TCL BEGINNER'S GUIDE

A SMOOTH START PART ONE

WHAT IS TCL ?

- Its pronounced (Tickle)

TCL (Tool Command Language) is a scripting language known for its simplicity and flexibility. It is commonly used for rapid prototyping, scripting, and embedded applications. This guide will cover the basics of TCL, including variables, input/output, control structures, and loops.

WHY DID IT BECAME INDUSTRY STANDARD ?

TCL has become the de facto standard embedded command language for Electronic Design Automation (EDA) applications. Whether you need to automate repetitive behavior, extend the functionality of an application, control multiple tools with a single script or create a custom GUI, TCL is your best choice.

BASIC INPUT AND OUTPUT

- Output using (**puts**)

➔The format is: puts “what you want to print”

➔If you want to print a variable value: puts “the value of your variable is $var”

- Input using (**gets**)

➔We use it with stdin format like: (gets stdin name), here the user will provide a string that will be stored in a variable called **name.**

***Note that TCL is much like python in dealing with variables**

BASIC INPUT AND OUTPUT

How the script looks like:

```
puts "Welcome, Can you tell me what is your name? "
gets stdin name
puts "Nice to meet you $name"
```

How we Execute it :

```
mohammad@DESKTOP-7CC1858:~/TCL$ tclsh ExampleOne.tcl
Welcome, Can you tell me what is your name?
```

Notice how the $ helped us printing the string

```
Mohammad
Nice to meet you Mohammad
mohammad@DESKTOP-7CC1858:~/TCL$
```

VARIABLES AND EXPRESSIONS

- TCL uses (set) to declare a variable: set x 10, here x is getting the value of 10

➔To assign a floating point number we just change the 10 to 10.0, here all calculations are done assuming floating point operations.

- Expressions in TCL are evaluated using (expr):

set a 5

set b 3

puts "Addition: [**expr** {$a + $b}]“

Everything inside “ “ is for printing, everything inside [] is for the expr and everything inside the {} is for evaluating.

VARIABLES AND EXPRESSIONS

Assigning values using (set), printing using (puts)

```
set x 10
set y 20
puts " The addition result of X and y is : [expr {$x + $y}]"
```

After execution it looks like a static calculator

```
mohammad@DESKTOP-7CC1858:~/TCL$ tclsh ExampleTwo.tcl
 The addition result of X and y is : 30
```

REFERENCES

- Tutorial point
- One Compiler
- TCL Manual

If you need help with more resources please feel free to contact me, or lets chat on Calendly :

Lets Chat