

# Sample Scripts

????

- [ip.sh](#) - ?? Public IP ????
- [whoamifuck.sh](#) - Linux ???????

## Auto-rar.sh

?????????? (\*.part0XX.rar)

```
#!/bin/bash

echo "-> Started: "`date +%m/%d/%y\ %H:%M\ %Z`
echo "-> As:"`whoami`

i=1
ARCHIVES="$@"
for f in $ARCHIVES; do
    PARTCHECK=$(expr "$f" : ".*\([Pp][Aa][Rr][Tt][0-9]\+\.\[Rr][Aa][Rr]\)")
    RARCHECK=$(expr "$f" : "\(.*\.\[Rr][Aa][Rr]\)")
    echo "-> Processing($i of $#): $f"

    if [ "$PARTCHECK" ] && [ -f $f ]; then
        echo "-> Extracting Multipart Archive"
        unrar x $f
        if [ $? -eq 0 ]; then
            FILES=$(expr "$f" : "\(.*\[Pp][Aa][Rr][Tt]\).*")
            echo "-> Extraction Successful"
            echo "-> Removing $FILES*.rar"
            rm $FILES*.rar
        else
            echo "-> **Extraction Failed"
            exit 1
        fi
    else
        echo "-> **Extraction Failed"
        exit 1
    fi
done
```

```

if [ "$RARCHHECK" ] && [ -f $f ]; then
    echo "-> Extracting Single Archive"
    unrar x $f
    if [ $? -eq 0 ]; then
        echo "-> Extraction Successful"
        echo "-> Removing $f"
        rm $f
    else
        echo "-> **Extraction Failed"
        exit 1
    fi
fi
fi
echo ""
i=$(( i+1 ))
done

```

## Domaincheck.sh

```

#!/bin/bash

#Specify all the domains you want to check
DOMAINS="google.com dribbble.com facebook.com youtube.com"

current_epoch=`date +%s`
for dm in $DOMAINS
do
    expiry_date=`whois $dm | egrep -i "Expiration Date:|Expires on"| head -1 | awk '{print $NF}'`
    echo -n " $dm - Expires on $expiry_date "
    expiry_epoch=`date --date="$expiry_date" +%s`
    epoch_diff=`expr $expiry_epoch - $current_epoch`
    days=`expr $epoch_diff / 86400`
    echo " $days days remaining. "
done

```

## Scripting Basics

# BASH SCRIPTING BASICS

```
1  #!/bin/bash
2
3  username="Jay"
4  filename=$3
5
6  read -p "Enter your username: " user
7  echo "Username: $user"
8
9  if [ "$EUID" -ne 0 ]; then
10     echo "You are not running this script as the root user."
11 else
12     echo "You are running this script as the root user."
13 fi
14
15 echo "Counting to 5:"
16 for i in {1..5}; do
17     echo "$i"
18 done
19
20
21 function greet() {
22     echo "Hello, $1!"
23 }
24 greet "Alice"
25
26 echo "Enter a number between 1 and 2: "
27 read num
28 case $num in
29     1) echo "You chose one." ;;
30     2) echo "You chose two." ;;
31     *) echo "Invalid choice." ;;
32 esac
33
34 if [ -e "$filename" ] && [ -d "$filename" ]; then
35     echo "File exists and is a directory."
36 else
37     echo "File does not exist or is not a directory."
38 fi
39
40 echo "First argument: $1"
41 echo "Second argument: $2"
42
43 cat nonexistent-file.txt 2> /dev/null
44 echo "Exit status: $?"
45
46 fruits=("Apple" "Orange" "Banana")
47 echo "Fruits: ${fruits[0]}"
48
49 declare -A capitals
50 capitals[USA]="Washington D.C."
51 capitals[France]="Paris"
52 echo "Capital of France: ${capitals[France]}"
53
54 current_date=$(date)
55 echo "Today's date is: $current_date"
56
57 echo "This is a sample text." > example.txt
58 find / -name hello.txt &> /dev/null
59
60 result=$(( expr 15 - 2 ))
61 echo $result
62
63 SRC="/path/to/foo.cpp"
64 BASEPATH=${SRC##*/}
65 echo $BASEPATH
66
67 trap 'echo "Received SIGTERM signal. Cleaning up."; exit' SIGTERM
68
69 # This is a single line comment
70
71 : ' this a multiline
72    comment'
```

Shebang Line

Variables

User Input

Conditional if Statement

For Loop

Functions

Conditional Case Statement

File Operations

Command Line Arguments

Exit Status Codes

Indexed Arrays

Associative Arrays

Command Substitution

Command Line Redirections

Arithmetic Operations

Parameter Expansion

Process Signal Handling

Comments

 bash script.sh

---

Revision #5  
Created 1 June 2020 07:44:00 by Admin  
Updated 11 November 2024 19:15:58 by Admin