "fillettrad" 1)
)
```

و يمكن توفير خطوتين في خطوة واحدة كالمثال التالي حيث قمنا بعملية حسابية و اعطينا متغير قيمة

Command: (/ 21 (setq q 7))

3

Command: !Q

7

- الدالة (entlast) تعطينا بيانات عن آخر عنصر

Command: (entlast)

<Entity name: 7efb40b0>

و اذا استخدمناها قبل رسم او استخدام اي عنصر فإنه يعطينا nil

Command: (entlast)

nil

و للحصول على معلومات تفصيلية نكتب

Command: (setq a (entget(entlast)))

```
((-1 . <Entity name: 7efb40b0>) (0 . "LINE") (330 . <Entity name: 7efb2cf8>) (5
. "116") (100 . "AcDbEntity") (67 . 0) (410 . "Model") (8 . "0") (100 .
"AcDbLine") (10 891.592 1203.92 0.0) (11 1789.65 627.174 0.0) (210 0.0 0.0 1.0))
```

و هناك رموز شائعه الاستخدام مثل

1-	اسم الكائن
0	نوع الكائن
1	قيمة النص
5	القبضه handle
6	اسم ال LINE TYPE
8	الطبقة
10	نقطة البداية او المركز (بحسب العنصر المحدد)
11	نقطة النهاية او المحاذاة
38	المسقط
39	السمك
40	نصف القطر او ارتفاع الن
62	اللون
67	عنصر تحديد خاص بحيز النقطة او علامة حيز الورقة
70	رقم اصدار ال ole
72	حالة العمل هل انت على ال layout ام على model
0	عندما تكون في 1 ، model عندما تكون في layout
284	نظام التظليل

• الدالة ssget و تقوم هذه الدالة بحفظ العناصر التي يحددها المستخدم لاستخدامها لاحقا

```
Command: (ssget)
Select objects: Specify opposite corner: 2 found
```

و يتم ربطها بمتغير عن طريق الامر setq

```
Command: (setq q (ssget))
Select objects: Specify opposite corner: 2 found
Select objects:
<Selection set: 2>
```

و يمكن استخدام العناصر المحددة لاحقا

```
Command: m
MOVE
Select objects: !q
<Selection set: 2>
```

و يمكن فاتره العناصر التي نريدها مثل اختيار الخطوط فقط

```
(SETQ XX(SSGET (LIST (CONS 0 "LINE"))))
```

• الدالة polar

يحدد نقطة بناء على نقطة أخرى و الزاوية التي تحدها و المسافة بينهما  
في هذا المثال : النقطة a و الزاوية 0.1 و المسافة عشرون

```
Command: (setq a (getpoint))
Command: (setq q (polar a 0.1 20))
(917.808 2660.4 0.0)
```

• الدالة assoc يبحث في السلسل المزدوجة association list عن زوج كل عنصر

```
Command: (setq country '((amrica israil) (almania itali) (iran turki)))
((AMRICA ISRAIL) (ALMANIA ITALI) (IRAN TURKI))
```

```
Command: (assoc 'amrica country)
(AMRICA ISRAIL)
```

و في حالة سؤاله عن عنصر غير موجود

```
Command: (assoc 'utobia country)
nil
```

• الدالة AND و هي تخبرنا اذا كان المتغير او المعطى به قيمة ام لا  
اذا اعطيته اكثر من متغير فيشرط ان يكون كلهم بهم قيمة ليعطيني t  
اذا كان ليس به قيمة يعطينا nil

```
Command: (setq a 103 b nil c "string")
"string"
```

```
Command: (and 1.4 a c)
```

T

```
Command: (and 1.4 a b c)
nil
```

```
Command: (and a)
T
```

```
Command: (and w)
Nil
```

```
Command: (AND 5)
T
```

• الدالة arx و تعطينا اسماء arx المحملة حاليا  
AutoCAD Runtime Extension

```
Command: (arx)
("acapp.arx" "accamera.arx" "acdbclkedit.arx" "acdblclkeditpe.arx" "acdimg.arx"
"acdwgrecovarx" "acdxnotify.arx" "aceplotx.arx" "acetlodr.arx"
"acetutil.arx" "acgsconfig.arx" "achapi17.dbx" "achlnkui.arx" "acidropmgr.arx")
```

```
"acismobj17.dbx" "acismui.arx" "aclayerp.arx" "acmfd.dll" "acpltstamp.arx"  
"acprevinput.arx" "acsecopt.arx" "acsign.arx" "acspacetrans.arx" "acstd.arx"  
"actaskbar.arx" "actp.arx" "acxrefesw.arx" "acxrefservice.arx" "movebak.arx"  
"oleaprot.arx" "vl.arx" "whohas.arx" "wscommcntraccon.arx")
```

● دالة arxload  
لتحميل ملف arxload

```
Command: (arxload anyfile.arx)
```

```
Command: (arxload "c:/program files/AutoCAD 2007/anyfile.arx")
```

● دالة arxunload  
لاغاء تحميل ملف arx

```
Command: (arxunload anyfile)
```

● دالة boundp  
تنظر هل يحتوي العنصر على قيمة ام لا  
اذا كان يحتوي على قيمة الناتج t  
اما اذا كان لا يحتوي على قيمة الناتج يكون nil

```
Command: (setq x 13 y nil)  
nil
```

```
Command: (boundp 'x)
```

```
T
```

```
Command: (boundp 'y)  
nil
```

● الدالة if الشرطية

و تستخدم لوضع شرط على حدث فمثلا لا ترسم دائرة الا اذا كان نصف القطر اكبر من عشرين  
و صيغتها

```
((الشرط) (حدث اذا تحقق الشرط)(حدث اخر اذا لم يتحقق الشرط))
```

```
IF (YOU UNDERSTAND ANY THING) ,( MEET ME)
```

```
Command: (setq z 1)
```

```
Command: (if (< z 3) (alert"islamna"))
```

```
Command: (setq e (getint))
```

```
Command: (if (< e 20) (princ "e is less than 20") (princ "e is not less 20"))  
e is not less 20"e is not less 20"
```

في السطر الاول انشأنا متغير يأخذ رقم من المستخدم  
السطر الثاني اذا كان الرقم الذي ادخله المستخدم اقل من عشرين اظهر رسالة "الرقم الذي ادخلته اصغر من  
عشرين" و اذا كان الرقم اكبر من عشرين اظهر رسالة "الرقم الذي ادخلته اكبر من عشرين"

```
Command: (SETQ A (GETINT))
```

Command: (SETQ B (GETINT))  
Command: (if (= a b) (setq b 5) (setq b 6))

السطر الاول و الثاني نطلب من المستخدم ادخال رقمي للمتغيرين A,B  
السطر الثالث اذا كانت قيمتي المتغيرين متساويين فاجعل قيمة المتغير الثاني 5  
و اذا لم يكونوا متساويين فاجعل قيمة المتغير الثاني 6

Repeat •  
تكرار عملية معينة عدد معين من المرات

Command: (SETQ A 100)  
Command: (REPEAT 4 (SETQ A (+ A 10)))  
140

في السطر الثاني قمنا بتكرار عملية اضافة 10 الى المتغير

While •

افعل شيء ما حتى يحدث حدث معين  
"استمر في سرقة الخزانة حتى يكتشف المدير"  
"استمر في الغياب حتى تلذغ اخر الشهر "

Command: (while (شرط)  
              ((الاوامر التي سينفذها البرنامج))

Command: (setq a 1)  
Command: (while (< a 20)  
(princ a)  
(princ "\n")  
(setq a (+ 1 a)))

السطر الاول نحدد قيمة للمتغير  
السطر الثاني نخبر البرنامج ان يستمر في العمل ما دام المتغير اصغر من العشرين  
السطر الثالث و الرابع اوامر للبرنامج و هنا نأمره بكتابة المتغير  
السطر الخامس اجعل المتغير يزيد بمقدار 1

و السطر  
(princ "\n")  
مكافئ ل (terpri) و الذي يقوم بالكتابه على السطر التالي

• دالة acad\_strlsort  
و هي ترتيب السلسل الحرفية ترتيبا ابجدياً (في هذا المثال رتب الاشهر ترتيبا ابجديا )

Command: (setq mont '("Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug"  
"Sep" "Oct" "Nov" "Dec"))  
("Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec")  
Command: (acad\_strlsort mont)  
("Apr" "Aug" "Dec" "Feb" "Jan" "Jul" "Jun" "Mar" "May" "Nov" "Oct" "Sep")

• دالة vl-file-delete  
لالغاء ملف معين و اذا كان في ملف SUPPORT لا نكتب المسار  
(vl-file-delete "newauto.bat")  
و يمكننا كتابة المسار  
(vl-file-delete "c:/newauto.bat")

• دالة ACAD\_COLORDLG

داله تعطيك نافذه تختار منها لون و في هذا المثال جعلنا اللون الافتراضي هو الاخضر 3

Command: (ACAD\_COLORDLG 3)

و لتخزين اللون لاستخدامه لاحقاً (اعطاء لون لطبقه مثلاً)

Command: (SETQ Q(ACAD\_COLORDLG 3))

DCL  
لجعل واجهة رسومية  
و لتحميل ملف DCL لابد من أمرين  
الاول وضع الملف DCL في ملفات الـ SUPPORT او اضافة المسار الى ملفات الاتوكاد  
DCL الثاني استدعاء ملف

```
(("setq Dcl_Id% (load_dialog "MyDialogs.dcl)
```

مثال

```
MyFirst : dialog {  
    label = "Hello World";  
    spacer;  
    : text {  
        label = "This is my first dialog.";  
        alignment = centered;  
    }  
    spacer;  
    ok_only;  
} //MyFirst
```

، وفي ملف الليسب نكتب

```
defun c:MyFirst (/ Dcl_Id%)  
  ( (princ "\nMyFirst") (princ  
; لتحميل الصندوق الحواري  
  setq Dcl_Id% (load_dialog "MyDialogs.dcl"))  
    (new_dialog "MyFirst" Dcl_Id%)  
      ; Dialog Actions  
      ( (start_dialog  
; لاغاء التحميل  
      unload_dialog Dcl_Id%)  
        (princ)  
      ); defun c:MyFirst
```

نقوم بتشغيل الليسب  
MYFIRST  
فيظهر امامنا





# اسیم و البرمجة

A word cloud visualization where the size of each programming language name corresponds to its popularity or prevalence. The most prominent word is "Java", which is rendered in a large, bold, dark green font. Surrounding "Java" are numerous other programming language names, each with a unique color. To the left of "Java", the word "JavaScript" is also quite large. Other notable words include "Python", "C++", "Haskell", "Ruby", "PHP", "Pascal", "Swift", "Mathematica", "Lisp", "Groovy", "Tcl", "SAS", "cT", "Max/MSP", "Dart", "Scala", "Haskell", "Cobol", "Erlang", "C#", "D", "Scal", "F#", "ML", "Assembly", "Scratch", "OpenEdge ABL", "ABAP", "Visual Basic", "Transact-SQL", "Clarion", "FoxPro", "Perl", "Logo", "Prolog", "PostScript", "Scheme", "R", "C++", "Haskell", "Go", "Fortran", "ActionScript", "Forth", "PL/SQL", "MATLAB", "C", "Lua", "TCL", "Delphi", "PL/I", and "Objective-C". The colors range from dark brown to bright yellow-green, creating a dense and colorful composition.

عمر سلیم

## 1. تغير كل العناصر الى اللون الابيض

Command: ( command " chprop" "all" "" "color" "7" "")

اعلام البرنامج اننا سنعطيه امر من اوامر الاتوكاد  
امر تغير الخصائص "chprop"  
اختيار تغير كل الخصائص "all"  
او يمكن كتابة last لتغير اللون لآخر عنصر فقط  
ضغط زر الادخال ""  
اختيار تغير خاصية اللون "color"  
اختيار تغير اللون الابيض "7"  
ضغط زر الادخال ("")

## 2. تغير لون العناصر التي على طبقة معينة

Command: (command "layer" "color" "7" "hvac" "")

اعلام البرنامج اننا سنعطيه امر من اوامر الاتوكاد  
امر الطبقات "layer"  
اختيار تغير خاصية اللون "color"  
اختيار تغير اللون الابيض "7"  
اختيار الطبقة "hvac"  
ضغط زر الادخال ("")

## 3. رسم خط

Command: (setq a (getpoint))  
Command: (setq b (getpoint))  
Command: (command "line" a b "")

السطر الاول و الثاني نحدد فيهم نقطتين  
السطر الثالث نرسم مستطيل بمعلومية ركnie  
و "" بمعنى ضغط زر الادخال

## 4. رسم مستطيل

Command: (setq a (getpoint))  
Command: (setq b (getpoint))  
Command: (command "rectangle" a b)

السطر الاول و الثاني نحدد فيهم نقطتين  
السطر الثالث نرسم مستطيل بمعلومية ركnie

## 5. طريقة اخرى لرسم مستطيل

Command: (setq a (getpoint a))  
Command: (setq b (getcorner a))  
Command: (setq c (list (car b)(cdr a)))  
Command: (setq d (list (car a)(cdr b)))  
Command: (command "line" a c b d a "")

اي نقطة تحدد بادعائي س, ص  
السطر الاول نحدد نقطة A

السطر الثاني نحدد ركن (النقطة B) بناء على النقطة الاولى  
 السطر الثالث نحدد نقطة (النقطة C) تأخذ من النقطة الثانية موقعها على محور السينات و من النقطة الاولى  
 موقعها على محور الصادات  
 السطر الرابع نحدد نقطة (النقطة D) تأخذ من النقطة الثانية موقعها على محور الصادات و من النقطة الاولى  
 موقعها على محور السينات (رسم بياني)  
 السطر الخامس نرسم اربع خطوط بين النقط الاربعه

#### 6. طريقة ثلاثة لرسم مستطيل

Command: (command "rectangle" "\_width" "2" "0,0" "10,10")

حيث نرسم مستطيل بسمك 2 و نقطتين معلومتين هما "10,10" "0,0"

#### 7. رسم دائرة

Command: (setq a (getpoint))

Command: (command "circle" a 200)

circle Specify center point for circle or [3P/2P/Ttr (tan tan radius)]:

Specify radius of circle or [Diameter]: 200

السطر الاول تحديد نقطة المركز  
السطر الثاني

اعلام البرنامج اننا سنعطيه امر من اوامر الاتوCAD (command

"circle" رسم دائرة

نقطة المركز a

نصف القطر (200)

#### 8. عند عمل bind لملف

```
(DEFUN C:EN()
  Command: (command "tilemode" 0)
  (COMMAND "ZOOM""E")
  (command "audit" "y")
  (setvar "bindtype" 1)
  (command "-xref" "b" "*")
  (COMMAND "PURGE""ALL""*""N")
  (COMMAND "QSAVE")
  (COMMAND "CLOSE")
)
```

(DEFUN C:EN() الدالة تعريف

Command: (command "tilemode" 0) التحويل الى layout

(COMMAND "ZOOM""E") عمل زoom للشاشة

(command "audit" "y") audit عمل

(setvar "bindtype" 1) تغير نوع البايند

(command "-xref" "b" "\*") bind أمر

(COMMAND "PURGE""ALL""\*""N") تنضيف اللوحة

(COMMAND "QSAVE") حفظ

(COMMAND "CLOSE") اغلاق

## غلق القوس )

### 9. كتابة نص

كتابة نص بنقطة معرفة سابقا و ارتفاع 200 و زاوية 0

```
Command: (SETQ A (GETPOINT))
(278.852 53.9469 0.0)
```

```
Command: (COMMAND "TEXT" A "200" "0" "AMRSELIM" )
```

### 10. تغيير ارتفاع نص

```
(defun C:chtext (/ a ts n index b1 b c d b2)
  (setq a (ssget))
  (setq ts (getreal "\nEnter new text size"))
  (setq n (sslength a))
  (setq index 0)
  (repeat n
    (setq b1 (entget (ssname a index))))
  (setq index (1+ index))
  (setq b (assoc 0 b1))
  (if (= "TEXT" (cdr b))
    (progn
      (setq c (assoc 40 b1))
      (setq d (cons (car c) ts))
      (setq b2 (subst d c b1))
      (entmod b2)))
    (princ)
  ))
```

### مصطلحات

الكلمة	الشرح
REAL OR FLOAT	ارقام حقيقة مثل 2.3 و 5
INTEGER	رقم صحيحة مثل 5 و 6 و 3 بدون ارقام عشرية و هي من 32767 الى 32768
LIST	مجموعة من الارقام المسلسلة ذات ترتيب معين (6 5 2 14 8)
string	نص "غزة رمز العزة"
symbol	عنصر مثل point

### برامح مساعدة

- برنامج متميز لعمل lisp و dialog component defined dcl (dcl) بطريقة سهلة و قوية و اجهه مرنية تشبه الفيجوال بيسك لكنه ليس مجاني

<http://www.beowulftech.com/lispgen/home.html>

### AUTOLISP اهم المواقع لتعلم

- [www.amrselim.net](http://www.amrselim.net)
- <http://www.afralisp.net/>
- <http://www.cadtutor.net>
- <http://www.jefferyspanders.com/autolisp.html>



# اسیم و البرمجة

عمر سليم

add.lsp	تضرط أحياناً لجمع أرقام موجودة على الشاشه أمامك، ومن الأفضل أن تقاوم بعملية الجمع مباشرة دون الاستعانة بالآلة الحاسبة، وداعلك سوى أن تتقى على الرقم مع زر الإدخال فترى الرقم في سطر الحاله، ثم تتقى على الرقم هذا البرنامج لكل من يحتاج إلى التحويل بين وحداتقياس المختلفة
ALLUNITS	يتميز ALLUNITS بطريقه السريعة للتحويل بين وحداتقياس المختلفة (وحدات قياس الزمن، الطول، المساحة، الحجم، الوزن، الزوايا)، كل ذلك باستخدام مربع حوار البرنامج الذي يمكن تشغيله كذلك الأمر بشفافية داخل بعض أوامر اوتوكاد التقليدية وتمرير القيم الناتجة عن عملية التحويل إلى سطر الأوامر مباشرة.
AntiRot.VLX AUTOLABEL.VLX barea bd BreakAll bylaybk.lsp	تدوير كائن أو مجموعة كائنات ترقيم تلقائي يستخدم لحساب مساحة منطقة عن طريق توقيع نقطة بها لرسم الدكت (تكيف) احداث قطع في الخطوط المتقطعة وذلك في نقاط التقاطع يجعل لون البلوك لون الایر يقوم البرنامج بحساب مساحات الأشكال في اوتوكاد، ويعطي نتيجة سريعة للحساب حيث يقوم بإنشاء متعدد خطوط Polyline حول الشكل المطلوب حساب مساحته وبطبيعة مستقلة لونها أصفر للتذيز في حال وجود أكثر من جزء ثم يطلب البرنامج نقطة لإدراج النص مع إمكانية التحكم بحجمه بشكل ديناميكي وعلى طبقة خاصة أيضاً.
CAR.LSP	يتحول حالة الكتابة POLYLINE نسخ كل الكلام المحدد و يمكنك بعد هذا نقلة لملف كتابي خارجي عمل حصر للكتل مسح البلوك عند الانتهاء من الملف و ارساله يقوم بـ AUDIT, BIND, LAYOUT ZOOM, SAVE, مسح كل النقاط Point الموجودة في الرسم دفعه واحدة فتح المستعرض على اللوحة الحالية FLEX
CASE.LSP circle2pl clipcopy CNT EBL.lsp EN.lsp epts.LSP EXDWG.LSP flex inch-cm.lsp INSORT layerhtm ll.lsp LLN.LSP openbak.lsp PEND.LSP prefix RBL.LSP RF.lsp SAVEA.LSP WBA.LSP wblockm.lsp WPIPE.LSP	يتحول لاثن الى سنتيمتر ادراج بلوكات يتم تدويرها مع الاحتفاظ باقئية العناصر الكتابية انشاء صفحة تحت بها اسماء الطبقات و خصائصها غلق كل الطبقات برنامج اوتوليس بسيط يحسب مجموع اطوال عدة خطوط Lines او متعددة خطوط Polylines فتح ملفات اوتوكاد ذات الهيئة BAK مباشرة رسم علامة القطع توليد ارقام متتالية يسبقها كلمة بيدل البلوك لرص البلوكات انشاء نسخ احتياطية من الرسومات جعل كل بلوك في لوحة منفصلة مع كتابة ورقة باسماء البلوكات تصدير كل بلوك في اللوحة الى لوحة منفصلة لرسم صاج تكيف

## ملخص للأوامر

<b>defun c:()</b>	تعريف الدالة	<b>atof</b>	تحويل النص الى رقم حقيقي
<b>zerop</b>	التأكد هل الناتج صفر ام لا	<b>atoi</b>	تحويل النص الى رقم طبيعي
<b>exp</b>	تعطى القيمة الاسية	<b>rtos</b>	تحويل الرقم الى نص
<b>~</b>	متمم العدد	<b>cvunit</b>	التحويل من وحدة الى وحدة اخرى
<b>MAX</b>	الرقم الاكبر	<b>TYPE</b>	تعطينا نوع المتغير
<b>MIN</b>	الرقم الاصغر	<b>princ</b>	اظهار رساله على سطر الاوامر
<b>REM</b>	المتبقي من القسمة	<b>ALERT</b>	اظهار رساله للمستخدم
<b>FIX</b>	تقريب لاقرب عدد صحيح و الغاء الكسور	<b>COMMAND</b>	لاعطاء اوامر الاتوكاد
<b>FLOAT</b>	تحول العدد الى كسر	<b>LOAD</b>	تحميل ملف
<b>LOG</b>	ابجاد اللوغاريتم	<b>setq</b>	تحديد قيمة لمتغير
<b>SQRT</b>	الجذر التربيعي	<b>getpoint</b>	تحديد نقطة
<b>sin</b>	جا الزاوية	<b>getdist</b>	يستخدم لمعرفه المسافة بين نقطتين
<b>cos</b>	جتا الزاوية	<b>distance</b>	يستخد لمعرفه المسافة بين نقطتين
<b>atan</b>	ظل الزاوية	<b>getint</b>	للحصول على رقم صحيح من المستخدم
<b>angle</b>	الزاوية	<b>getreal</b>	للحصول على اي رقم صحيح او كسر
<b>minusp</b>	الرقم المعطى سالب ام لا	<b>getstring</b>	للحصول على نص من المستخدم
<b>inters</b>	نقطة تلاقى اربع نقط	<b>ascii</b>	اعطاء رقم الاسكى للحرف
<b>/=</b>	لا يساوى	<b>chr</b>	اعطاء الحرف المقابل لرقم الاسكى
<b>eq</b>	السلسلتين متساويتين ام لا	<b>atom</b>	و تعطى nil اذا كان سلسله و + اذا لم يكن
<b>equal</b>	مقارنة بين سلسلتين	<b>eval</b>	تخبرني بالقيمة المسجلة في المتغير
<b>SUBSTR</b>	لاظهار عدد معين من الاحرف من سلسله نصية	<b>quote</b>	تجعل الحروف كبيرة
<b>strcat</b>	تضم جملتين في جملة واحدة	<b>setvar</b>	اعطاء قيمة لمتغير الاتوكاد
<b>car</b>	عرض اول رقم في السلسلة	<b>getvar</b>	الحصول على قيمة متغير الاتوكاد
<b>cadr</b>	عرض ثانى رقم في السلسلة	<b>entlast</b>	تعطينا بيانات عن آخر عنصر
<b>caddr</b>	عرض ثالث رقم في السلسلة	<b>ssget</b>	بحفظ العناصر التي يحددها المستخدم
<b>cdr</b>	عرض السلسلة بدون الرقم الاول	<b>polar</b>	يحدد نقطة بناء على نقطة اخري والزاوية
<b>nth</b>	عرض عنصر معين من السلسلة الرقمية	<b>AND</b>	المتغير او المعطى به قيمة ام لا
<b>cons</b>	اضافة عنصر لاول القائمه	<b>arx</b>	و تعطينا اسماء arx المحملة حاليا
<b>apply</b>	تنفيذ دالة معينة على السلسلة الحرفية	<b>arxload</b>	تحميل ملف arxload
<b>append</b>	تضم السلسل الرقمية في سلسلة واحدة	<b>arxunload</b>	الاغاء تحميل ملف arx
<b>reverse</b>	عكس السلسلة الرقمية	<b>boundp</b>	هل يحتوي العنصر على قيمة ام لا
<b>listp</b>	هل العنصر المعطى سلسلة رقمية ام لا	<b>if</b>	لووضع شرط على حدث
<b>strlen</b>	عدد حروف الجملة	<b>Repeat</b>	تكرار عملية معينة عدد معين من المرات
	للشرح : حمل كتاب شرح الاتوليسب من	<b>While</b>	افعل شيء ما حتى يحدث حدث معين
		<b>acad_strlsort</b>	و هي ترتيب السلسل الحرفية ترتيبا ابجديا



## Ruby

روبي (أو ياقوت) (بالإنجليزية: Ruby) هي لغة برمجة كائنة ديناميكية متعددة الاستخدام تستخدم في تطوير الواقع الإلكتروني وتطبيقات الهاتف المحمول . وتمتاز اللغة بكونها لغة شبيهة نقية كما تمتاز باحتواها على كثير من خواص اللغات الوظيفية.

صممت لغة Ruby كي تكون بسيطة وسهلة في كتابة التعليمات البرمجية.

تعتبر لغة Python سهلة التعلم للمبتدئين.

صمم تطبيق Ruby on Rails( Rails) باستخدام لغة Ruby، والذي يستخدم في جمع المعلومات من خادم الويب أو البحث في قاعدة البيانات على شبكة الإنترنت.

ويستخدم هذا التطبيق على موقع الكتروني مثل: Shopify GitHub Groupon و Scribd

حسنا الجميل اننا يمكننا ان نبرمج اشياء للريفيت بلغة البرمجة روبي

<http://revitruby.com/>

هذه الاسطر يجب كتابتها في روبي لنبدأ في التعامل مع الريفيت

```
load_assembly 'RevitAPI'
```

```
load_assembly 'RevitAPIUI'
```

```
include Autodesk::Revit
```

```
include Autodesk::Revit::UI
```

```
include Autodesk::Revit::DB
```

```
include Autodesk::Revit::DB::Architecture
```

مرجع لبرمجة الريفيت ب روبي



PROGRAMMING  
Language

winner of the Ruby Logo Contest

<https://github.com/hakonhc/RevitRubyShell>

[www.youtube.com/watch?v=3rCu1acxwR0](http://www.youtube.com/watch?v=3rCu1acxwR0)

## Matlab

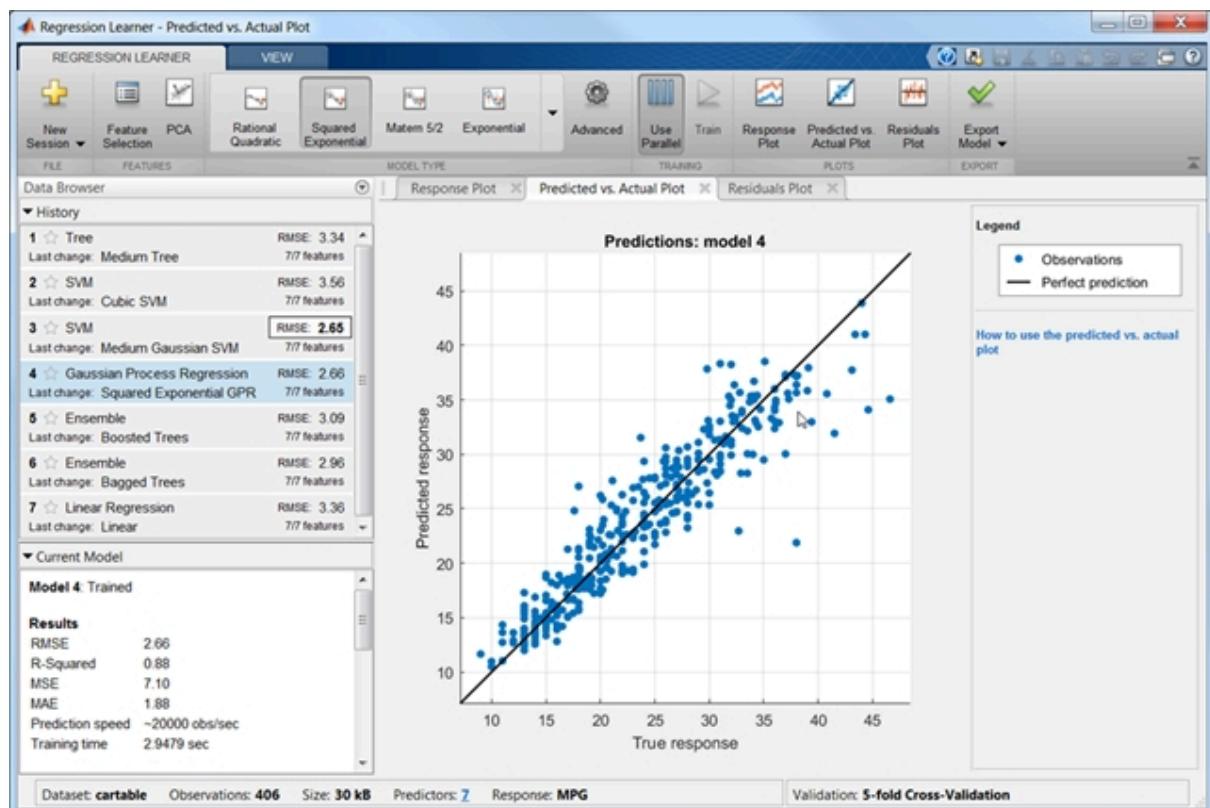
برنامج الماتلاب هو برنامج هندي (وله مجالات أخرى) يقوم بعمليات تحليل وتمثيل البيانات من خلال معالجة تلك البيانات تبعاً لقاعدة البيانات الخاصة به، فمثلاً يستطيع البرنامج عمل التفاضل differentiation و التكامل و كذلك المعادلات التفاضلية Integration و كذلك يقوم بحل المعادلات الجبرية Algebraic Equations ذات الرتب العليا والتي قد تصل من الصعوبة ما تصل، ليس فقط ذلك بل يستطيع البرنامج عمل التفاضل الجزئي، ويقوم بعمل عمليات الكسر الجزئي Partial fraction بسهولة ويسر والتي تستلزم وقتاً كبيراً لعملها بالطرق التقليدية، هذا من الناحية الأكاديمية، أما من الناحية التطبيقية يستطيع البرنامج العمل في جميع المجالات الهندسية مثل أنظمة التحكم Control System، وفي مجال الميكانيكا Mechanical Field، وكذلك محاكاة الإلكترونيات Electronics وصناعة السيارات Automotive Industry، وكذلك مجال الطيران والدفاع الجوي Aerospace and Defense، والكثير من التطبيقات الهندسية.

فعم القدام السريع في التكنولوجيا أصبحت الحاجة ملحة على تعلم مثل هذا البرنامج حتى نصبح في سباق التنافس الصناعي.

لغة الماتلاب

هي لغة ذات أداء عالي للحسابات التقنية، حيث تدمج التخمين ، و البرمجة بطريقة سهلة الاستعمال في بيئتها حيث المشاكل و الحلول تكون معبرة في مجموعة رموز رياضية ذات علاقة ببعضها.

لغة Matlab تطورت على مر السنين مع زيادة استعمالها مثلاً في محيط الجامعات حيث هي تركيبات وسائل قياسية لحصول المبتدئين و المتقدمين في الرياضيات ، الهندسة ، و العلوم في الصناعة ، ماتلاب وسيلة اختيار لابحاث الإنتاجية العالمية التطوير و التحليل .



## JavaScript

هي لغة قابلة للتنفيذ من جهة الخادم والعميل طورت من قبل شركة **Netscape** ، والتي استمدت جزءاً كبيراً من تراكيب جملها البرمجية من لغة **C**.

يمكن أن تستخدم عبر عدة متصفحات الكترونية، وتعتبر لغة أساسية في تطوير رسوميات متحركة وتفاعلية على شبكة الإنترنت.

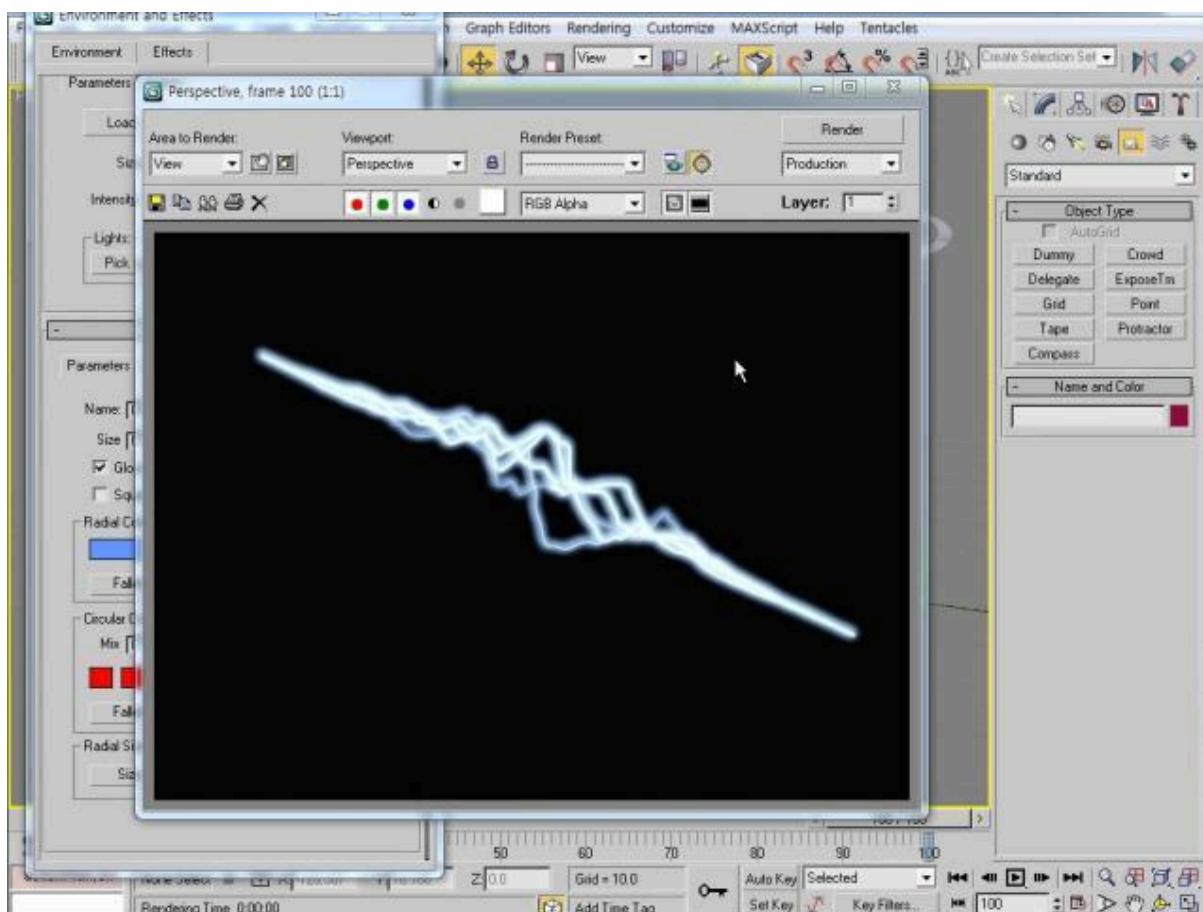
تستخدم أيضاً في تطوير الألعاب وإنشاء تطبيقات سطح المكتب.

تكون مترجمات لغة **Java script** متضمنة ضمن الإضافات الخارجية (**extensions**) لمتصفح الإنترنت **Google chrome** .**Adobe's Creative suite** و **Apple safari** **Adobe acrobat**

[https://www.youtube.com/watch?v=IzRMKTIJvOM&list=PLNMim060\\_nUKWvwo0iXClVbyIP0oGo6lu&pp=gAQBiAQB](https://www.youtube.com/watch?v=IzRMKTIJvOM&list=PLNMim060_nUKWvwo0iXClVbyIP0oGo6lu&pp=gAQBiAQB)

## MAXScript

لغة برمجة **scripting** مدمجة في التري دي ماكس Autodesk® 3ds Max® and Autodesk® 3ds Max®  
.Design



### Objects and Classes in Object-Oriented Programming

Inheritance and Polymorphism ■

Properties, Methods, Operators, and Literals ■

و يعطيك القدرة على :

```
-- create a sphere, convert it to a mesh, and animate all vertices
s=sphere radius:100
convertToMesh s
animateVertex s #all
-- on each frame, move each vertex around randomly
animate on
( for v=1 to s.numverts do
( v_pos_track="$"+s.name+".Vertex_"+(v as string)+".controller"
v_pos=execute(v_pos_track)
at time 0 v_pos_last=v_pos.value
for t = 1f to 20f do
( at time t
( v_pos.value=v_pos_last+(random [-5,-5,-5] [5,5,5])
v_pos_last=v_pos.value
)
)
)
)
```

- السيطرة على عناصر البرنامج من خلال سطر الأوامر
- تسجيل حركاتك في البرنامج باستخدام اوامر **maxscript**
- بناء معدلات تحكم في الأضاءة و العناصر في الثري دي ماكس
- عمل سكريبتات على هيئة زر اير

عشرات السكريبتات جاهزة

<http://www.scriptspot.com/3ds-max/scriptS>

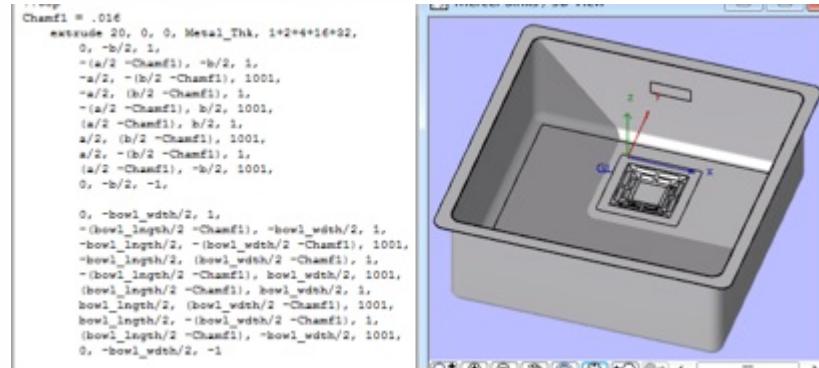
## GDL

### أختصار Geometric Descriptive Language

وهي لغة برمجة وظيفية تستند إلى BASIC

تم إنشاءه لتشجيع المهندسين المعماريين على استخدام هذه اللغة لبناء الأجسام الهندسية الخاصة بهم في ارشيكاد، وتوسيع إمكانيات التصميم والعرض.

احتياجات التصميم الخاص بك، والخلفية الخاصة بك في البرمجة ومعرفتك الهندسة الوصفية تؤثر حيث تبدأ في GDL.



<http://gdl.graphisoft.com/gdl-basics>

## Processing



لغة برمجة مفتوحة المصدر وهي تعتبر لغة برمجة كافية كما أنها بيئة تطوير متکاملة تم بناؤها من أجل الأشخاص المهتمين بالتصميم المرئي والفنون الإلكترونية الأخرى. هذه اللغة تهدف أيضا إلى تعليم مبادئ البرمجة بصيغة مرئية. تم إنشاء هذه اللغة من قبل كيسى ريس (Casey Reas) وبن فرای (Ben Fry) والذين كانوا سابقا في معهد ماساتشوستس للتكنولوجيا.

أحد أهم أهداف هذه اللغة هي مساعدة الغير-مبرجين على البدء بتعلم البرمجة وذلك من خلال مشاهدة النتائج بصيغة مرئية فورية. تم بناء هذه اللغة باستخدام القدرات المرئية للغة جافا وتطورها.

يتضمن مع لغة برمجة بروسينج بيئة تطوير بسيطة تدعى سكتش-بوك (بالإنجليزية: sketchbook) من ضمن الميزات بأنه يمكنك تصدير التطبيق المصمم بها إلى ثلاثة أنظمة تشغيل الويندوز واللينكس والماك بضغط الاختصار **Export Application** أو من **File** ثم **Ctrl+E**

<https://processing.org/>

## Xamarin



Xamarin هو بيئة عمل أو تطبيق أو سمه ما تشاء أصبح ضمن أحد تطبيقات Microsoft ويساعدك على إنشاء تطبيق على مختلف الأنظمة بلغة C# دون الحاجة لعناء تعلم مختلف اللغات لمختلف الأنظمة ، ويوفر لك كامل الأدوات والآليات التي تحتاجها في إنشاء تطبيق إحترافي والجدير بالذكر بأن التطبيق يكون تطبيق Native ( فطري حسب ترجمة قول ? ) والمقصود بها بأن التطبيق ينشأ كتطبيق حقيقي كأنك قمت ببرمجه بنفس لغة الأم لبرمجة التطبيق دون استخدام محاكيات خارجية أو غيره ، ليس كباقي الخيارات المتوفرة والتي من أشهرها PhoneGap والذي يجعلك تجهز تطبيق بـ HTML5 والتي من أكبر عيوبها أنك لن تستطيع الوصول لجميع موارد الجهاز والتي تقيدك كثيراً في بعض الأمور .

## BimQL

لغة البرمجة **BimQL** لغة مفتوحة المصدر تعمل على [BimServer.org](http://BimServer.org) لروية الملفات



[source code for the plugin on github](#)

## لغة البرمجة رينج Ring programming language



موقع اللغة <http://ring-lang.net> <http://ring-lang.sourceforge.net>

مصمم اللغة : المهندس محمود سمير فايد من مصر و مصمم ايضا تقنية البرمجة بدون كود PWCT من مواليـد ديسـمبر عام 1986 ، تخرـج عام 2008 من كلـية الـهندـسة الإـلـكتـرونـيـة قـسـم هـنـدـسـة وـلـوـمـ الـحـاسـب يـعـمل فـي مـجـال الـبرـمـجـة وـالـبـحـث الـعـلـمـي وـالـآن هو باـحـث فـي جـامـعـة الـمـلـك سـعـود .

سمعت عن البرمجة بدون كود في بدايتها و انبهرت بسهولتها و تعلمها و حصلت على شهادة من جامعة الملك سعود

فـلـما ظـهـرـت لـغـة الـرـينـج كان من الـطـبـيعـي ان اـبـدا فـي تـعـلـمـها وـجـدـتـها بـسـيـطـة وـمـفـيـدة وـقـرـيـبة من لـغـة الـإـنـسـان وـيمـكـنـ بها عـمـلـ بـرـامـجـ مـفـيـدةـ فـيـ مـجـالـ نـمـذـجـةـ مـعـلـومـاتـ الـبـنـاءـ الذـىـ اـهـتمـ بـهـ وـمـخـلـفـ الـمـجاـلـاتـ الـأـخـرـىـ ايـضـاـ.

من مزايا اللغة

### • لـغـة الـبرـمـجـة الـرـينـج هـيـ لـغـةـ مـخـصـصـةـ لـلـتـطـبـيقـاتـ General Applicationsـ عـامـةـ وـمـتـعـدـدـةـ الـأـغـرـاضـ Purposeـ

يمـكـنـ اـسـتـخـادـهـاـ لـتـطـوـيرـ التـطـبـيقـاتـ سـوـاءـ كـانـتـ Desktopـ اوـ Webـ اوـ Mobileـ وـهـيـ لـغـةـ جـديـدةـ ظـهـرـتـ فـيـ عـامـ 2016ـ

الـلـغـةـ مـبـتـكـرـةـ تـحـتـويـ عـلـىـ العـدـيدـ مـنـ الأـفـكـارـ الـجـديـدـةـ وـتـسـتـفـيدـ أـيـضـاـ فـيـ تصـمـيمـهـاـ مـنـ 9ـ لـغـاتـ بـرـمـجـةـ أـخـرـىـ سـبـقـتهاـ وـهـيـ تـشـبـهـ إـلـىـ حدـ كـبـيرـ لـغـاتـ مـثـلـ Pythonـ وـ Rubyـ إـلـاـ انـهـاـ صـمـمـتـ لـتـكـونـ اـكـثـرـ بـسـاطـةـ وـمـرـونـةـ .

تـدـعـمـ الـلـغـةـ الـعـدـيدـ مـنـ أـنـمـاطـ الـبـرـمـجـةـ مـثـلـ الـ Functionalـ وـالـ OOPـ وـالـ Proceduralـ وـالـ Reflectionـ Metaprogrammingـ

وـتـتـمـيزـ الـلـغـةـ عـنـ جـمـيعـ لـغـاتـ الـبـرـمـجـةـ الـأـخـرـىـ بـدـعـمـ مـتـمـيزـ لـلـ Natural Declarativeـ وـالـ Programmingـ

تـعـلـمـ الـلـغـةـ عـلـىـ مـخـلـفـ الـأـنـظـمـةـ مـثـلـ Windowsـ وـ Linuxـ وـ macOSـ وـ hـيـ مـجـانـيـةـ مـفـتوـحةـ المـصـدرـ Free Open Sourceـ

الـلـغـةـ تـحـتـ رـخـصـةـ MITـ مـاـ يـتـيحـ لـكـ اـسـتـخـادـهـاـ فـيـ عـلـىـ تـطـبـيقـاتـ مـجـانـيـةـ اوـ تـجـارـيـةـ مـنـ اـجـلـ الـرـيـحـ المـادـيـ اللـغـةـ تـعـتـمـدـ فـيـ طـرـيـقـةـ عـلـمـهـاـ عـلـىـ وـجـودـ مـتـرـجـ Compilerـ يـقـوـمـ بـتـرـجـمـةـ الـكـوـدـ الـخـاصـ بـالـلـغـةـ بـحـيـثـ يـتـمـ تـنـفـيـذـهـ مـنـ خـلـالـ آـلـةـ اـفـتـراـضـيةـ Virtual Machineـ

صـمـمـتـ مـنـ اـجـلـ الـلـغـةـ وـقـدـ تـمـ بـنـاءـ مـتـرـجـمـ الـلـغـةـ الـآـلـةـ الـاـفـتـراـضـيـةـ مـنـ خـلـالـ لـغـةـ Cـ

تـسـهـلـ عـلـيـكـ الـلـغـةـ تـرـجـمـةـ وـتـشـغـيلـ الـتـطـبـيقـاتـ مـبـاشـرـةـ حـيـثـ تـمـ دـمـجـ الـمـتـرـجـمـ وـالـآـلـةـ الـاـفـتـراـضـيـةـ مـعـاـ

- مع امكانية استخدام كل منهم على حده عند الحاجة لذلك
- اللغة صممت لتكون بسيطة وصغيرة الحجم ومرنة وسريعة وفقا لهذا الترتيب من الأولويات.
- تأتى اللغة مع الكثير من المكتبات التى تجعلها لغة عملية قابلة للاستخدام فى مختلف المجالات وتسهل عليك اللغة عمل مكتبات جديدة
- اما من خلال كتابتها باستخدام اللغة نفسها او من خلال دعم مكتبات كتبت مسبقا بلغات مثل C و C++
- بيئه العمل الخاصة باللغة (المحرر الخاص بها - مصمم النماذج - وغيرها الكثير من المكتبات والادوات)
- تم كتابتها باستخدام لغة الرينج Ring نفسها فهي لغة عملية وليس مجرد لغة اكاديمية او بحثية لتطبيق افكار جديدة فقط.
- صممت اللغة ايضا من اجل بناء الجيل الجديد من تقنية البرمجة بدون كود من خاللها.

و يشرفنا استضافة المهندس محمود  
بداية نشكرك على هذا العمل العظيم و نحب ان نسئلوك حضرتك

**ما الذي جذبك للبرمجة ؟**

البداية كانت حين اردت تلوين شاشة نظام الدوس MS-DOS السوداء الى اللون الازرق فعلمتهى والدى كيف اقوم بذلك من خلال برنامج بسيط بلغة Clipper ومن هنا بدأ الشغف بتعلم المزيد فحصلت على الكتب الخاصة باللغة من مكتبة والدى وبدأت يوميا اقوم بالقراءة والتطبيق العملي ... المتعة تكمن انك تطبق افكار وتنقلها من مجرد خيال الى شيء واقعي يمكن ان تراه وتتفاعل معه ... لمن لا يعرف الدوس هو نظام التشغيل الذى انتجته Microsoft قبل ال Windows ... ولمن لا يعرف كلينير ... هي لغة برمجة متخصصة فى انظمة قواعد البيانات كانت شائعة الاستخدام تحت نظام الدوس.

**ما هي أهمية البرمجة ؟**

تكمن قوة البرمجة فى اختصار الوقت والدقة وعدم تكرار افسنا والقدرة على التعامل مع كم كبير من البيانات والمعلومات وتحقيق اهدافنا بشكل أسرع ... تنوع مجالات البرمجة من الانظمة الى البرامج والتطبيقات والألعاب والادوات المختلفة يجعلها وسيلة فعالة للتغير شكل حياتنا فى مختلف المجالات ... بدأ هذا العلم (هندسة وعلوم الحاسوب) مستقىدا من مختلف المجالات والتخصصات ثم تحول الامر ليساهم فى تقدمها جميعا ويلعب دوره فى حياة كل إنسان تقريبا فاليوم لا يخلو من التعامل مع البرامج والتطبيقات بشكل مباشره او غير مباشرة من خلال الاستفادة من اثرها.

**لماذا الاولين سورس ؟؟**

لانه أفضل وسيلة لكي تعمل في المجال الذي تحب في الوقت الذي تحب مع الاشخاص الذين تختارهم ... هو فرصة لكل من يرغب في التعلم ... او يرغب في الابداع واظهار موهبته ... او يرغب في عمل شيء يحبه لانه يريد القيام به ... اري انه وسيلة للبرمجة والتطوير ترفع من كفاءة المبرمجين وتزيد من خبرتهم وتساهم في تطور المجتمع بشكل سريع.

**كيف نبدأ في تعلم لغة برمجة جديدة ؟**

ان تختارها اولا لهدف واضح مثل تعلم البرمجة او عمل مشروع ما او استكشاف عالم جديد ... ثم تختارها بشكل صحيح يتناسب مع هذا الهدف فاذا كنت تريد مثل عمل تطبيق Web فمن المنطقى ان تختار لغة مناسبة لهذا المجال ... اما اذا كنت تريد تعلم البرمجة لأول مرة فمن الافضل ان تبدأ بلغة سهلة وبسيطة تجعلك تحب المجال ... اما اذا كنت تريد استكشاف عالم جديد فمن الحكمة ان تختار لغة برمجة جديدة بالنسبة لك و تستخدم افكار و مفاهيم لم تتعامل معها من قبل.  
بعد تحديد الهدف والاختيار تتعلم اسasisيات البرمجة و تطبقها بشكل صحيح من خلال امثلة بسيطة حتى تستوعبها ... الهدف هنا ان تعرف المفاهيم الاساسية مثل التعامل مع البيانات Data Structure و جمل التحكم Control Structure و انماط البرمجة

المختلفة Programming Paradigm ثم بعد ذلك الاساسيات المتعلقة بالمجال الذى ستعمل فيه سواء كان Database او Games على سبيل المثال ... بعد تعلم اساسيات البرمجة بشكل عام واساسيات التخصص الذى ستختاره بشكل خاص ... تاتى مرحلة العمل على مشاريع كبيرة لكي تتعلم مختلف الجوانب المتعلقة باتمام اعمال كبيرة الحجم ... ثم اخيرا مرحلة ان تقوم بعمل مشاريع قام بطلبيها الاخرين منك ... وبالتالي البرمجة تجمع بين العلم والفن والتكنولوجيا وهي رحلة تعلم بلا نهاية حيث يمكن ان تزداد خبرة وتحصل على المزيد من القدرات من خلال القراءة والتطبيق المستمر والتفكير الابداعي.

### ما الخطوة القادمة لحضرتك ؟

حاليا وصلت بلغة الرينج الى مستوى جيد بدخولها الى اول 100 لغة برمجة على مستوى العالم لكن فى الواقع تلك مجرد بداية ولدينا طموح اكبر من ذلك بكثير مما سيترتب عليه المزيد من التطوير والعمل على تحديث اللغة بشكل مستمر والسعى نحو عمل قفزات فى تصميمها تجذب لها شريحة اكبر من المطورين وتزيد من قدرات اللغة بجانب ذلك اعمل على تطوير الجيل الجديد من تقنية البرمجة بدون كود باستخدام لغة الرينج وهو الان فى مرحلة متقدمة وشارف على الانتهاء وان شاء الله رب العالمين سيكون متاح لنا خلال عام 2018

### شرح لغة البرمجة رينج Ring programming language

يتوفر العديد من المصادر لتعلم اللغة بداية من المرجع الأساسي الخاص بها (1885 صفحة) باللغة الانجليزية

[https://sourceforge.net/projects/ring-lang/files/Ring%201.6/Fayed\\_RingDoc\\_1.6.pdf/download](https://sourceforge.net/projects/ring-lang/files/Ring%201.6/Fayed_RingDoc_1.6.pdf/download)

وانطلاقا الى مجموعة من المصادر باللغة العربية

(1) دروس تعليمية من تقديم م/سارة حمدي

<https://ringprogramming4arab.wordpress.com/%D8%A7%D9%84%D9%85%D8%AD%D8%AA%D9%88%D9%8A-2>

(2) دروس تعليمية من تقديم أ/أحمد حسونة

(3) دروس تعليمية من تقديم أ/ علاء محمد

<https://www.youtube.com/watch?v=uSv5rJp6snw&list=PLgc0fyJgivssosIJu4kJGk5avljaMzYHK>

(4) بالإضافة الى دروس تعليمية من تقديم م/ محمود فايد

<https://www.youtube.com/watch?v=UCoXewxgvbl&list=PLpQiqjcu7CuFc027iGHaBLPCZHuzCHkBC>

## مصادر لتعلم البرمجة

1. موقع لتعلم اساسيات البرمجة للاطفال و للمبتدئين من [mit](https://scratch.mit.edu) كما يعلم برمجة الجوال
2. موقع تعليمي تفاعلي لتعلم لغات برمجة مثل ruby & python <https://www.codecademy.com> دون الحاجة لتنصيب اي برنامج على جهازك
3. موقع مفتوح لتعلم البرمجة <https://code.org>
4. دوره البايثون للمهندس محمود عوف <http://ain-academy.com/KFS-ca/lecture.php?num=27>
5. موقع من مايكروسوفت لتعلم البرمجة مثل C# و الفيجوال بيسك <https://msdn.microsoft.com>
6. موقع [edx](#) به العديد من الدورات البرمجية القوية
7. dynamo لتعلم الـ ifc
8. ستجد ifc

  - [BIMserver](#) : <http://bimserver.org>
  - [ifcOpenShell](#) : <http://ifcopenshell.org>
  - [ifcPlusPlus](#): <http://www.ifcplusplus.com>
  - [IFC-SDK](#) : <http://www.osor.eu/projects/ifc-sdk>
  - [Open IFC Tools](#) : <http://www.openifctools.org>
  - [xBIM](#): <http://www.openbim.org>

9. لبرمجة الريفيت <http://revitapisearch.com>

هنا ستجد كل ما تحتاج لبرمجة ADDINS للريفيت [www.autodesk.com/developrevit](http://www.autodesk.com/developrevit)

## كلمات حول الكتاب

بعد دراسة لغة الاوتوليس من قبل الاستاذ محمد شوقي والاستاذ رضا فوزى اصبح كتاب الاوتوليس وبرشامة الاوتوليس الذين قمت بعملهما المرجع لدى عندما اقوم بعمل او تعديل اى ليسب فجزاك الله عنى وعن كل رسام يسترشد بما تفعله خير الجزاء (بسم الله الرحمن الرحيم ) و قل اعملوا فسيري الله عملكم و رسوله و المؤمنون

محمود فوزي

ابدا كلامي بشكرك على هذا المجهود المشكور من قبلك في كيفية شرح لغة الاوتوليس التي كانت من اللغات الثقيلة من زمان و لكنك من طريقة شرحك للاوتوليس و تمكنك من اللغة قدرت توصل الي ايسر طريقة لفهم اللغة و جزاك الله عنا كل خير الاستاذ حسين محمد رئيس قسم الرسم في مكتب افافير و سيفاك

### وآخر دعوانا أن الحمد لله رب العالمين

في الختام اتمنى ان اكون وقفت في وضع قدمك على اول الطريق لتعلم البرمجة ، و اذا وجدت خطأ فاعلمنى به ، او اصلاحه مشكورا

وما بها من خطأ ومن خلل اذنت في إصلاحه لمن فعل  
لكن بشرط العلم والإنصاف فذا وذا من أجمل الأوصاف  
والله يهدي سُبُّل السَّلَام سُبْحَانَهُ بِحَلْهِ اعتصامي