



TCL/TK

tutorialspoint

S I M P L Y E A S Y L E A R N I N G

www.tutorialspoint.com



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



<https://twitter.com/tutorialspoint>

About the Tutorial

Tcl is a general purpose multi-paradigm system programming language. It is a scripting language that aims at providing the ability for applications to communicate with each other.

On the other hand, Tk is a cross platform widget toolkit used for building GUI in many languages.

This tutorial covers various topics ranging from the basics of the Tcl/ Tk to its scope in various applications.

Audience

This tutorial is designed for all those individuals who are looking for a starting point of learning Tcl/ Tk. Therefore, we cover all those topics that are required for a beginner and an advanced user.

Prerequisites

Before proceeding with this tutorial, it is advisable for you to understand the basic concepts of computer programming. This tutorial is self-contained and you will be able to learn various concepts of Tcl/Tk even if you are a beginner. You just need to have a basic understanding of working with a simple text editor and command line.

Disclaimer & Copyright

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute, or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness, or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

	About the Tutorial	i
	Audience	i
	Prerequisites	i
	Disclaimer & Copyright	i
	Table of Contents	ii
1.	TCL – OVERVIEW	1
	Features of Tcl	1
	Applications.....	1
2.	TCL – ENVIRONMENT SETUP	2
	Local Environment Setup	2
	Text Editor	2
	The Tcl Interpreter.....	3
	Installation on Windows.....	3
	Installation on Linux	3
	Installation on Debian based Systems	4
	Installation on Mac OS X.....	4
	Installation from Source Files	4
3.	TCL – SPECIAL VARIABLES.....	6
	Examples for using Tcl Special Variables	7
4.	TCL – BASIC SYNTAX.....	9
	First Tcl Program.....	9
	Comments	9
	Identifiers.....	10
	Reserved Words	10

	Whitespace in Tcl	11
5.	TCL – COMMANDS	12
	Command Substitution	13
	Variable Substitution	13
	Backslash Substitution	13
6.	TCL – DATA TYPES	14
	Simple Tcl Objects	14
	String Representations	15
	List	15
	Associative Array	16
	Handles	16
7.	TCL – VARIABLES	17
	Variable Naming	17
	Dynamic Typing	17
	Mathematical Expressions	18
8.	TCL – OPERATORS	20
	Arithmetic Operators	20
	Relational Operators	22
	Logical Operators	23
	Bitwise Operators	25
	Ternary Operator	27
	Operators Precedence in Tcl	27
9.	TCL – DECISIONS	30
	Tcl - If Statement	31
	Tcl – If else Statement	32

	The if...else if...else Statement	34
	Tcl – Nested If Statement	35
	Tcl – Switch Statement	36
	Tcl – Nested Switch Statement	39
	The? : Operator	40
10.	TCL – LOOPS.....	42
	Tcl – While Loop	43
	Tcl – For Loops.....	44
	Tcl – Nested Loops.....	46
	Loop Control Statements	48
	Tcl – Break Statement.....	48
	Tcl – Continue Statement	50
	The Infinite Loop	51
11.	TCL – ARRAYS.....	52
	Size of Array	52
	Array Iteration.....	52
	Associative Arrays	53
	Indices of Array	53
	Iteration of Associative Array	54
12.	TCL – STRINGS.....	55
	String Representations	55
	String Escape Sequence	55
	String Command.....	56
13.	TCL – LISTS.....	63
	Creating a List.....	63

Appending Item to a List.....	64
Length of List.....	64
List Item at Index.....	64
Insert Item at Index.....	65
Replace Items at Indices.....	65
Set Item at Index.....	66
Transform List to Variables.....	66
Sorting a List.....	66
14. TCL – DICTIONARY.....	68
Size of Dict.....	68
Dictionary Iteration.....	69
Value for Key in Dict.....	69
All Keys in Dict.....	70
All Values in Dict.....	70
Key Exists in Dict.....	70
15. TCL – PROCEDURES.....	72
Procedures with Multiple Arguments.....	72
Procedures with Variable Arguments.....	73
Procedures with Default Arguments.....	73
Recursive Procedures.....	74
16. TCL – PACKAGES.....	75
Creating Package.....	75
17. TCL – NAMESPACES.....	77
Creating Namespace.....	77
Nested Namespaces.....	77

	Importing and Exporting Namespace.....	78
	Forget Namespace.....	79
18.	TCL – FILE I/O.....	80
	Opening Files.....	80
	Closing a File.....	80
	Writing a File.....	81
	Reading a File.....	81
19.	TCL – ERROR HANDLING.....	83
	Error Syntax.....	83
	Catch Syntax.....	83
20.	TCL – BUILT-IN FUNCTIONS.....	85
	Math Functions.....	85
	System Functions.....	87
21.	TCL – REGULAR EXPRESSIONS.....	90
	Syntax.....	90
	Multiple Patterns.....	91
	Switches for Regex Command.....	92
22.	TK – OVERVIEW.....	94
	Features of Tk.....	94
	Applications Built in Tk.....	94
23.	TK – ENVIRONMENT.....	95
	The Tk Interpreter.....	95
	Installation on Windows.....	95
	Installation on Linux.....	96

Installation on Debian Based Systems	96
Installation on Mac OS X.....	97
Installation from Source Files	97
Examples for Using Tcl Special Variables	99
24. TK – WIDGETS OVERVIEW	101
Creating a Widget	101
Widget Naming Convention.....	101
Color Naming Convention.....	101
Dimension Convention	101
Common Options	102
25. TK – BASIC WIDGETS	105
Tk - Label Widget	105
Tk – Button Widget.....	106
Tk – Entry Widgets.....	109
Tk – Message Widget.....	110
Tk – Text Widget.....	111
Tk – Top Level Widgets	113
26. TK – LAYOUT WIDGETS.....	115
Tk – Frame Widget.....	115
Tk – Place Widget	116
Tk – Pack Widget	117
Tk – Grid Widget.....	119
27. TK – SELECTION WIDGETS.....	122
Tk – Radio Button Widget.....	122
Tk – Check Button Widget	123

	Tk – Menu Widget	125
	Tk – Listbox Widget	127
28.	TK – CANVAS WIDGETS	131
	Options.....	131
	Widgets for Drawing in Canvas	132
	Tk – Canvas Line Widget	132
	Tk - Canvas Arc Widget	134
	Tk – Canvas Rectangle Widget	135
	Tk – Canvas Oval Widget	136
	Tk – Canvas Polygon Widget	137
	Tk - Canvas Text Widget	138
	Tk – Canvas Bitmap Widget	139
	Tk – Canvas Image Widget	140
29.	TK – MEGA WIDGETS	143
	Tk – Dialog Widget.....	143
	Tk – Spinbox Widget	144
	Tk – Combobox Widget	145
	Tk – Notebook Widget	146
	Tk – Progressbar Widget.....	147
	Tk – Treeview Widget	148
	Tk – Scrollbar Widget.....	150
	Tk – Scale Widget	151
30.	TK – FONTS	154
	Options.....	154
31.	TK – IMAGES	156

	Options.....	156
32.	TK – EVENTS.....	159
	Event Binding	161
33.	TK – WINDOWS MANAGER	162
	Creating Window.....	164
	Destroying Window.....	164
34.	TK – GEOMETRY MANAGER	165
	Positioning and Sizing.....	165
	Grid Geometry.....	165

1. Tcl – Overview

Tcl is shortened form of **Tool Command Language**. John Ousterhout of the University of California, Berkeley, designed it. It is a combination of a scripting language and its own interpreter that gets embedded to the application, we develop with it.

Tcl was developed initially for Unix. It was then ported to Windows, DOS, OS/2, and Mac OSX. Tcl is much similar to other unix shell languages like Bourne Shell (Sh), the C Shell (csh), the Korn Shell (sh), and Perl.

It aims at providing ability for programs to interact with other programs and also for acting as an embeddable interpreter. Even though, the original aim was to enable programs to interact, you can find full-fledged applications written in Tcl/Tk.

Features of Tcl

The features of Tcl are as follows:

- Reduced development time.
- Powerful and simple user interface kit with integration of TK.
- Write once, run anywhere. It runs on Windows, Mac OS X, and almost on every Unix platform.
- Quite easy to get started for experienced programmers; since, the language is so simple that they can learn Tcl in a few hours or days.
- You can easily extend existing applications with Tcl. Also, it is possible to include Tcl in C, C++, or Java to Tcl or vice versa.
- Have a powerful set of networking functions.
- Finally, it's an open source, free, and can be used for commercial applications without any limit.

Applications

Tcl is a general-purpose language and you can find Tcl everywhere. It includes,

- Scalable websites that are often backed by databases.
- High performance web servers build with TclHttpd.
- Tcl with CGI based websites.
- Desktop GUI applications.
- Embedded applications.

2. Tcl – Environment Setup

Try it Option

You really do not need to set up your own environment to start learning Tcl programming. Reason is very simple, we already have set up Tcl Programming environment online, so that you can execute all the Tcl examples online at the same time when you are doing your theory work. This gives you confidence in what you are reading and to check the result with different options. Feel free to modify any example and execute it online.

Try following example using '**Try it**' option available at the top right corner of the sample code box:

```
#!/usr/bin/tclsh

puts "Hello, World!"
```

For most of the Tcl examples given in this tutorial, you will find **Try it** option, so just make use of it and enjoy your learning. For Tk examples, you will need to have a console to see graphical results; so, we recommend to have your own Tk setup.

Local Environment Setup

If you are willing to set up your environment for Tcl, you need the following two software applications available on your computer:

- (a) Text Editor
- (b) Tcl Interpreter.

Text Editor

This will be used to type your program. Examples of a few text editors include Windows Notepad, OS Edit command, Brief, Epsilon, EMACS, and vim or vi.

Name and version of a text editor can vary on different operating systems. For example, Notepad will be used on Windows, and vim or vi can be used on windows as well as Linux or UNIX.

The files you create with your text editor are called source files and contain program source code. The source files for Tcl programs are named with the extension **".tcl"**.

Before starting your programming, make sure you have one text editor in place and you have enough experience to write a computer program, save it in a file, build it, and finally execute it.

The Tcl Interpreter

It is just a small program that enables you to type Tcl commands and have them executed line by line. It stops execution of a tcl file, in case, it encounters an error unlike a compiler that executes fully.

Let's have a helloWorld.tcl file as follows. We will use this as a first program, we run on a platform you choose.

```
#!/usr/bin/tclsh

puts "Hello World!"
```

Installation on Windows

Download the latest version for windows **installer** from the list of Active Tcl binaries available. The active Tcl community edition is free for personal use.

Run the downloaded executable to install the Tcl, which can be done by following the on screen instructions.

Now, we can build and run a Tcl file say helloWorld.tcl by switching to folder containing the file using 'cd' command and then execute the program using the following steps

```
C:\Tcl> tclsh helloWorld.tcl
```

We can see the following output.

```
C:\Tcl> helloWorld
```

C:\Tcl is the folder, I am using to save my samples. You can change it to the folder in which you have saved Tcl programs.

Installation on Linux

Most of the Linux operating systems come with Tcl inbuilt and you can get started right away in those systems. In case, it's not available, you can use the following command to download and install Tcl-Tk.

```
$ yum install tcl tk
```

Now, we can build and run a Tcl file say helloWorld.tcl by switching to folder containing the file using 'cd' command and then execute the program using the following steps:

```
$ tclsh helloWorld.tcl
```

We can see the following output:

```
$ hello world
```

Installation on Debian based Systems

In case, it's not available in your OS, you can use the following command to download and install Tcl-Tk:

```
$ sudo apt-get install tcl tk
```

Now, we can build and run a Tcl file say helloWorld.tcl by switching to folder containing the file using 'cd' command and then execute the program using the following steps:

```
$ tclsh helloWorld.tcl
```

We can see the following output:

```
$ hello world
```

Installation on Mac OS X

Download the latest version for Mac OS X **package** from the list of Active Tcl binaries available. The active Tcl community edition is free for personal use.

Run the downloaded executable to install the Active Tcl, which can be done by following the on screen instructions.

Now, we can build and run a Tcl file say helloWorld.tcl by switching to folder containing the file using 'cd' and then execute the program using the following steps:

```
$ tclsh helloWorld.tcl
```

We can see the following output:

```
$ hello world
```

Installation from Source Files

You can use the option of installing from source files when a binary package is not available. It is generally preferred to use Tcl binaries for Windows and Mac OS X, so only compilation of sources on unix based system is shown below.

- Download the **source files**.
- Now, use the following commands to extract, compile, and build after switching to the downloaded folder.

```
$ tar xzf tcl8.6.1-src.tar.gz
$ cd tcl8.6.1
$ cd unix
$ ./configure --prefix=/opt --enable-gcc
$ make
```

```
$ sudo make install
```

Note: Make sure, you change the file name to the version you downloaded on commands 1 and 2 given above.

3. Tcl – Special Variables

In Tcl, we classify some of the variables as special variables and they have a predefined usage/functionality. The list of special variables is listed below.

Special Variable	Description
argc	Refers to a number of command-line arguments.
argv	Refers to the list containing the command-line arguments.
argv0	Refers to the file name of the file being interpreted or the name by which we invoke the script.
env	Used for representing the array of elements that are environmental variables.
errorCode	Provides the error code for last Tcl error.
errorInfo	Provides the stack trace for last Tcl error.
tcl_interactive	Used to switch between interactive and non-interactive modes by setting this to 1 and 0 respectively.
tcl_library	Used for setting the location of standard Tcl libraries.
tcl_pkgPath	Provides the list of directories where packages are generally installed.
tcl_patchLevel	Refers to the current patch level of the Tcl interpreter.
tcl_platform	Used for representing the array of elements with objects including byteOrder, machine, osVersion, platform, and os.
tcl_precision	Refers to the precision i.e. number of digits to retain when converting to floating-point numbers to strings. The default value is 12.
tcl_prompt1	Refers to the primary prompt.
tcl_prompt2	Refers to the secondary prompt with invalid commands.
tcl_rcFileName	Provides the user specific startup file.
tcl_traceCompile	Used for controlling the tracing of bytecode compilation. Use 0 for no output, 1 for summary, and 2 for detailed.
tcl_traceExec	Used for controlling the tracing of bytecode execution. Use 0 for no output, 1 for summary, and 2 for detailed.
tcl_version	Returns the current version of the Tcl interpreter.

The above special variables have their special meanings for the Tcl interpreter.

Examples for using Tcl Special Variables

Let's see some examples for special variables.

Tcl Version

```
#!/usr/bin/tclsh

puts $tcl_version
```

When you run the program, you will get a similar output as shown below:

```
8.5
```

Tcl Environment Path

```
#!/usr/bin/tclsh

puts $env(PATH)
```

When you run the program, you will get a similar output as shown below:

```
/web/com/GNUstep/Tools:/usr/GNUstep/Local/Tools:/usr/GNUstep/System/Tools:/usr/
local/sml/bin:/usr/local/flex/bin:/usr/local/gcc-
4.8.1/bin:/usr/share/java:./usr/share/java:/usr/lib/jvm/java/lib:/usr/lib/jvm/
java/jre/lib:/usr/local/bin:/usr/local/mozart/bin:/usr/local/go/bin:/usr/local/
factor:/usr/local/groovy-2.1.7/bin:/opt/Pawn/bin:/usr/local/icon-
v950/bin:/usr/local/lib/mono/4.0:/usr/lib64/qtC.3/bin:/usr/local/bin:/bin:/usr/
bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/Pawn/bin:/usr/local/dart/bin:/usr/loca
l/julia/usr/bin:/usr/local/julia:/usr/local/scriptbasic/bin
```

Tcl Package Path

```
#!/usr/bin/tclsh

puts $tcl_pkgPath
```

When you run the program, you will get a similar output as shown below:

```
/usr/lib64/tcl8.5 /usr/share/tcl8.5 /usr/lib64/tk8.5 /usr/share/tk8.5
```

Tcl Library

```
#!/usr/bin/tclsh  
  
puts $tcl_library
```

When you run the program, you will get a similar output as shown below:

```
/usr/share/tcl8.5
```

Tcl Patch Level

```
#!/usr/bin/tclsh  
  
puts $tcl_patchLevel
```

When you run the program, you will get a similar output as shown below:

```
8.5.7
```

Tcl Precision

```
#!/usr/bin/tclsh  
  
puts $tcl_precision
```

When you run the program, you will get a similar output as shown below:

```
0
```

Tcl Startup File

```
#!/usr/bin/tclsh  
  
puts $tcl_rcFileName
```

When you run the program, you will get a similar output as shown below:

```
~/.tclshrc
```

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://store.tutorialspoint.com>