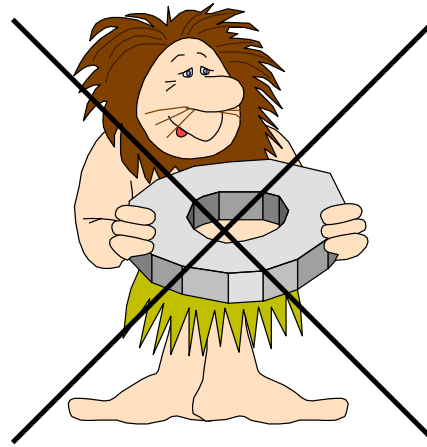
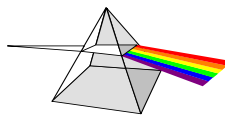


# AutoCAD

*Plus jamais ça !*



# Programmation AutoLISP de BASE



**Christian REB CAO/DAO Services**

1 rue des Fleurs / 68230 ZIMMERBACH

☎ 03 89 58 98 16 / ☎ 03 89 58 40 20

✉ reb.c@wanadoo.fr

Web: perso.wanadoo.fr / rebcao

Conseil - Formation - Développement

Votre Spécialiste AutoCAD

© Christian REB

# **Programme du module AutoLISP programmation de Base**

## **1. Le langage de programmation AutoLISP**

Introduction  
Présentation

## **2. Les fonctions AutoLISP en mode commande**

Les opérateurs de base  
Initialiser une variable  
Rappel, utilisation d'une variable

## **3. Quelques fonctions AutoLISP élémentaires**

Fonctions de création de liste  
Fonctions de manipulation de liste  
Fonctions d'extraction de liste

.../...

## **4. Exercice N° 1 : construire un rectangle**

Analyse du problème  
Déclaration et définition des éléments  
Les fonctions de saisie  
Création d'un fichier programme  
L'éditeur Visual LISP  
Chargement et test du programme AutoLISP  
Créer une nouvelle commande dans AutoCAD  
Calcul des points et construction du rectangle  
Amélioration du programme

## **5. Exercice N° 2 : construire une pièce paramétrée**

Présentation et analyse du sujet  
Déclaration et définition des éléments  
Création du fichier programme  
Chargement et test du programme AutoLISP  
Calcul des points  
Construction automatique de la pièce  
Amélioration du programme



# **Programme du module AutoLISP programmation de Base**

## **1. Le langage de programmation AutoLISP**

Introduction  
Présentation

## **2. Les fonctions AutoLISP en mode commande**

Les opérateurs de base  
Initialiser une variable  
Rappel, utilisation d'une variable

## **3. Quelques fonctions AutoLISP élémentaires**

Fonctions de création de liste  
Fonctions de manipulation de liste  
Fonctions d'extraction de liste

.../...

## **4. Exercice N° 1 : construire un rectangle**

Analyse du problème  
Déclaration et définition des éléments  
Les fonctions de saisie  
Création d'un fichier programme  
L'éditeur Visual LISP  
Chargement et test du programme AutoLISP  
Créer une nouvelle commande dans AutoCAD  
Calcul des points et construction du rectangle  
Amélioration du programme

## **5. Exercice N° 2 : construire une pièce paramétrée**

Présentation et analyse du sujet  
Déclaration et définition des éléments  
Création du fichier programme  
Chargement et test du programme AutoLISP  
Calcul des points  
Construction automatique de la pièce  
Amélioration du programme





# Langage de programmation AutoLISP : Présentation

Ce langage est une implémentation du langage LISP (XLISP mis au point par David BETZ), il est intégré au logiciel AutoCAD.

LISP est le langage de référence utilisé pour la recherche et le développement dans les domaines de l'intelligence artificielle et des systèmes experts

AutoLISP permet aux utilisateurs et aux concepteurs d'AutoCAD d'écrire des MACROS et des fonctions dans un langage évolué très puissant adapté aux applications graphiques, sans aucun ajout ou coût supplémentaire (hors mis la formation).

AutoLISP est facile à apprendre et flexible

Le fait de pouvoir créer des macros va accroître votre productivité.

Avec AutoCAD on dispose également de la possibilité de programmation en Langage C, mécanisme appelé ADS (AutoCAD Development System). AUTODESK s'engage à prendre en charge AutoLISP à long terme. ADS est proposé comme alternative. ADS est utilisé par les sociétés qui développent des applicatifs métier AutoCAD.

Pour approfondir vos connaissances AutoLISP, vous avez la possibilité d'acheter des ouvrages comme : LISP de WINSTON et HORN, LOOKING at LISP, AutoLISP et personnalisation des menus, Editions Reynald GOULET, Les Editions SYBEX.





# Langage de programmation AutoLISP : Introduction

La mise en oeuvre d'AutoLISP est très simple, soit on utilise des fonctions directement au clavier "en mode commande", soit on regroupe ces macros dans un fichier avec un simple éditeur de texte (Bloc notes, Wordpad), mieux en utilisant l'éditeur Visual LISP à partir de la version 2000, ou encore en personnalisant des icônes dans les barres d'outils.

Chaque fonction est encadrée par des parenthèses.

Si le langage est effectivement simple à mettre en oeuvre, il est assez contraignant dans les recherches d'erreurs (Debugage), simplifié aujourd'hui grâce à l'éditeur Visual LISP.

Les programmes seront interprétés par AutoCAD, ce qui a pour avantage une compatibilité "totale" entre les différentes version AutoCAD. En effet, on peut envisager d'utiliser encore aujourd'hui des programmes AutoLISP utilisés avec une version 12, par exemple. Et inversement, il serait possible d'utiliser un programme LISP de la version 2004, du moment que l'on ne fait pas appel à des commandes ou des fonctions inexistantes dans la version 12.

La mise à disposition de ces nouvelles fonctions (commandes) peut être effectuée manuellement (chargement à la demande) ou automatiquement (toujours disponible dans tous les dessins).





## Fonctions AutoLISP en mode commande : Les Opérateurs de BASE

Il existe plusieurs fonctions destinées aux opérations courantes de calcul de test. Voici quelques exemples :

( * 10 3 ... )	retourne <b>30</b> , c'est le <b>produit</b> de nombre entier ou réel
( + 21 2 3 5.0 ... )	retourne <b>31.0</b> , c'est la <b>somme</b> de nombre réel ou entier
( 1+ 7 )	retourne <b>8</b> , c'est l' <b>incrément +1</b> du nombre unique entier ou réel qui suit
( 1- 6 )	retourne <b>5</b> , c'est l' <b>incrément-1</b> du nombre unique entier ou réel qui suit
( / 10 3 ... )	retourne <b>3</b> , c'est la <b>division</b> d'un entier par un entier retourne un entier
( / 10 3.0 ... )	retourne <b>3.33333</b> , c'est la <b>division</b> d'un entier par un réel retourne un réel
( = 10 3 ... )	retourne <b>nil</b> , c'est la relation <b>égal à...</b> ici cette relation est FAUSSE
( = 2.0 2 ... )	retourne <b>T</b> , c'est la relation <b>égal à...</b> cette fois ci VRAI
( /= 5 4 ... )	retourne <b>T</b> , c'est la relation <b>différent de...</b> ou non égal à...
( < 10 3 ... )	retourne <b>nil</b> , c'est la relation <b>strictement inférieur à...</b>
( <= 5 12 ... )	retourne <b>T</b> , c'est la relation <b>inférieur ou égal à...</b>
( > 10 3 ... )	retourne <b>T</b> , c'est la relation <b>strictement supérieur à...</b>
( >= 5 12 ... )	retourne <b>nil</b> , c'est la relation <b>supérieur ou égal à...</b>



AutoLISP retourne souvent certains messages, et tout particulièrement : **nil** qui veut dire FAUX ou RIEN  
**T (True)** qui veut dire VRAI





## Fonctions AutoLISP en mode commande : Initialiser une variable

La fonction **SETQ** est une des fonctions de BASE d'AutoLISP. Elle permet l'assignation de valeur à un nom d'élément (appelé aussi variable). Il est possible de manipuler n'importe quel type d'élément AutoLISP tel que : fonction, réel, entier, chaîne alphanumérique, résultat d'opération, etc. Voici quelques exemples :

( <b>setq</b> <b>N1</b> <b>3</b> )	retourne <b>3</b> , c'est le dernier élément évalué
( <b>setq</b> <b>N1</b> <b>3</b> <b>N2</b> <b>7</b> )	retourne <b>7</b> , c'est le dernier élément évalué
( <b>setq</b> <b>N2</b> ( <b>*</b> <b>12</b> <b>11.25</b> ) )	retourne <b>135.0</b>
( <b>setq</b> <b>N1</b> ( <b>+</b> <b>N1</b> <b>7</b> ) )	retourne <b>10.0</b>
( <b>setq</b> <b>N1</b> ( <b>/</b> <b>10</b> <b>3.0</b> ) )	retourne <b>3.33333</b>
( <b>setq</b> <b>N2</b> ( <b>*</b> <b>N1</b> <b>3</b> ) )	retourne <b>10.0</b> (les calculs ont une précision de <b>10<sup>-15</sup></b> )
( <b>setq</b> <b>FM</b> <b>*</b> )	retourne <b>&lt;Subr: #1490fd6&gt;</b> , assigne <b>FM</b> à une fonction
( <b>setq</b> <b>PL1</b> "etage 1 <del>de</del> ctricite <del>non</del> phase")	retourne l'ensemble de la chaîne de caractère
( <b>setq</b> <b>PL2</b> "etage 1 <del>re</del> seau informatique <del>tele</del> com")	retourne l'ensemble de la chaîne de caractère



Attention à l'imbrication des parenthèses. Il faut autant de parenthèses ouvrantes que fermantes.  
C'est la cause la plus fréquente d'erreur dans les programmes





## Fonctions AutoLISP en mode commande : Rappeler une variable

Toutes ces variables peuvent être utilisées dans les commandes d'AutoCAD en ajoutant devant le nom de la variable : **!**  
Il est possible également d'afficher la valeur de cette variable sous forme de texte AutoCAD. Cette possibilité ne concerne que des variables de type réel, entier ou alphanumérique. Voici quelques exemples :

*Commande :* **CERCLE**, placer le centre, puis à rayon taper : **!N1** AutoCAD dessine un cercle de rayon **N1**

*Commande :* - **CALQUE** (au clavier) option Etablir, puis taper : **!PL1** AutoCAD crée et rend courant le calque **PL1**  
idem avec **!PL2**

*Commande :* -**TEXTE** (texte simple), à "Texte :", taper : **!PL1** AutoCAD retourne le même texte **!PL1**  
pour que le texte évalue la variable **PL1**, il faut activer une commande système **TEXTEVAL 1**  
cette fois, AutoCAD renvoie le texte assigné à la variable **PL1**



Attention le caractère **!** ne sera utilisé devant une variable AutoLISP qu'en mode commande clavier, ou macro jamais dans un programme.







## Fonctions AutoLISP élémentaires : Principe

En programmation AutoLISP le plus petit élément est appelé un **ATOME** (par exemple : N1, N2, PL1)

Quelque soit la fonction utilisée l'AutoLISP cherche toujours à interpréter les instructions qui lui sont fournies.

Un élément n'ayant pas encore été défini sera considéré à **nil** pour AutoLISP (valeur FAUSSE)

Le langage AutoLISP est une succession de manipulation de LISTES de fonctions et d'arguments (simple ou imbriquées). Ces listes sont toujours entre parenthèses. Voici des exemples que AutoLISP peut retourner :

( setq    X       M    )	liste contenant une fonction (SETQ) et <b>M</b> à assigner à la variable <b>X</b>
( 215    31    45    )	format AutoLISP d'un point AutoCAD en X,Y,Z
( "boite" "cylindre" "sphère" )	Liste de chaîne de caractères
( a1 b2 c1 d2 e1 f1 g2 )	Liste de variables AutoLISP
( pt ( ( x1 y1 z1 ) ( x2 y2 z2 ) ( x3 y3 z3 ) ) )	liste imbriquée avec d'autres listes



**Attention au nombre de parenthèses pour la dernière liste. Il faut autant de parenthèses fermantes que d'ouvrantes.**





## Fonctions AutoLISP élémentaires : Création de LISTES

Les fonctions permettant la création des listes AutoLISP sont :

**SETQ**      **assigne une variable avec une valeur**

**LIST**      **regroupe les éléments entre parenthèses en une liste**

Exemples :

( **setq** L1 ' ( a b c d e ( f g ) h ) )

le **QUOTE** ' permet de créer la liste sans évaluer les arguments  
retourne ( a b c d e ( f g ) h )

( **list** N1 PL1 "axe" "mecanique" XY )

retourne ( 3.3333 "etage 1 dectricite ~~non~~ phase" "axe" "mécanique" nil )

( **setq** L1 (list N1 PL1 "axe" "mécanique" XY) )

assigne à L1 le contenu de la liste générée par la fonction **LIST** (L1 est une liste)  
retourne ( 3.3333 "etage 1 dectricite" "axe" "mécanique" nil )

( **setq** L2 (list L1 "archi" "topo") )

assigne à L2 le contenu de la liste générée par la fonction **LIST**  
retourne ( ( 3.3333 "etage- 1 dectricite" "axe" "mécanique" nil ) "archi" "topo" )



**Un élément non initialisé retournera  
toujours nil**





## Fonctions AutoLISP élémentaires : Création de LISTES

La fonction permettant d'ajouter des éléments dans une liste, est :

**CONS** créer une liste et/ou une paire pointée et ajouter les éléments entre parenthèses dans une liste (au début de la liste).

Exemples :

( setq L2 ( cons "Bordure\_route" L2 ) )      retourne ( "Bordure\_route" ( 3.3333 "etage 1 dectricite" "axe" "mécanique" nil ) "archi" "topo" )

( setq L3 ( cons 1 "Bordure\_route" ) )      retourne ( 1 . "Bordure\_route" ) crée une paire pointée

( setq L3 ( cons ( cons 2 "Axe\_route" ) L3 ) )      retourne une liste contenant une nouvelle paire pointée  
( ( 2 . "Bordure\_route" ) ( 1 . "Bordure\_route" ) )

Un point AutoCAD est toujours représenté sous forme d'une liste de 2 ou 3 éléments ( X Y Z ), par exemple :

( setq PT ( list 125.23 45.69 612.35 ) )      retourne ( 125.23 45.69 612.35 )

( cons PT L2 )      retourne ( ( 125.23 45.69 612.35 ) "Bordure\_route" ( 3.3333 "etage 1 dectricite" "axe" "mécanique" nil ) "archi" "topo" )

*Pour le dernier exemple, comme SETQ n'est pas utilisé la valeur retournée par AutoLISP ne sera pas conservée.*





# Fonctions AutoLISP élémentaires : Manipulation et extraction des LISTEs

La manipulation des listes est la clé du langage AutoLISP. Les fonctions de manipulations et d'extraction de listes sont :

<b><i>SETQ</i></b>	<i>assigner une valeur</i>
<b>CAR</b>	extraction du premier élément de la liste
<b>CDR</b>	retourne le contenu d'une liste sans le premier élément
<b>CADR</b>	extraction du deuxième élément d'une liste
<b>CADDR</b>	extraction du troisième élément d'une liste
<b>CADDDR</b>	extraction du quatrième élément d'une liste
<b>REVERSE</b>	renverse l'ordre d'une liste

Pour illustrer ces fonctions nous utiliserons la liste du point **PT ( 125.23 45.69 612.35 )**





# Fonctions AutoLISP élémentaires : Manipulation et extraction des LISTEs

soit la liste du point : **PT** ( **125.23** **45.69** **612.35** ) *qui correspond à ( X Y Z )*

Extraction et assignation de la **valeur X** (1er élt.) : ( **setq** **X** ( **car** **PT** ) ) retourne **125.23**

Obtenir la liste **PT** **sans le premier élément X** : ( **cdr** **PT** ) retourne ( **45.69** **612.35** )

Extraction et assignation de la **valeur Y** (2e élt.) : ( **setq** **Y** ( **cadr** **PT** ) ) retourne **45.69**

Extraction et assignation de la **valeur Z** (3e élt.) : ( **setq** **Z** ( **caddr** **PT** ) ) retourne **612.35**

Extraction et assignation du **4e élément** : ( **setq** **N** ( **caddr** **PT** ) ) retourne **nil** ( fin de liste )

Extraction et assignation du **1er élt de la liste L2** : ( **setq** **E1** ( **car** **L2** ) ) retourne **"Bordure\_route"**

**Redéfinir la liste L2** en enlevant le 1er élément : ( **setq** **L2** ( **cdr** **L2** ) ) retourne ( ( **3.3333** **"etage 1 detricite"** **"axe""mécanique"** **nil** ) **"archi"** **"topo"** )

**Renverser l'ordre de la liste L2** : ( **setq** **L2** ( **reverse** **L2** ) ) retourne ( **"topo"** **"archi"** ( **3.3333** **"etage 1 detricite""axe""mécanique"** **nil** ) )





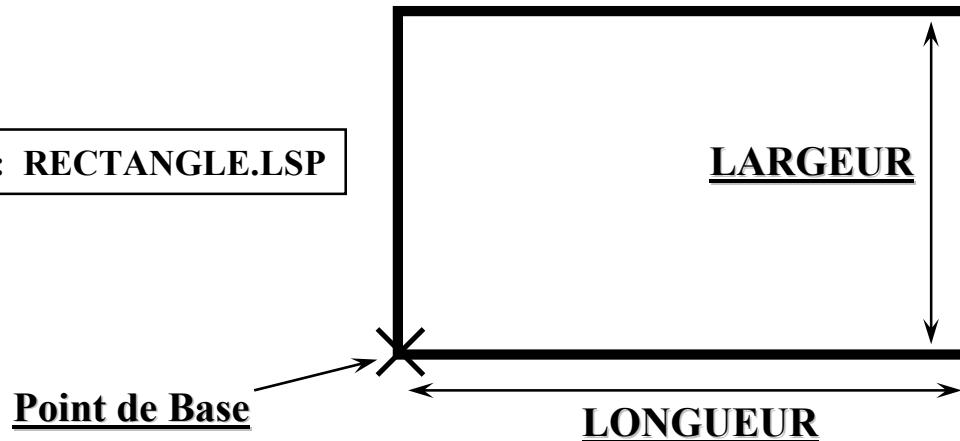
## AutoLISP travaux dirigés, 1er Exercice : Construction d'un rectangle

L'exercice consiste à créer une fonction pour **construire un rectangle** en indiquant la **longueur** puis la **hauteur**.

L'analyse sommaire du problème me permet d'identifier les objectifs suivants :

1. Je veux créer un fichier programme (AutoLISP)
2. Je veux positionner le rectangle en donnant un point à l'écran
3. Je veux définir sa longueur au clavier ou à l'écran
4. Je veux définir sa largeur au clavier ou à l'écran
5. Je ne veux pas changer sa rotation (rectangle orthogonal au SCU courant)

Nom du programme : RECTANGLE.LSP



Quelque soit le langage de programmation, il est recommandé de commenter au maximum le fichier programme. Commentaire d'ordre informel, mais également des dessins sommaires représentant les points clés (point, position, variables)

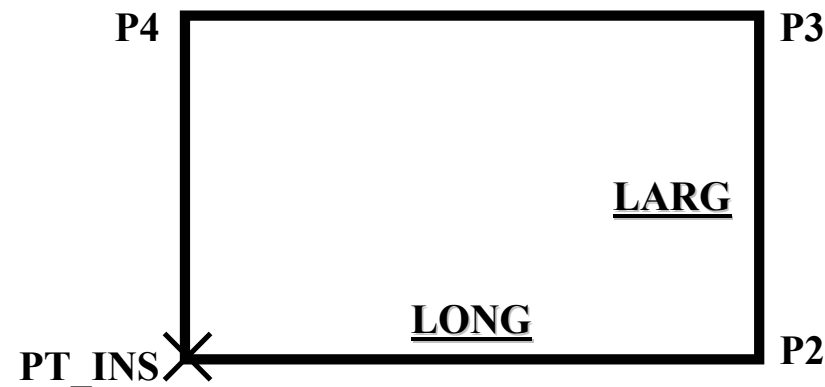




## AutoLISP travaux dirigés : Définition des variables

Avant toute programmation, il faut définir les éléments de base pour le programme, en particulier le nom du programme, les variables, etc. Exemple :

<b>RECTANGL.LSP</b>	<b>nom du programme AutoLISP. L'extension normale est LSP, mais AutoLISP accepte n'importe quelle extension qui devra être précisée lors du chargement du programme</b>
<b>PT_INS</b>	<b>point de base du rectangle (toujours le coin bas gauche)</b>
<b>LONG</b>	<b>longueur du rectangle</b>
<b>LARG</b>	<b>hauteur du rectangle</b>
<b>P2</b>	<b>Point bas droit du rectangle</b>
<b>P3</b>	<b>point haut droit du rectangle</b>
<b>P4</b>	<b>point haut gauche du rectangle</b>





## AutoLISP travaux dirigés : Fonctions de saisie

Les fonctions de saisie permettent aux programmes AutoLISP de dialoguer avec l'utilisateur. Ces fonctions sont destinées à récupérer une réponse de l'utilisateur, puis de l'assigner à une variable (avec SETQ). Ces fonctions marquent généralement une pause. Il s'agit des fonctions suivantes :

<b>GETANGLE</b>	retourne la valeur de l'angle indiqué au clavier ou à l'écran en <b>RADIANS</b>
<b>GETDIST</b>	retourne la distance indiquée au clavier ou entre deux points à l'écran
<b>GETINT</b>	attend une valeur entière comprise entre +32 767 et- 32768
<b>GETREAL</b>	attend une valeur réelle
<b>GETPOINT</b>	attend la saisie d'un point au clavier ou à l'écran
<b>GETSTRING</b>	attend l'entrée d'une chaîne de caractère (voir N.B. ci dessous)

*(voir manuel AutoLISP pour d'autres fonctions GET...)*

Chaque **GET...** permet d'afficher un message à l'écran. Ce message entre guillemets peut utiliser des caractères de contrôle précédés par \ comme :

<b>\n</b>	saut de ligne à l'écran
<b>\t</b>	tabulation (décale de 8 espaces à droite)
<b>\r</b>	affiche sur la même ligne
<b>\\</b>	retourne le caractère \
<b>\"</b>	retourne le " et le mémorise dans la chaîne de caractères
<b>\nnn</b>	retourne le caractère dont le code OCTAL est nnn

Il est possible de faire précéder un **GET...** par la fonction de préparation **INITGET** (**INITGET 2**). Cette fonction permet de verrouiller les réponses à l'aide de codes : **1**=pas d'entrée nulle / **2**=interdit valeur Zéro / **4**=interdit valeur négative / **8**=pas de contrôle des limites / **16**=non utilisé / **32**=trace les lignes et boîtes élastiques / **64**=interdit l'entrée Z / **128**=aléatoire.







## Exemple de table ASCII

Chaque caractères utilisés dans les applications DOS, WINDOWS, UNIX utilisent des tables de correspondances. La table la plus utilisée dans le monde PC est appelée table ASCII (Américan National Standard Code for Information Interchange). Elle est limitée à 255 codes (codifiés sur 8 bits). Voici la correspondance des codes de 32 à 175 les plus utilisés :

Déc.	Car.	Octal	56	8	70	81	Q	121	106	j	152	131	â	203	156	£	234
32	<espace>	40	57	9	71	82	R	122	107	k	153	132	ä	204	157	¥	235
33	!	41	58	:	72	83	S	123	108	l	154	133	à	205	158	P	236
34	"	42	59	;	73	84	T	124	109	m	155	134	å	206	159	f	237
35	#	43	60	<	74	85	U	125	110	n	156	135	ç	207	160	á	240
36	\$	44	61	=	75	86	V	126	111	o	157	136	ê	210	161	í	241
37	%	45	62	>	76	87	W	127	112	p	160	137	ë	211	162	ó	242
38	&	46	63	?	77	88	X	130	113	q	161	138	è	212	163	ú	243
39	'	47	64	@	100	89	Y	131	114	r	162	139	ï	213	164	ñ	244
40	(	50	65	A	101	90	Z	132	115	s	163	140	î	214	165	Ñ	245
41	)	51	66	B	102	91	[	133	116	t	164	141	ì	215	166	ª	246
42	*	52	67	C	103	92	\	134	117	u	165	142	Ä	216	167	º	247
43	+	53	68	D	104	93	]	135	118	v	166	143	Å	217	168	¿	250
44	,	54	69	E	105	94	^	136	119	w	167	144	É	220	169	—	251
45	-	55	70	F	106	95	_	137	120	x	170	145	æ	221	170	¬	252
46	.	56	71	G	107	96	`	140	121	y	171	146	Æ	222	171	½	253
47	/	57	72	H	110	97	a	141	122	z	172	147	ô	223	172	¼	254
48	0	60	73	I	111	98	b	142	123	{	173	148	ö	224	173	¡	255
49	1	61	74	J	112	99	c	143	124		174	149	ò	225	174	«	256
50	2	62	75	K	113	100	d	144	125	}	175	150	û	226	175	»	257
51	3	63	76	L	114	101	e	145	126	~	176	151	ù	227			
52	4	64	77	M	115	102	f	146	127	Esc/Del	177	152	ÿ	230			
53	5	65	78	N	116	103	g	147	128	Ç	200	153	Ö	231			
54	6	66	79	O	117	104	h	150	129	ü	201	154	Ü	232			
55	7	67	80	P	120	105	i	151	130	é	202	155	é	233			

### Exemple de codification de F (70) :

128	64	32	16	8	4	2	1
0	1	0	0	0	1	1	0

### Codification de W (87) :

128	64	32	16	8	4	2	1
0	1	0	1	0	1	1	1

### Codification OCTAL de ù (151) :

151	8	18	8
7	18	2	2





## **AutoLISP travaux dirigés : Création d'un programme**

**Les programmes AutoLISP doivent respecter quelques règles simples :**

**Facile à analyser...**

**Contenir le maximum de commentaires. Un commentaire est toujours précédé par ;  
Un commentaire peut également être placé en fin de ligne**

**Respecter le format ASCII (format texte, jamais dans un format natif tel que WORD)**

**Il faut aérer au maximum le programme (lignes blanches, commentaires). Une ligne blanche n'a pas besoin de commencer avec un ;**

**Il faut indenter le programme par niveau de fonction (tabulations)**

**Représenter schématiquement ce que fait le programme (dans le cas de fonction de dessin automatique)**

**Lister et commenter l'ensemble des variables utilisées dans le programme (évite de tenir un cahier)**





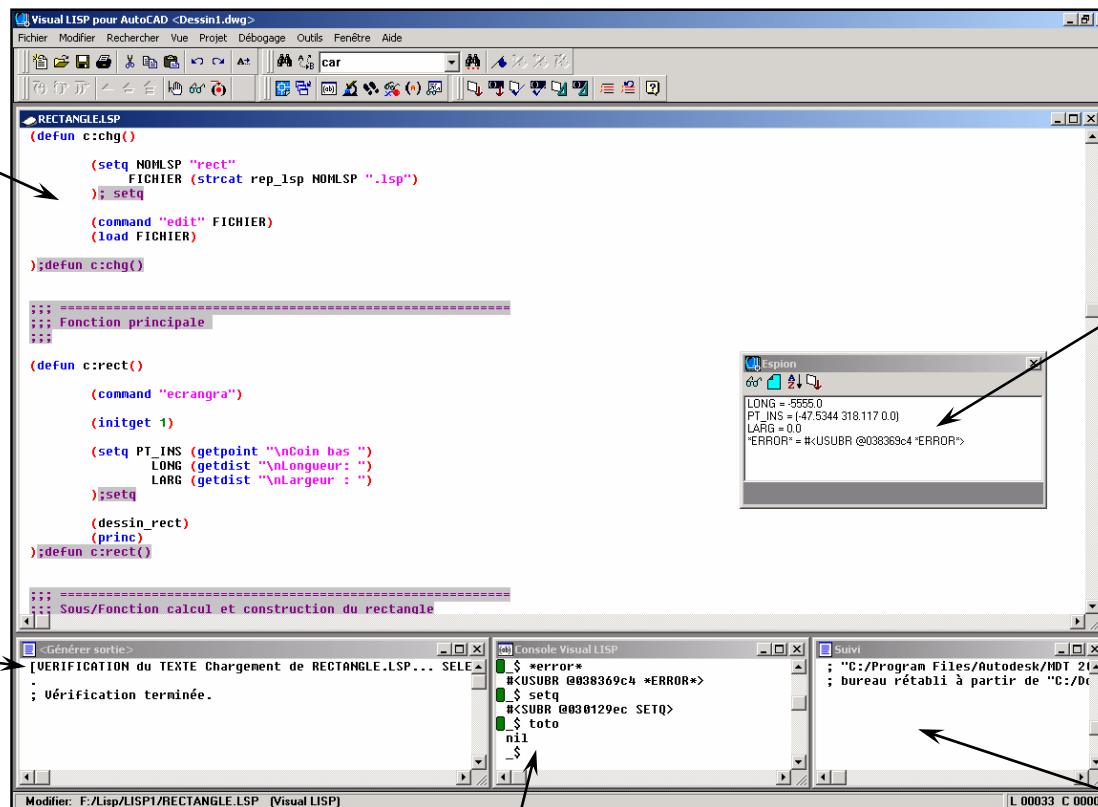
# L'éditeur Visual LISP : Présentation

L'éditeur Visual LISP est un outil de développement des programmes. Il intègre des fonctions de présentation, d'affichage en fonction des éléments, de test...

Zone Programme source

Zone Espion, permet de visualiser les variables ou fonctions en cours.

« Générer sortie » Zone des messages, tests, vérification



Historique de certaines fonctions Visual LISP.

Zone permettant de tester des syntaxes, des fonctions, vérifier les valeurs. Correspond à la zone commande AutoCAD.

Outils : AutoLISP > Editeur Visual LISP



Christian REB CAO/DAO Services  
1 rue des Fleurs / 69230 ZIMMERBACH  
Tél. : 03 89 58 98 16 / FAX : 03 89 58 40 20  
E Mail : reb.c@wanadoo.fr  
Web : perso.wanadoo.fr/rebcao

Conseil- Formation- Développement  
Votre Spécialiste AutoCAD

Programmation AutoLISP de Base

Page N° 17  
LISP4 NEW.PPT  
31/08/2005



## L'éditeur Visual LISP : La zone programme

L'éditeur Visual LISP affiche chaque type d'éléments avec une couleur différente. Ce qui Exemple :

<b>Commentaires</b> ;;;	<b>fond gris</b>
<b>Parenthèses</b> ( )	<b>rouge</b>
<b>Fonction LISP</b>	<b>bleue</b>
<b>Chaine de caractères</b> " ... "	<b>rose</b>
<b>Nom d'éléments (fonction, variable)</b>	<b>noir</b>
<b>Valeur numérique</b>	<b>vert</b>

```
RECTANGLE.LSP
(defun dessin_rect( / P2 P3 P4 )

  ;-----
  ; calcul
  ;-----

  (setq P2 (LIST (+ (car PT_INS) LONG) (cadr PT_INS) 0)
        P3 (LIST (car P2) (+ (cadr P2) LARG) 0)
        P4 (LIST (car PT_INS) (cadr P3) 0))

  ;-----
  ; tracé
  ;-----

  (setq old_osmode (getvar "OSMODE"))

  (setvar "OSMODE" 0)
  (setvar "cmdecho" 0)

  (command "polyline" PT_INS "_w" 0 0 P2 P3 P4 "c")

  (setvar "OSMODE" old_osmode)

); defun dessin_rect
```





## AutoLISP travaux dirigés : Création du fichier RECTANGLE.LSP

Utiliser VISUAL LISP ou un éditeur de texte standard : BLOC NOTES, WORD PAD (en mode texte pas de format natif).

```
;;;-----  
;;;   
;;;      Fichier RECTANGLE.LSP  
;;;      Construction d'un rectangle  
;;;   
;;;      COIN bas gauche, LONGUEUR, LARGEUR  
;;;      le 28/04/2004, par le BIDOUILLEUR FOU  
;;;-----  
;;;   
;;;   
;;;      P4                                P3  
;;;      ++++++  
;;;      +                                +  
;;;      +                                +  
;;;      LARG +                            +  
;;;      +                                +  
;;;      +      LONG                      +  
;;;      ++++++  
;;;      PT_INS                            P2  
;;;   
;;;   
;;; 
```



Faites précéder les commentaires par `;;;` car dans la pratique il arrive souvent de devoir mettre temporairement une ligne en commentaire avec un seul `;`





## AutoLISP travaux dirigés : Création du fichier RECTANGLE.LSP (suite...)

Lorsque la partie ci dessous est entrée, enregistrer, puis tester le déroulement et l'affichage dans le dessin.

```
;;;-----  
;;; Point d'insertion du rectangle  
;;;  
(setq PT_INS (getpoint "\nCoin bas gauche du rectangle" )  
  
;;;-----  
;;; Longueur du rectangle  
;;;  
(setq LONG (getdist "\nLongueur: " ) )  
  
;;;-----  
;;; Largeur du rectangle  
;;;  
(setq LARG (getdist "\nLargeur : " ) )
```



Charge le programme dans  
le dessin en cours



Bascule dans le dessin  
en cours



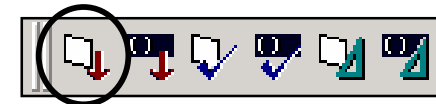


# AutoLISP travaux dirigés : Charger un programme AutoLISP

Il existe plusieurs façons de charger un programme AutoLISP : en mode manuel, semi automatique ou automatique. La Fonction de chargement est **LOAD** suivie du chemin + nom de fichier entre guillemets. Le séparateur des chemins sera le /

## Mode MANUEL

menu Visual LISP Menu Outils : Charger le texte dans l'éditeur, ou l'icône



au clavier taper : ( load "c:/chemin/nom du fichier" )

si l'extension du fichier n'est pas précisée, AutoLISP recherche un fichier .LSP

exemple : ( load "rectangle" )

charge le fichier RECTANGL.LSP

( load "c:/formation/AutoCAD/piece.prg" ) charge le fichier PIECE.PRG

## Mode SEMI AUTOMATIQUE

par le menu Outils : Charger une application...

Charge à la demande les fichiers AutoLISP qui sont préparés dans une liste utilisateur

## Mode AUTOMATIQUE

AutoLISP recherche au chargement ou à la création de chaque dessin un fichier utilisateur ACADDOC.LSP (ce fichier n'existe pas en standard, il faut le créer sauf si un applicatif l'a déjà créé). Dans ce fichier il faut rajouter une ligne pour charger les programmes AutoLISP : ( load "rectangle" ) ou ( load " c:/formation/AutoCAD/rectangle"). Pour les anciennes versions (V14 et moins).





## AutoLISP travaux dirigés : Tester le programme

Dans l'état actuel du programme, il sera nécessaire de le charger à chaque fois que l'on aura besoin de dessiner un rectangle.

D'autre part, seul les fonctions de saisie sont activées, c'est pourquoi après chargement les trois questions défileront et resteront sans effets à l'écran.

Si le programme "PLANTE", dans ce cas localiser la ligne où se produit l'erreur. Vérifier si le nombre de parenthèses et de guillemets, le nom des fonctions AutoLISP, etc. Utiliser les fonctions Visual LISP : S'il manque un guillemet la suite prend une couleur rose... Les parenthèses seront contrôlées avec Menu Modifier : Apparier les parenthèses... On peut également utiliser la vérification du programme entier ou de la sélection.



Si l'ensemble se passe bien, AutoLISP retourne à la fin de la dernière question la valeur entrée pour LARG.

*Afin d'éviter trop d'erreurs dans un programme AutoLISP, il est préférable de procéder par étape. Tester le bon fonctionnement à chacune des nouvelles étapes : Affichage, saisie, calcul, dessin, etc.*







## AutoLISP travaux dirigés : Création d'une fonction COMMANDE

Afin d'éviter de charger à chaque fois le programme lors de la construction d'un RECTANGLE, créons une nouvelle **COMMANDE** dans AutoCAD. Il s'agit d'encadrer la partie saisie dans une fonction appelé **DEFUN**. Cette fonction précédée par **C:** (commande) permettra ensuite d'utiliser une nouvelle commande dans le dessin courant. Dès que l'on utilise une telle fonction, il est souhaitable d'indenter le programme par niveau de fonction (tabulations).

**( DEFUN FONC () ... )**

fonction utilisable par d'autres fonctions AutoLISP.

**( DEFUN C:FONC () ... )**

fonction de commande qui ne peut pas être appelé par une autre fonction AutoLISP. Ne peut être utilisée que sous forme de commande AutoCAD.

```
;;;
;;;
;;;-----
;;; Fonction principale
;;;
(defun C:RECT ()

    ;;;-----
    ;;; Point d'insertion du rectangle
    ;;;
    (setq PT_INS (getpoint "\nCoin bas gauche du rectangle" ) )

    ;;;-----
    ;;; Longueur du rectangle
    ;;;
    (setq LONG (getdist "\nLongueur: " ) )

    ;;;-----
    ;;; Largeur du rectangle
    ;;;
    (setq LARG (getdist "\nLargeur : " ) )

) ;defun C:RECT
```





# AutoLISP travaux dirigés : Les variables LOCALES et GLOBALES

Dans une fonction AutoLISP on distingue deux types de variables. Ces variables peuvent être utilisées par diverses fonctions de programmes AutoLISP différents. Ces variables sont :

**LOCALES**

propre à une fonction. Retourne nil dès que l'on quitte cette fonction. Une variable doit toujours être déclarée ( / var1 var2 )

**GLOBALES**

restent actives dans le dessin courant, peuvent être utilisées par d'autres fonctions

```
;;;-----  
;;; Fonction principale  
;;;  
(defun C:RECT ( / PT_INS LONG LARG )  
  
    ;;;-----  
    ;;; Point d'insertion du rectangle  
    ;;;  
    (setq PT_INS (getpoint "\nCoin bas gauche du rectangle" ) )  
  
    ;;;-----  
    ;;; Longueur du rectangle  
    ;;;  
    (setq LONG (getdist "\nLongueur: ") )  
  
    ;;;-----  
    ;;; Largeur du rectangle  
    ;;;  
    (setq LARG (getdist "\nLargeur : ") )  
  
);defun C:RECT
```





## AutoLISP travaux dirigés : Calcul des POINTS

La fonction C:RECT sera appelée fonction principale du programme. Ajoutons une nouvelle fonction pour calculer les points du rectangle. Cette fonction sera appelée à partir de C:RECT en utilisant des variables LOCALES. Dans un premier temps, ajouter l'appel de la fonction DESSIN\_RECT dans la fonction principale C:RECT.

```
;;;-----  
;;; Fonction principale  
;;;  
(defun C:RECT( / PT_INS LONG LARG )  
  
    ;;;-----  
    ;;; Point d'insertion du rectangle  
    ;;;  
    (setq PT_INS (getpoint "\nCoin bas gauche du rectangle" ) )  
  
    ;;;-----  
    ;;; Longueur du rectangle  
    ;;;  
    (setq LONG (getdist "\nLongueur: " ) )  
  
    ;;;-----  
    ;;; Largeur du rectangle  
    ;;;  
    (setq LARG (getdist "\nLargeur : " ) )  
  
    ( dessin_rect PT_INS LONG LARG )  
  
);defun C:RECT
```





## AutoLISP travaux dirigés : Calcul des POINTS

La fonction de calcul permettra à d'autres programmes AutoLISP de l'utiliser en passant les arguments PT\_INS, LONG, LARG. Les variables P2, P3 et P4 sont des variables locales de la fonction DESSIN\_RECT.

```
(setq LARG (getdist "\nLargeur : ") )

( dessin_rect PT_INS LONG LARG )

);;; defun C:RECT

;;;-----
;;; Dessin du rectangle
;;;

(defun dessin_rect( pt_ins long larg / P2 P3 P4 )

    ;;;-----
    ;;; calcul
    ;;;-----

    (setq      P2 (LIST (+ (car PT_INS) LONG) (cadr PT_INS)      0)
              P3 (LIST (car P2)                (+ (cadr P2) LARG) 0)
              P4 (LIST (car PT_INS)            (cadr P3)        0)
    ) ;setq

);defun dessin_rect
```





## **AutoLISP travaux dirigés : Tester le programme et le tracé du rectangle**

Afin de contrôler l'exactitude des points, vérifions en mode commande les quatres POINTS : PT\_INS, P2, P3 et P4, pour cela :

1. Charger le programme, puis taper RECT
2. Répondre aux questions Position, longueur, largeur
3. Utiliser la commande Polyligne pour tracer le rectangle en répondant successivement :

**!PT\_INS !P2 !P3 !P4**

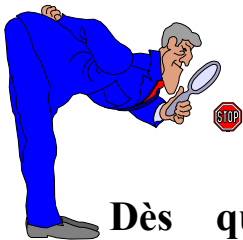
4. Pour finir le rectangle, il faut utiliser l'option clore, "C" (attention à l'épaisseur courante)
5. Prévoir éventuellement la mise à 0 (zéro) de l'épaisseur de la polyligne. (option internationale `_w +` largeur de départ = 0 et largeur de fin = 0)





## AutoLISP travaux dirigés : Construction du rectangle

La fonction AutoLISP : **COMMAND** permet d'utiliser presque toutes les commandes standards d'AutoCAD. Utiliser les éléments de la commande polyligne tester précédemment, pour construire le rectangle. La fonction **COMMAND** accepte une instruction spéciale: **PAUSE** qui permet à l'utilisateur d'entrer une valeur ou un point au clavier ou à l'écran.



Dès que la fonction est complète, tester le fonctionnement du programme. Noter tous les problèmes rencontrés. Vérifier que la construction fonctionne avec un petit ou un grand rectangle.

```
(defun dessin_rect( pt_ins long larg / P2 P3 P4 )

;;; -----
;;; calcul
;;; -----

(setq      P2 (LIST (+ (car PT_INS) LONG) (cadr PT_INS)      0)
          P3 (LIST (car P2)                (+ (cadr P2) LARG) 0)
          P4 (LIST (car PT_INS)            (cadr P3)          0)
);setq

;;; -----
;;; dessin
;;; -----

( command "polyline" "_w" 0 0 pt_ins p2 p3 p4 "c" )

);defun dessin_rect
```



Attention, n'oublier pas les guillemets pour le nom de la commande et les différentes options ( "polyline" "c" ). Sans les guillemets AutoLISP cherche à évaluer la variable qui retourne le plus souvent : nil et "plante" le programme.





# AutoLISP travaux dirigés : Amélioration du programme, nouvelles fonctions

Un certain nombre de fonctions sont destinées à améliorer le fonctionnement des programmes AutoLISP. Il s'agit d'étudier les fonctions : **SETVAR** (modifie les variables systèmes), **GETVAR** (lit les variables systèmes), **PRINC** (affiche le contenu d'une variable AutoLISP), **\*ERROR\*** (fonction de gestion d'erreur), **ALERT** (affiche une case de dialogue à l'écran), **PROMPT** (affichage de messages textes à l'écran), **COMMAND** (appel de commandes standards AutoCAD).

Lorsque l'on observe toutes les phases dans l'utilisation du programme RECTANGL.LSP on constate certains défauts liés à :

## L'AFFICHAGE

1. Le défilement des commandes à l'écran
2. L'affichage de **nil** au chargement du programme RECTANGL.LSP
3. L'affichage de **nil** à la fin de la fonction RECT
4. Les marques de construction peuvent être visibles aux coins du rectangle
5. Lorsque l'écran texte est affiché, il faut basculer manuellement sur le graphique

## LA CONSTRUCTION

6. Les accrochages objets permanents perturbent la construction du rectangle

## LE DEROULEMENT DU PROGRAMME

7. Affichage parasite en cas d'annulation dans la commande RECT (surtout avec V12)
8. Il n'y a pas d'aide sur l'utilisation du programme au chargement (quoi faire ?)
9. Il est possible de rentrer une valeur nulle aux questions Pt Insertion, Longueur, Largeur





## AutoLISP travaux dirigés : Amélioration de l'affichage

La fonction **SETVAR** permet de modifier les variables systèmes d'AutoCAD. Voir le tableau de correspondance des variables système sous **MODIFVAR**.

```
;;;-----  
;;; Fonction principale  
;;;  
(defun C:RECT( / PT_INS LONG LARG )  
  
    (setvar "cmdecho" 0 ) ;;; CMDECHO 0 supprime les messages des commandes  
    (graphscr) ;;; fonction de basculement en écran graphique  
  
    ;;;-----  
    ;;; Point d'insertion du rectangle  
    ;;;  
    (setq PT_INS (getpoint "\nCoin bas gauche du rectangle" ) )  
    ;;;-----  
    ;;; Longueur du rectangle  
    ;;;  
    (setq LONG (getdist "\nLongueur: " ) )  
    ;;;-----  
    ;;; Point d'insertion du rectangle  
    ;;;  
    (setq LARG (getdist "\nLargeur : " ) )  
    ( dessin_rect PT_INS LONG LARG )  
  
    (princ) ;;; évite l'affichage nil à la fin de RECT  
    ;;; permet d'afficher le contenu de variables (PRINC LONG)  
  
);defun C:RECT
```





# Exemple de Variables Systèmes d'une version AutoCAD 2004

## A

ACADLSPASDOC	Détermine si AutoCAD charge le fichier acad.lsp dans chaque dessin ou uniquement dans le premier dessin ouvert au cours d'une session AutoCAD.
ACADPREFIX	Contient, le cas échéant, le chemin d'accès spécifié dans la variable d'environnement ACAD, muni de séparateurs de chemin si nécessaire.
ACADVER	Contient le numéro de version d'AutoCAD.
ACISOUTVER	Contrôle la version ACIS des fichiers SAT créés à l'aide de la commande SAUVEACIS.
ADCSTATE	Détermine si DesignCenter est activé ou désactivé. Destinée aux développeurs qui doivent déterminer l'état via AutoLISP.
AFLAGS	Définit les drapeaux d'attributs du code binaire de la commande ATTDEF.
ANGBASE	Définit l'angle de base 0 en accord avec le SCU courant.
ANGDIR	Définit la direction positive à partir de l'angle 0 relativement au SCU courant.
APBOX	Active ou désactive la boîte d'ouverture AutoSnap.
APERURE	Définit, en pixels, la hauteur de la cible d'accrochage.
AREA	Enregistre la dernière aire calculée par une commande AIRE, LISTE ou LISTEBD.
ATTDIA	Contrôle si la commande INSERER utilise une boîte de dialogue pour la saisie de la valeur d'attribut.
ATTMODE	Contrôle le mode d'affichage des attributs.
ATTREQ	Détermine si la commande INSERER utilise des paramètres par défaut pendant l'insertion de blocs.
AUDITCTL	Contrôle si la commande CONTROLE crée un rapport d'analyse (fichier ADT).
AUNITS	Définit les unités des angles.
AUPREC	Définit le nombre de positions décimales pour les unités d'angle.
AUTOSNAP	Contrôle le marqueur, les textes info et le magnétisme AutoSnap.

## B

BACKZ	Enregistre le décalage du plan de délimitation arrière par rapport au plan cible de la fenêtre active.
BINDTYPE	Détermine comment les noms de références externes sont traités lorsque celles-ci sont liées ou modifiées.
BLIPMODE	Définit si les marques temporaires sont visibles.

## C

CDATE	Définit la date et l'heure
CECOLOR	Définit la couleur des nouveaux objets.
CELTSCALE	Définit le facteur d'échelle du type de ligne de l'objet courant.
CELTYPE	Définit le type de ligne des nouveaux objets.
CELWEIGHT	Définit l'épaisseur des lignes des nouveaux objets.
CHAMFERA	Définit la distance du premier chanfrein.
CHAMFERB	Définit la distance du deuxième chanfrein.
CHAMFERC	Définit la longueur du chanfrein.
CHAMFERD	Définit la distance du quatrième chanfrein.
CHAMMODE	Définit la méthode d'entrée utilisée par AutoCAD pour créer les chanfreins.
CIRCLERAD	Définit le rayon par défaut d'un cercle.
CLAYER	Définit le calque courant.
CMDACTIVE	Enregistre le code binaire indiquant qu'une commande ordinaire, une commande transparente, un script ou une boîte de dialogue est active.
CMDECHO	Détermine si AutoCAD renvoie des messages de confirmation et des données pendant l'exécution de la fonction AutoLISP (commande).
CMDNAMES	Affiche le nom des commandes active et transparente.
CMLJUST	Spécifie la justification d'une multiligne.
CMLSCALE	Contrôle la largeur totale d'une multiligne.
CMLSTYLE	Définit le style de la multiligne.
COMPASS	Détermine si la boussole 3D est active ou inactive dans la fenêtre courante.
COORDS	Contrôle la mise à jour des coordonnées sur la ligne d'état.
CLOTSTYLE	Détermine le style de tracé courant pour les nouveaux objets.
CPROFILE	Contient le nom du profil courant.
CTAB	Renvoie le nom de l'onglet courant (objet ou présentation) dans le dessin. Permet à l'utilisateur d'identifier l'onglet actif.
CURSORSIZE	Détermine la taille du réticule (curseur) en tant que pourcentage de la taille de l'écran.
CVPORT	Définit le numéro d'identification de la fenêtre active.

## D

DATE	Enregistre les date et heure courantes.
<b>DBCSTATE</b>	Enregistre l'état du Gestionnaire de connexion BD, s'il est actif ou non.
DBMOD	Indique l'état de modification du dessin à l'aide du code binaire.
DCTCUST	Affiche le chemin d'accès et le nom de fichier du dictionnaire personnalisé courant.
DCTMAIN	Affiche le nom de fichier du dictionnaire orthographique personnalisé courant.
DEFLPLSTYLE	Spécifie le style de tracé par défaut pour les nouveaux calques.
DEFFLSTYLE	Spécifie le style de tracé par défaut pour les nouveaux calques.
DELOBJ	Contrôle si les objets utilisés pour créer d'autres objets sont conservés ou supprimés de la base de données du dessin.
DEMANDLOAD	Spécifie, le cas échéant, quand l'interrogation d'AutoCAD charge une application de partie tierce lorsqu'un dessin contient des objets personnalisés créés sous cette application.
DIASAT	Enregistre la méthode employée pour quitter la dernière boîte de dialogue utilisée.
DIMADEC	Définit le nombre de positions de précision affichées pour les cotes angulaires.
DIMALT	Contrôle l'affichage des unités alternantes dans les cotes.
DIMALTD	Contrôle le nombre de positions décimales pour les unités alternantes.
DIMALTF	Définit le multiplicateur pour les unités alternantes.
DIMALTRND	Détermine l'arrondissement des unités alternantes.
DIMALTTD	Définit le nombre de positions décimales dans les valeurs de tolérance des unités alternatives d'une cote.
DIMALTTZ	Bascule la suppression des zéros pour les valeurs de tolérance.
DIMALTU	Définit les unités de format des unités alternatives de tous les membres de familles de styles de cote à l'exception des angulaires.
DIMALTZ	Vérifie la suppression des zéros pour les valeurs de cote des unités alternantes
DIMAPOST	Spécifie un préfixe ou un suffixe de texte (ou les deux) pour les unités alternatives de tous les types de cotes (sauf angulaires).
DIMASO	Contrôle l'associativité des objets de cote.
<b>DIMASSOC</b>	Gère l'associativité des objets de cote.
DIMASZ	Définit la taille des pointes de flèches de la ligne de cote et de la ligne de repère.
DIMATFIT	Détermine comment le texte de cote et les flèches sont agencés quand ces éléments ne peuvent pas tenir entre les lignes d'attache.
DIMAUNIT	Définit le format des unités pour les cotes angulaires.
DIMAZIN	Supprime les zéros des cotes angulaires.
DIMBLK	Définit les pointes de flèche affichées aux extrémités des lignes de cote ou de repère.
DIMBLK1	Lorsque la valeur de DIMSAH est Active, définit la pointe de flèche de la deuxième extrémité de la ligne de cote.
DIMBLK2	Lorsque la valeur de DIMSAH est Active, définit la pointe de flèche de la deuxième extrémité de la ligne de cote.
DIMCEN	Contrôle le dessin des marques centrales de cercle ou d'arc, et des lignes médianes au moyen des commandes COTCEN, COTDIA, et COTRAYON.
DIMCLRD	Attribue des couleurs aux lignes de cote, aux pointes de flèches et aux lignes de repère.
DIMCLRE	Affecte des couleurs aux lignes d'attache.
DIMCLRT	Attribue des couleurs au texte des cotes.
DIMDEC	Définit le nombre de décimales affichées dans les cotes exprimées en unités principales.
DIMDLE	Etend la ligne de cote au-delà de la ligne d'attache lorsque des barres obliques sont tracées à la place des pointes de flèche.
DIMDLI	Définit l'espacement des lignes de cote pour les cotes de ligne de base.
DIMDSEP	Définit le caractère à utiliser comme séparateur décimal lors de la création de cotes dont les unités sont exprimées dans le format décimal.
DIMEXE	Détermine dans quelle mesure la ligne d'attache peut être étendue au-delà de la ligne de cote.
DIMEXO	Spécifie le décalage des lignes d'attache par rapport aux points d'origine.
DIMFIT	Obsolète. Remplacé par DIMATFIT et DIMTMOVE
DIMFRAC	Définit le format des fractions quand DIMLUNIT a pour valeur 4 ou 5.
DIMGAP	Définit l'espace autour du texte de cote lorsque la ligne de cote est interrompue pour faire tenir tout ce texte.
DIMJUST	Contrôle la justification horizontale du texte de cote.
DIMLDRBLK	Spécifie le type de flèche pour les lignes de repère.
DIMLFAC	Définit un facteur d'échelle pour les cotes linéaires.
DIMLIM	Génère des limites de cote comme texte par défaut.



## Exemple de Variables Systèmes d'une version AutoCAD 2004

DIMLUNIT	Définit les unités de tous les types de cote, sauf les cotes angulaires.
DIMLWD	Assigne une épaisseur aux lignes de cote.
DIMLWE	Assigne une épaisseur aux lignes d'attache.
DIMPOST	Définit un préfixe ou un suffixe de texte (ou les deux) pour une cote.
DIMRND	Arrondit toutes les distances de cotation à la valeur spécifiée.
DIMSAH	Contrôle l'affichage des pointes de flèche de ligne de cote.
DIMSCALE	Définit le facteur d'échelle global appliqué aux variables de cotation qui spécifient des tailles, des distances ou des décalages.
DIMSD1	Contrôle la suppression de la première ligne de cote.
DIMSD2	Contrôle la suppression de la deuxième ligne de cote.
DIMSE1	Supprime l'affichage de la première ligne d'attache.
DIMSE2	Supprime l'affichage de la deuxième ligne d'attache.
DIMSHO	Contrôle la redéfinition des cotes au cours des déplacements interactifs.
DIMSOXD	Supprime le tracé des lignes de cote en dehors des lignes d'attache.
DIMSTYLE	Définit le style de cote courant.
DIMTAD	Définit la position verticale du texte par rapport à la ligne de cote.
DIMTDEC	Fixe le nombre de décimales des valeurs de tolérance à afficher pour les unités de cotation principales.
DIMTFAC	Définit un facteur d'échelle qui sert à calculer la hauteur du texte pour les fractions et les tolérances de cote.
DIMTIH	Contrôle la position du texte de cote dans les lignes d'attache pour tous les types de cote sauf les cotes en coordonnées.
DIMTIX	Trace un texte entre les lignes d'attache.
DIMTM	Définit la limite de tolérance minimum (ou inférieure) du texte de cote lorsque DIMITOL ou DIMLIM est activée.
DIMTMOVE	Définit les règles de déplacement du texte de cote.
DIMTOFL	Contrôle le tracé d'une ligne de cote entre les lignes d'attache, même si le texte est à l'extérieur de lignes d'attache.
DIMTOH	Lorsqu'elle est activée, cette variable définit la position du texte placé à l'extérieur des lignes d'attache.
DIMTOL	Ajoute des tolérances au texte de cote.
DIMTOLJ	Définit la justification verticale des valeurs de tolérance en fonction du texte de cote nominal.
DIMTP	Définit la limite de tolérance maximum (ou supérieure) du texte de cote lorsque DIMITOL ou DIMLIM est activée.
DIMTSZ	Définit la taille des barres obliques tracées à la place des pointes de flèche pour la cotation linéaire, de rayon et de diamètre.
DIMTVP	Contrôle la position verticale du texte de cote au dessus ou au dessous de la ligne de cote.
DIMTXSTY	Spécifie le style de texte de la cote.
DIMTXT	Définit la hauteur du texte de cote, sauf si le style de texte courant a une hauteur fixe.
DIMTZIN	Contrôle la suppression des zéros dans les valeurs de tolérance.
DIMUNIT	Obsolète. Remplacé par DIMLUNIT et DIMFRAC
DIMUPT	Définit le fonctionnement du curseur pour le texte positionné par l'utilisateur.
DIMZIN	Contrôle la suppression des zéros de la valeur d'unité principale.
DISPSILH	Contrôle l'affichage des contours des solides en mode filaire.
DISTANCE	Enregistre la distance calculée par la commande DISTANCE.
DONUTID	Définit la valeur par défaut du diamètre intérieur d'un anneau.
DONUTOD	Définit la valeur par défaut du diamètre extérieur d'un anneau.
DRAGMODE	Contrôle l'affichage des objets déplacés de manière interactive
DRAGP1	Définit la fréquence d'échantillonnage lors de la régénération en cours de déplacement interactif.
DRAGP2	Définit la fréquence d'échantillonnage lors de l'entrée en cours de déplacement interactif rapide.
DWGCHECK	Détermine si la dernière modification d'un dessin a été effectuée avec un produit autre qu'AutoCAD.
DWGCODEPAGE	Possède la même valeur que la variable SYSCODEPAGE (pour des raisons de compatibilité).
DWGNAME	Enregistre le nom du dessin entré par l'utilisateur.
DWGPREFIX	Enregistre le préfixe d'unité/répertoire du dessin.
DWGTITLED	Indique si le dessin courant a été nommé.

## E

EDGMODE	Définit de quelle façon les commandes AJUSTER et PROLONGE déterminent les arêtes de coupe et de contours.
ELEVATION	Enregistre l'élévation courante par rapport au SCU courant de la fenêtre courante dans l'espace courant.
<b>ERRNO</b>	Affiche le numéro du code d'erreur approprié lorsqu'un appel de la fonction AutoLISP génère une erreur détectée par AutoCAD. Les applications AutoLISP peuvent examiner la valeur courante d'ERRNO à l'aide de la formule (getvar "errno").
EXPERT	Contrôle si certains messages sont affichés.
EXPLMODE	Définit si la commande DECOMPOS prend en charge des blocs à échelle non uniforme (NUS).
EXTMAX	Enregistre le point supérieur droit de l'étendue du dessin.

EXTMIN	Enregistre le point inférieur gauche de l'étendue du dessin.
EXTNAMES	Définit les paramètres des noms d'objet existants (comme les types de ligne et les calques) stockés dans les tables de symboles.

**F**

FACETRATIO	Contrôle le rapport de linéarité des facettes des solides ACIS cylindriques et coniques.
FACETRES	Ajuste le lissage des objets ombrés et des objets rendus dont les lignes cachées ont été supprimées.
FILEDIA	Supprime l'affichage des boîtes de dialogue de fichiers.
FILLETRAD	Enregistre le rayon de raccord courant
FILLMODE	Indique si les multilignes, les traces, les solides, les hachures (y compris pour le remplissage de solides) et les polygones étendus sont remplis.
FONTALT	Spécifie la police de rechange à utiliser lorsque le fichier de polices spécifié est introuvable.
FONTMAP	Spécifie le fichier de mappage de police à utiliser.
FRONTZ	Enregistre le décalage du plan de délimitation avant par rapport au plan cible de la fenêtre active.
FULLOPEN	Indique si le dessin courant est partiellement ouvert.

## G

<b>GFANG</b>	Indique l'angle d'un remplissage avec gradient. Les valeurs acceptées sont comprises entre 0 et 360 degrés.
<b>GFCLR1</b>	Indique la couleur d'un remplissage avec gradient d'une couleur ou la première couleur d'un remplissage avec gradient de deux couleurs.
<b>GFCLR2</b>	Indique la seconde couleur d'un remplissage avec gradient de deux couleurs.
<b>GFCLRUM</b>	Définit la couleur comme une teinte (mêlée avec du blanc) ou une ombre (mêlée avec du noir) dans un

<b>GFCLRSTATE</b>	Indique si un remplissage avec gradient utilise une ou deux couleurs.
<b>GFNAME</b>	Indique le motif d'un remplissage avec gradient. Les valeurs acceptées sont comprises entre 1 et 9.
<b>GFSHIFT</b>	Indique si le motif d'un remplissage avec gradient est centré ou décalé en haut à gauche.
<b>GRIDMODE</b>	Spécifie si une grille est activée ou non.
<b>GRIDUNIT</b>	Spécifie l'espacement de grille (X et Y) de la fenêtre active.
<b>GRIPBLOCK</b>	Gère l'affectation de poignées dans les blocs.
<b>GRIPCOLOR</b>	Définit la couleur des poignées non sélectionnées (représentées par des contours).
<b>GRIPHOT</b>	Définit la couleur des poignées sélectionnées (représentées par des boîtes remplies).
<b>GRIPHOVER</b>	Contrôle la couleur de remplissage d'une poignée lorsque le curseur s'arrête dessus. Les valeurs admises sont comprises entre 1 et 255.
<b>GRIPOBJLIMIT</b>	Supprime l'affichage des poignées lorsque le jeu de sélection d'origine comprend plus d'objets que le nombre spécifié.
<b>GRIPS</b>	Permet l'utilisation des poignées du jeu de sélection en mode Etirer, Déplacer, Rotation, Echelle et Miroir.
<b>GRIPSIZE</b>	Définit la taille, en pixels, du cadre tracé pour afficher la poignée.
<b>GRIPTIPS</b>	Contrôle l'affichage des astuces liées aux poignées lorsque le curseur se trouve au-dessus des poignées d'objets personnalisés prenant en charge les astuces.

H

<b>HALOGAP</b>	Détermine l'espace à afficher pour indiquer qu'un objet est masqué par un autre.
<b>HANDLES</b>	Indique si les poignées d'objet sont activées et accessibles par les applications.
<b>HIDEPRECISION</b>	Contrôle la précision du masquage et de l'ombrage.
<b>HIDETEXT</b>	Indique si les objets texte créés à l'aide des commandes TEXTE, TXTDYN ou TEXTMULT sont traités lors de l'exécution de la commande CACHER.
<b>HIGHLIGHT</b>	Gère la mise d'objets en surbrillance ; n'affecte pas les objets sélectionnés au moyen de poignées.
<b>HPANG</b>	Indique l'angle des hachures.
<b>HPASSOC</b>	Détermine si les motifs de hachure et les remplissages avec gradient sont associatifs.
<b>HPBOUND</b>	Contrôle le type d'objet créé par les commandes FHACH et CONTOUR.
<b>HPDOUBLE</b>	Spécifie si les hachures définies par l'utilisateur sont doublées.
<b>HPNAME</b>	Définit un nom de motif de hachurage par défaut.
<b>HPSCALE</b>	Spécifie un facteur d'échelle pour les hachures.
<b>HPSPACE</b>	Spécifie l'espacement entre les hachures pour les motifs simples définies par l'utilisateur.
<b>HYPERLINKBASE</b>	Indique le chemin d'accès utilisé pour tous les hyperliens relatifs du dessin.



**Christian REB CAO/DAO Services**  
1 rue des Fleurs / 68230 ZIMMERBACH  
Tél. : 03 89 58 98 16 / FAX : 03 89 58 40 20  
E Mail: reb.c@wanadoo.fr  
Web : perso.wanadoo.fr/rebcao

**Conseil- Formation- Développement**  
**==== Votre Spécialiste AutoCAD =====**

# Programmation AutoLISP de Base

Exemple de Variables Systèmes d’une version AutoCAD 2004

IMAGEHLT	Contrôle la mise en surbrillance de toute l'image tramée ou de son cadre seulement.
INDEXCTL	Contrôle si les index de calque et d'espace sont créés et enregistrés dans des fichiers dessin.
INETLOCATION	Contient l'emplacement Internet utilisé par la commande NAVIGATEUR et la boîte de dialogue Naviguez sur l'Internet.
INSBASE	Stocke le point de base d'insertion défini par la commande BASE.
INSNAME	Définit le nom de bloc par défaut pour la commande INSERER.
INSUNITS	Lorsque vous faites glisser un bloc à partir d'AutoCAD DesignCenter, indique l'une des valeurs d' unité de dessin.
INSUNITSDEFSOURCE	Définit la valeur des unités de contenu source.
INSUNITSDEFTARGET	Définit la valeur des unités de dessin cible.
INTERSECTIONCOLOR	Détermine la couleur des polygones d'intersection.
INTERSECTIONDISPLAY	Définit l'affichage des polygones d'intersection.
ISAVEBAK	Accélère les sauvegardes incrémentielles, en particulier pour les dessins de grande taille.
ISAVEPERCENT	Détermine la quantité d'espace non utilisé toléré dans un fichier dessin.
ISOLINES	Spécifie le nombre de lignes isométriques par surface pour les objets.

L	
LASTANGLE	Enregistre l'angle final du dernier arc entré.
LASTPOINT	Enregistre le dernier point entré.
LASTPROMPT	Enregistre la dernière invite affichée sur la ligne de commande.
LAYOUTREGENCTL	Indique comment la liste d'affichage est mise à jour dans l'onglet Objet et dans les onglets de présentation.
LENSLENGTH	Enregistre la distance focale (en millimètres) appliquée dans les vues en perspective pour la fenêtre active.
LIMCHECK	Gère la création d'objets en dehors des limites du dessin.
LIMMAX	Enregistre la limite supérieure droite du dessin pour l'espace courant.
LIMMIN	Enregistre la limite inférieure gauche du dessin pour l'espace courant.
LISPINIT	Lorsqu'une interface à document unique est activée, indique si les fonctions et les variables AutoLISP sont conservées quand vous ouvrez un nouveau dessin
LOCALE	Affiche le code de langage ISO de la version d'AutoCAD en cours d'exécution.
LOCALROOTPREFIX	Enregistre le chemin d'accès complet au dossier racine où les fichiers personnalisables locaux sont installés.
LOGFILEMODE	Spécifie si le contenu de la fenêtre texte doit être écrit dans un fichier journal.
LOGFILENAME	Indique le nom et le chemin d'accès au fichier journal.
LOGFILEPATH	Spécifie le chemin d'accès des fichiers journaux de tous les dessins d'une session.
LOGINNAME	Affiche le nom de l'utilisateur configuré ou entré lors du chargement d'AutoCAD.
LTSCALE	Définit le facteur d'échelle de type de ligue global.
LUNITS	Définit les unités linéaires.
LUPREC	Enregistre le nombre de positions décimales affichées pour les unités linéaires.
LWDEFAULT	Définit la valeur de l'épaisseur des lignes par défaut.
LWDISPLAY	Contrôle si l'épaisseur des lignes est affichée sur l'onglet Objet ou Présentation.
LWUNITS	Contrôle si les unités d'épaisseur des lignes sont affichées en pouces ou en millimètres.

M	
MAXACTVP	Définit le nombre maximum de fenêtres pouvant être actives simultanément dans l'affichage.
MAXSORT	Définit le nombre maximum de noms de symbole ou de fichier à trier par commandes de listage.
MBUTTONPAN	Contrôle la fonction du troisième bouton ou de la troisième roulette du périphérique de pointage.
MEASUREINIT	Définit les unités de dessin anglaises ou métriques.
MEASUREMENT	Définit les unités de dessin anglaises ou métriques pour le seul dessin courant.
MASQMODIF	Ferme le Gestionnaire de connectivité aux BD.
MENUCTL	Gère le changement de page du menu écran.
MENUECHO	Définit les bits de contrôle du message et de la confirmation visuelle du menu.
MENUNAME	Contient le nom du fichier de menu et son chemin d'accès.
MIRRTEXT	Gère le reflet du texte par MIROIR.
MODEMACRO	Affiche une chaîne de texte sur la ligne d'état.
MTEXTED	Définit les éditeurs de texte principal et secondaire à utiliser pour les objets texte multiligne.
MTEXTFIXED	Contrôle l'aspect de l'éditeur de texte multiligne.
MTJIGSTRING	Définit le contenu du texte d'exemple affiché à l'emplacement du curseur lorsqu'une commande TEXTMULT est lancée.
MYDOCUMENTSPREFIX	Enregistre le chemin d'accès complet au dossier Mes documents de l'utilisateur actuellement connecté.

N	
NOMUTT	Supprime l'affichage d'un message qui devrait normalement rester visible.
O	
OBSCUREDCOLOR	Indique la couleur des lignes obscurcies.
OBSCUREDLTYPE	Indique le type de ligne des lignes obscurcies.
OFFSETDIST	Définit la distance de décalage par défaut.
OFFSETGAPTYP	Détermine comment décaler les polygones quand un espace est créé suite au décalage des segments de polygone individuels.
OLEHIDE	Gère l'affichage des objets OLE sous AutoCAD.
OLEQUALITY	Détermine le niveau de qualité par défaut des objets OLE incorporés.
OLESTARTUP	Détermine si l'application source d'un objet OLE incorporé se charge lors du traçage.
ORTHOMODE	Restreint le mouvement du curseur à une direction perpendiculaire.
OSMODE	Définit les modes d'accrochage aux objets en utilisant les codes binaires.
OSNAPCOORD	Contrôle si les coordonnées entrées sur la ligne de commande écrasent l'accrochage des objets en cours d'exécution.

P	
PALETTEOPAQUE	Détermine si les fenêtres peuvent être transparentes.
PAPERUPDATE	Contrôle l'affichage d'une boîte de dialogue d'avertissement quand vous essayez d'imprimer une présentation avec un format de papier différent de la valeur par défaut pour le fichier de configuration du traceur.
PDMODE	Gère l'affichage des objets point.
PDSIZE	Définit la taille d'affichage des objets point.
PEDITACCEPT	Supprime l'affichage de l'invite L'objet sélectionné n'est pas une polygone de la commande PEDIT.
PELLIPSE	Définit le type d'ellipse créé par la commande ELLIPSE.
PERIMETER	Mémoire le dernier périmètre calculé par une commande AIRE, LISTE ou LISTEBD.
PFACEVMAX	Définit le nombre maximum de sommets par face.
PICKADD	Détermine si les sélections suivantes remplacent le jeu de sélection courant ou s'y ajoutent.
PICKAUTO	Gère le fenêtrage automatique lorsque le message Choix des objets s'affiche.
PICKBOX	Définit la hauteur de la cible de sélection d'objet.
PICKDRAG	Contrôle la méthode de dessin d'une fenêtre de sélection.
PICKFIRST	Contrôle si la sélection d'objets s'effectue avant ou après l'émission d'une commande.
PICKSTYLE	Contrôle la sélection des groupes et des hachures associatives.
PLATFORM	Indique la version d'AutoCAD utilisée.
PLINEGEN	Définit la génération des motifs de type de ligne autour des sommets d'une polygone2D.
PLINETYPE	Spécifie si AutoCAD utilise des polygones bidimensionnelles optimisées.
PLINEWID	Enregistre la largeur de polygone par défaut.
PLOTID	Obsolète, elle sert uniquement à préserver l'intégrité des scripts et des routines LISP antérieurs à AutoCAD2000.
PLOTROTMODE	Contrôle l'orientation des tracés.
PLOTTER	Obsolète. Elle sert uniquement à préserver l'intégrité des scripts et des routines LISP antérieurs à AutoCAD2000.
PLQUIET	Contrôle l'affichage des boîtes de dialogue optionnelles et des erreurs non bloquantes pour le tracé en différé et les scripts.
POLARADDANG	Contient les angles polaires définis par l'utilisateur.
POLARANG	Définit l'incrément des angles polaires.
POLARDIST	Définit l'incrément d'accrochage quand la variable système SNAPSTYL a pour valeur 1 (résolution polaire).
POLARMODE	Contrôle les paramètres de repérage polaire et d'accrochage aux objets.
POLYSIDES	Définit le nombre de côtés par défaut pour la commande POLYGON.
POPUPS	Affiche l'état du pilote d'écran actuellement configuré.
PRODUCT	Renvoie le nom du produit
PROGRAM	Renvoie le nom du programme
PROJECTNAME	Assigne un nom de projet au dessin courant.
PROJMODE	Définit le mode de projection courant pour les opérations d'ajustement ou de prolongation.
PROXYGRAPHICS	Spécifie si les images des objets externes sont enregistrées dans le dessin.



Exemple de Variables Systèmes d’une version AutoCAD 2004

PROXYNOTICE	Affiche un avis à chaque ouverture d'un dessin contenant des objets personnalisés créés sous une application qui n'est pas présente.	SPLFRAME	Gère l'affichage des splines et des polygones à lissage de spline.
PROXYSHOW	Gère l'affichage des objets externes dans un dessin.	SPLINESEGS	Définit le nombre de segments de ligne à générer pour chaque polygone d'ajustage à courbe spline.
PROXYWEBSEARCH	Spécifie la façon dont AutoCAD recherche les activateurs d'objets.		
PSLTSCALE	Contrôle l'échelle du type de ligne de l'espace papier.	SPLINETYPE	Définit le type de courbe générée par l'option Pspline de la commande PEDIT.
PSPROLOG	Définit un nom de section pour l'en-tête lu dans le fichier acad.psf lors de l'exécution de la commande SAUVEPS.	STANDARDSVIOLATION	Détermine si l'utilisateur est averti des violations de normes qui existent dans le dessin courant, lors de la création ou la modification d'un objet non normalisé.
PSQUALITY	Détermine la qualité de rendu des images PostScript.	STARTUP	Détermine si la boîte de dialogue Créer un nouveau dessin est affichée lors du lancement d'un nouveau dessin à l'aide des commandes NOUVEAU et RAPNOUV.
PSTYLEMODE	Indique si le dessin courant est en mode dépendant des couleurs ou style de tracé nommé.	SURFTAB1	Définit le nombre de tabulations à générer pour les commandes SURFREGL et SURFEXTR.
PSTYLEPOLICY	Contrôle si la propriété couleur d'un objet est associée à son style de tracé.	SURFTAB2	Définit la densité du maillage dans la direction N pour les commandes SURFREV et SURFGAU.
PSVPSCALE	Définit le facteur d'échelle d'affichage des fenêtres nouvellement créées.	SURFTYPE	Contrôle le type de lissage que doit effectuer l'option Lisser surface de la commande PEDIT.
		SURFU	Définit la densité de surface pour l'option Lisser surface de la commande PEDIT dans la direction M.
PUCSBASE	Mémoire le nom du SCU qui définit l'origine et l'orientation des paramètres de SCU orthogonal dans l'espace papier uniquement.	SURFV	Définit la densité de surface pour l'option Lisser surface de la commande PEDIT dans la direction M.
Q		SYSCODEPAGE	Indique la page de code système spécifié dans le fichier acad.xmlf
QTEXTMODE	Contrôle l'affichage du texte.		
R		T	
RASTERPREVIEW	Détermine si les aperçus d'image BMP sont enregistrés avec le dessin.	TABMODE	Gère l'utilisation du mode Tablette.
REFEDITNAME	Indique si une session de modification des références est en cours dans un dessin et mémorise le nom du fichier de références.	TARGET	Enregistre l'emplacement du point cible de la fenêtre active.
REGENMODE	Gère la régénération automatique du dessin.	TDCREATE	Mémoire la date et l'heure locale de création du dessin.
RE INT	Réinitialise le numériseur, le port du numériseur et le fichier acad.pgp.	TDINDWG	Enregistre la durée totale d'édition.
REMEMBERFOLDERS	Gère le chemin par défaut de l'option Rechercher dans ou Enregistrer des boîtes de dialogue standard de sélection de fichiers.	TDUCREATE	Mémoire la date et l'heure universelle de création du dessin.
REPORTERROR	Contrôle si un rapport d'erreurs peut être envoyé à Autodesk lorsqu'AutoCAD se ferme subitement.	TDUPDATE	Enregistre l'heure et la date de la dernière mise à jour ou du dernier enregistrement.
ROAMABLEROOTPREFIX	Enregistre le chemin d'accès complet au dossier racine où les fichiers personnalisables itinérants sont installés.	TDUSRTIMER	Enregistre le temps écoulé à l'horloge de l'utilisateur.
RTDISPLAY	Contrôle l'affichage des images tramées pendant ZOOM ou PAN.	TDUUPDATE	Mémoire la date et l'heure universelle de la dernière mise à jour/du dernier enregistrement.
S		TEMPPREFIX	Contient le nom du répertoire des fichiers temporaires.
SAVEFILE	Enregistre le nom du fichier courant de sauvegarde automatique.	TEXTVAL	Gère la méthode d'évaluation des chaînes de texte.
SAVEFILEPATH	Indique le chemin d'accès au répertoire où sont stockés tous les fichiers d'enregistrement automatique de la session AutoCAD.	TEXTFILL	Gère le remplissage des polices TrueType au cours du traçage, de l'exportation au moyen de la commande SAUVEPS et du rendu.
SAVENAME	Stocke le nom de fichier et le chemin d'accès du dessin courant, quand vous avez enregistré ce dernier.	TEXTQLTY	Définit la finesse des contours de texte pour les polices TrueType pendant le traçage, l'exportation avec la commande SAUVEPS et le rendu.
SAVETIME	Définit, en minutes, l'intervalle de sauvegarde automatique.	TEXTSIZE	Définit la taille par défaut des nouveaux objets texte tracés avec le style de texte courant.
SCREENBOXES	Enregistre le nombre de boîtes dans la zone menu de la zone graphique.	TEXTSTYLE	Définit le nom du style de texte courant.
SCREENMODE	Enregistre un code binaire indiquant l'état graphique ou texte de l'affichage AutoCAD.	THICKNESS	Définit l'épaisseur 3D courante.
SCREENSIZE	Enregistre la taille, en pixels, de la fenêtre active (X et Y).	TILEMODE	Définit l'onglet Objet ou le dernier onglet de présentation en tant qu'onglet courant.
SDI	Détermine si AutoCAD s'exécute avec une interface monodocument ou multidocument.	TOOLTIPS	Contrôle l'affichage des textes info.
SHADEGE	Gère l'ombrage des arêtes lors du rendu.	TPSTATE	Détermine si la fenêtre Palettes d'outils est active ou inactive.
SHADEDF	Définit le taux de lumière réfléchié diffuse par rapport à la lumière ambiante.	TRACKPATH	Contrôle l'affichage des chemins d'alignement du repérage polaire et d'accrochage aux objets.
SHORTCUTMENU	Définit si les menus contextuels des modes Default, Edition et Commande sont disponibles dans la zone de dessin.	TRAYICONS	Détermine si un bac s'affiche sur la barre d'état.
SHPNAME	Définit un nom de forme par défaut.	TRAYNOTIFY	Détermine si les notifications de service s'affichent dans le bac de la barre d'état.
SIGWARN	Détermine si un avertissement est affiché à l'ouverture d'un fichier auquel une signature numérique est jointe.	TRAYTIMEOUT	Détermine la durée d'affichage (en secondes) des notifications de service.
SKETCHINC	Définit l'incrément d'enregistrement de la commande MAINLEV.	TREEDEPTH	Spécifie la profondeur maximale, à savoir combien de fois l'index espace à structure arborescente peut se diviser en branches.
SKPOLY	Détermine si la commande MAINLEV génère des lignes ou des polygones.	TREEMAX	Limite l'utilisation de la mémoire lors de la régénération d'un dessin en réduisant le nombre maximum de noeuds dans l'index spatial (arbre octal).
SNAPANG	Définit l'angle d'accrochage et de rotation de la grille dans la fenêtre active.	TRIMMODE	Contrôle si AutoCAD ajuste les arêtes sélectionnées pour les chanfreins et les raccords.
SNAPBASE	Définit le point d'origine du pas de grille de la fenêtre courante par rapport au SCU courant.	TSPACEFAC	Contrôle l'espacement des lignes de texte multiligne, exprimé par un facteur de la hauteur du texte.
SNAPISOPAIR	Gère le plan isométrique de la fenêtre active.	TSPACEYPE	Contrôle le type d'espacement des lignes utilisé pour le texte multiligne.
SNAPMODE	Active et désactive le mode Résolution.	TSTACKALIGN	Contrôle l'alignement vertical du texte empli.
SNAPSTYL	Définit le style d'accrochage pour la fenêtre active.	TSTACKSIZE	Contrôle le pourcentage de hauteur d'une fraction de texte superposée par rapport à la hauteur courante du texte sélectionné.
SNAPTYP	Définit le style de résolution de la fenêtre courante.		
SNAPUNIT	Définit l'espacement d'accrochage pour la fenêtre active.		
SOLIDCHECK	Active et désactive la validation des solides pour la session AutoCAD courante.		
SORTENTS	Contrôle les opérations d'ordre de tri d'objets de la commande OPTIONS (à partir de l'onglet Sélection)		





U

UCSAXISANG	Mémoire l'angle par défaut lors de la rotation du SCU sur un de ses axes à l'aide de l'option X, Y ou Z de la commande SCU.
UCSBASE	Mémoire le nom du SCU qui définit l'origine et l'orientation des paramètres de SCU orthogonal.
UCSFOLLOW	Génère une vue plane à chaque passage d'un SCU à l'autre.
UCSICON	Affiche l'icône du SCU dans la fenêtre courante.
UCSNAME	Mémoire le nom du système de coordonnées courant pour la fenêtre courante dans l'espace courant.
UCSORG	Mémoire le point d'origine du système de coordonnées courant pour la fenêtre courante dans l'espace courant.
UCSORTHO	Détermine si le paramètre SCU orthogonal associé est restauré automatiquement quand une vue orthogonale est restaurée.
UCSVIEW	Détermine si le SCU courant est enregistré avec une vue existante.
UCSVP	Détermine si le SCU des fenêtres actives reste fixe ou change pour refléter le SCU de la fenêtre active.
UCSXDIR	Mémoire la direction X du SCU courant pour la fenêtre courante de l'espace courant.
UCSXDIR	Mémoire la direction X du SCU courant pour la fenêtre courante de l'espace courant.
UNDOCTL	Mémoire un code binaire indiquant l'état des options Auto et Contrôle de la commande ANNULER.
UNDOMARKS	Enregistre le nombre de marques placées dans le flot de commandes ANNULER par l'option Marque.
UNITMODE	Gère le format d'affichage des unités.
USERI1-5	Enregistre et récupère les valeurs entières.
USERR1-5	Enregistre et récupère les nombres réels.
USERS1-5	Enregistre et récupère les données de chaîne de texte.

V

VIEWCTR	Enregistre le centre de vue dans la fenêtre active.
VIEWDIR	Enregistre la direction de visualisation de la fenêtre active.
VIEWMODE	Contrôle le mode de visée pour la fenêtre active à l'aide d'un code binaire.
VIEWSIZE	Enregistre la hauteur de vue dans la fenêtre active.
VIEWTWIST	Enregistre l'angle de bascule de vue pour la fenêtre active.
VISRETAIN	Contrôle la visibilité, la couleur, le type de ligne, l'épaisseur des lignes et les styles de tracé (si PSTYLEPOLICY a pour valeur 0) des calques dépendants de références externes et spécifie si les changements de chemin de références externes imbriquées sont enregistrés.
VSMAX	Enregistre le coin supérieur droit de l'écran virtuel de la fenêtre active.
VSMIN	Enregistre le coin inférieur gauche de l'écran virtuel de la fenêtre active.

W

WHIPARC	Détermine si l'affichage des cercles et des arcs est lissé ou non.
WHIPTHREAD	Détermine l'utilisation d'un processeur supplémentaire (traitement multithread) pour améliorer la vitesse des opérations comme ZOOM et PAN qui redessinent ou régénèrent le dessin.
WMFBKGN	Contrôle l'arrière plan du métafichier Windows résultant de la commande WMFOUT et du format métafichier d'objets situés dans le Presse papier ou glissés puis déposés
WMFFOREGND	Gère l'attribution d'une couleur de premier plan aux objets AutoCAD dans d'autres applications
WORLDUCS	Indique si le SCU est identique au SCG.
WORLDVIEW	Détermine si les entrées en 3DORBIT, VUEDYN et POINTVUE sont relatives au SCG (par défaut), au SCU courant ou au SCU indiqué dans la variable système UCSBASE.
WRITESTAT	Indique si un fichier dessin est en lecture seule ou si les opérations d'écriture y sont autorisées. Cette variable est destinée aux développeurs qui doivent déterminer l'état d'écriture via AutoLISP.

X

XCLIPFRAME	Contrôle la visibilité des contours de délimitation de xref.
XEDIT	Détermine si les références peuvent ou non être éditées dans le dessin courant si celui-ci est référencé par un autre dessin.
XFADECTL	Contrôle l'intensité de l'estompage des références modifiées dans le dessin.
XLOADCTL	Active et désactive le chargement immédiat et contrôle s'il charge le dessin d'origine ou une copie.
XLOADPATH	Crée un chemin d'accès pour l'enregistrement de copies temporaires des fichiers xref de chargement sur demande.
XREFCTL	Gère l'écriture des fichiers .xlg par AutoCAD (fichiers journaux de référence externe).
XREFNOTIFY	Gère les notifications concernant les xrefs mises à jour ou manquantes.

Z

ZOOMFACTOR	Contrôle le changement incrémentiel dans le zoom avec la roulette de la souris Intellimouse, à l'avant ou à l'arrière.
------------	--





## AutoLISP travaux dirigés : Amélioration de la construction

Il faut être très attentif à l'environnement dans lequel la fonction AutoLISP peut évoluer. Dans ce type de construction, les accrochages aux objets risquent de perturber gravement le tracé du rectangle. Le rectangle peut être totalement déformé en raison d'une trop petite taille ou d'objets voisins sur lesquels AutoCAD s'accroche.

```
(defun dessin_rect( pt_ins long larg / P2 P3 P4 )
  ;;-----
  ;; calcul
  ;;-----
  (setq  P2 (LIST (+ (car PT_INS) LONG) (cadr PT_INS) 0)
        P3 (LIST (car P2) (+ (cadr P2) LARG) 0)
        P4 (LIST (car PT_INS) (cadr P3) 0)
  );setq
  ;;-----
  ;; dessin
  ;;-----

  (setq  old_osmode (getvar "osmode")   ;; OSMODE sauve l'état des accrochages
        old_blipmode (getvar "blipmode") ;; BLIPMODE sauve l'état des marques
  );setq

  (setvar "osmode" 0) ;; inactive les accrochages
  (setvar "blipmode" 0) ;; inactive les marques

  ( command "polyline" pt_ins p2 p3 p4 "c" )

  (setvar "osmode" old_osmode) ;; restitue les anciens accrochages
  (setvar "blipmode" old_blipmode) ;; restitue l'état initial des marques

);defun dessin_rect
```





## AutoLISP travaux dirigés : Gestion d'erreur et message d'aide

La fonction spéciale **\*ERROR\*** permet de créer une fonction personnalisée de gestion des erreurs dans un programme AutoLISP.

La fonction **PRINC**, **PROMPT** et **ALERT** permettent d'afficher le contenu de la variable ou du texte à l'écran.

```
) ;defun dessin_rect

;;;-----
;;; Gestion des erreurs
;;;

(defun *error* (err)                                     ;;; définition de la fonction *ERROR*

    (princ err)                                           ;;; affichage de l'erreur stockée dans (err)

    (alert "du genou ! tu t'es gouré!!!")                ;;; affiche la case de dialogue avec le message

) ;defun *error*                                           ;;; fin de la fonction *ERROR*

;;; ***** Fin programme RECTANGLE.LSP *****
;;;

(prompt "\nTaper RECT pour construire un rectangle Longueur / Largeur...")
(princ)                                                    ;;; évite d'afficher nil au chargement de RECTANGLE.LSP
```



Lorsque le programme est en cours d'élaboration, l'activation de la fonction **\*ERROR\*** rend très difficile la localisation des erreurs qui provoquent un "plantage" du programme. Dans ce cas, neutraliser provisoirement la fonction avec **;** ou en réinitialisant la fonction à **NIL** : **(setq \*ERROR\* NIL)**





## AutoLISP travaux dirigés : Exemple de Corrigé

```
;;;*****  
;;;   fichier  RECTANGL.LSP, construction d'un RECTANGLE  
;;;   en indiquant LONGUEUR/HAUTEUR  
;;;   créé le 28/04/2004, par LE BIDOUILLEUR FOU  
;;;*****
```

```
;;;   * ----- *  
;;;   | P4                P3 |  
;;;   |                    |  
;;;   |                    |  
;;;   |                    |  
;;;   |                    | LARG  
;;;   |                    |  
;;;   |                    |  
;;;   | PT_INS      LONG    P2 |  
;;;   * ----- *
```

(.../...)







## AutoLISP travaux dirigés : Exemple de Corrigé (suite...)

```
;;? -----  
;;; Fonction RECT  
  
( defun c:rect( / pt_ins long larg )  
  
    (setvar "cmdecho" 0)      ;;; supprime les messages des commandes  
    (command "ecrangra")      ;;; commande de basculement en écran graphique  
  
    ;;? -----  
    ;;; Point d'insertion du rectangle  
    (initget 3)  
    ( setq pt_ins ( getpoint "\nPosition coin bas gauche du rectangle : " ) )  
  
    ;;? -----  
    ;;; Longueur du rectangle  
    (initget 3)  
    ( setq long ( getdist pt_ins "\nLongueur du rectangle : " ) )  
  
    ;;? -----  
    ;;; Largeur du rectangle  
    (initget 3)  
    ( setq larg ( GETDIST pt_ins "\nHauteur du rectangle : " ) )  
  
    (dessin_rect pt_ins long larg)  
    (princ)  
  
); defun RECT
```

(.../...)





## AutoLISP travaux dirigés : Exemple de Corrigé (suite...)

```
;;? -----  
;;; construction du rectangle  
  
(defun dessin_rect( pt_ins long larg / p2 p3 p4)  
  
    ;;? -----  
    ;;; calcul des points  
  
    (setq    p2 (list (+ (car pt_ins) long)    (cadr pt_ins)    0 )  
              p3 (list (car p2)                (+ (cadr pt_ins) larg) 0 )  
              p4 (list (car pt_ins)            (cadr p3)          0 )  
    );setq  
  
    ;;? -----  
    ;;; dessin du rectangle  
  
    (setq    old_osmode    (getvar "osmode")    ;;; sauve l'état des accrochages  
              old_blipmode (getvar "blipmode")  ;;; sauve l'état des marques  
    );setq  
  
    (setvar "osmode" 0)    ;;; inactive les accrochages  
    (setvar "blipmode" 0)  ;;; inactive les marques  
  
    (command "polyline" pt_ins "_w" 0 0 p2 p3 p4 "c") ;;; trace le rectangle  
  
    (setvar "osmode" old_osmode)    ;;; restitue les anciens accrochages  
    (setvar "blipmode" old_blipmode) ;;; restitue l'état initial des marques  
  
);defun dessin_rect
```

(.../...)





## AutoLISP travaux dirigés : Exemple de Corrigé (FIN)

```
;;; -----  
;;; gestion des erreurs  
;;;  
  
      (defun *error* (err)  
        (prompt "\n\n\t* * * ERREUR * * *")  
        (prompt (strcat "\n\nType = " erreur) )  
        (alert "du genou, tu t'es gouré!!!...")  
        (alert erreur)  
      );defun *error*  
  
;;;***** FIN *****  
  
(prompt "\nTaper  RECT  pour construire un rectangle LONGUEUR/LARGEUR")  
  
(princ)
```

**(Fin)**





# **Les erreurs de syntaxe dans les programmes AutoLISP**

**Si le langage AutoLISP est simple, le développement reste "artisanal". Avec les outils standards mis à notre disposition, il est très fréquent d'avoir des erreurs dans les programmes AutoLISP. Même avec Visual LISP, il faut faire la chasse aux erreurs.**

**Les erreurs les plus fréquentes sont :**

- 1. Fautes de frappe dans le nom d'une fonction ou d'une variable**
- 2. Non respect des paramètres requis par une fonction**
- 3. Non respect du format des données (entier, réel, chaîne)**
- 4. Oublier ou trop de parenthèses**
- 5. Oublier ou trop de guillemets**

**Sans un outil tel que Visual LISP, il existe peu de solution pour trouver ces erreurs de syntaxe. (seul aspect négatif de ce langage).**

**Pour ceux qui ne possède pas l'éditeur Visual LISP, une astuce, consiste à utiliser le compilateur AutoLISP d'une version 11 ou 12. La compilation détecte les erreurs à l'intérieur du programme au moment de sa compilation.**

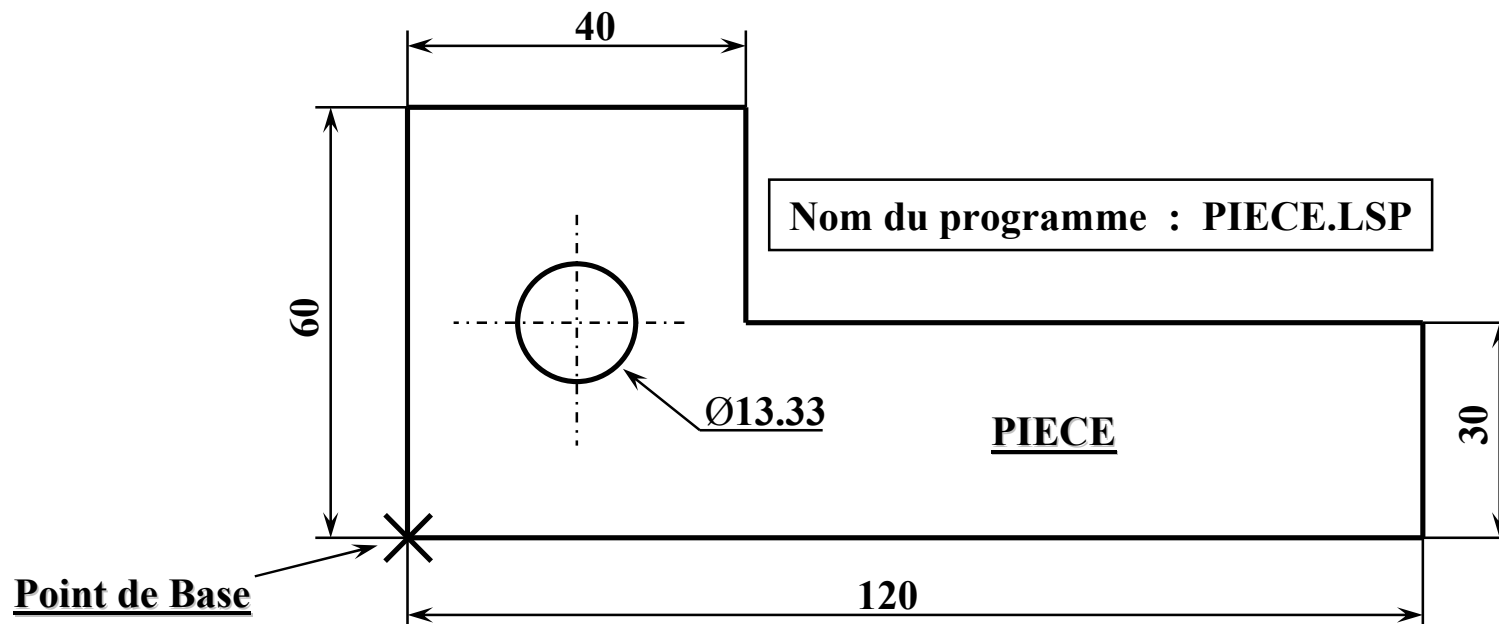




## AutoLISP travaux dirigés, Exercice N° 2 : Construire une pièce paramétrée

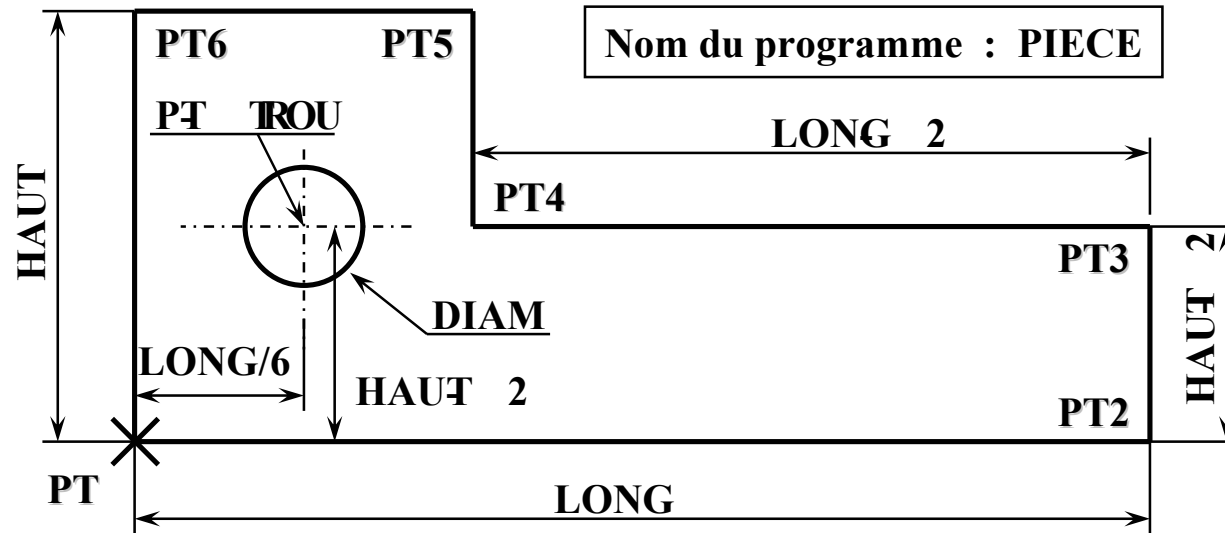
L'exercice consiste à créer une fonction pour **construire une pièce de forme répétitive** en indiquant la **longueur** puis la **hauteur**. L'analyse sommaire du problème me permet d'identifier les points suivants :

1. Je veux créer un fichier programme (AutoLISP)
2. Je veux positionner la pièce en donnant un point à l'écran
3. Je veux définir sa longueur au clavier ou à l'écran
4. Je veux définir sa hauteur au clavier ou à l'écran
5. Je veux définir automatiquement les autres dimensions à partir des contraintes de la pièce
6. Je veux coter automatiquement mon dessin
7. Je veux que la gestion de calques soit faite automatiquement (pièce, axe, cotes, etc)





## AutoLISP travaux dirigés : Définition des variables et contraintes



### Déclaration et définition des éléments et variables :

**PIECE.LSP**

**PT**

**LONG**

**HAUT**

**PT2**

**PT3**

**PT4**

**PT5**

**PT6**

**Nom du fichier AutoLISP**

**Point d'insertion de la pièce**

**Longueur de la pièce**

**Hauteur de la pièce ( ATTENTION :  $HAUT \geq LONG/3$  )**

**coin bas droit de la pièce**

**coin milieu haut droit la pièce**

**coin centre milieu de la pièce**

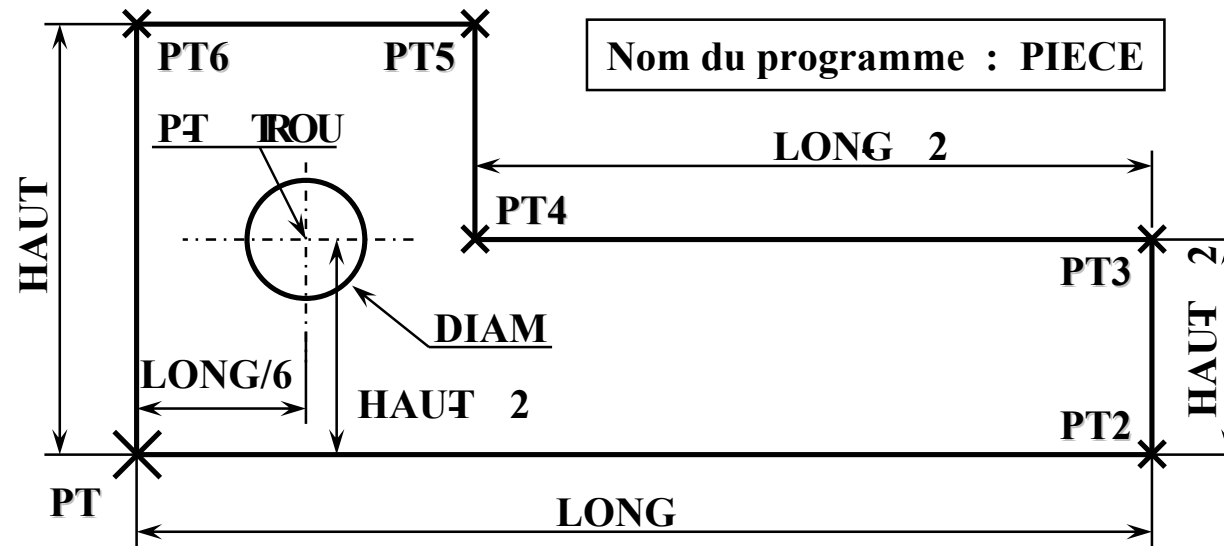
**coin centre haut de la pièce**

**coin haut gauche de la pièce**





## AutoLISP travaux dirigés : Définition des variables et contraintes (suite...)



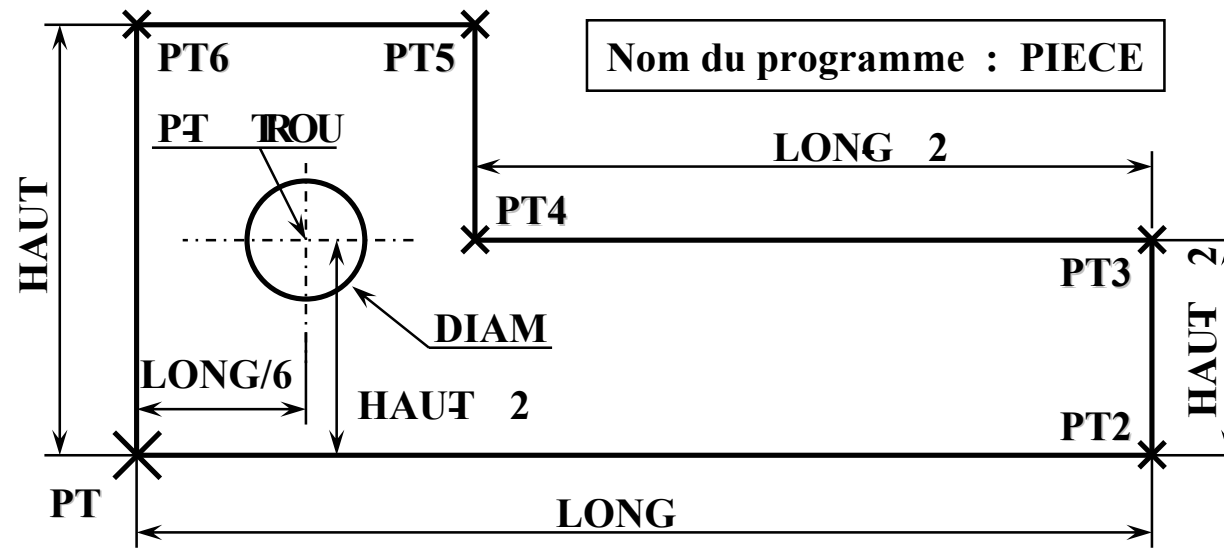
### Définition des contraintes :

HAUT 2	distance PT2 à PT3 = $HAUT / 2$
LONG 2	distance PT3 à PT4 = $LONG * 2/3$
PT TROU	point centre du trou, distance X de PT à PT TROU = $LONG / 6$ distance Y de PT à PT TROU = $HAUT / 2$
DIAM	diamètre du trou = $LONG / 9$





## AutoLISP travaux dirigés : Définition des variables et contraintes (suite...)



### Définition des extrémités des axes

PA1 et PA2

Extrémité gauche et droite de l'axe horizontal du trou

PA3 et PA4

Extrémité bas et haut de l'axe vertical du trou

### Définition des éléments de cotation :

PC G

position cote 10 unités vers la gauche

PC D

position cote 10 unités vers la droite

PC H

position cote 10 unités vers le haut

PC B

position cote 10 unités vers le bas

### Définition des calques :

PIECE

tous les objets de la pièce (couleur BLANC)

AXE

les axes (couleur ROUGE, type de ligne AXES)

COT

toutes les cotes (couleur BLANC, paramétrer le style ISO)







## Travaux dirigés : Fonction conditionnelle

La fonction **IF** permet de vérifier l'état de certaines variables. La condition IF considère généralement deux solutions : SI FAUX alors... ou SI VRAI alors.... Voici quelques exemples :

```
( if ( = N1 0 )
      ( prompt "\nValeur = 0, opération annulée" )      ;;; si VRAI
      ( prompt "\nValeur correcte, opération en cours" )  ;;; si FAUX
    ) ; if
```

**Attention, avec IF si plusieurs fonctions doivent être exécutées dans le cas VRAI ou FAUX, il faut encadrer l'ensemble des fonctions avec PROGN .**

```
( if ( = N1 0 )
      ( prompt "\nValeur = 0, opération annulée" )      ;;; si VRAI
      ( PROGN ( prompt "\nValeur correcte, opération en cours" ) ;;; si FAUX
                (setq VAR1 N1)
                (dessin_rect PT_INS LN LG)
              ) ; progn
    ) ; if
```





## Travaux dirigés : Fonction de boucle

La fonction **WHILE** permet de rester dans une boucle tant que la condition mini maxi n'est pas atteinte. ( TANT QUE... FAIRE...)

```
(setq N 1 MAX 50 )           ;;; affecte les valeurs de TEST

( WHILE      (< N MAX)         ;;; début et condition pour la boucle

      (prompt "\nEtat compteur : ") ;;; affiche à l'écran le texte entre guillemets
      (princ N)                 ;;; affiche à l'écran la valeur de N
      (setq N (1+ N))           ;;; incrémente la valeur de N (+1)

);while                       ;;; fin de la boucle
```

Il est possible à chaque passage de boucle de faire appel à des fonctions.

Une boucle peut également imbriquer d'autres boucles.

AutoLISP quitte la boucle lorsque la condition finale est rencontrée ( dès que  $N = MAX$  ).





## Travaux dirigés : Fonction calcul d'un point "polaire"

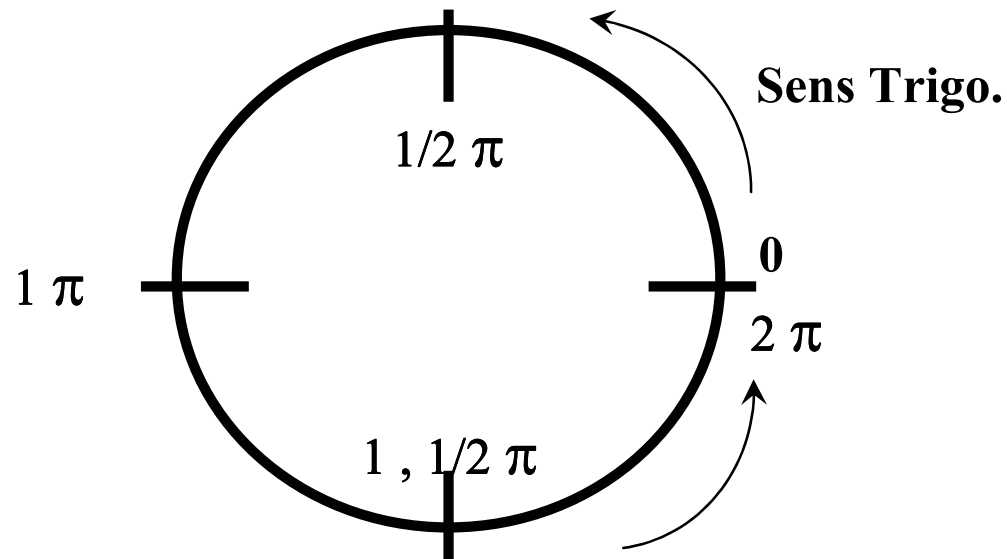
La fonction **POLAR** permet la définition d'un point en indiquant le point de référence l'angle en radians et la distance.

( **POLAR**    **PT**    **ANGLE**    **DISTANCE** )

L'angle doit toujours être exprimé en radians. La référence  $0^\circ$  se trouve toujours à 3H00 et le sens de définition est le sens trigonométrique.

Il existe une variable constante appelée **PI** ( **pi** ). Il est possible d'utiliser directement cette variable pour les calculs d'angle. Valeur de **PI** avec AutoLISP :

**3 , 141 592 653 589 793**





(.../...)



## Programmation AutoLISP de Base



## AutoLISP travaux dirigés : Exemple de Corrigé N° 2 (suite...)

```
;;; Les contraintes
;;; HAUT 2          Distance PT2 PT3 = HAUT / 2
;;;
;;; LONG 2          Distance PT3 PT4 = LONG * 2/3
;;;
;;; PT TROU          Point centre du trou      = LONG / 6 de PT en X
;;;                                     HAUT 2 de PT en Y
;;;
;;; DIAM             Diamètre du trou = LONG / 9

;;; Les traits d'axe
;;; PA1              Extrémité gauche de l'axe horizontal du trou
;;; PA2              Extrémité droite de l'axe horizontal du trou
;;; PA3              Extrémité bas de l'axe vertical du trou
;;; PA4              Extrémité haut de l'axe vertical du trou

;;; La cotation
;;; PE G             Position cote ... gauche (... DIAM de la pièce)
;;; PE D             Position cote ... droite (... DIAM de la pièce)
;;; PE H             Position cote en haut (... DIAM de la pièce)
;;; PE B             Position cote en bas (... DIAM de la pièce)

;;; Les CALQUES
;;; PIECE            La pièce en BLANC
;;; AXE              Les axes en ROUGE / type de ligne AXES
;;; COT              Les cotes en BLANC utilisant le style de cotation ISO
```

(.../...)





## AutoLISP travaux dirigés : Exemple de Corrigé N° 2 (suite...)

```
;;; =====  
;;; Gestion des ERREURS LISP  
;;;  
  
(defun *ERROR*(erreur)          ;;;  définition de la fonction *ERROR*  
    (prompt "\n\t* * * ERREUR * * *")  
    (prompt "\nType = ")  
    (prompt erreur)             ;;;  affiche le type d'erreur  
); defun
```

(.../...)





## AutoLISP travaux dirigés : Exemple de Corrigé N° 2 (suite...)

```
;;; =====  
;;; Fonction principale  
;;;  
  
(defun c:piece( / PT LONG HAUT )  
  
    (setvar "CMDECHO" 1)                ;;; évite l'affichage d'écho des commandes  
    (command "ecranga")                 ;;; bascule en écran graphique  
  
    (setq PT (getpoint "\nCoin bas gauche ")) ;;; saisie du Point d'insertion  
  
    (initget 2)                          ;;; interdit une entrée Zéro ou Nulle  
    (setq LONG (getdist pt "\n\nLongueur pièce: ") ;;; saisie de la longueur  
            HAUT 0)                      ;;; remet haut à Zéro  
    ); setq  
  
    (while (< HAUT (/ LONG 3))            ;;; verrouille la condition HAUT >= LONG/3  
        (prompt "\nHauteur trop faible, Minimum = ") ;;; avertissement à l'utilisateur  
        (princ (/ long 3))                ;;; affiche la valeur mini autorisée  
        (setq HAUT (getdist pt "\n\nLargeur pièce: ")) ;;; demande d'entrer à nouveau la hauteur  
    ); while                               ;;; fin de la boucle  
  
    (desspiece PT LONG HAUT )             ;;; appel de la fonction de dessin  
  
    (princ)                               ;;; évite l'affichage NIL  
  
); defun piece()
```

(.../...)





## AutoLISP travaux dirigés : Exemple de Corrigé N° 2 (suite...)

```
;;; =====  
;;; Sous/Fonction calcul et construction de la pièce  
;;;  
  
(defun desspiece( PT LONG HAUT / PT2 PT3 PT4 PT5 PT6 PT TROU DIAM PL_AXE PL_COT  
                  PA1 PA2 PA3 PA4 PE DPE GPE BPE H )  
  
  (setq PT2 (LIST (+ (car PT) LONG)                ;;; Coin bas droit de la pièce  
                  (cadr PT)  
                  0)  
        PT3 (LIST (car PT2)                          ;;; coin au dessus de PT2  
                  (+ (cadr PT2) (/ HAUT 2))  
                  0)  
        PT4 (LIST (+ (car PT3) (* (/ LONG 3) 2))    ;;; coin milieu centre de la pièce  
                  (cadr PT3)  
                  0)  
        PT5 (LIST (car PT4)                          ;;; coin au dessus de PT4  
                  (+ (cadr PT4) (/ HAUT 2))  
                  0)  
        PT6 (LIST (car PT)                          ;;; coin haut gauche de la pièce  
                  (cadr PT5)  
                  0)  
        PT TROU (LIST (+ (car PT) (/ LONG 6))        ;;; centre du trou  
                      (cadr PT3)  
                      0)  
        DIAM      (/ LONG 9.0)                      ;;; diamètre du trou  
  
        PL_AXE     "axe"                             ;;; nom du calque des traits d'axe  
        PL_COT     "cot"                             ;;; nom du calque des cotations
```

(.../...)







## AutoLISP travaux dirigés : Exemple de Corrigé N° 2 (suite...)

```
PA1 (LIST      { (car PT  TROU) diam)      ;;; extrémité gauche de l'axe horizontal du trou
      (cadr PT  TROU)
      0)

PA2 (LIST      (+ (car PT  TROU) diam)      ;;; extrémité droite de l'axe horizontal du trou
      (cadr PT  TROU)
      0)

PA3 (LIST      (car PT  TROU)              ;;; extrémité bas de l'axe vertical du trou
      { (cadr PT  TROU) diam)
      0)

PA4 (LIST      (car PT  TROU)              ;;; extrémité haut de l'axe vertical du trou
      (+ (cadr PT  TROU) diam)
      0)

);setq

(if (= (getvar "dimscale") 0 )              ;;; vérifie la valeur de l'échelle des cotes
    (setq ecart_cot 10)                    ;;; définition de la valeur de ECART_COT
    (setq ecart_cot (* (getvar "dimscale") 10 ) )
);if

(setq PE  D (LIST      (+ (car PT2) ecart_cot)      ;;; position ligne de cote à droite de la pièce
      (cadr PT2)
      0)

      PE  G (LIST      { (car PT) ecart_cot)        ;;; position ligne de cote à gauche de la pièce
      (cadr PT)
      0)

      PE  B (LIST      (car PT2)                  ;;; position ligne de cote en bas de la pièce
      { (cadr PT2) ecart_cot)
      0)

      PE  H (LIST      (car PT3)                  ;;; position ligne de cote en haut de la pièce
      (+ (cadr PT3) ecart_cot)
      0)

);setq
```

(.../...)





## AutoLISP travaux dirigés : Exemple de Corrigé N° 2 (suite...)

```

(setq  old_osmode      (getvar "OSMODE")      ;;; sauvegarde l'état des accrochages
      old_blipmode    (getvar "blipmode")    ;;; sauvegarde l'état des marques
);setq
(setvar "OSMODE" 0) ;;; annule tous les accrochages
(setvar "BLIPMODE" 0) ;;; annule les marques
(command "CALQUE" "E" "PIECE" "") ;;; crée et rend le calque de la pièce courant
(command "_PLINE" pt "_w" 0 0 PT2 PT3 PT4 PT5 PT6 "c") ;;; dessine la pièce
(command "_CIRCLE" PF TROU "D" DIAM ) ;;; dessine le cercle
(command "_LAYER" "E" "AXE" "tl" "axes" "axe" "co" "1" "axe" "") ;;; crée et rend le calque courant
(command "_LINE" PA1 PA2 "") ;;; dessine l'axe horizontal
(command "_LINE" PA3 PA4 "") ;;; dessine l'axe vertical
(command "_LAYER" "E" "COT" "")
(command "_dim1" "_horizontal" PT PT2 PE B"") ;;; cotation de la pièce
(command "_dim1" "_horizontal" PT6 PT5 (polar pt6 (/ pi 2.0) ecart_cot) "") ;;; cotation de la pièce
(command "_dim1" "_vertical" PT PT6 PE G"") ;;; cotation de la pièce
(command "_dim1" "_vertical" PT2 PT3 PE D"") ;;; cotation de la pièce
(command "_dim1" "_diameter" (polar PF TROU (* (/ pi 4) 7) (/ diam 2.0) ) ;;; cotation de la pièce
      ""
      (polar PF TROU (* (/ pi 4) 7) diam )

);command

(setvar "OSMODE" old_osmode) ;;; restitue les anciens d'accrochages
(setvar "BLIPMODE" old_blipmode) ;;; restitue les marques
);defun despiece()

;;; =====
;;; Message affiché au chargement du programme
;;;
(prompt "\n\nTaper: PIECE pour dessiner une piece") ;;; message d'aide au chargement du programme
(princ) ;;; évite l'affichage NIL au chargement du prog.

```

**(Fin)**

