

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.

CONDITIONAL STATEMENTS

- Like other programming languages TCL has conditional structures like:
- If {condition} {
Body} elseif {condition}
{body} else {body}

CONDITIONAL STATEMENTS

This is how the script would look like:

```
set x 10
set y 11
if {$x > $y} {
  puts "x is greater than y"
} else {
  puts "y is greater than x"
}
```

As we can see y is greater than x, so if the conditions are written well we will see the second print statement

This is the output of the execution:

```
mohammad@DESKTOP-7CC1858:~/TCL$ tclsh ExampleThree.tcl
y is greater than x
```

CONDITIONAL STATEMENTS

- We can simplify the conditional statements, if – else statements in precise to ternary operator form.

```
[expr {$x > $y ? "x is greater than y" : "y is greater than x"}]
```

After the question mark we have the output if this statement is true and after the colon we have the output if the statement is false

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](#)