

Day 2: TechSaksham Training

Date: 12/02/2025

Operators

1. **Arithmetic Operators** – Used for mathematical operations.
2. **Logical Operators** – Used for comparison of conditions:
 - AND = `&&`
 - Less than = `<`
 - Greater than or equal to = `>=`
 - Less than or equal to = `<=`
 - OR = `|` (backquote, below ESC button)
 - NOT = `!`
3. **Comparison Operators** – Compares values:
 - Equal = `==`
 - Greater than = `>`
4. **Bitwise Operators** – Perform operations on binary values or conditions.
 - XOR – If both conditions are the same, it is true; otherwise, false.

Note: To print a variable in output, use double quotes (""); if not, use single quotes ('').

Conditional Statements

Syntax for `if` condition:

```
if [ condition ];  
then  
    commands / expressions  
fi
```

•

Syntax for `if-else` condition:

```
if [ condition ];  
then  
    commands / expressions  
else  
    commands / expressions  
fi
```

•

Example: Check if a number is even or odd

```
#!/bin/bash
num=5
res=$(( num % 2 ))
if [ $res -eq 0 ];
then
    echo "$num is even"
else
    echo "$num is odd"
fi
```

Loops

For Loop

Syntax:

```
for variable in list
do
    expression
done
```

-

Example:

```
for i in 1 2 3 4
do
    echo "$i"
done
```

-

While Loop

Syntax:

```
initialization/start
while [ condition ]
do
    expression
    increment/decrement
done
```

-

Until Loop

Syntax:

```
initialization/start
until [ condition ]
do
    expression
    increment/decrement
done
```

-

Nested Loops

Example: Print multiplication tables from 1 to 10

```
#!/bin/bash
for ((i=1;i<=10;i++))
do
    for ((j=1;j<=10;j++))
    do
        res=$(( i * j ))
        echo "$i x $j = $res"
    done
done
echo " "
```

-

Pattern Printing

Pattern 1:

```
*****
*
*
*
*
```

Functions

Defining a function:

```
function name_of_function()
{
    echo "Hello"
}
```

-

Calling a function:

name_of_function

-

Using arguments in functions:

```
function my_function()  
{  
    echo "Argument 1: $1"  
    echo "Argument 2: $2"  
}
```

-

Return Values:

```
return_value=$?  
echo "Return value is: $return_value"
```

-

Additional Notes:

- Use `$0` to get the script name.
- Use `$#` to get the count of arguments.

This document summarizes everything covered in Day 2 of the TechSaksham training. 🚀