

# Tarun's Bash Scripting Notes – Hindi Edition (with Examples and Descriptions)

---

## 1 वेरिएबल्स और यूज़र इनपुट (Variables & Input)

### विवरण:

वेरिएबल्स में डेटा को स्टोर किया जाता है। इनपुट `read` से लिया जाता है।

### ◊ उदाहरण:

```
name="Tarun"           # वेरिएबल में वैल्यू स्टोर हो रही है
echo "Hello $name"      # वेरिएबल की वैल्यू टर्मिनल पर दिखा रहा है

read city               # यूज़र से input ले रहा है और city वेरिएबल में स्टोर कर रहा है
echo "आपका शहर है: $city" # स्टोर की गई वैल्यू को टर्मिनल पर प्रिंट कर रहा है
```

## 2 शर्तें (If-Else Conditions)

### विवरण:

शर्तों के आधार पर निर्णय लेना।

### ◊ उदाहरण:

```
if [ $age -gt 18 ]; then      # चेक कर रहा है: age 18 से ज़्यादा है क्या?
    echo "वयस्क"
elif [ $age -eq 18 ]; then    # अगर age = 18 हो, तब यह चलेगा
    echo "ठीक 18 साल के हो"
else
    echo "अवयस्क"             # उपरोक्त दोनों false हों तो
fi                             # if स्टेटमेंट का अंत
```

## 3 लूप्स (Loops - for, while)

### विवरण:

लूप का उपयोग किसी कार्य को बार-बार करने के लिए होता है।

### ◊ For loop:

```
for i in {1..5}; do          # i को 1 से 5 तक increment करता है
    echo "नंबर: $i"           # हर बार i की वैल्यू प्रिंट करता है
```

```
done                # loop का अंत
```

#### ◇ While loop:

```
i=1
while [ $i -le 5 ]; do
    echo $i
    i=$((i+1))
done
```

---

## 4) केस स्टेटमेंट (Case Statement)

### 📖 विवरण:

मल्टीपल विकल्पों के लिए उपयोग होता है।

#### ◇ उदाहरण:

```
read choice
case $choice in
    1) echo "Option 1";;
    2) echo "Option 2";;
    *) echo "गलत विकल्प";;
esac
```

---

## 5) फ़ंक्शन्स (Functions)

### 📖 विवरण:

कोड ब्लॉक जिसे बार-बार बुलाया जा सकता है।

#### ◇ उदाहरण:

```
greet() {                # function शुरू होता है 'greet' नाम से
    echo "नमस्ते $1!"     # जो भी value पास करें, वो $1 में आएगी
}
greet "Tarun"            # Function को कॉल करते हुए "Tarun" भेजा गया
```

---

## 6) ऐरे (Arrays)

### 📖 विवरण:

ऐरे एक जैसे कई वैल्यूज़ को स्टोर करने के लिए।

#### ◇ उदाहरण:

```
fruits=("सेब" "केला" "आम")
echo ${fruits[0]}      # सेब
echo ${fruits[@]}      # सभी फलों को दिखाएगा
```

---

## 7 कमांड लाइन आर्गुमेंट्स (Arguments)

### 📖 विवरण:

स्क्रिप्ट को चलाते समय टर्मिनल से वैल्यू भेजना।

#### ◊ उदाहरण:

```
echo "Script name: $0"
echo "First argument: $1"
echo "All arguments: $@"
```

---

## 8 फाइल हैंडलिंग (File Handling)

### 📖 विवरण:

फाइल बनाना, लिखना, पढ़ना और जोड़ना।

#### ◊ उदाहरण:

```
echo "Hello" > file.txt      # नई फाइल बनाएगा
echo "More" >> file.txt      # उसी फाइल में जोड़ देगा
cat file.txt                 # फाइल को पढ़ेगा
```

---

## 9 स्ट्रिंग ऑपरेशन (String Manipulation)

### 📖 विवरण:

स्ट्रिंग के अंदर से टुकड़ा निकालना या उसकी जानकारी लेना।

#### ◊ उदाहरण:

```
text="LinuxScript"
echo ${#text}      # लंबाई - 11
echo ${text:0:5}   # Substring - Linux
```

---

## 10 ट्रैप और सिग्नल्स (Trap and Signals)

### 📖 विवरण:

Ctrl+C जैसा interruption handle करना।

#### ◊ उदाहरण:

```
trap "echo Ctrl+C दबाया गया!" SIGINT
while true; do
    sleep 1
done
```

## 1.1 ऑपरेटर्स (Operators)

### 📖 विवरण:

तुलना, गणना और लॉजिकल conditions के लिए।

प्रकार	उदाहरण	उपयोग
गणितीय	<code>\$((5+3))</code>	जोड़, घटाव
तुलना	<code>-gt, -lt</code>	तुलना करना
लॉजिकल	<code>-a, -o</code>	AND, OR
स्ट्रिंग	<code>-z, -n</code>	खाली या भरा

## 1.2 स्क्रिप्ट को रन करना और डिबग करना

```
chmod +x script.sh      # executable बनाओ
./script.sh             # स्क्रिप्ट चलाओ
bash -x script.sh       # debug mode
```

## 1.3 इंटरैक्टिव मेनू (Select Menu)

### 📖 विवरण:

यूज़र से विकल्प दिलवाने के लिए।

#### ◊ उदाहरण:

```
echo "मेनू:"
select opt in "तारीख" "फाइल लिस्ट" "बाहर"; do
    case $REPLY in
        1) date;;
        2) ls;;
        3) break;;
        *) echo "गलत इनपुट";;
```

```
esac  
done
```

---

## 🧠 अभ्यास सुझाव:

- रोज़ एक छोटा script बनाएं
- `man` कमांड से हर Linux टूल को explore करें
- Cron Jobs और log मॉनिटरिंग scripts बनाने की कोशिश करें

---

## 👋 अंतिम बात:

Bash scripting से आप Linux को automation से control कर सकते हैं।

**अभ्यास करें, mini projects बनाएं, और shell का master बनें!**