Tcsh Shell Scripting Cheatsheet

A comprehensive cheat sheet for Tcsh shell scripting, covering syntax, commands, and best practices to help you write effective and efficient scripts.

Tcsh Basics

Shell Invocation

`tcsh` - Starts a new Tcsh shell.
`tcsh script.tcsh` - Executes a Tcsh script.
`tcsh -f script.tcsh` - Executes script without sourcing `.cshrc`.
`tcsh -x script.tcsh` - Executes script with tracing (debugging).
`tcsh -v script.tcsh` - Executes script with verbose output.

Basic Syntax

`#! /bin/tcsh`	Shebang line, specifies the interpreter for the script.
`set variable = value`	Variable assignment.
`echo $variable`	Prints the value of a variable.
`# comment`	Single-line comment.
`exit`	Exits the script.

Input/Output Redirection

`>`	Redirects output to a file (overwrites).
`>>`	Appends output to a file.
`<`	Redirects input from a file.
`|`	Pipes output to another command.

Variables and Expressions

Variable Types

Tcsh primarily uses string variables. Numbers are also treated as strings unless used in numerical expressions.

Variable Assignment and Usage

`set variable = value`	Assigns `value` to `variable`.
`set variable = ($item1 $item2 ...)`	Assigns a list of items to `variable`.
`echo $variable`	Prints the value of the variable.
`echo $variable[n]`	Prints the nth element of the list variable (index starts at 1).
`unset variable`	Unsets a variable.

Expressions

`@ variable = expression`	Evaluates an arithmetic expression and assigns the result to `variable`.
`@ variable++`	Increments `variable`.
`@ variable--`	Decrements `variable`.
`@ variable += value`	Adds `value` to `variable`.
`@ variable -= value`	Subtracts `value` from `variable`.

Control Structures

Conditional Statements

`if (condition) then commands endif`

`if (condition) then commands else commands endif`

`if (condition) then commands else if (condition) then commands else commands endif`

Looping Statements

`foreach variable (wordlist) commands end`

`while (condition) commands end`

Switch Statement

`switch (variable) case value1: commands breaksw case value2: commands breaksw default: commands breaksw endsw`

Built-in Commands

File Manipulation

`ls`	Lists files and directories.
`mkdir directory`	Creates a new directory.
`rm file`	Removes a file.
`rmdir directory`	Removes an empty directory.
`cp source destination`	Copies a file.
`mv source destination`	Moves or renames a file.

Process Control

`ps`	Displays running processes.
`kill pid`	Terminates a process.
`jobs`	Lists background jobs.
`fg %jobid`	Brings a background job to the foreground.
`bg %jobid`	Sends a job to the background.

String Manipulation

`string length $variable`	Returns the length of the string in variable.
`string index $variable position`	Returns the character at the given position in the string.
`string range $variable start end`	Returns a substring from start to end index.