

# AUTOLISP

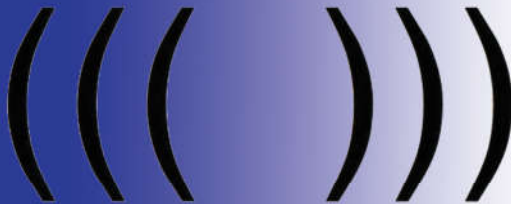
لیسپ یکی از زبان‌های برنامه‌نویسی کامپیوتر است و به خاطر مناسب بودن آن با اتوکد، این زبان را برای اتوکد انتخاب کرده‌اند. می‌توان گفت بهترین زبان برای برنامه‌نویسی در اتوکد، زبان برنامه‌نویسی اتولیسپ است چون این زبان به راحتی می‌تواند چند عنصر را در یک متغیر تعریف کند (مانند مختصات X,Y,Z یک نقطه) و در هر زمان می‌توانید به هر کدام از عناصر داخل متغیر دسترسی داشته باشید. البته این مزیت یکی از مزیت‌هایی است که این زبان نسبت به زبان‌های برنامه‌نویسی دیگر دارد.

کامل‌ترین مرجع کاربردی اتولیسپ (برنامه‌نویسی در اتوکد) ترجمه و تألیف: مهندس رسول محمدی

کامل‌ترین مرجع کاربردی

## اتولیسپ

(برنامه‌نویسی در اتوکد)



ترجمه و تألیف:

مهندس رسول محمدی



# **کامل ترین مرجع کاربردی اتولیسپ ( برنامه نویسی در اتوکد )**

**تألیف و ترجمه:**

مهندس رسول محمدی

**سال انتشار:** زمستان ۱۳۹۷

**سایت منتشر کننده:**

<https://www.selfamooz.ir>

**طراح جلد:**

مونا جواهری

تقدیم به پدر و مادرم

پدر و مادر، همسر

که عشق را معنای دوباره بخشیدند

## مقدمه

اگر بخواهم زبان برنامه‌نویسی اتولیسپ برای اتوکد را تعریف کنم، می‌توانم بگویم اتولیسپ زبانی است که با علم به آن می‌توانید از کارهای تکراری کاملاً جلوگیری کنید و خطای انسانی را به صفر برسانید و تا ۹۰ درصد در زمان صرفه‌جویی کنید. در بازار کار امروز کسانی موفق هستند که بتوانند کاری که به آنها محول می‌شود را در زمان کمتر و با هزینه کمتر و همین‌طور با دقت بیشتری انجام دهند. یکی از راه‌های رسیدن به این ۳ مهم، داشتن علم برنامه‌نویسی می‌باشد.

لیسپ یکی از زبان‌های برنامه‌نویسی کامپیوتر است و به خاطر مناسب بودن آن با اتوکد، این زبان را برای اتوکد انتخاب کرده‌اند. می‌توان گفت بهترین زبان برای برنامه‌نویسی در اتوکد، زبان برنامه‌نویسی اتولیسپ است چون این زبان به راحتی می‌تواند چند عنصر را در یک متغیر تعریف کند (مانند مختصات  $X,Y,Z$  یک نقطه) و در هر زمان می‌توانید به هر کدام از عناصر داخل متغیر دسترسی داشته باشید. البته این مزیت یکی از مزیت‌هایی است که این زبان نسبت به زبان‌های برنامه‌نویسی دیگر دارد.

اصولاً کتاب آموزشی می‌تواند سه مشخصه داشته باشد. این سه مشخصه عبارتند از: آموزش محور، مثال محور و پروژه محور. در مورد مشخصه آموزش محور بودن نیز دو مشخصه دیگر وجود دارد که عبارتند از: دانش محور بودن و مهارت محور بودن. در این کتاب بعد از توضیحات ابتدایی، توابع مختلف به طور کامل توضیح داده شده است (آموزش محور) و همین‌طور برای هر تابع یک مثال از کاربرد آن تابع آورده شده است (مثال محور). در فصل آخر، ۱۲ برنامه کاربردی نوشته شده است و به توضیح جزء به جزء آن‌ها پرداخته شده است (پروژه محور).

در مورد مشخصه‌های آموزش محور بودن (دانش محور و مهارت محور) کتاب نیز ذکر این نکته خالی از لطف نیست که بعد از خواندن و انجام مثال‌ها و تمرینات این کتاب، هم دانش شما عزیزان نسبت به برنامه‌نویسی بالا رفته و هم به مهارت قابل قبولی در این زمینه دست خواهید یافت.

در تألیف این کتاب سعی شده از گفتن مطالب اضافی صرف نظر شود و فقط مطالبی گفته شود که در رشد آموزش و مهارت در برنامه‌نویسی شما عزیزان مفید باشد.

این کتاب برای افرادی مناسب است که به مهارت نسبی در به کار بردن از دستورات اتوکد رسیده باشند و به خوبی

بتوانند از دستورات اتوکد استفاده کنند و درک درستی از این نرم افزار داشته باشند. همچنین افرادی که به برنامه نویسی علاقه دارند و خواهان برنامه نویسی در اتوکد هستند. اما از همه مهم تر این کتاب برای کسانی مؤثر است که به دنبال ذخیره وقتشان هستند و می خواهند در کمترین زمان ممکن به بیشترین راندمان و نتیجه دلخواه در زمینه ترسیم در اتوکد دست یابند.

در انتهای هر فصل، نقشه ذهنی آن فصل ارائه شده است. اگر بخواهم نقشه ذهنی را تعریف کنم به شرح زیر است: نقشه ذهنی، نموداری درختی است که برای بیان کردن کلمات، ایده ها، فعالیت ها یا موارد دیگر مربوط به یک کلید واژه یا ایده مورد استفاده قرار می گیرد. این نمودار برای تولید، سازماندهی و ساختاردهی ایده ها و افکار به کار بسته می شود و در حل مسئله، فرآیند تصمیم گیری و نوشتن می تواند مورد استفاده قرار بگیرد. عناصر یک نقشه ذهنی به صورت ادراکی و بر اساس اهمیت مطالب در قالب گروه ها، شاخه ها و قسمت های متفاوت قرار می گیرند. نقشه های ذهنی را می توان برای مرور حافظه نیز به کار بست.

از دلایل استفاده نقشه ذهنی، درگیر کردن هر دو نیمکره مغز برای به خاطر سپاری بهتر است.

در ادامه توضیحات کوتاهی در مورد هر فصل آمده که به شما دیدی کلی در مورد هر فصل می دهد:

**فصل اول – آشنایی با محیط برنامه نویسی اتولیسپ:** قبل از استفاده از هر محیطی بهتر است آن را شناخته تا بهتر بتوان با آن ارتباط برقرار کرد. چگونگی وارد شدن به محیط و نحوه بارگذاری فایل های لیسپ و همچنین نکاتی در مورد شروع برنامه نویسی، از مطالب این فصل می باشد.

**فصل دوم – توابع و کلیات اولیه برنامه نویسی:** نحوه شروع برنامه و استفاده از توابع اولیه برای نوشتن برنامه های کوچک و ابتدایی از مطالب این فصل می باشد.

**فصل سوم – کار با لیست های کوچک:** لیست های کوچک مثل مختصات یک نقطه (که شامل سه عنصر است) از پرکاربردترین لیست هایی است که در برنامه نویسی لیسپ با آنها سر و کار داریم. طریقه استفاده از هر یک از عناصر لیست های کوچک و چگونگی ساخت یک لیست، مطالبی هستند که در این فصل می آموزید.

**فصل چهارم: انواع داده در اتولیسپ:** از نظر کاربران برنامه‌ای مفید است که بتوان به بهترین شکل ممکن ورودی‌ها را از کاربر دریافت کند و از آن در برنامه استفاده کند.

در این فصل بعد از اینکه با انواع داده آشنا شدید، با توابعی آشنا می‌شوید که انواع داده‌ها را از کاربر گرفته و از آنها در برنامه استفاده می‌کنند. همین‌طور با توابع شرطی و ساخت حلقه به طور کامل آشنا خواهید شد.

**فصل پنجم – توابع تبدیل:** در این فصل توابعی توضیح داده می‌شوند که بتوان با آنها، واحدهای مختلف (طول و زاویه) را به هم تبدیل کرد. توابع چاپ متن نیز آموزش داده می‌شود.

همچنین توابعی آموزش داده می‌شود که بتوان با آنها داده‌های مختلف را به هم تبدیل کرد.

**فصل ششم – کار با متغیرهای سیستمی:** متغیرهای سیستمی تنظیمات و رفتار دستورات و پنجره‌های اتوکد را کنترل می‌کنند و با روشن، خاموش و یا عدد تنظیم می‌شوند. طریقه تنظیم آنها و دریافت اطلاعات، از مطالبی است که در این فصل می‌آموزید.

همچنین گدهایی می‌آموزید که بتوانید با وارد کردن آنها از Object Snap ها در برنامه‌تان استفاده کنید.

**فصل هفتم – کار با لیست موجودیت موضوع:** شاید برایتان سؤال ایجاد شود که اگر بخواهیم موضوعاتی که قبلاً ترسیم شده را با برنامه‌ای که می‌نویسیم انتخاب و تغییر دهیم چه باید بکنیم؟ نگران این موضوع نباشید، زیرا در این فصل توابعی می‌آموزید که بتوان موضوعات مختلف را در قسمت‌های مختلف اتوکد انتخاب کرده و موجودیت آنها را به دست آورده و مطابق با خواسته خود تغییر دهید.

**فصل هشتم – کار با فایل:** یکی از خاصیت‌های برنامه‌های پویا و حرفه‌ای این است که بتوان با آن به فایل‌های خارجی مانند فایل‌های اتوکد (dwg)، فایل‌های لیسپ (lsp)، فایل‌های Notepad (txt)، فایل‌های اکسل (xls یا xlsx) و ... دسترسی پیدا کرد. ساخت فایل، ویرایش، پاک کردن موضوعات داخل آن و ... از مطالبی است که در این فصل خواهید آموخت.

همچنین توابع جستجو در جدول (لایه‌ها، استایل نوشته و ...)، از توابعی است که در این فصل می‌آموزید.

**فصل نهم – اشکال زدایی:** توابعی که در این فصل فرا می‌گیرید کمک می‌کند که خطا در برنامه‌تان را به حداقل برسانید و راحت‌تر بتوانید ایراد برنامه‌تان را بیابید.

**فصل دهم – توابع تکمیلی:** در این فصل توابع زیادی را فرا خواهید گرفت که با دانستن آنها زودتر به هدف‌تان در برنامه‌نویسی می‌رسید.

**فصل یازدهم – چند برنامه کاربردی:** فصل پایانی کتاب به نوشتن و توضیح جزء به جزء چندین برنامه اختصاص یافته است. این فصل به شما کمک می‌کند که چگونگی استفاده از توابع و دستورات مختلف را در برنامه‌نویسی درک کنید.

در انتها اگر سؤال، پیشنهاد یا انتقاد سازنده‌ای در مورد کتاب داشتید، می‌توانید با ایمیل [Rasoul.mohammadi62@gmail.com](mailto:Rasoul.mohammadi62@gmail.com) و یا شماره ۰۹۱۲۶۲۲۷۲۱۰ در ارتباط باشید.



## فهرست

فصل ۱ - آشنایی با محیط برنامه‌نویسی اتولیسپ	۱
۱-۱ - وارد شدن به محیط برنامه‌نویسی اتولیسپ	۱
۲-۱ - پرانتزها	۳
۳-۱ - دو راه دیگر برای یافتن پرانتز مورد نظر (Parentheses Matching)	۴
۴-۱ - بارگذاری فایل‌های Lisp و Macro	۵
فصل ۲ - توابع و کلیات اولیه برنامه‌نویسی	۱۲
۱-۲ - تابع Defun	۱۲
۲-۲ - علامت دابل کوتیشن (")	۱۳
۳-۲ - متغیرها	۱۳
۱-۳-۲ - نمونه‌هایی از نام متغیرها	۱۴
۴-۲ - لیست	۱۵
۵-۲ - علامت سمی کالِن (;)	۱۶
۶-۲ - تابع Prompt	۱۶
۷-۲ - تابع (Terpri) و یا n\	۱۷
۸-۲ - تابع Setq	۱۸
۹-۲ - تابع Getpoint	۲۰
۱۰-۲ - تابع Command	۲۰
۱۱-۲ - تابع Getcorner	۲۱
فصل ۳ - کار با لیست‌های کوچک	۲۵
۱-۳ - تابع Car	۲۵
۲-۳ - تابع Cadr	۲۶
۳-۳ - تابع Caddr	۲۶
۴-۳ - تابع List	۲۷
۵-۳ - تابع Cdr	۲۹
فصل ۴ - انواع داده در اتولیسپ	۳۲

۳۲	۱-۴ - انواع داده.....
۳۳	۲-۴ - Nil.....
۳۳	۳-۴ - تابع GetString.....
۳۴	۴-۴ - تابع Getreal.....
۳۵	۵-۴ - تابع Getint.....
۳۵	۶-۴ - تابع Getdist.....
۳۶	۷-۴ - چهار عمل اصلی.....
۳۷	۸-۴ - تابع Abs.....
۳۸	۹-۴ - تابع Sqrt.....
۳۸	۱۰-۴ - عملگرها.....
۳۸	۱-۱۰-۴ - عملگر مساوی (=).....
۳۸	۲-۱۰-۴ - عملگر کوچک تر (<).....
۳۹	۳-۱۰-۴ - عملگر کوچک تر مساوی (<=).....
۳۹	۴-۱۰-۴ - عملگر بزرگ تر (>).....
۳۹	۵-۱۰-۴ - عملگر بزرگ تر مساوی (>=).....
۳۹	۶-۱۰-۴ - عملگر نامساوی (/).....
۳۹	۱۱-۴ - تابع If,Then,Else.....
۴۱	۱۲-۴ - رابط‌های منطقی And و Or.....
۴۲	۱۳-۴ - تابع Cond.....
۴۲	۱۴-۴ - تابع While.....
۴۷	فصل ۵ - توابع تبدیل.....
۴۷	۱-۵ - تابع Angle.....
۴۸	۲-۵ - نحوه محاسبه زاویه توسط برنامه.....
۴۹	۳-۵ - تابع Getangle.....
۵۰	۴-۵ - تابع Getorient.....
۵۱	۵-۵ - تبدیل واحدها در اتوکد (طول و زاویه).....
۵۲	۶-۵ - تفاوت توابع Getangle و Getorient.....

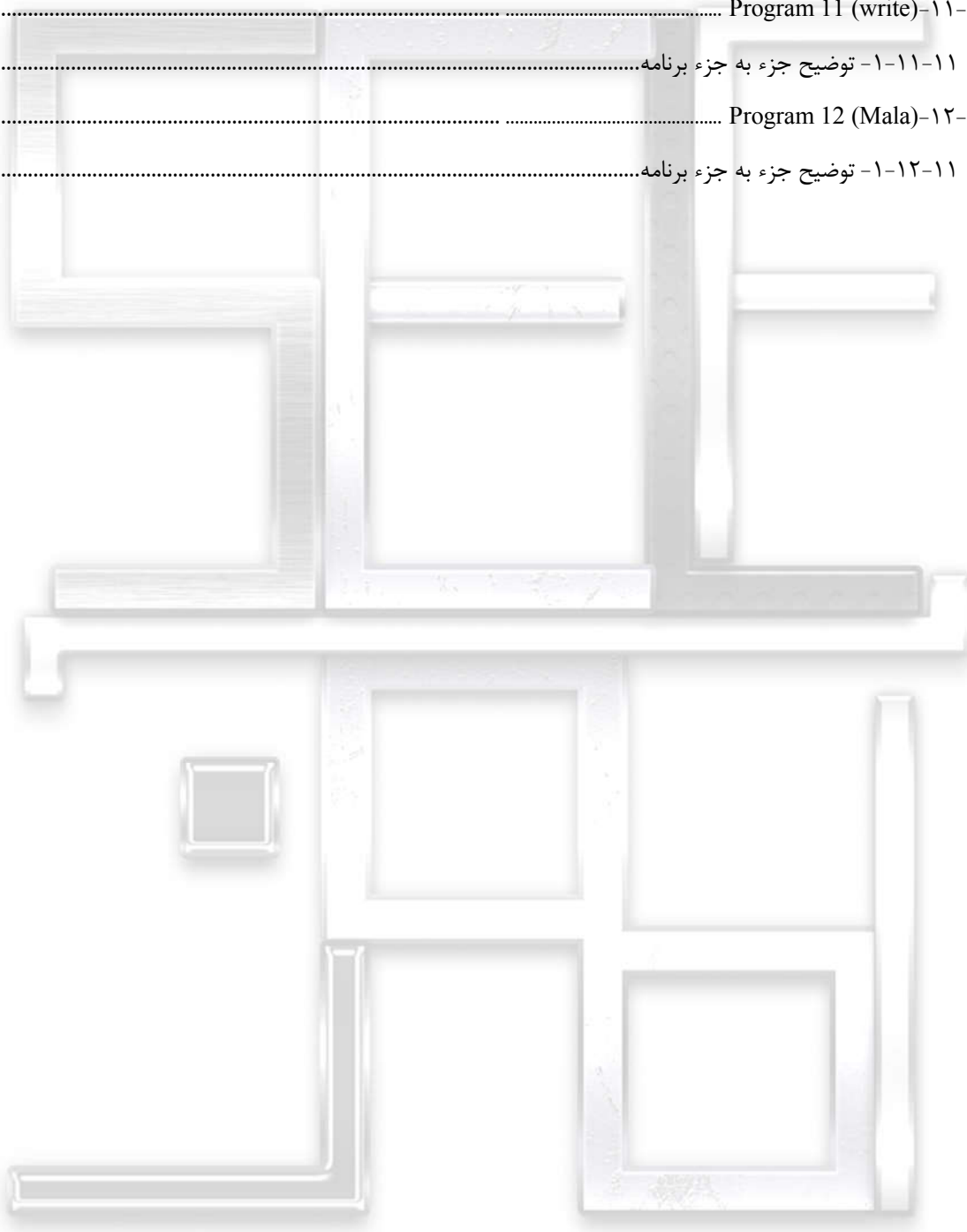
۵۴	.....	Angtos تابع	۷-۵
۵۶	.....	Rtos تابع	۸-۵
۵۷	.....	Distof تابع	۹-۵
۵۸	.....	۱۰-۵-توابع چاپ متن	
۵۸	.....	Princ تابع	۱۱-۵
۵۹	.....	Atof تابع	۱۲-۵
۶۰	.....	Atoi تابع	۱۳-۵
۶۱	.....	Itoa تابع	۱۴-۵
۶۴	.....	فصل ۶- کار با متغیرهای سیستمی	
۶۴	.....	Setvar تابع	۱-۶
۶۵	.....	Getvar تابع	۲-۶
۶۵	.....	متغیرهای سیستمی	۳-۶
۶۶	.....	Aunits متغیر سیستمی	۱-۳-۶
۶۶	.....	Auprec متغیر سیستمی	۲-۳-۶
۶۶	.....	Lunits متغیر سیستمی	۳-۳-۶
۶۷	.....	Luprec متغیر سیستمی	۴-۳-۶
۶۷	.....	Orthomode متغیر سیستمی	۵-۳-۶
۶۷	.....	Blipmode متغیر سیستمی	۶-۳-۶
۶۸	.....	Osmode متغیر سیستمی	۷-۳-۶
۶۸	.....	Dragmode متغیر سیستمی	۸-۳-۶
۶۸	.....	Strcat تابع	۴-۶
۶۹	.....	Strcase تابع	۵-۶
۷۲	.....	۶-۶- کدهای دستورات کمکی (Object Snap)	
۷۸	.....	فصل ۷- کار با لیست موجودیت موضوع	
۷۸	.....	Ssget تابع	۱-۷
۸۳	.....	Entsel تابع	۲-۷
۸۳	.....	Entlast تابع	۳-۷

۸۳	تابع Sslength	۴-۷
۸۴	تابع Ssname	۵-۷
۸۵	تابع Entget	۶-۷
۸۶	کدهای وابسته موضوعات	۷-۷
۸۷	تابع Assoc	۸-۷
۸۷	تابع Cons	۹-۷
۸۹	تابع Subst	۱۰-۷
۹۰	تابع Entmod	۱۱-۷
۹۱	تابع Entdel	۱۲-۷
۹۱	تابع Entmake	۱۳-۷
۹۶	کار با فایل	فصل ۸
۹۶	تابع Repeat	۱-۸
۹۷	تابع Progn	۲-۸
۹۸	تابع Open	۳-۸
۹۹	تابع Close	۴-۸
۱۰۰	تابع Read-Line	۵-۸
۱۰۱	تابع Write-line	۶-۸
۱۰۱	جدول نمادها	۷-۸
۱۰۲	تابع Tblsearch	۸-۸
۱۰۳	تابع Tblnext	۹-۸
۱۰۴	تابع Tblobjname	۱۰-۸
۱۰۷	اشکال زدایی برنامه	فصل ۹
۱۰۸	تابع Initget	۱-۹
۱۱۱	توضیحی کوتاه در مورد بیت‌ها	۲-۹
۱۱۳	تابع Getkword	۳-۹
۱۱۵	تابع Findfile	۴-۹
۱۱۶	تابع Getfiled	۵-۹

۱۱۹	.....Nth تابع ۹-۶
۱۲۲	.....فصل ۱۰-توابع تکمیلی
۱۲۲	.....Member تابع ۱۰-۱
۱۲۳	.....Append تابع ۱۰-۲
۱۲۳	.....Mapcar تابع ۱۰-۳
۱۲۴	.....Apply تابع ۱۰-۴
۱۲۵	.....Lambda تابع ۱۰-۵
۱۲۶	.....Foreach تابع ۱۰-۶
۱۲۷	.....۷-کُد 4-۷
۱۲۸	.....Strlen تابع ۱۰-۸
۱۲۸	.....Length تابع ۱۰-۹
۱۲۹	.....Substr تابع ۱۰-۱۰
۱۳۰	.....Chr تابع ۱۱-۱۰
۱۳۰	.....Ascii تابع ۱۲-۱۰
۱۳۰	.....Read-char تابع ۱۳-۱۰
۱۳۱	.....Write-char تابع ۱۴-۱۰
۱۳۲	.....Acad_strlsort تابع ۱۵-۱۰
۱۳۳	.....Alert تابع ۱۶-۱۰
۱۳۳	.....Atan تابع ۱۷-۱۰
۱۳۴	.....Cos تابع ۱۸-۱۰
۱۳۴	.....Cvunit تابع ۱۹-۱۰
۱۳۵	.....Distance تابع ۲۰-۱۰
۱۳۶	.....Expt تابع ۲۱-۱۰
۱۳۶	.....Fix تابع ۲۲-۱۰
۱۳۶	.....Float تابع ۲۳-۱۰
۱۳۷	.....Gcd تابع ۲۴-۱۰
۱۳۷	.....Graphscr تابع ۲۵-۱۰

۱۳۷	.....Initdia تابع ۱۰-۲۶
۱۳۸	.....Last تابع ۱۰-۲۷
۱۳۹	.....Log تابع ۱۰-۲۸
۱۳۹	.....Max تابع ۱۰-۲۹
۱۴۰	.....Min تابع ۱۰-۳۰
۱۴۰	.....Revers تابع ۱۰-۳۱
۱۴۱	.....Sin تابع ۱۰-۳۲
۱۴۱	.....Textscr تابع ۱۰-۳۳
۱۴۵	فصل ۱۱- چند برنامه کاربردی
۱۴۵	..... Program 1 (PG)-۱۱-۱
۱۴۵	..... توضیح جزء به جزء برنامه ۱۱-۱-۱
۱۴۶	..... Program 2 (FS)-۱۱-۲
۱۴۷	..... توضیح جزء به جزء برنامه ۱۱-۲-۱
۱۴۷	..... Program 3 (Drawre)-۱۱-۳
۱۴۸	..... توضیح جزء به جزء برنامه ۱۱-۳-۱
۱۴۹	..... Program 4 (Square)-۱۱-۴
۱۵۰	..... توضیح جزء به جزء برنامه ۱۱-۴-۱
۱۵۱	..... Program 5 (DC)-۱۱-۵
۱۵۲	..... توضیح جزء به جزء برنامه ۱۱-۵-۱
۱۵۳	..... Program 6 (DF)-۱۱-۶
۱۵۴	..... توضیح جزء به جزء برنامه ۱۱-۶-۱
۱۵۶	..... Program 7 (Adim)-۱۱-۷
۱۵۶	..... توضیح جزء به جزء برنامه ۱۱-۷-۱
۱۵۸	..... Program 8 (Extodxf)-۱۱-۸
۱۵۹	..... توضیح جزء به جزء برنامه ۱۱-۸-۱
۱۶۱	..... Program 9 (TZ)-۱۱-۹
۱۶۲	..... توضیح جزء به جزء برنامه ۱۱-۹-۱

۱۶۴.....	Program 10 (RT)-۱۰-۱۱
۱۶۴.....	۱-۱۰-۱۱- توضیح خط به خط برنامه
۱۶۷.....	Program 11 (write)-۱۱-۱۱
۱۶۷.....	۱-۱۱-۱۱- توضیح جزء به جزء برنامه
۱۶۸.....	Program 12 (Mala)-۱۲-۱۱
۱۶۹.....	۱-۱۲-۱۱- توضیح جزء به جزء برنامه





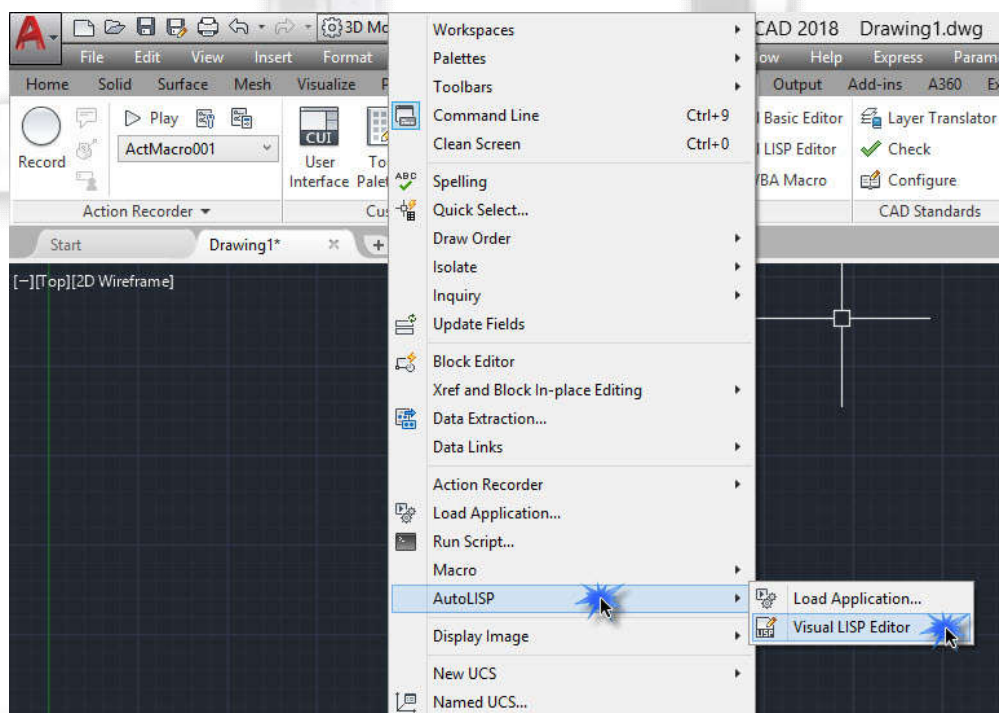
## فصل ۱- آشنایی با محیط برنامه‌نویسی اتولیسپ

مطالبی که در این فصل می‌آموزید:

- ✓ چگونگی وارد شدن به محیط برنامه‌نویسی
- ✓ نحوه بارگذاری فایل‌های لیسپ و ماکرو
- ✓ همچنین نکاتی در مورد شروع برنامه‌نویسی

### ۱-۱- وارد شدن به محیط برنامه‌نویسی اتولیسپ

اگر از اتوکدهای نسخه قدیمی یا کلاسیک استفاده کنید، از طریق منوی Tools و مسیر زیر وارد محیط برنامه‌نویسی شوید:



شکل ۱-۱: نحوه وارد شدن به محیط برنامه‌نویسی در اتوکدهای نسخه قدیمی





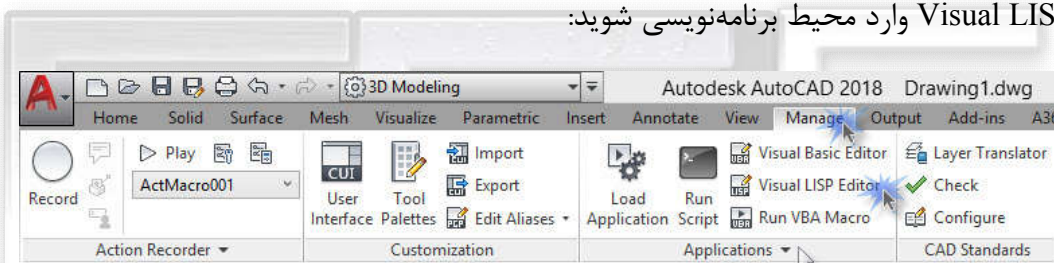
فصل ۱- آشنایی با محیط برنامه‌نویسی  
اتولیسپ

تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

کامل‌ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه‌نویسی در اتوکد)

اگر از اتوکدهای نسخه جدید استفاده کنید، از طریق سربرگ Manage و پنل Applications و انتخاب دستور

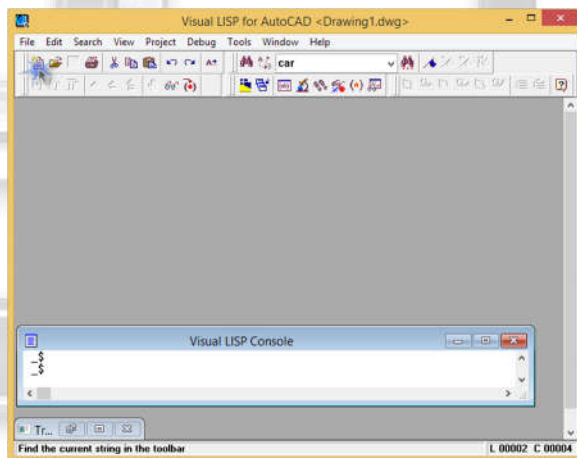
Visual LISP Editor وارد محیط برنامه‌نویسی شوید:



شکل ۱-۲: نحوه وارد شدن به محیط برنامه‌نویسی در اتوکدهای نسخه جدید

همین‌طور می‌توانید با نوشتن دستور VLIDE در خط فرمان و فشردن Enter، وارد محیط برنامه‌نویسی شوید.

وقتی وارد این محیط شوید با یک پنجره خالی که امکان نوشتن برنامه را ندارد مواجه خواهید شد.



شکل ۱-۳: پنجره خالی و باز کردن صفحه جدید

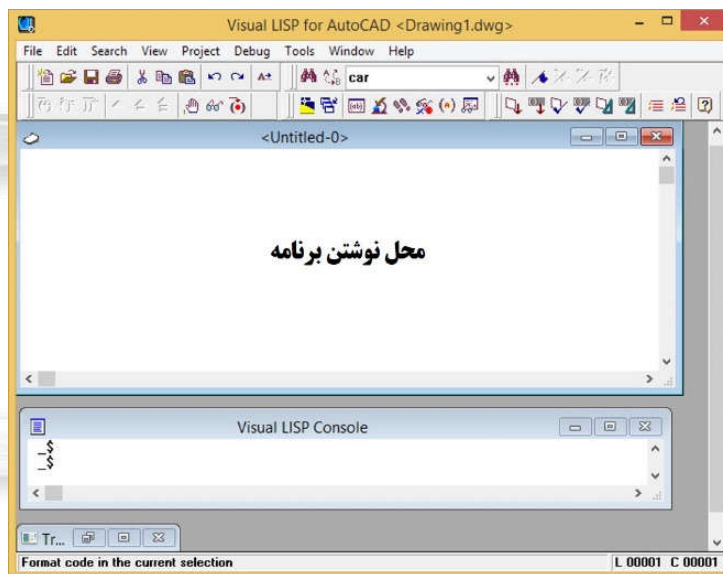
برای اینکه یک صفحه جدید برای برنامه‌نویسی ایجاد کنید، روی دکمه New کلیک کنید.



کامل ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

فصل ۱- آشنایی با محیط برنامه نویسی  
اتولیسپ



شکل ۱-۴: محل نوشتن برنامه

## ۱-۲- پرانتزها

در برنامه نویسی اتولیسپ، نوشتن پرانتزها در ابتدا و انتهای هر خط برنامه فوق العاده ضروری است (اگر پرانتز کم یا زیاد باشد، برنامه اجرا نمی شود)؛ به طوری که باید (تأکید می کنم، باید) در ابتدای هر جمله از برنامه، پرانتز را باز و بعد از تمام شدن آن، پرانتز را ببندید.

برای روشن شدن این موضوع، در صفحه برنامه یک پرانتز باز کرده و با دکمه Enter، کُرِسِر را به چند خط پایین تر منتقل کنید. حالا پرانتز را ببندید. به محض بستن پرانتز، کُرِسِر در یک لحظه به پشت پرانتزی که در ابتدا نوشته بودید منتقل می شود.

مزیت این کار در این است که متوجه شوید آن جمله از برنامه از کجا شروع شده است و کمک می کند که بهتر برنامه تان را اشکال زدایی کنید.



کامل ترین مرجع کاربردی آموزش

اتولیسیپ

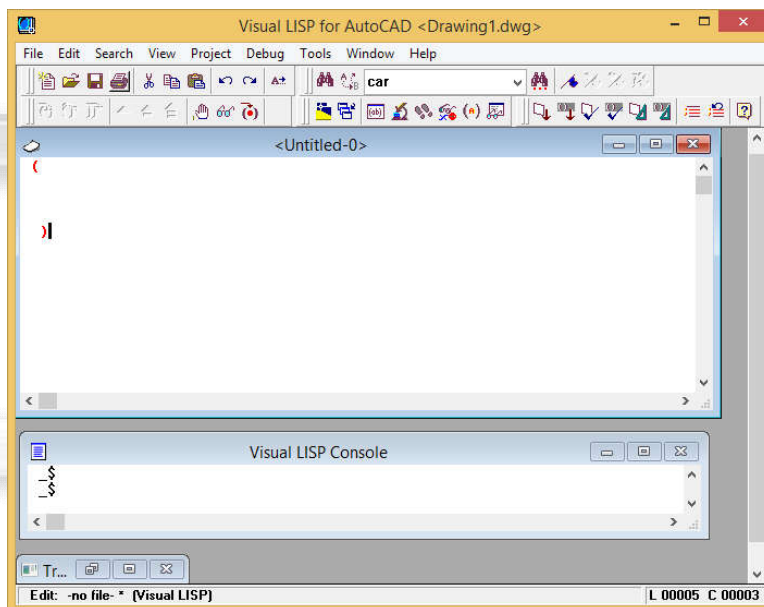
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی

۰۹۱۲۶۲۲۷۲۱۰

فصل ۱- آشنایی با محیط برنامه نویسی

اتولیسیپ



شکل ۱-۵: نوشتن پرانتز در برنامه

**نکته ۱-۱:** اتولیسیپ امکان استفاده از چندین پرانتز را در داخل هم فراهم کرده است. به جمله زیر توجه کنید:

`(setq a (list 2 4 5))`

**نکته ۱-۲:** به منطق کردن دو پرانتز در ابتدا و انتهای جمله‌ها اصطلاحاً *Parentheses Matching* گفته می‌شود.

### ۱-۳- دو راه دیگر برای یافتن پرانتز مورد نظر (*Parentheses Matching*)

۱- کُرسر را به پشت اولین پرانتز برده و دوبار کلیک چپ کنید. با این کار همه موضوعات بین دو پرانتز

انتخاب می‌شوند و متوجه می‌شوید که آن جمله برنامه از کجا شروع و در کجا به پایان رسیده است.

**نکته ۱-۳:** اگر کُرسر را به پشت آخرین پرانتز ببرید و همین کار را انجام دهید، نتیجه یکی است.



کامل ترین مرجع کاربردی آموزش

اتولیسپ

(برنامه نویسی در اتوکد)

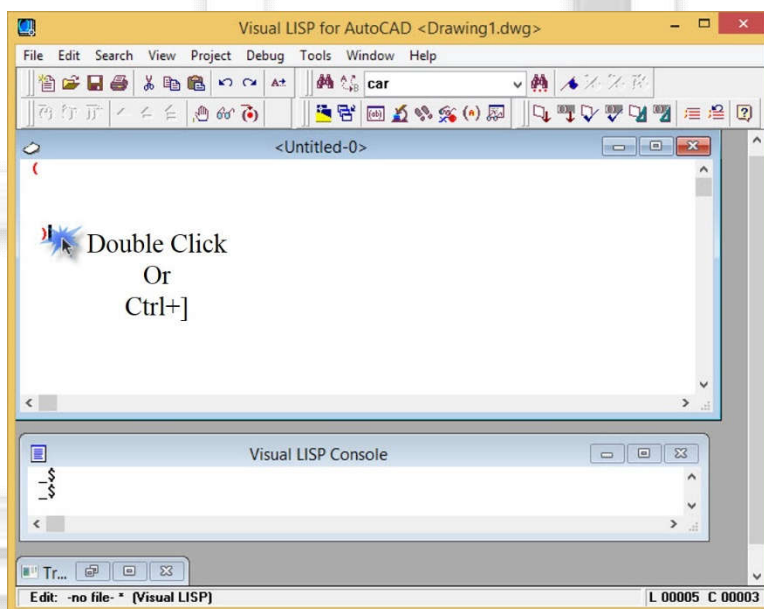
تألیف و ترجمه: رسول محمدی

۰۹۱۲۶۲۲۷۲۱۰

فصل ۱- آشنایی با محیط برنامه نویسی

اتولیسپ

۲- کُرِسِر را به پشت پرانتز باز ببرید و کلیدهای ترکیبی `[+Ctrl]` را با هم فشار دهید. مشاهده خواهید کرد که کُرِسِر به انتهای آن جمله منتقل می شود. حال اگر کلیدهای ترکیبی `[+Ctrl]` را با هم فشار دهید، کُرِسِر به سر جای خود (اول آن جمله) باز می گردد.



شکل ۱-۶: Parentheses Matching

#### ۴-۱ بارگذاری فایل های Lisp و Macro

در مورد برنامه لیسپ، قبلاً و در ابتدای کتاب توضیحاتی ارائه شد. فایل های ماکرو نیز از نظر کارکرد شبیه به فایل های لیسپ می باشند، با این تفاوت که برنامه های ماکرو با زبان برنامه نویسی ویژوال بیسیک نوشته می شوند. همین طور از نظر رابط کاربری بهتر از برنامه های لیسپ می باشند چون در ماکرونویسی می توانید برای ارتباط با کاربر به جای دریافت اطلاعات در خط فرمان از پنجره های طراحی شده استفاده کنید.

البته در برنامه نویسی لیسپ نیز می توان برای وارد کردن اطلاعات، پنجره طراحی کرد ولی نیاز به آشنا بودن سطحی به برنامه نویسی با زبان C می باشید. برای آشنا شدن با این نوع برنامه نویسی، DLC AutoLISP را در



فصل ۱- آشنایی با محیط برنامه‌نویسی  
اتولیسپ

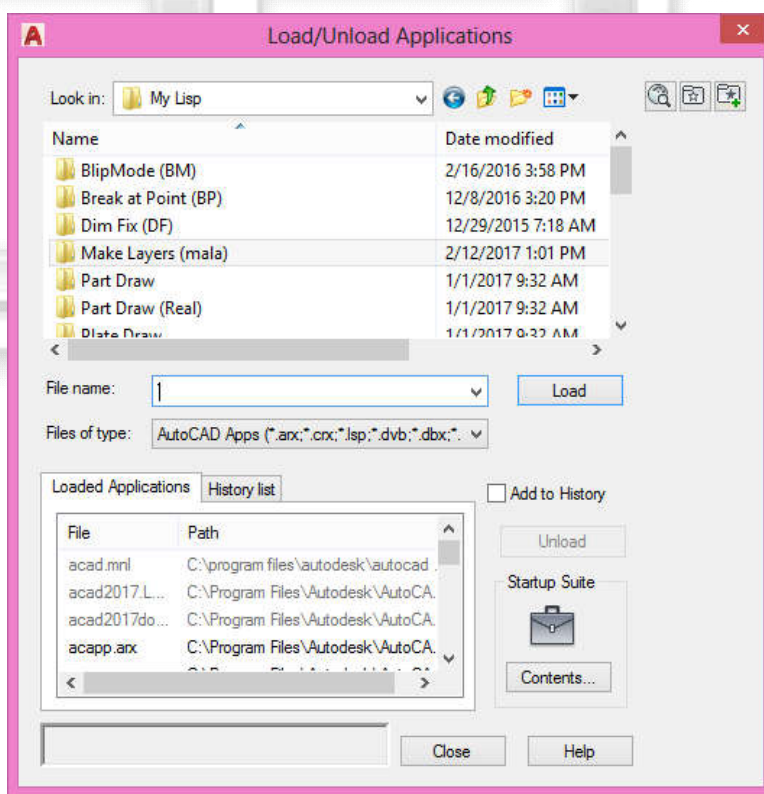
تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

کامل‌ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه نویسی در اتوکد)

اینترنت جستجو کنید.

وقتی برنامه را نوشتید، برای استفاده دائم یا موقت از برنامه باید آن را بارگذاری کنید. در اتوکدهای نسخه جدید برای بارگذاری کردن فایل‌های Lisp و Macro در اتوکد، ابتدا از سربرگ Manage، پَنِل Applications، دستور Load Application را اجرا کنید و یا ap را در خط فرمان تایپ کنید (که روش ساده‌تری است) و اینتر را فشار دهید.

پنجره Load/Unload Applications باز می‌شود.



شکل ۱-۷: Load/Unload Applications

نکته ۱-۴: با نوشتن ap در خط فرمان در همه نسخه‌های اتوکد، پنجره Load/Unload Applications باز می‌شود.





کامل ترین مرجع کاربردی آموزش

اتولیسپ

(برنامه نویسی در اتوکد)

فصل ۱- آشنایی با محیط برنامه نویسی

اتولیسپ

تألیف و ترجمه: رسول محمدی

۰۹۱۲۶۲۲۷۲۱۰

به توضیح قسمت های مختلف این پنجره می پردازیم:

Look in: در این قسمت نام پوشه ای که داخل آن قرار دارید و در پنجره پایین آن نام فایل هایی که در این پوشه وجود دارند، مشخص می شود.

نکته ۱-۵: از دکمه های کنار این قسمت می توانید برای جستجو در پوشه های مختلف، ساخت پوشه جدید و ... استفاده کنید.

File name: در این قسمت نام فایلی که انتخاب کردید مشخص می شود.

دکمه Load: بعد از انتخاب فایل و فشردن این دکمه، فایل مورد نظر در اتوکد بارگذاری خواهد شد.

Files of type: در این قسمت می توانید نوع فایلی که قصد بارگذاری آن را دارید، انتخاب کنید.

سربرگ Loaded Applications: در این سربرگ نام و مسیر تمام فایل هایی که قبلاً بارگذاری شده را می توانید ببینید.

سربرگ History list: در این سربرگ می توانید با استفاده از Drag & Drop برنامه مورد نظر را در این قسمت بیاندازید. اگر تیک گزینه Add to history انتخاب نشده باشد، برنامه بارگذاری می شود ولی به سربرگ اضافه نمی شود.

Add to history: با انتخاب این دکمه، برنامه هایی که به سربرگ History List اضافه کرده اید، بارگذاری می شوند.

دکمه Remove/ Unload: این دکمه وقتی از سربرگ Loaded Applications فایلی را انتخاب کنید، به دکمه

Unload تبدیل می شود و به این معنی است که اگر این دکمه را فشار دهید، فایلی را که انتخاب کردید از لیست

این قسمت حذف می شود و در بارگذاری های بعدی اتوکد نمی توانید از آن استفاده کنید.

وقتی از سربرگ History list فایلی را انتخاب کنید، به دکمه Remove تبدیل می شود و به این معنی است که



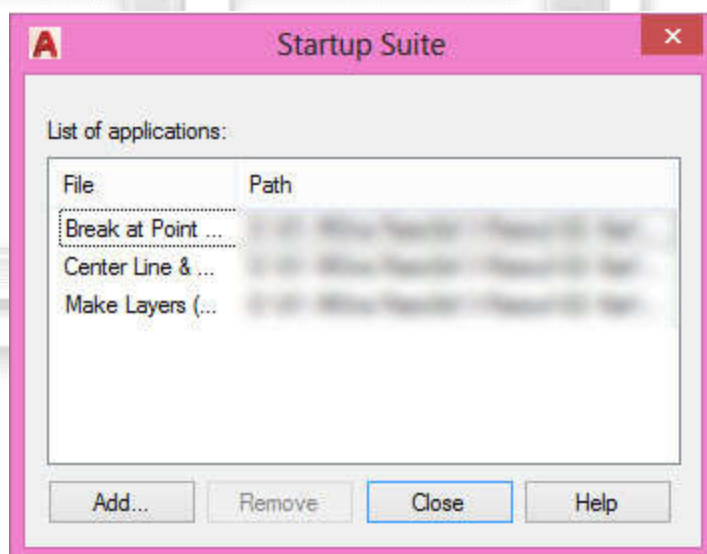


اگر این دکمه را فشار دهید، فایلی را که انتخاب کردید از لیست این قسمت حذف می شود.

قسمت Startup Suite: وقتی در این قسمت روی دکمه Content کلیک کنید، پنجره Startup Suite باز می شود.

در این پنجره اگر روی دکمه Add کلیک کنید، می توانید فایل مورد نظر را در این قسمت فراخوانی کنید تا در بارگذاری های بعدی اتوکد بتوانید از آن استفاده کنید.

اگر فایلی را در این قسمت انتخاب کنید و دکمه Remove را فشار دهید، فایل مورد نظر از این قسمت حذف می شود و در بارگذاری های بعدی اتوکد نمی توان از آن استفاده کرد.



شکل ۱-۸: پنجره Startup Suite

اگر در پنجره Load/ Unload Applications و در قسمت Look in روی فایل Lisp و Macro دوبار کلیک کنید (به شرط آنکه آن فایل را قبلاً بارگذاری نکرده باشید) پنجره ی زیر باز می شود که به توضیح دکمه های آن می پردازم.





کامل ترین مرجع کاربردی آموزش

اتولیسپ

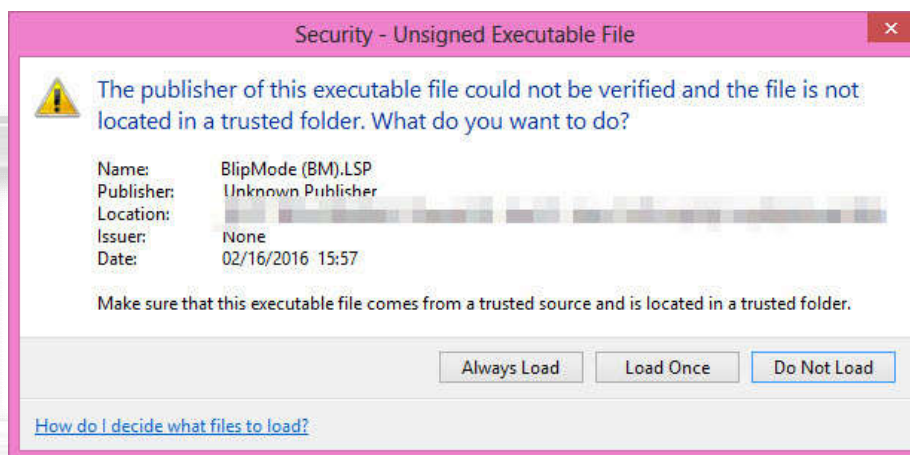
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی

۰۹۱۲۶۲۲۷۲۱۰

فصل ۱- آشنایی با محیط برنامه نویسی

اتولیسپ



شکل ۹-۱: پنجره Security

Always Load: با کلیک روی این دکمه فایل مورد نظر برای همیشه بارگذاری می شود.

Load Once: با کلیک روی این دکمه فایل مورد نظر فقط برای یک بار بارگذاری می شود.

Do Not Load: با کلیک روی این دکمه فایل بارگذاری نمی شود.

نکته ۱-۶: اگر فایل *Lisp* و *Macro* را در پوشه *Support* (در مسیر نصب اتوکد) کپی کنید، دیگر نیازی به بارگذاری آن نیست چون هنگام بالا آمدن نرم افزار، اتوکد تمامی فایل های موجود در این پوشه را خوانده و در صورت لزوم بارگذاری می کند. فقط به این مطلب دقت کنید که تعداد فایل های موجود در این پوشه زیاد نشود چون در این صورت بالا آمدن اتوکد با تأخیر انجام می شود.



## نقشه ذهنی فصل اول

پرانته‌ها

حتماً ابتدا و انتهای جمله

آشنایی با محیط  
برنامه نویسی اتولیسپ

دائم  
موقت

AP

بارگذاری فایل

خط فرمان

VLIDE

وارد شدن به محیط

اتوکدهای قدیمی Tools=> AutoLISP=> Visual LISP Editor

اتوکدهای جدید Manage=> Applications=> Visual LISP Editor

این کتاب الکترونیکی، حاصل ۹ ماه تلاش مستمر و چندین سال تجربه مؤلف و صاحب اثر می باشد.

لطفاً از تکثیر و کپی کردن آن جداً خودداری کنید.

  
**رسول محمدی**

  ۰۹۱۲۶۲۲۷۲۱۰



[rasoul.mohammadi62@gmail.com](mailto:rasoul.mohammadi62@gmail.com)



[www.selfamooz.ir](http://www.selfamooz.ir)

**در کانال ها و شبکه های اجتماعی ما را با نام**

**@selfamooz**

**دنبال کنید.**



کامل ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

فصل ۲- توابع و کلیات اولیه  
برنامه نویسی

## فصل ۲- توابع و کلیات اولیه برنامه نویسی

مطالبی که در این فصل می آموزید:

- ✓ نحوه شروع برنامه نویسی
- ✓ استفاده از توابع اولیه برای نوشتن برنامه های کوچک و ابتدایی
- ✓ چگونه مختصات نقاط را در اتوکد گرفته و به متغیرها اختصاص دهید.
- ✓ با علامت های مرسوم و مورد استفاده در برنامه نویسی اتولیسپ آشنا خواهید شد.
- ✓ از دستورات اتوکد به راحتی در برنامه تان استفاده کنید.

### ۲-۱- تابع Defun

اولین تابعی که در اتولیسپ استفاده می شود، تابع Defun است. این تابع، در ابتدای همه برنامه ها نوشته می شود و به معنی شروع یک برنامه جدید است.

حالت کلی این تابع به شکل زیر است:

(Defun Sym ([Arguments] [/ Variables...])

Sym: یک نام که برنامه را تعریف می کند.

Arguments: نام متغیر

Variable: نام متغیری که به صورت محلی داخل یک برنامه قرار می گیرد.

(defun c:draw ()

نکته ۲-۱: ممکن است در یک فایل چند برنامه نوشته شود. با این تابع می توان فهمید که برنامه جدید از کجا شروع شده است.



کامل ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

فصل ۲- توابع و کلیات اولیه  
برنامه نویسی

نکته ۲-۲: بعد از نوشتن *Defun*، *C* را نوشته و بعد از آن نام برنامه را بنویسید (نام برنامه همان نامی است که بعد از *Load* کردن برنامه، در خط فرمان می نویسید)

اگر در داخل پرانتزهای بعد از نام برنامه، یک اسلش (/) بگذارید و بعد نام متغیرهایی که در آن برنامه تعریف کردید را بنویسید، مقادیر آن متغیرها فقط تا موقعی اعتبار دارند که از برنامه خارج نشده باشید.

```
(defun c:draw (/ p0 p1 a)
```

نکته ۲-۳: اگر بخواهید متغیرها را داخل پرانتز بنویسید، باید دقت کنید که بعد از اسلش (/) یک فاصله بگذارید و بعد متغیرها را بنویسید. در غیر این صورت برنامه اجرا نخواهد شد.

## ۲-۲- علامت دابل کوتیشن (")

هر متن یا نوشته ای که بین دو علامت دابل کوتیشن (") قرار بگیرد، در خط فرمان نوشته می شود.

```
(defun c: draw ()  
  (setq s (getstring "Write Text: "))  
)
```

بعد از نوشتن برنامه روی دکمه *Load active edit window* (↓) از نوار ابزار *Tools* کلیک کنید و روی دکمه *Activate AutoCAD* (🚩) کلیک کرده تا به محیط اتوکد منتقل شوید و یا بدون اینکه روی پنجره برنامه کلیک کنید، به محیط اتوکد رفته و *draw* را در خط فرمان بنویسید و کلید اینتر را فشار دهید و نتیجه را در خط فرمان ببینید.

خواهید دید که جمله «Write Text:» در خط فرمان ظاهر می شود.

## ۲-۳- متغیرها

متغیرها نشان دهنده مقادیر دیگری هستند که در خود ذخیره کرده اند. متغیرها را برای این تعریف کنید که بعداً



بتوانید از مقادیری که درون خود ذخیره می کنند، در قسمت های دیگر برنامه استفاده کنید.

نکته ۲-۴: در نام متغیرها بهتر است از حروف A-Z و ارقام ۰-۹ استفاده شود.

نکته ۲-۵: از علائم در ابتدای نام متغیرها استفاده نمی شود.

۲-۳-۱- نمونه هایی از نام متغیرها

P0

A

Ang

در اتولیسپ می توان چند مقدار را در داخل یک متغیر قرار داد.

برنامه زیر را بنویسید:

```
(defun c:draw())
```

```
(setq a (list 3 6 8))
```

```
)
```

بعد از نوشتن برنامه روی دکمه Load active edit window (↓) از نوار ابزار Tools کلیک کنید و روی دکمه

Activate AutoCAD (🖱️) کلیک کرده تا به محیط اتوکد منتقل شوید و یا بدون اینکه روی پنجره برنامه کلیک

کنید، به محیط اتوکد رفته و draw را در خط فرمان بنویسید و کلید اینتر را فشار دهید و نتیجه را در خط فرمان

ببینید.

حال در خط فرمان a! را نوشته و کلید اینتر را فشار دهید. خواهید دید که در خط فرمان (۳ ۶ ۸) ظاهر می شود.

حال برنامه زیر را بنویسید:

```
(defun c:draw ())
```

```
(setq s (getstring "Text: "))
```



```
)
```



کامل ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

فصل ۲- توابع و کلیات اولیه  
برنامه نویسی

بعد از نوشتن برنامه روی دکمه Load active edit window (  ) از نوار ابزار Tools کلیک کنید و روی دکمه Activate AutoCAD (  ) کلیک کرده تا به محیط اتوکد منتقل شوید و یا بدون اینکه روی پنجره برنامه کلیک کنید، به محیط اتوکد رفته و draw را در خط فرمان بنویسید و کلید اینتر را فشار دهید و نتیجه را در خط فرمان ببینید.

حال در خط فرمان s! را بنویسید و کلید اینتر را فشار دهید. خواهید دید که در خط فرمان اسمی که نوشتید ظاهر می شود.

نکته ۲-۶: اگر در خط فرمان اتوکد علامت تعجب (!) را در ابتدای نام هر متغیر بنویسید و کلید اینتر را فشار دهید، محتویات داخل متغیر نشان داده می شود. برای روشن شدن این موضوع، اعداد بعد از list را تغییر دهید و یا واژه دیگری وارد کنید و مراحل بالا را تکرار کنید و بعد در خط فرمان اتوکد a! یا s! را بنویسید و کلید اینتر را فشار داده و نتیجه را ببینید.

## ۲-۴- لیست

یکی از برتری های زبان برنامه نویسی لیسپ نسبت به زبان های برنامه نویسی دیگر این است که یک متغیر، می تواند چندین مقدار را در خود جای دهد (این مقادیر می توانند عدد یا متن باشند). مثلاً اگر یک برنامه بنویسید که بخواهد یک نقطه در صفحه اتوکد انتخاب کنید، این نقطه شامل مختصات (X,Y,Z) می باشد که مختصات این نقطه تماماً در یک متغیر تعریف می شود و می توان به هر کدام از اعداد X,Y,Z به تنهایی و در صورت نیاز دسترسی پیدا کرد.

لیست ها حتی ممکن است بیشتر از ۳ مقدار (عنصر) باشند.





کامل ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

فصل ۲- توابع و کلیات اولیه  
برنامه نویسی

## ۲-۵- علامت سیمی کالِن (;)

در برنامه هر متن یا نوشته‌ای بعد از این علامت، خوانده نمی‌شود و از این علامت برای نوشتن یک نکته یا نوشتن طرز کار قسمتی از برنامه استفاده می‌شود.

برنامه زیر را بنویسید و به تغییر رنگ قسمت بعد از علامت (;) توجه کنید. نوشته‌هایی که بعد از این علامت قرار می‌گیرند، به صورت سایه روشن خاکستری رنگ هستند.

به جای نوشتن علامت سیمی کالِن (;) می‌توانید از دستور Comment block (☰) در نوار ابزار Tools استفاده کنید و برای برداشتن این علامت از دستور Uncomment block (☲) در نوار ابزار Tools استفاده کنید.

```
(defun c:draw ()
  (setq a (getpoint "Pick Point: "));For Get Point
)
```

نکته ۲-۷: گاهی اوقات ممکن است بخواهید قسمتی از برنامه‌ای که نوشتید، اجرا نشود. برای این منظور می‌توانید از این علامت استفاده کنید.

## ۲-۶- تابع Prompt

از این تابع برای ظاهر شدن یک متن در خط فرمان اتوکد استفاده می‌شود.

حالت کلی این تابع به شکل زیر است:

(Prompt Msg)

Msg: پیامی را که می‌خواهید نوشته شود.

```
(defun c:Draw ()
  (Prompt "I Am a Programer")
)
```



کامل ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

فصل ۲- توابع و کلیات اولیه  
برنامه نویسی

برنامه بالا را نوشته و اجرا کنید. خواهید دید که نوشته I Am a Programer در خط فرمان ظاهر می شود.  
نکته ۲-۸: منظور از نوشتن و اجرای برنامه این است که برنامه را در پنجره لیسپ نوشته و بعد از نوشتن برنامه روی دکمه Load active edit window (↓) از نوار ابزار Tools کلیک کنید و روی دکمه Activate AutoCAD (🖱️) کلیک کرده تا به محیط اتوکد منتقل شوید و یا بدون اینکه روی پنجره برنامه کلیک کنید، به محیط اتوکد بروید و draw را در خط فرمان بنویسید و کلید اینتر را فشار دهید و نتیجه را در خط فرمان ببینید.

## ۲-۷- تابع (terpri) و یا \n

این تابع باعث عوض شدن خط می شود و در خط فرمان، نوشته بعدی روی خط بعدی ظاهر می شود. این تابع باعث می شود برنامه نوشته شده توسط شما بیشتر شبیه توابع اتوکد شود.  
برنامه زیر را بنویسید:

```
(defun c:draw ()
  (setq P1 (Prompt "Hello World "))
  (setq P2 (Prompt "I Am a Programer "))
)
```

برنامه را اجرا کنید. خواهید دید که دو نوشته در کنار هم و در یک خط قرار می گیرند.

حال به شکل زیر برنامه را بنویسید:

```
(defun c:draw ()
  (setq P1 (Prompt "Hello World ")) (terpri)
  (setq P2 (Prompt "I Am a Programer "))
)
```

برنامه را اجرا کنید. خواهید دید که هر کدام از نوشته ها در خط جداگانه ظاهر می شود.





کامل ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

فصل ۲- توابع و کلیات اولیه  
برنامه نویسی

بهترین روش برای نوشتن برنامه، به شکل زیر است:

```
(defun c:draw ()
  (setq P1 (Prompt "\nHello World "))
  (setq P2 (Prompt "\nI Am a Programer "))
)
```

## ۲-۸- تابع Setq

برای اختصاص یک مقدار به یک متغیر یا ثابت از این تابع استفاده می شود. در برنامه نویسی اتولیسپ با علامت مساوی (=) نمی توان مقداری را مساوی با مقدار دیگر قرار داد و باید از تابع Setq استفاده شود (به جز چند تابعی که به آنها اشاره خواهد شد).

حالت کلی این تابع به شکل زیر است:

```
(Setq Sym Expr [Sym Expr]...)
```

Sym: یک اسم که به متغیر اختصاص می دهید.

نکته ۲-۹: بهتر است از اسمی استفاده کنید که آن متغیر را معرفی کند. مثلاً اگر می خواهید یک نقطه تعریف کنید از Point یا P استفاده کنید.

Expr: مقدار متغیر

```
(defun c:draw ()
  (setq a 20)
)
```

در برنامه بالا، مقدار ۲۰ به متغیر a اختصاص پیدا می کند.

نکته ۲-۱۰: در برنامه نویسی لیسپ باید به ترتیب قرارگیری متغیرها دقت کرد.

```
(defun c:draw ()
```



(setq a 20)

(setq b a)

)

در این برنامه مقدار ۲۰ به متغیر a نسبت داده خواهد شد و در خط بعد این برنامه، مقدار a به متغیر b نسبت داده خواهد شد.

برنامه را اجرا کنید و بعد از اجرای برنامه در خط فرمان a! را نوشته و اینتر را فشار دهید.

خواهید دید که برنامه مقدار ۲۰ را برمی گرداند. حال عبارت b! را نوشته و اینتر را فشار دهید.

خواهید دید که برنامه دوباره مقدار ۲۰ را برمی گرداند، زیرا متغیر b مقدار متغیر a را به خود گرفته است.

حال به جای عدد ۲۰ در برنامه، عبارت زیر را بنویسید و نتیجه را ببینید:

“I Am a Programer”

اگر چند متغیر داشته باشید، می توانید به جای چند بار نوشتن تابع Setq، یکبار آنرا نوشته و متغیرها را بعد از آن بنویسید.

(defun c:draw ()

(setq a 20

b a

c 12)

)

یا

(defun c:draw ()

(setq a 20 b a c 12)

)



کامل ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

فصل ۲- توابع و کلیات اولیه  
برنامه نویسی

## ۲-۹- تابع Getpoint

از این تابع برای انتخاب یک نقطه بر روی صفحه گرافیکی اتوکد استفاده می شود. با انتخاب یک نقطه، مختصات (X,Y,Z) آن نقطه دریافت می شود. اگر همراه این تابع، از تابع Setq استفاده کنید، مختصات آن نقطه به متغیر مورد نظر اختصاص پیدا می کند.

حالت کلی این تابع به شکل زیر است:

(Getpoint [Pt] [Msg])

Pt: یک نقطه ۲ بعدی یا ۳ بعدی

Msg: اگر بخواهید این تابع با پیغامی مبنی بر اینکه کاربر چه کاری انجام دهد، همراه شود، بعد از این تابع بین دو دابل کوتیشن، پیغام را بنویسید.

(defun c:draw ()

(setq p0 (getpoint "\nPick Point "))

(setq p1 (getpoint "\nPick Next Point "))

)

برنامه بالا را بنویسید و اجرا کنید. بعد از اجرا، نقاط را در صفحه گرافیکی اتوکد انتخاب کنید.

حال p0! یا p1! را در خط فرمان بنویسید و اینتر کنید.

## ۲-۱۰- تابع Command

یکی از کارآمدترین و پرکاربردترین توابع در برنامه نویسی اتولیسپ، تابع Command می باشد. با استفاده از این تابع می توانید به تمام دستورات اتوکد دسترسی داشته باشید و از آنها استفاده کنید.

حالت کلی این تابع به شکل زیر است:

(Command [Arguments]...)



Arguments: دستور و متغیرهایی که بعد از تابع می آید.

```
(defun c:draw ()
  (setq p0 (getpoint "\nPick Point "))
  (setq p1 (getpoint "\nPick Next Point "))
  (command "line" p0 p1 "")
)
```

برنامه بالا را بنویسید و اجرا کنید. بعد از اجرا، دو نقطه در صفحه گرافیکی انتخاب کنید. بعد از انتخاب دومین نقطه خواهید دید که یک خط از نقطه اول به نقطه دوم ترسیم خواهد شد.

در برنامه بالا دو دابل کوتیشن آخر در تابع command، عمل Enter یا Esc را برای خروج از تابع انجام می دهند.

```
(defun c:draw ()
  (setq p0 (getpoint "\nPick Point "))
  (setq p1 (getpoint "\nPick Next Point "))
  (setq p2 (getpoint "\nPick Next Point "))
  (setq p3 (getpoint "\nPick Next Point "))
  (command "line" p0 p1 p2 p3 "c")
)
```

در برنامه بالا حرف c در بین دو دابل کوتیشن، عمل Close را انجام می دهد و نقطه p3 را به p0 متصل می کند. توجه: به شما تبریک می گویم، زیرا اولین برنامه کامل خود را نوشتید. مهارت در برنامه نویسی ملزم به داشتن تلاش و پشتکار و تمرین زیاد است. مطمئن باشید که با تمرین زیاد و ممارست در این کار به مهارت لازم برای برنامه نویسی خواهید رسید.

## ۲-۱۱- تابع Getcorner

از این تابع برای دریافت مختصات یک نقطه استفاده می شود. این تابع شبیه به تابع Getpoint می باشد، با این



کامل‌ترین مرجع کاربردی آموزش  
اتولیسپ  
(برنامه نویسی در اتوکد)

تألیف و ترجمه: رسول محمدی  
۰۹۱۲۶۲۲۷۲۱۰

فصل ۲- توابع و کلیات اولیه  
برنامه‌نویسی

تفاوت که برای اجرای این تابع حتماً باید قبلاً یک نقطه بر روی صفحه گرافیکی اتوکد انتخاب شده باشد.

حالت کلی این تابع به شکل زیر است:

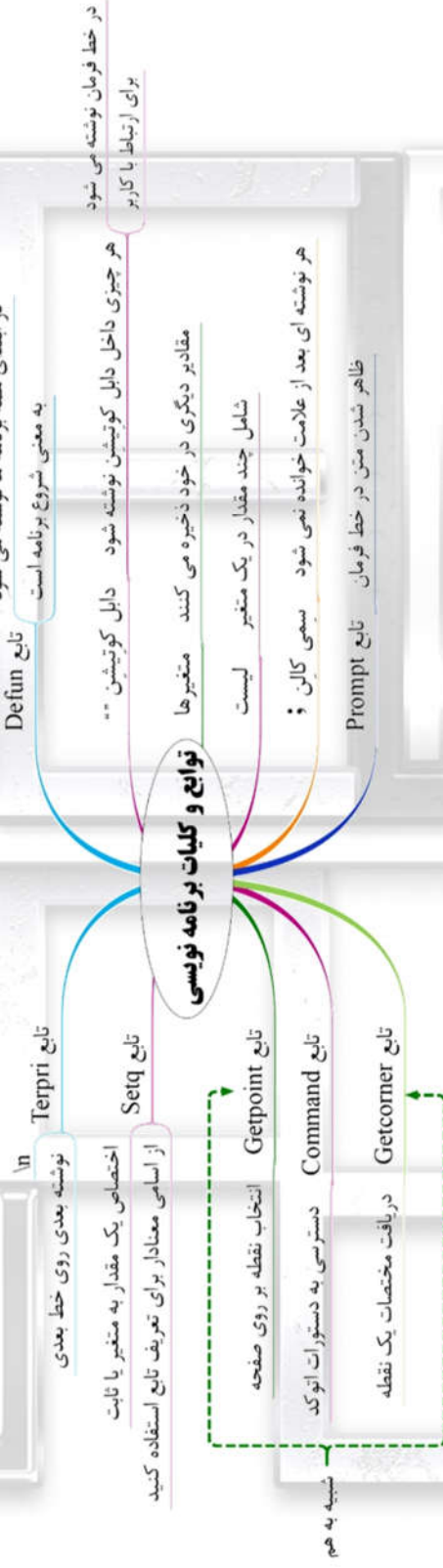
(Getpoint [Pt] [Msg])

Pt: یک نقطه ۲ بُعدی یا ۳ بُعدی

Msg: اگر بخواهید این تابع با پیغامی مبنی بر اینکه کاربر چه کاری انجام دهد، همراه شود، بعد از این تابع بین دو دابل کوتیشن، پیغام را بنویسید.

نکته ۱-۲: اگر در برنامه بخواهید از دستور *Rectagle* استفاده کنید، به طوری که نقطه دوم، مختصات مشخصی نداشته باشد (منظور این است که بتوانید هر نقطه دلخواهی را بر روی صفحه گرافیکی اتوکد انتخاب کنید)، از این تابع استفاده کنید.

## نقشه ذهنی فصل دوم







این کتاب الکترونیکی، حاصل ۹ ماه تلاش مستمر و چندین سال تجربه مؤلف و صاحب اثر می‌باشد.

لطفاً از تکثیر و کپی کردن آن جداً خودداری کنید.

رسول محمدی 

  ۰۹۱۲۶۲۲۷۲۱۰

 [rasoul.mohammadi62@gmail.com](mailto:rasoul.mohammadi62@gmail.com)

 [www.selfamooz.ir](http://www.selfamooz.ir)

در کانال‌ها و شبکه‌های اجتماعی ما را با نام

@selfamooz

دنبال کنید.