



# :\$cripting with MikroTik

By **SOUMIL GUPTA BHAYA**

**Mikortik Certified Trainer**

# What is Scripting ?

- A scripting language is a programming language that employs a high-level construct to interpret and execute one command at a time.
- Scripts are programs written for a special run-time environment that automate the execution of tasks.
- They are distinct from the core code of the application.
- Often created or at least modified by the end-user.

# Why Do We Need Scripting ?

- Automation of tasks based on events.
- Repetition of regular tasks.
- Batch creation and processing of lists, addresses etc.
- Monitoring of router.

# Basic Scripting

# Commands/Syntax

- Simple Command:

- Syntax:

- Operators:

- Variables:

- /ip address print
- /interface Ethernet set ether1 disabled=yes
- :log info "hello";
- :set VariableName 10;
- :put \$VariableName;
- "+", "-", "=", "<=", "!=", "&&", "||" Etc.
- Arithmetic, Relational, Logical, Bitwise etc.
- global - accessible from all scripts created by current user. < :global Gvar >
- local - accessible only within the current scope. < :local Lvar >

# Basic Scripting

- Variable Declaration:

- Basics:

- Conditional Statements:

- Parentheses:

## Commands/Syntax:

- Declaration of variables at start of script.
- Always keep Scope of variables in mind. (Global/Local)
- `:global Var -> Variable Declaration`
- `:put $Var -> Prints Value of "Var" in console`
- `:log info/warning/error "Hello"; -> Prints Hello in logs.`
- `#comments` are not executed in the script.
- `:if (<condition>) do={} else={}`
- Eg: `:if ($a>$b) do{[:put ($a-$b)]} else={[:put ($b-$a)]}`
- `{}` Defines Scope.
- `[]` Command Substitution.
- `()` Sub Expression/Grouping.
- In the above Example, everything within the `{}` falls within the scope of the `do/ else` commands. `[]` is used for command within command. `()` is for grouping of arithmetic operation.

# Structuring a Script

- Declaration of Variables
- Setting of Values
- Body

```
:global ddnsuser "theddnsusername"
:global ddnspass "theddnspassword"
:global theinterface "interfacename"
:global ddnshost blabla.dyndns.org
:global ipddns [:resolve $ddnshost];
:global ipfresh [ /ip address get [/ip address find interface=$theinterface ] address ]
:if ( [ :typeof $ipfresh ] = nil ) do={
    :log info ("DynDNS: No ip address on $theinterface .")
} else={
    :for i from=( [:len $ipfresh] - 1) to=0 do={
        :if ( [:pick $ipfresh $i] = "/" ) do={
            :set ipfresh [:pick $ipfresh 0 $i];
        } } :if ($ipddns != $ipfresh) do={
            :log info ("DynDNS: IP-DynDNS = $ipddns")
            :log info ("DynDNS: IP-Fresh = $ipfresh")
            :log info "DynDNS: Update IP needed, Sending UPDATE...!"
            :global str
            "/nic/update\?hostname=$ddnshost&myip=$ipfresh&wildcard=NOCHG&mx=NOCHG&ba
ckmx=NOCHG"
            /tool fetch address=members.dyndns.org src-path=$str mode=http user=$ddnsuser \
                password=$ddnspass dst-path=("/DynDNS.".$ddnshost)
            :delay 1
            :global str [/file find name="DynDNS.$ddnshost"];
            /file remove $str
            :global ipddns $ipfresh
            :log info "DynDNS: IP updated to $ipfresh!"
        } else={
            :log info "DynDNS: dont need changes";
        }
    }
}
```

**Don't Forget Comments!**

# Basic Scripting: Examples

- Regular Backup Scheduling
  - Creating Backups/Export Files
  - Sending Files Via Email Tool or FTP
  - Scheduling The Script

```
:put "Running System Backup
:system backup save name=RegBack;
:put "System Backed Up Successfully";
:tool e-mail send
to="xxxxxxxx@gmail.com" body="test
mail" subject="$[/system identity get
name] $[/system clock get time]
$[/system clock get date] Alert"
from="xxxxxxxx@gmail.com"
server="$[:resolve "smtp.gmail.com"]"
tls=yes file="RegBack";
:put "System Backup Successfully E-
mailed"
```

```
system scheduler add name="daily
Backup" interval=24:00:00 s
tart-time=00:00:00 on-event=":system
script run stsbkp;" disabled=no
```

# Basic Scripting: Examples

- Change Gateway to Backup Interface

- Increasing the default route distance of DHCP Client
- Sending SMS to inform about successful change in gateway

```
:ip dhcp-client set 0 default-route-distance=5;
```

```
:tool sms send port=Airtel channel=3 phone-number="9051634972" message="Gateway changed to 3g";
```

- Change Gateway back to default Interface

- Decreasing the default route distance of DHCP Client
- Sending SMS to inform about successful change in gateway

```
:ip dhcp-client set 0 default-route-distance=0;
```

```
:tool sms send port=Airtel channel=3 phone-number="9051634972" message="Gateway Changed to LAN";
```



# Advanced Scripting

## Commands/Syntax

- **Loops:**
  - **Do..While** -> :do {<commands>} while=(<conditions>);
  - **For** -> :for <var> from=<integer> to=<integer> step=<integer> do={<commands>}
  - **For Each** -> :foreach <var> in=<array> do={ <commands> };
- **System Parameters:**
  - **System Identity** -> :global SystemIdentity [/system identity get name]
  - **System Clock** -> :global SystemDate [/system clock get date]
  - **CPU Load** -> :global CpuLoad [/system resource get cpu-load]
- **Functions:**
  - Functions can be called in a script. Global functions can be called in any script, thus saving excess coding.
  - Eg:

```
:global myFunc do={ :return ($a + $b)}
:put [$myFunc a=6 b=2]
output->8
```

# Advanced Scripting

## More Commands

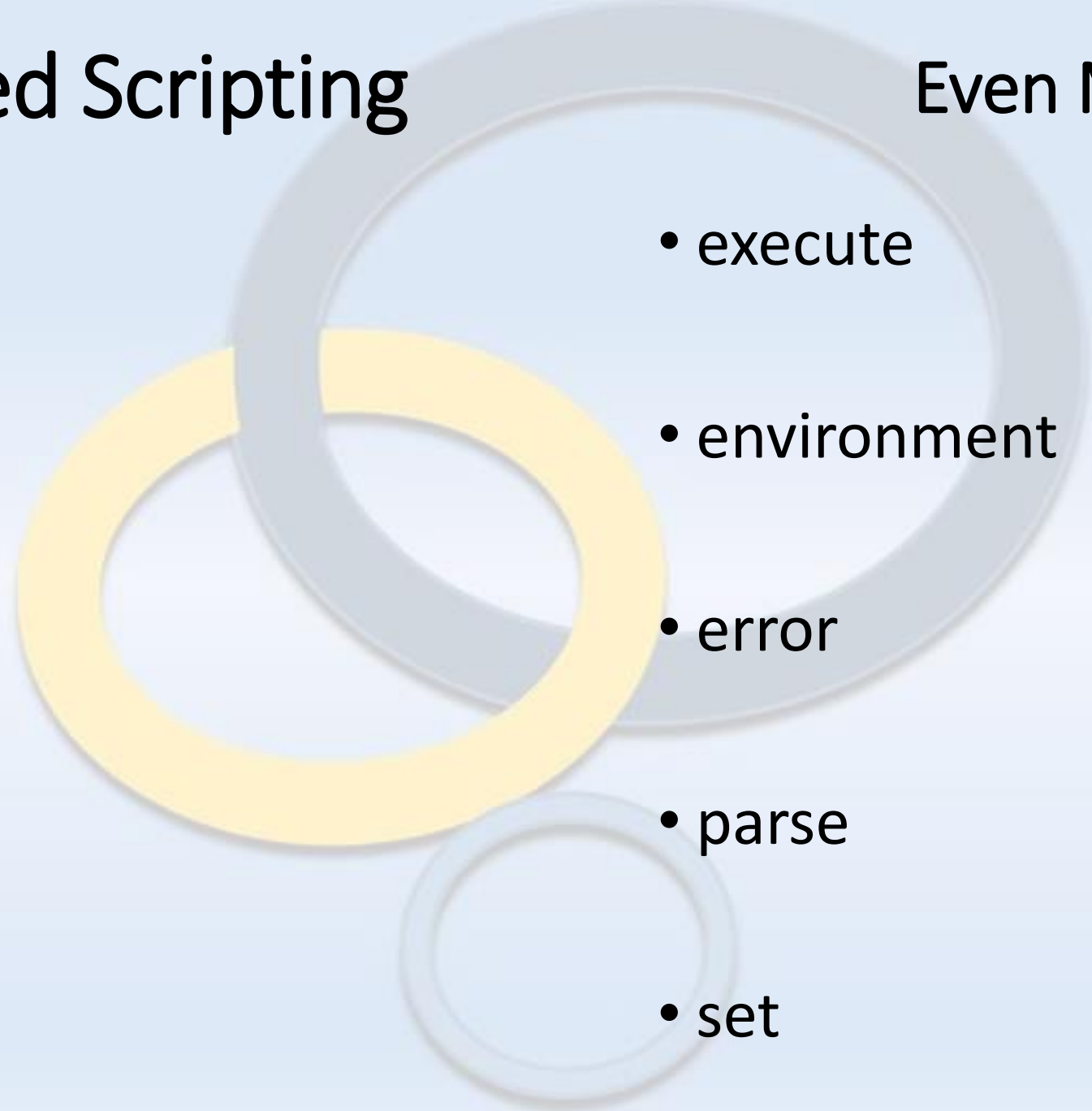
- Resolve
  - Returns the IP address resolution for a DNS lookup.
  - `[:resolve xyz.com]`
- Find
  - Finds the first occurrence of the search term.
  - `[:find <str> <tofind>]`
  - Eg: `:put [/interface find name~"ether"]`
- Append
  - Appends data to previous file.
  - Eg: `:tool sms inbox print append file="text log";`
- Pick
  - Returns specific sections of provided value.
  - `:pick <str> <start> <end>`
  - `:put [:pick "abcde" 1 3]`
- Get
  - Get selected items parameter value.
  - `get <id> <param>=<value>`

# Advanced Scripting

- beep
- delay
- len
- typeof
- time

# Even More

- execute
- environment
- error
- parse
- set



# Advanced Scripting : Examples

## Example of a Loop

- Deleting all received messages from inbox.

```
:local smscount;  
:set smscount [:tool sms inbox print count-only];  
:local i;  
:set i 1;  
:tool sms inbox print append file="text log";  
:for j from=$i to=$smscount step=1 do={:tool sms  
inbox print; :tool sms inbox remove numbers=0;};  
:set smscount [tool sms inbox print count-only];
```

# Advanced Scripting

- Blocking websites using DNS address
- Creating the address list.
- Creating the Firewall Rule..
- Picking each entry in the “host” list.
- Resolving the DNS given in the comment
- Putting the address in the address list.

```
/ip firewall address-list add address=0.0.0.0  
comment=www.blinknet.com  
list=host_blinknet
```

```
/ip firewall filter add chain=forward dst-  
address-list=host_blinknet action=drop
```

# Another Example

```
:local list  
:local comment  
:local newip  
:local oldip  
# Loop through each entry in the address list.  
:foreach i in=[/ip firewall address-list find] do={  
:set list [:pick [/ip firewall address-list get \${i} list] 0 5]  
# If they're 'host_', then we've got a match - process it  
:if (\${list} = "host_") do={  
:set comment [/ip firewall address-list get \${i} comment]  
:set oldip [/ip firewall address-list get \${i} address]  
# Resolve it and set the address list entry accordingly.  
: if (\${newip} != \${oldip}) do={:set newip [:resolve \${comment}]  
/ip firewall address-list set \${i} address=\${newip}} }
```

# Advanced Scripting : Examples

## Example of Fetching System Information

- Setting Parameters to variables.
- Displaying in console and/or Logs.
- Sending the fetched data via SMS to Admin

```
:local cpuload;  
:set cpuload [system resource get cpu-load];  
:local uptime;  
:set uptime [system resource get uptime ];  
:local version;  
:set version [system resource get version ];  
:local freemem;  
:set freemem [system resource get free-memory ];  
:local freq;  
:set freq [system resource get cpu-frequency ];  
:local sysid;  
:set sysid [system identity get name];  
:put "Router Name: $sysid | Up Time: $uptime | Cpu Load: $cpuload  
% | Version: $version | Free Memory: $freemem | CPU Frequency:  
$freq ";  
:tool sms send port=Airtel channel=3 phone-number="9051634972"  
message="Router Name: $sysid | Up Time: $uptime | Cpu Load:  
$cpuload % | Version: $version | Free Memory: $freemem | CPU  
Frequency: $freq ";
```

# What Can Be Achieved



- Automated system monitoring.
- Advanced Load Balancing.
- Extending Functionality.
- Easing of repetitive jobs.
- Much more, only limited by your imagination.
- A lot of resource available in the web.



# Thank You for Your Attention

Questions???