مقدمة لتعلم الAUTOLISP

هي احدى لغات البرمجة الخاصة بشركة Autodesk ، مثلها مثل ObjectARX ،. VBA و هي تعطي قوة للاتوكاد ، و تمكنك من الرسم بسرعة ، و اختصار الخطوات و توفير الوقت.
كل ما يمكنك فعله في الاتوكاد ، يمكنك بكل سهولة أن تفعله بالاتوليسب مزيدا لانتاجك .
Lisp is a programmable programming language.
و هي مشتقة من لغة lisp و هي احد لغات الذكاء الصناعي و التى هي من انتاج جون ماكرثي في الخمسينات ، و كان اول ظهور لل AUTOLISP في النسخة AUTOCAD 2.18 في شهر يناير 1986.
و هي اختصار

LISt Processing.

Or LISt Programming

و يقال انها اختصار

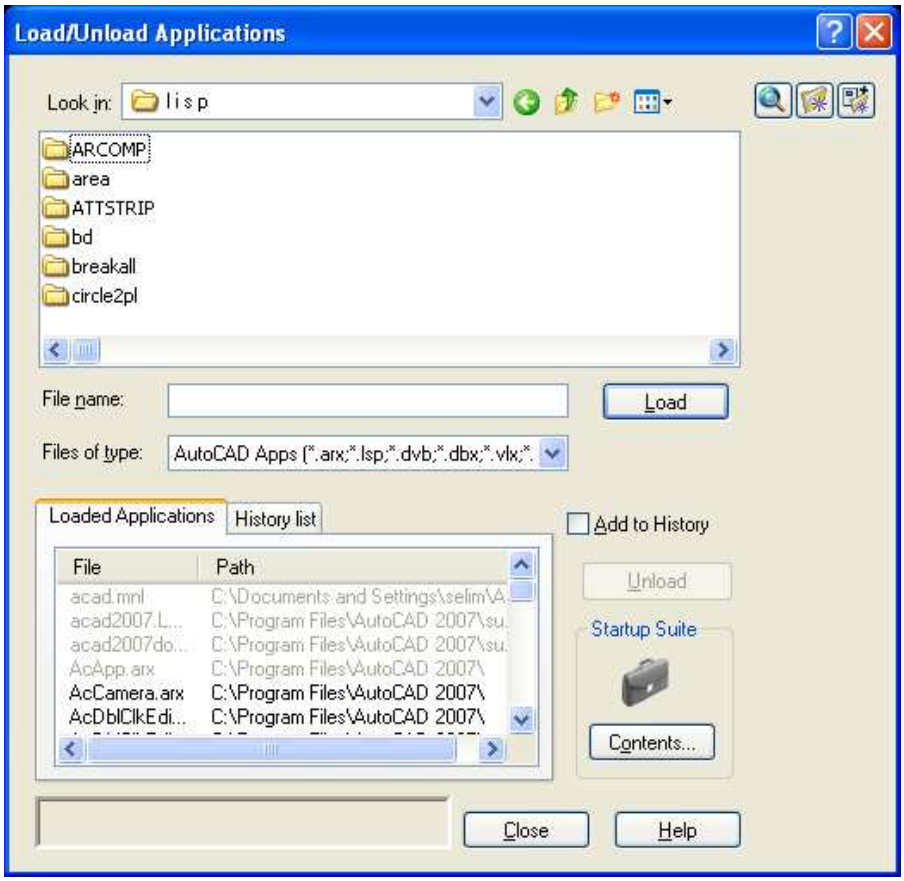
"Lots of Stupid Parentheses".

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

لتحميل الليسب

من قائمة TOOLS أختر

LOAD APPLICATION



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

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

CONTENTS

او اكتب aapload في سطر الاوامر

```
Command: appload
```

و لكتابه 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 و الصغيرة .
2) تكتب لغة الاتوليسب في اقواس و لا بد ان يكون عدد الاقواس اليمنى مساوي 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 الداله 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 محجوزة 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)
; error: bad function: 4
```

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

وصف للكود ;;;
مقدمة لمقطع في الليسب ;;
توضيح لجزء صغير ;

```
Command: (+ 4 2 ) ;comment
6
```

و الامتدادات لهذا الملف قد تكون:
Lsp الملف العادي و يمكن تحريرة بالمفكره العادية بالويندوز
mnl.*
Fas Vlx ملف ليسب محول للغة الاله
و اذا كان لدينا ثلاث ملفات بنفس الاسم بالامتدادات السابقة فان الاتوكاد يحمل vlx ثم fas ثم lsp

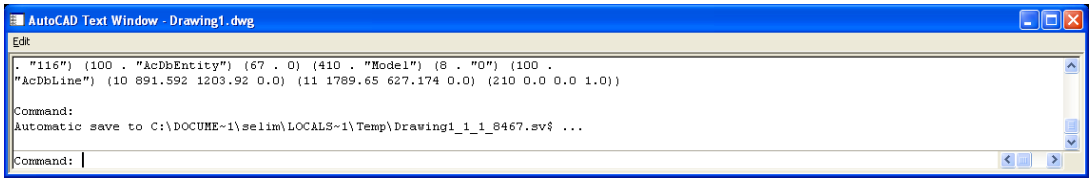
و لتشفير ملف الليسب (و ان كنت لا انصح بهذا . لجعل المنفعة عامة)
- من قائمة tools إفتح visual lisp editor
- في شاشة ال console اكتب الأمر التالي:
("vlisp-compile '1sa "d:/try.lsp" "d:/try.fas)
حيث المسار الأول الأول مسار الملف المرغوب تشفيره والثاني مسار الملف المشفر

اذا واجهتك مشاكل راسلني على ENGWEB@GMAIL.COM
او اكتبها في المدونة www.amrselim.net
او اشترك في القائمة البريدية http://groups.google.com/group/DRAFTSMAN

عمر سليم
رسام هندسي

البداية مع Autolist

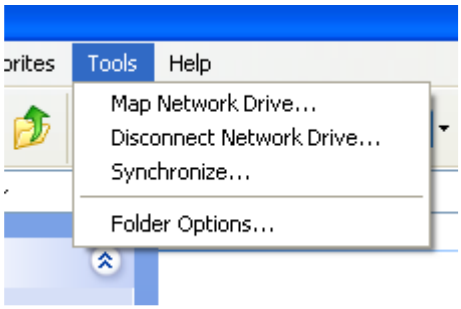
1)يمكن كتابة الاوامر مباشرة في سطر الاوامر الخاص بالاتوكاد command line
و يفهم الاتوكاد ان اي امر يبدأ بالاقواس () هو يتبع لغة الاتوليسب



2) او كتابة الاوامر في ملف txt مع تغير الامتداد من TXT الى lsp
و هو المعتاد ليسهل تبادلها و التعديل بها .

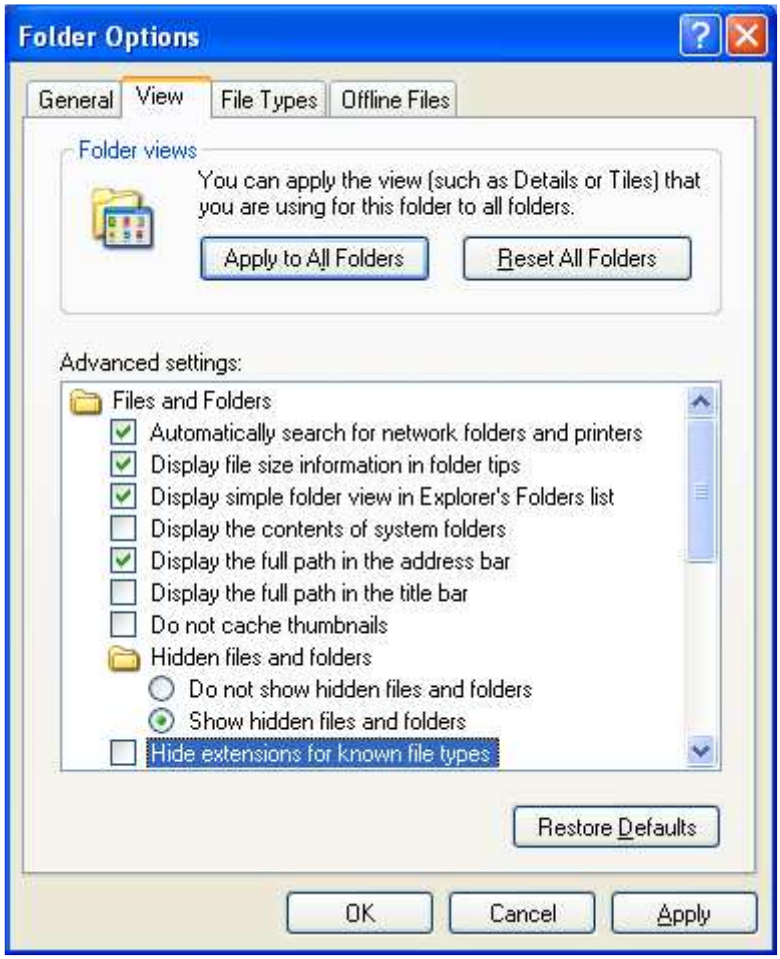
w.lsp

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



ثم نختار view

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



و الصيغة العامة

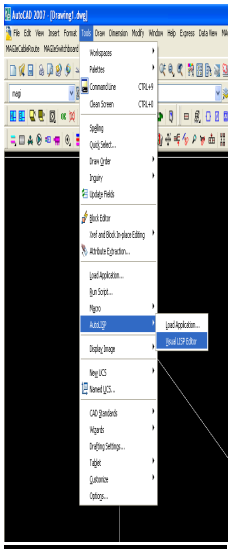
```
(defun c:اسم الداله(نكتب هنا المتغيرات الخاصه بالداله او يمكن تركها فارغه )
نكتب هنا الدوال و الاوامر
)
```

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

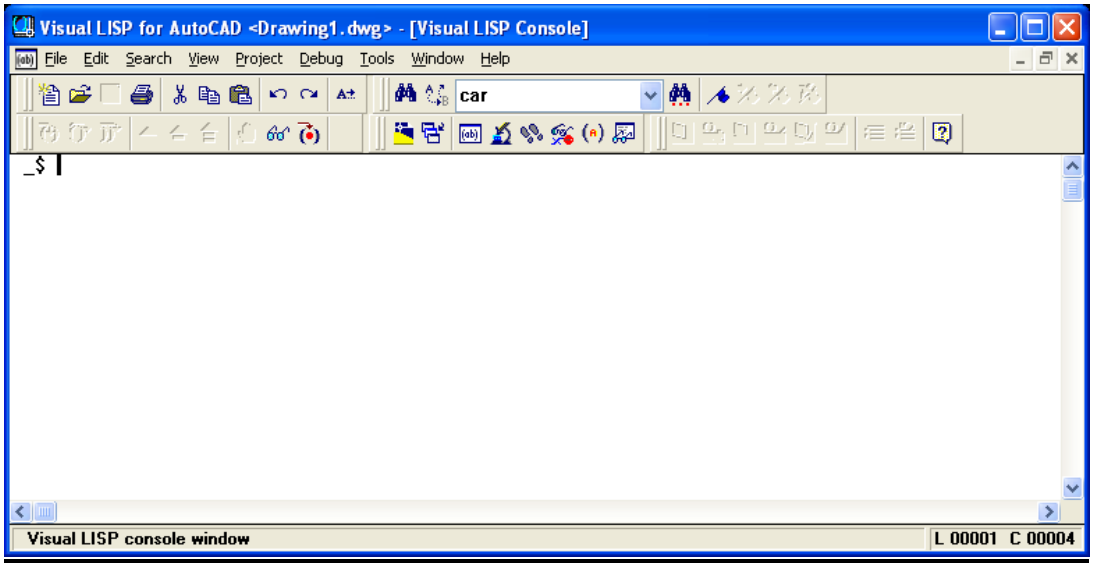
مثال بسيط نقوم فيه برسم مستطيل (سيتم شرح المثال لاحقا)

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

3) الدخول الى بيئه برمجة visual lisp editor
و تعطي قوة للتعديل و معالجة الاخطاء
من قائمة Tools أختر Autolisp ثم Visual Lisp Editor



تظهر امامك شاشة VISUAL LISPFOR AUTOCAD



الدوال الحسابية:

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

- الجمع +

المثال التالي نجمع فية الرقم الاول على الرقم الثاني .
مع العلم اننا (عكس اكثر لغات البرمجة) نكتب علامة الجمع اولا

```
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 ثلاث 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 ثلاث 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 ثلاث 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 الاتوليسب تجاهلت الكسور حيث ان القسمة كانت بين رقمين حقيقين integer
و عند القسمه على كسر فانه 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 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 رساله الخطأ التالية

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

و اهمية ترك مسافة بين الرقم و الداله و الا فسقرأها على انها كلمه واحده 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 لينج لنا 11

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

- دالة اضافة واحد

و هي داله مستقله تضيف رقم واحد الى العدد

```
Command: (1+ 6)
7
```

و عند كتابه رقم أخر غير الواحد تظهر لي رساله تقول انه لا توجد داله 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 انسب رقم يقبل عليه الرقمين القسمة عليه

```
Command: (gcd 100 15)
5
```

- الدالة zerop

للتأكد هل الناتج صفر ام لا .
في حاله الصفر يكون الناتج t
و في حاله غير هذا يكون الناتج nill

```
Command: (zerop 0)
T
```

```
Command: (zerop 9)
Nil
```

- الدالة abs

و تعطي القيمة المطلقة للرقم (تقلب السالب لموجب)

```
Command: (abs -4)
4
```

```
Command: (abs 4)
4
```

- الدالة exp

و تعطي القيمة الاسيه

```
Command: (exp 1)
2.71828
```

```
Command: (exp 4)
54.5982
```

و عند اعطاء البرنامج عدد من المتغيرات يخالف المتوقع تظهر لنا رساله 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 الاكبر

```
Command: (MAX 950 314)
950
```

- الدالة MIN

و هي تعطينا الرقم الاصغر

```
Command: (MIN 3000 1000)
1000
```

- الدالة REM

ايجاد REMINDER(المُذكر) للارقام

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 يتجاهل الكسور؟؟

```
Command: (/ 22 3)
7
```

تأمل المثال التالي عند استخدام الدالة FLOAT

```
Command: (/ (FLOAT 22) 3)
7.33333
```

- الدالة LOG

ايجاد اللوغريتم

```
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 رساله 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 اخري بالاستعانة بداله 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 نعطي 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 nillاذا كانت النتيجة خاطئه

- يساوي =

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

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

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

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

عند اعطائه ثلاث قيم فيشترط 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 اولا

```
Command: (< "e" "t")
```

```
T
```

- اصغر من او يساوي

```
Command: (<= 5 8)
T
```

- الدالة eq

السلسلتين متساويين ام لا
اذا كانا متساوين يعطينا t
غير متساويين 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 فنقارن بين الفرق بين a b و 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 المقتصة .

السلاسل 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 بعلامه '

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

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

```
Command: (car '())
nil
```

- **الدالة cadr**

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 '* '( 2 3 4 ))
24
```

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

- دالة APEND

و هي تضم السلاسل 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 اعطائها 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 و الارقام و الرموز

```
Command: (strlen "destory israil")
14
```

دوال التحويل

- دالة atof

لتحويل النص الى رقم حقيقي float
عند تخزين رقم مثل 314 على هيئه string فان العمليات الحسابية لا تؤثر عليه و لا يتجاوب 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:
```

للتحويل من الmodel الى layout

```
Command: (command "tilemode" 0)
```

للتحويل من الlayout 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 تريد تحميه" "رساله خطأ عند عدم وجود الملف")

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

(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 داله و اعطائها اسم للاستدعاء
و هي اختصار DEFIN FUNCTION تعريف داله

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

حيث n هنا متغير
و يوجد نوعين من المتغيرات
متغير يستخدم بواسطة الدالة
و متغير محلي يستخدم في الداله فقط و يتم سبقه بعلامة /

```
Command: (defun s (/n)(……..))
```

متغيرين احدهما محلي

```
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 الاتوكاد
السطر الثاني : رساله 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 للامر zoom, Extents
و تفيد هذه الدالة و امثالها في حاله نسيان المسافة بين اختصاري z , e

- الدالة setq

تحديد قيمة لمتغير ، و هي اختصار set-quote
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 لمعرفه المسافة بين نقطتين حيث نعطيه احدى النقطتين ثم نحدد الثانية بالماوس او ندخلها كتابه

```
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 اننا هنا نعطيها النقطتين مباشرة)

```
Command: (distance a b)
262.061
```

- الدالة getint

و تستخدم للحصول على رقم صحيح من المستخدم

```
(getint "\Enter 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 حرف ، اذا كتبت اكثر 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 \\ لتتمكن 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 مبتعدا عن النقطة على هيئه مستطيل (للفهم اكثر 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 الاسكي للحرف

```
Command: (ascii "b")
98
```

و في حاله عطائه كلمة يعطينا الاسكي لاول حرف فقط

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

- الدالة chr

اعطاء الحرف المقابل لرقم الاسكي

```
Command: (chr 97(
"a"
```

- الدالة atom

و تعطي 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 قيمه لمتغير الاتوكاد

مثل cmdecho الخاص بالاستجابات الخاصه بسطر الاوامر

```
(setvar "cmdecho" 1)
```

و filedia عرض مربعات الحوار التى 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)
```

Snapunit و هو متغير يتحكم في المسافة بين الجذب snap

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

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

```
Command: taskbar
Enter new value for Taskbar <1>: 0
```

- الدالة getvar

الحصول على قيمة متغير الاتوكاد

```
(if (< (getvar "filletrad") 1)
  (setvar "filletrad" 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 أخر عنصر

```
Command: (entlast)
<Entity name: 7efb40b0>
```

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

```
Command: (entlast)
nil
```

و للحصول على معلومات تفصيليه 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 model ام على layout
0 عندما تكون في 1 , model عندما تكون في layout
284 نظام التظليل

- الدالة ssget

و تقوم هذه الدالة بحفظ العناصر التى 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 يكمن فلتره العناصر التى نريدها مثل اختيار الخطوط فقط

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

- الدالة polar

يحدد نقطة بناء على نقطة اخرى و الزاوية التى نحددها 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 السلاسل 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 فيشترط ان يكون كلهم بهم قيمة ليعطيني t
اذا كان ليس به قيمه 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
```

- دالة arx

و تعطينا اسماء arx المحملة حاليا
AutoCAD Runtime Extension

```
Command: (arx)
("acapp.arx" "accamera.arx" "acdblclkedit.arx" "acdblclkeditpe.arx" "acdim.arx"
"acdwgrecovery.arx" "acdxnotify.arx" "aceplotx.arx" "acetlodr.arx"
"acetutil.arx" "acgsconfig.arx" "achapi17.dbx" "achlnkui.arx" "acidropmgr.arx"
"acismobj17.dbx" "acismui.arx" "aclayerp.arx" "acmgd.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

لالغاء 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

Command: (if (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 أنشأنا متغير يأخذ رقم من المستخدم
السطر الثاني اذا كان الرقم الذي ادخلة المستخدم اقل من عشرين اظهر رسالة "الرقم الذي ادخلته اصغر من عشرين" و اذا كان الرقم اكبر من عشرين اظهر رساله "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 عمليه 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 شئ ما حتى يحدث حدث معين
"استمر في سرقة الخزانة حتى يكتشفك المدير"
"استمر في الغياب حتى تلدغ اخر الشهر "

```
Command: (while (شرط)
 ((الاوامر التى 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 ترتب السلاسل الحرفية ترتيبا ابجدياً (في هذا المثال رتبت 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")
```

- دالةvl-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 لاستخدامة لاحقاً (اعطاء لون لطبقه مثلا)

```
Command: (SETQ Q(ACAD_COLORDLG 3)
```

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

(("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
```

، و في 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)
؛ لالغاء التحميل
(%unload_dialog Dcl_Id)
(princ)
defun c:MyFirst؛(
```

نقوم بتشغيل الليسب
MYFIRST
فيظهر امامنا

امثلة برمجية

1. تغير كل العناصر الى اللون الابيض

```
Command: ( command "_chprop" "all" "" "color" "7" "")
```

(command اعلام البرنامج اننا سنعطيه امر من اوامر الاتوكاد
"_chprop" امر تغير الخصائص
"all" اختيار تغير كل الخصائص

او يمكن كتابه last لتغير اللون لاخر عنصر فقط

"" ضغط زر الادخال
"color" اختيار تغير خاصية اللون
"7" اختيار تغير اللون الابيض
"") ضغط زر الادخال

2. تغير لون العناصر التى على طبقة معينة

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

(command اعلام البرنامج اننا سنعطيه امر من اوامر الاتوكاد
"layer" امر الطبقات
"color" اختيار تغير خاصية اللون
"7" اختيار تغير اللون الابيض
"hvac" اختيار الطبقة
"") 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 ركنيه
و "" بمعني 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 ركنيه

5. طريقة اخرى لرسم مستطيل

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

اي نقطة تحدد باحداثي 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 ثالثة لرسم مستطيل

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

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

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

(command اعلام البرنامج اننا سنعطيه امر من اوامر الاتوكاد
"circle" رسم دائرة
a نقطة المركز
200(نصف القطر

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 زووم للشاشة
(command "audit" "y") عمل audit
(setvar "bindtype" 1) تغير نوع البايند
(command "-xref" "b" "*") أمر bind
(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. تغير 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

الشرح	الكلمة
ارقام حقيقية مثل 2و2.3و-5	REAL OR FLOAT
رقم صحيحة مثل 5و6و3 بدون ارقام عشرية و هي من 32767 الى -32768	INTEGER
مجموعة من الارقام المسلسلة ذات ترتيب معين (6 5 2 14 8)	LIST
نص "غزة رمز العزة"	string
عنصر مثل point	symbol

برامج مساعدة

- DCL&Lisp Generator برنامج متميز لعمل lisp و dialog component defined) dcl) بطريقة سهلة و قويه و واجهه مرئيه تشبه الفيجوال بيسك لكنه ليس مجاني

http://www.beowolftech.com/lispgen/home.html

اهم المواقع لتعلم AUTOLISP

- www.amrselim.net
- http://www.afralisp.net/
- http://www.cadtutor.net
- http://www.jefferypsanders.com/autolisp.html

أشهر LISP

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

ملخص للأوامر

لتحويل النص الى رقم حقيقي	atof	تعريف الدالة	defun c:اسم الدالة()
لتحويل النص الى رقم طبيعي	atoi	للتأكد هل الناتج صفر ام لا	zerop
لتحويل الرقم الى نص	rtos	تعطي القيمة الاسيه	exp
التحويل من وحدة الى وحدة اخرى	cvunit	متمم العدد	~
تعطينا نوع المتغير	TYPE	الرقم الاكبر	MAX
اظهار رساله على سطر الاوامر	princ	الرقم الاصغر	MIN
اظهار رساله للمستخدم	ALERT	المتبقي من القسمة	REM
لاعطاء اوامر الاتوكاد	COMMAND	تقريب لاقرب عدد صحيح و الغاء الكسور	FIX
لتحميل ملف	LOAD	تحول العدد الى كسر	FLOAT
تحديد قيمة لمتغير	setq	ايجاد اللوغريتم	LOG
لتحديد نقطة	getpoint	الجذر التربيعي	SQRT
يستخدم لمعرفه المسافة بين نقطتين	getdist	جا الزاوية	sin
يستخدم لمعرفه المسافة بين نقطتين	distance	جتا الزاوية	cos
للحصول على رقم صحيح من المستخدم	getint	ظل الزاوية	atan
للحصول على اي رقم صحيح او كسر	getreal	الزاوية	angle
للحصول على نص من المستخدم	getstring	الرقم المعطى سالب ام لا	minusp
اعطاء رقم الاسكي للحرف	ascii	نقطة تلاقي اربع نقط	inters
اعطاء الحرف المقابل لرقم الاسكي	chr	لا يساوي	/=
و تعطي nil اذا كان سلسله و t اذا لم يكن	atom	السلسلتين متساويين ام لا	eq
تخبرني بالقيمة المسجلة في المتغير	eval	مقارنة بين سلسلتين	equal
تجعل الحروف كبيرة	quote	لاظهار عدد معين من الاحرف من سلسله نصية	SUBSTR
اعطاء قيمه لمتغير الاتوكاد	setvar	تضم جملتين في جملة واحدة	strcat
الحصول على قيمة متغير الاتوكاد	getvar	عرض اول رقم في السلسلة	car
تعطينا بيانات عن أخر عنصر	entlast	عرض ثاني رقم في السلسلة	cadr
بحفظ العناصر التى يحددها المستخدم	ssget	عرض ثالث رقم في السلسلة	caddr
يحدد نقطة بناء على نقطة اخرى و الزاوية	polar	عرض السلسلة بدون الرقم الاول	cdr
المتغير او المعطى به قيمة ام لا	AND	عرض عنصر معين من السلسلة الرقمية	nth
و تعطينا اسماء arx المحملة حاليا	arx	اضافة عنصر لاول القائمة	cons
لتحميل ملف arxload	**arxload**	تنفيذ دالة معينة على السلسلة الحرفية	apply
لالغاء تحميل ملف arx	arxunload	تضم السلاسل الرقمية في سلسلة واحدة	append
هل يحتوي العنصر على قيمة ام لا	boundp	عكس السلسلة الرقمية	reverse
لوضع شرط على حدث	if	هل العنصر المعطى سلسلة رقمية ام لا	listp
تكرار عمليه معينة عدد معين من المرات	Repeat	عدد حروف الجملة	strlen
افعل شئ ما حتى يحدث حدث معين	While	للشرح : حمل كتاب شرح الاتوليسب من	
و هي ترتب السلاسل الحرفية ترتيبا ابجدياً	acad_strlsort	www.amrselim.net	

في الختام اتمنى ان اكون وفقت في وضع قدمك على اول الطريق لتعلم لغة AUTOLISP ، و اذا وجدت خطأ فاعلمنى به ، او اصلحه مشكورا

وما بها من خطأ ومن خلل أذنت في إصلاحه لمن فعل

لكن بشرط العلم والإنصاف فذا وذا من أجمل الأوصاف

والله يهدي سُبل السلامِ سبحانه بحبله اعتصامي

كلمات حول الكتاب

بعد دراسة لغة الاوتوليسب من قبل الاستاذ محمد شوقى والاستاذ رضا فوزى
اصبح كتاب الاوتوليسب وبرشامة الاوتوليسب الذين قمت بعملهماهما 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 فسيرى الله عملكم و رسوله و المؤمنون

محمود فوزي

ابدا كلامي بشكرك علي هذا المجهود المشكور من قبلك في كيفية شرح لغة الاوتوليسب
التي كانت من اللغات الثقيلة من زمان و لكنك من طريقة شرحك للاوتوليسب 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 انفايرو سيفك

و أخر دعوانا أن الحمد لله رب العالمين

اتمنى ان يكون الكتاب قد نال رضاك

لمزيد من المعلومات عن كاتب هذا الكتاب و للتواصل زر الصفحة التالية
/http://newmilk.wordpress.com/about

كتبه :
عمر سليم