



AutoCAD Autolisp tutorials for beginners – Visual lisp

Description

AutoCAD, the leading software for computer-aided design (CAD) and drafting, offers a wide range of customization options to enhance productivity and streamline design workflows. AutoLISP, a programming language specific to AutoCAD, empowers users to create custom commands and automate repetitive tasks, opening up a world of possibilities for efficient design processes. In this comprehensive tutorial, we will dive into AutoCAD [Autolisp](#) tutorials for beginners, covering the fundamentals of AutoLISP and providing step-by-step guidance to master customization and automation in AutoCAD.

Key Takeaways

- AutoLISP is a programming language specifically designed for AutoCAD customization and automation.
- With AutoLISP, users can create custom commands, automate repetitive tasks, and extend AutoCAD's functionality.
- Understanding AutoLISP syntax, data types, variables, and control flow is crucial for writing effective AutoLISP programs.
- Custom commands can be defined using the `(defun)` function, while automation can be achieved by scripting repetitive tasks.
- AutoLISP offers the flexibility to interact with external data sources, enhancing the integration and data processing capabilities of AutoCAD.

AutoCAD Autolisp: An Introduction

AutoLISP is a dialect of the Lisp programming language specifically tailored for AutoCAD. With AutoLISP, you can create custom commands, automate repetitive tasks, and extend the functionality of AutoCAD beyond its out-of-the-box capabilities. Learning AutoLISP opens up a world of possibilities to

streamline your design process and boost productivity.

Getting Started with AutoLISP

To begin your AutoLISP journey, it's™ important to understand the basics. Here are some key concepts you should grasp:

1. **Syntax and Expressions:** AutoLISP follows a simple and consistent syntax, consisting of functions, variables, and expressions. Understanding how to structure your code is crucial for creating effective AutoLISP programs.
2. **Data Types:** AutoLISP supports various data types, including numbers, strings, lists, and symbols. Each data type has its own unique characteristics and usage, enabling you to manipulate and process different types of data within your programs.
3. **Variables and Scope:** Variables store and manage data within AutoLISP programs. Understanding variable scope and how to declare, assign, and use variables is essential for creating robust and flexible programs.
4. **Control Flow:** AutoLISP provides constructs such as conditionals (if, cond), loops (while, repeat), and iteration (foreach) to control the flow of your programs. Mastering these control flow mechanisms empowers you to create dynamic and responsive AutoCAD solutions.

Creating Custom Commands

One of the primary benefits of AutoLISP is the ability to create custom commands tailored to your specific needs. By defining your own commands, you can streamline repetitive tasks and automate complex operations. Here's™ an overview of the steps involved in creating custom commands:

1. **Defining a Command:** To create a custom command, you need to define its name, arguments, and functionality. This involves using the `(defun)` function and specifying the input parameters, code block, and return values for your command.
2. **Implementing Functionality:** Within your custom command, you can leverage AutoCAD's™ extensive set of built-in functions and commands to manipulate drawings, entities, layers, and more. You can also use variables, conditionals, and loops to add logic and interactivity to your commands.
3. **Loading and Accessing Commands:** Once you've™ defined your custom commands, you need to load them into AutoCAD's™ environment. This ensures that your commands are accessible and can be executed when needed. You can either load them manually or configure AutoCAD to load them automatically on startup.

Automation with AutoLISP

AutoLISP is not only useful for creating custom commands but also for automating repetitive tasks and workflows. By harnessing the power of AutoLISP automation, you can save significant time and effort in your design process. Here are a few areas where AutoLISP automation can be beneficial:

1. **Batch Processing:** AutoLISP allows you to automate repetitive tasks that involve processing multiple files or performing the same actions on multiple drawings. By writing scripts that utilize AutoLISP functions, you can achieve batch processing and apply changes uniformly across

multiple drawings.

2. **Data Extraction and Reporting:** AutoLISP provides functions to extract data from drawings, such as dimensions, properties, and [attributes](#). You can use these functions to generate reports, perform calculations, or export data to external formats for further analysis.
3. **Drawing Generation:** With AutoLISP, you can automate the creation of drawings based on predefined templates or rules. By leveraging functions to generate entities, set properties, and apply standards, you can quickly generate complex drawings with consistent formatting.

Top 25 of the most commonly used AutoLISP functions along with their descriptions:

1. `(setq variable value)`: Sets the value of a variable to a specified value.
2. `(+ num1 num2 ...)`: Performs addition of one or more numbers and returns the sum.
3. `(- num1 num2 ...)`: Subtracts one or more numbers and returns the difference.
4. `(* num1 num2 ...)`: Multiplies one or more numbers and returns the product.
5. `(/ num1 num2 ...)`: Divides one or more numbers and returns the quotient.
6. `(if condition then [else])`: Performs conditional branching based on a condition.
7. `(cond (test1 result1) (test2 result2) ...)`: Executes different expressions based on multiple conditions.
8. `(equal value1 value2)`: Checks if two values are equal.
9. `(not value)`: Returns the logical negation of a value.
10. `(and condition1 condition2 ...)`: Performs logical AND operation on multiple conditions.
11. `(or condition1 condition2 ...)`: Performs logical OR operation on multiple conditions.
12. `(setq variable value)`: Assigns a value to a variable.
13. `(setq variable (function arguments))`: Stores the result of a function in a variable.
14. `(length list)`: Returns the number of elements in a list.
15. `(nth index list)`: Retrieves the element at a specified index in a list.
16. `(car list)`: Returns the first element of a list.
17. `(cdr list)`: Returns a new list with the first element removed.
18. `(cons item list)`: Adds an item to the beginning of a list.
19. `(list item1 item2 ...)`: Creates a new list with the specified items.
20. `(append list1 list2 ...)`: Combines multiple lists into a single list.
21. `(member item list)`: Checks if an item is present in a list and returns the sublist from that item onwards.
22. `(reverse list)`: Reverses the order of elements in a list.
23. `(subst new old list)`: Replaces occurrences of an element with a new value in a list.
24. `(apply function arguments)`: Calls a function with a list of arguments.
25. `(mapcar function list)`: Applies a function to each element of a list and returns a new list with the results.

These functions form the foundation of AutoLISP programming and can be used to perform a wide range of operations and manipulations within AutoCAD.

FAQ: AutoCAD Autolisp Tutorials for Beginners

1. How do I use AutoLISP in AutoCAD?

To use AutoLISP in AutoCAD, you need to write AutoLISP programs or scripts and load them into AutoCAD. AutoLISP programs are typically saved with a .LSP file extension. Once loaded, you can execute AutoLISP commands and functions within AutoCAD's command line or through custom commands.

2. How do you write AutoLISP?

To write AutoLISP code, you can use a plain text editor, such as Notepad, or a specialized code editor like Visual Studio Code. AutoLISP code consists of expressions, functions, and variables. You define functions using the `(defun)` statement, assign values to variables using the `(setq)` statement, and use various AutoLISP functions to perform operations on drawings and entities.

3. What is an AutoLISP in AutoCAD?

AutoLISP is a programming language specific to AutoCAD. It allows you to customize and extend the functionality of AutoCAD by creating custom commands and automating repetitive tasks. With AutoLISP, you can manipulate drawings, entities, layers, properties, and perform calculations and data extraction.

4. What is the difference between AutoCAD VBA and AutoLISP?

AutoCAD VBA (Visual Basic for Applications) and AutoLISP are two programming languages used for customizing AutoCAD. The main difference is that VBA is based on the Visual Basic programming language, while AutoLISP is a dialect of the Lisp programming language. VBA offers a more traditional procedural programming approach, while AutoLISP has a functional programming style.

5. What is the difference between AutoLISP and Visual LISP?

AutoLISP and Visual LISP are often used interchangeably, but there is a slight difference. AutoLISP refers to the programming language itself, whereas Visual LISP is the integrated development environment (IDE) within AutoCAD for writing, editing, and managing AutoLISP code. Visual LISP provides additional tools and features to enhance the AutoLISP programming experience.

6. What is the application of AutoLISP?

AutoLISP has various applications in AutoCAD. It enables users to create custom commands, automate repetitive tasks, generate complex drawings, extract data, perform calculations, and enhance the overall productivity and efficiency of the design process. AutoLISP allows users to tailor AutoCAD to their specific needs and streamline their workflow.

7. What editor is used for AutoLISP?

AutoLISP code can be written using any plain text editor, such as Notepad or Notepad++. However, for a more convenient and feature-rich experience, you can use the Visual LISP Editor provided within AutoCAD. The Visual LISP Editor offers syntax highlighting, code navigation, debugging tools, and integration with AutoCAD's environment.

8. What is the command function in AutoLISP?

The command function in AutoLISP is used to invoke built-in AutoCAD commands within an AutoLISP program. By using the command function, you can execute AutoCAD commands, such as Line, Circle, Zoom, or any other command available in AutoCAD's command line, directly from your AutoLISP code.

9. What are the loops in AutoLISP?

AutoLISP provides loops such as `while`, `repeat`, and `foreach` to control the flow of execution within a program. The `while` loop repeatedly executes a block of code while a specified condition is true. The `repeat` loop executes a block of code for a specific number of iterations. The `foreach` loop iterates over a list and performs an operation on each element.

10. How do I write a script in AutoCAD?

To write a script in AutoCAD, you can use a plain text editor to create a text file containing a series of AutoCAD commands. Each command should be written on a new line. Save the file with a `.SCR` extension. Then, within AutoCAD, you can run the script by using the `SCRIPT` command and specifying the path to the script file.

11. What is the best language for AutoCAD?

AutoLISP is considered one of the best languages for customizing and automating AutoCAD due to its direct integration with the software and its extensive set of functions tailored for CAD operations. However, other languages like Visual Basic for Applications (VBA), .NET languages (C#, VB.NET), and Python can also be used for AutoCAD customization, depending on your specific requirements and preferences.

12. Can we automate AutoCAD?

Yes, AutoCAD can be automated using programming languages like AutoLISP, VBA, .NET languages, and Python. These languages allow you to create scripts, custom commands, and programs that automate repetitive tasks, perform batch operations, generate drawings, extract data, and more. Automation in AutoCAD significantly enhances productivity and efficiency in the design process.

13. What language are AutoCAD macros written in?

AutoCAD macros can be written in AutoLISP or VBA. AutoLISP macros are typically saved with the `.LSP` extension, while VBA macros use the `.DVBA` extension. Macros are small snippets of code that

perform specific tasks within AutoCAD and can be executed with a single command or keystroke.

14. What is Setq in AutoLISP?

In AutoLISP, `setq` is a function used to assign a value to a variable. The name `setq` stands for "set quote." It takes two arguments: a symbol (variable name) and an expression (value). For example, `(setq myVariable 10)` assigns the value 10 to the variable named `myVariable`. You can later reference this variable throughout your program.

15. What is the introduction of AutoLISP programming?

The introduction of AutoLISP programming refers to the initial steps of learning and understanding the fundamentals of AutoLISP. It involves familiarizing yourself with the syntax, data types, variables, functions, control flow structures, and concepts specific to AutoLISP. It is the starting point to grasp the power and potential of AutoLISP for customization and automation in AutoCAD.

16. What is the file extension for AutoLISP?

The file extension for AutoLISP programs is typically `.LSP`. When you save an AutoLISP program, you should use this extension to indicate that the file contains AutoLISP code. AutoCAD recognizes files with the `.LSP` extension as AutoLISP programs and allows you to load and execute them within its environment.

17. How do I edit an AutoLISP file?

To edit an AutoLISP file, you can open it using any text editor, such as Notepad or Notepad++. Simply locate the `.LSP` file on your computer, right-click on it, and select "Open with" or "Open" depending on your operating system. Choose the text editor of your choice, and the AutoLISP file will open, allowing you to make modifications to the code.

18. Where are LISP routines in AutoCAD?

LISP routines, which are AutoLISP programs, can be stored in various locations in AutoCAD. By default, AutoCAD looks for LISP routines in the Support File Search Path specified in the Options menu. This path typically includes directories such as the AutoCAD installation folder and the user's support file folder. You can also load LISP routines from specific locations using the `APPLLOAD` command.

19. How do I open Visual LISP in AutoCAD?

To open the Visual LISP Editor in AutoCAD, you can use the `VLIDE` command. Simply type "VLIDE" in the AutoCAD command line and press Enter. The Visual LISP Editor will open, providing you with a dedicated environment for writing, editing, and managing your AutoLISP code within AutoCAD.

20. How do I get rid of AutoLISP in AutoCAD?

If you want to remove AutoLISP programs or functionality from AutoCAD, you can unload or delete the associated LISP routines. To unload a specific AutoLISP program, use the `UNLOAD` command followed by the name of the LISP file. To permanently remove AutoLISP programs, you can delete the associated `.LSP` files from their storage location on your computer.

21. What are the advantages of using AutoLISP in AutoCAD?

Using AutoLISP in AutoCAD offers several advantages. Firstly, it allows you to customize AutoCAD to suit your specific needs and workflows. You can create custom commands, automate repetitive tasks, and streamline your design process. AutoLISP also provides access to a wide range of built-in functions and features, enabling you to manipulate drawings, entities, and properties with ease. Additionally, AutoLISP promotes efficiency and productivity by reducing manual effort and enabling batch processing and data extraction.

22. Can AutoLISP programs interact with external data sources?

Yes, AutoLISP programs can interact with external data sources. By utilizing functions like `open`, `read-line`, and `write-line`, you can read and write data from and to text files, CSV files, databases, or even web services. This capability allows you to integrate AutoCAD with other software systems and leverage external data for automation, reporting, or analysis purposes.

23. Is programming experience necessary to learn AutoLISP?

No, programming experience is not necessarily required to learn AutoLISP. AutoLISP is known for its simplicity and beginner-friendly syntax. However, having a basic understanding of programming concepts such as variables, functions, and control flow will undoubtedly be beneficial. With dedicated practice, hands-on examples, and access to learning resources, beginners can grasp AutoLISP concepts and gradually enhance their programming skills.

24. Are there any limitations to what can be achieved with AutoLISP in AutoCAD?

While AutoLISP is a powerful tool for customization and automation in AutoCAD, there are certain limitations to consider. AutoLISP is primarily focused on 2D drafting and basic 3D modeling tasks. More complex operations, such as advanced 3D modeling, may require additional programming languages or APIs. Additionally, AutoLISP has performance limitations compared to lower-level languages, but these limitations are rarely a significant concern for typical AutoCAD customization tasks.

25. How can I debug and troubleshoot AutoLISP programs?

AutoLISP provides various methods for debugging and troubleshooting programs. One approach is to use the `(print)` function to output specific values or messages at different stages of your program to track the execution flow. Another method is to use the Visual LISP Editor's debugging features, such as setting breakpoints, stepping through code, and examining variables' values. Additionally,

referring to AutoLISP documentation, online forums, and communities can help you resolve common issues and learn from experienced users.

Resources for Learning AutoLISP

As a beginner, it's essential to have access to reliable resources to aid your AutoLISP learning journey. Here are some recommended sources to deepen your understanding and master AutoLISP:

- [Autodesk AutoCAD Documentation](#)
- [AutoCAD AutoLISP Developer's Guide](#)
- [AutoCAD AutoLISP Tutorials on YouTube](#)
- [AutoCAD Customization Forums](#)

By exploring these resources, practicing hands-on exercises, and engaging with the AutoCAD community, you can enhance your AutoLISP skills and become proficient in customization and automation.

Conclusion

Mastering AutoLISP opens up a world of possibilities for AutoCAD users, enabling them to customize their workflows and automate time-consuming tasks. By following the tutorials and understanding the fundamentals of AutoLISP, beginners can gradually enhance their skills and unlock the full potential of AutoCAD's customization features.

Through the creation of custom commands, users can streamline their design processes and improve efficiency. Automation with AutoLISP allows for the execution of repetitive tasks, batch processing, and data extraction, resulting in significant time savings. Additionally, the ability to interact with external data sources expands the possibilities for integration and analysis.

With dedication, practice, and exploration of available resources, beginners can become proficient in AutoCAD Autolisp and leverage this powerful tool to enhance their productivity and effectiveness in the world of CAD and design. So, embrace the power of AutoLISP and embark on a journey of customization and automation in AutoCAD.