

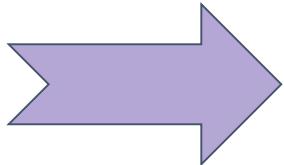
# Conditional Statements

NPTEL

# Conditional Statements (if-else)



**if, else, and elif**



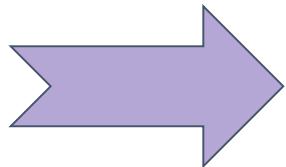
Conditional statements allow you to execute certain parts of the code based on conditions.

NPTEL

# Conditional Statements (if-else)



**if, else, and elif**



Conditional statements allow you to execute certain parts of the code based on conditions.

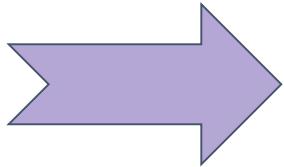
Syntax

NPTEL

# Conditional Statements (if-else)



## if, else, and elif



Conditional statements allow you to execute certain parts of the code based on conditions.

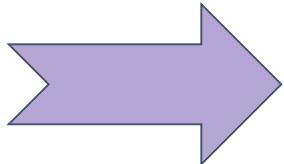
## Syntax

```
if [condition]; then  
    Statement 1  
elif [condition]; then  
    Statement 2  
else  
    Statement 3  
fi
```

# Conditional Statements (if-else)



**if, else, and elif**



Example-1: Checking File Existence

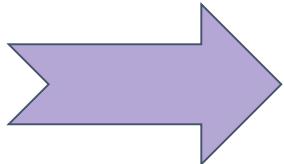
```
#!/bin/bash
if [ -f "file1" ]; then
    echo "File exists."
else
    echo "File does not exist."
fi
```

NPTEL

# Conditional Statements (if-else)



**if, else, and elif**



## Example-1: Checking File Existence

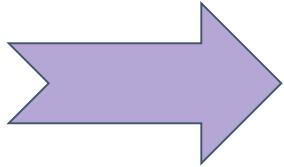
```
#!/bin/bash
if [ -f "file1" ]; then
    echo "File exists."
else
    echo "File does not exist."
fi
```

```
[root@localhost Test2]# ls
condition1.sh name.sh
[root@localhost Test2]# ./condition1.sh
File does not exist.
[root@localhost Test2]# touch file1
[root@localhost Test2]# ls
condition1.sh file1 name.sh
[root@localhost Test2]# ./condition1.sh
File exists.
[root@localhost Test2]#
```

# Conditional Statements (if-else)



if, else, and elif



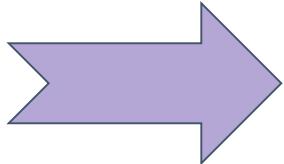
Example-2: Checking Numbers

```
#!/bin/bash
num=10
if [ $num -gt 5 ] ; then
    echo "$num is greater than 5"
elif [ $num == 5 ] ; then
    echo "$num is equal to 5"
else
    echo "$num is less than 5"
fi
```

# Conditional Statements (if-else)



## if, else, and elif



### Example-2: Checking Numbers

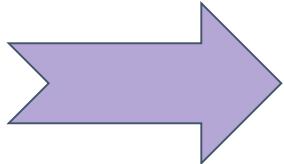
```
#!/bin/bash
num=10
if [ $num -gt 5 ]; then
    echo "$num is greater than 5"
elif [ $num == 5 ]; then
    echo "$num is equal to 5"
else
    echo "$num is less than 5"
fi
```

```
[root@localhost Test2]# cat checkNumbers.sh
#!/bin/bash
num=10
if [ $num -gt 5 ]; then
    echo "$num is greater than 5"
elif [ $num == 5 ]; then
    echo "$num is equal to 5"
else
    echo "$num is less than 5"
fi
[root@localhost Test2]# ./checkNumbers.sh
10 is greater than 5
[root@localhost Test2]#
```

# Conditional Statements (if-else)



if, else, and elif



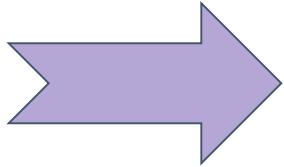
Example-2: Checking Numbers

```
#!/bin/bash
num=5
if [ $num -gt 5 ] ; then
    echo "$num is greater than 5"
elif [ $num == 5 ] ; then
    echo "$num is equal to 5"
else
    echo "$num is less than 5"
fi
```

# Conditional Statements (if-else)



## if, else, and elif



### Example-2: Checking Numbers

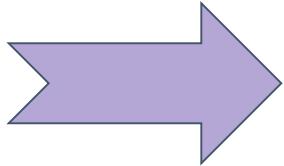
```
#!/bin/bash
num=5
if [ $num -gt 5 ]; then
    echo "$num is greater than 5"
elif [ $num == 5 ]; then
    echo "$num is equal to 5"
else
    echo "$num is less than 5"
fi
```

```
[root@localhost Test2]# cat checkNumbers.sh
#!/bin/bash
num=5
if [ $num -gt 5 ]; then
    echo "$num is greater than 5"
elif [ $num == 5 ]; then
    echo "$num is equal to 5"
else
    echo "$num is less than 5"
fi
[root@localhost Test2]# ./checkNumbers.sh
5 is equal to 5
[root@localhost Test2]#
```

# Conditional Statements (if-else)



if, else, and elif



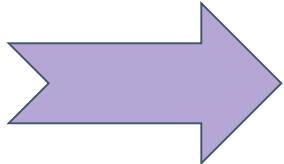
Example-2: Checking Numbers

```
#!/bin/bash
num=3
if [ $num -gt 5 ] ; then
    echo "$num is greater than 5"
elif [ $num == 5 ] ; then
    echo "$num is equal to 5"
else
    echo "$num is less than 5"
fi
```

# Conditional Statements (if-else)



## if, else, and elif



### Example-2: Checking Numbers

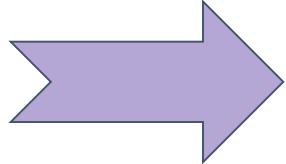
```
#!/bin/bash
num=3
if [ $num -gt 5 ]; then
    echo "$num is greater than 5"
elif [ $num == 5 ]; then
    echo "$num is equal to 5"
else
    echo "$num is less than 5"
fi
```

```
[root@localhost Test2]# cat checkNumbers.sh
#!/bin/bash
num=3
if [ $num -gt 5 ]; then
    echo "$num is greater than 5"
elif [ $num == 5 ]; then
    echo "$num is equal to 5"
else
    echo "$num is less than 5"
fi
[root@localhost Test2]# ./checkNumbers.sh
3 is less than 5
[root@localhost Test2]#
```

# Conditional Statements (if-else)



**if, else, and elif**



**Example-3: Checking the grade of student with user input**

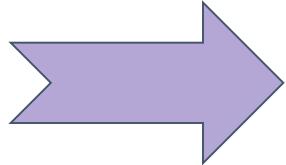
```
#!/bin/bash
echo "Enter your marks:"
read marks
if [ $marks -ge 90 ]; then
    echo "Grade: A"
elif [ $marks -ge 80 ]; then
    echo "Grade: B"
else
    echo "Grade: C"
fi
```

NPTEL

# Conditional Statements (if-else)



## if, else, and elif



### Example-3: Checking the grade of student with user input

```
#!/bin/bash
echo "Enter your marks:"
read marks
if [ $marks -ge 90 ]; then
    echo "Grade: A"
elif [ $marks -ge 80 ]; then
    echo "Grade: B"
else
    echo "Grade: C"
fi
```

```
[root@localhost Test2]# ./grade.sh
Enter your marks:
95
Grade: A
[root@localhost Test2]# ./grade.sh
Enter your marks:
45
Grade: C
[root@localhost Test2]# ./grade.sh
Enter your marks:
80
Grade: B
[root@localhost Test2]#
```



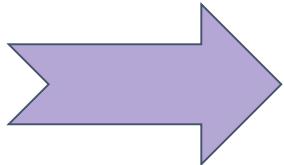
# Loops in Shell Scripting

NPTEL

# Loops in Shell Scripts



## For loop, While loop



Loops allow you to execute a block of code multiple times.

NPTEL

# Loops in Shell Scripts



## For loop

Example-1: Looping Through a List of Numbers

```
#!/bin/bash
for i in 1 2 3 4 5; do
    echo "Number: $i"
done
```

# Loops in Shell Scripts



## For loop

### Example-1: Looping Through a List of Numbers

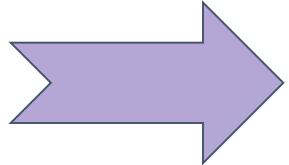
```
#!/bin/bash
for i in 1 2 3 4 5; do
    echo "Number: $i"
done
```

```
[root@localhost Test2]# cat forLoop.sh
#!/bin/bash
for i in 1 2 3 4 5; do
    echo "Number: $i"
done
[root@localhost Test2]# ./forLoop.sh
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
[root@localhost Test2]#
```

# Loops in Shell Scripts



## While loop



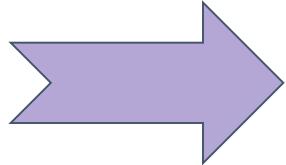
### Example-1: Count the Numbers

```
#!/bin/bash
count=1
while [ $count -le 5 ];
do
  echo "Count: $count"
  count=$((count + 1))
Done
```

# Loops in Shell Scripts



## While loop



### Example-1: Count the Numbers

```
#!/bin/bash
count=1
while [ $count -le 5 ];
do
    echo "Count: $count"
    count=$((count + 1))
Done
```

```
[root@localhost Test2]# cat whileLoop.sh
#!/bin/bash
count=1
while [ $count -le 5 ]; do
    echo "Count: $count"
    count=$((count + 1))
done
[root@localhost Test2]# ./whileLoop.sh
Count: 1
Count: 2
Count: 3
Count: 4
Count: 5
[root@localhost Test2]# █
```

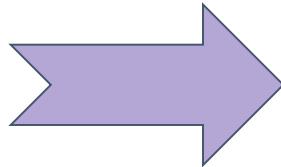
# Functions in Shell Scripting

NPTEL

# Functions in Shell Scripts



## Defining and Using Functions



Functions in shell scripting help modularize code and reuse sections.

NPTEL

# Functions in Shell Scripts



## Defining and Using Functions

### Basic Function

```
#!/bin/bash
my_function() {
    echo "This is a
function."
}
my_function
```

NPTEL

# Functions in Shell Scripts



## Defining and Using Functions

### Basic Function

```
#!/bin/bash
my_function() {
    echo "This is a
function."
}
my_function
```

```
[root@localhost Test2]# cat function.sh
#!/bin/bash
my_function() {
    echo "This is a function."
}
my_function
[root@localhost Test2]# ./function.sh
This is a function.
[root@localhost Test2]#
```

# Functions in Shell Scripts



## Defining and Using Functions

### Basic Function

```
#!/bin/bash
my_function() {
    echo "This is a
function."
}
my_function
```

```
[root@localhost Test2]# cat function.sh
#!/bin/bash
my_function() {
    echo "This is a function."
}
[root@localhost Test2]# ./function.sh
[root@localhost Test2]#
```

# Functions in Shell Scripts



## Returning Values from Functions

Function with Return Value

```
#!/bin/bash
add() {
    sum=$(( $1 + $2 ))
    return $sum
}
add 5 10
echo "The sum is $?"
```

NPTEL

# Functions in Shell Scripts



## Returning Values from Functions

### Function with Return Value

```
#!/bin/bash
add() {
    sum=$(( $1 + $2 ))
    return $sum
}
add 5 10
echo "The sum is $?"
```

Exit status of the last  
executed command

NPTEL

# Functions in Shell Scripts



## Returning Values from Functions

### Function with Return Value

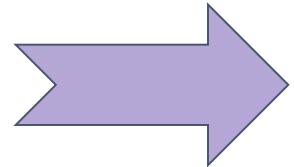
```
#!/bin/bash
add() {
    sum=$(( $1 + $2 ))
    return $sum
}
add 5 10
echo "The sum is $?"
```

```
[root@localhost Test2]# cat functionReturn.sh
#!/bin/bash
add() {
    sum=$(( $1 + $2 ))
    return $sum
}
add 5 10
echo "The sum is $?"
[root@localhost Test2]# ./functionReturn.sh
The sum is 15
[root@localhost Test2]# █
```

# Functions in Shell Scripts



## Functions



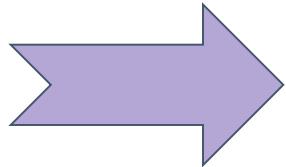
### Example: Factorial Calculation

```
#!/bin/bash
factorial() {
    if [ $1 -le 1 ]; then
        echo 1
    else
        prev=$(factorial $(( $1 - 1 )))
        echo $(( $1 * prev ))
    fi
}
echo "Factorial of 5 is: $(factorial 5)"
```

# Functions in Shell Scripts



## Functions



### Example: Factorial Calculation

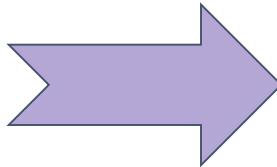
```
#!/bin/bash
factorial() {
    if [ $1 -le 1 ]; then
        echo 1
    else
        prev=$(factorial $(( $1 - 1 )))
        echo $(( $1 * prev ))
    fi
}
echo "Factorial of 5 is: $(factorial 5)"
```

```
[root@localhost Test2]# cat factorial.sh
#!/bin/bash
factorial() {
    if [ $1 -le 1 ]; then
        echo 1
    else
        prev=$(factorial $(( $1 - 1 )))
        echo $(( $1 * prev ))
    fi
}
echo "Factorial of 5 is: $(factorial 5)"
[root@localhost Test2]# ./factorial.sh
Factorial of 5 is: 120
[root@localhost Test2]#
```

# Real-World Use Case: Automation Script



## Disk Space Monitoring Script



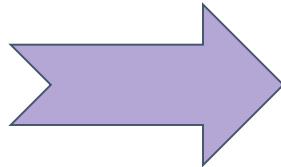
A script to monitor disk space and send alerts if usage exceeds a threshold.

```
df -h | awk 'NR>1 {print $5 " " $1}' | while
read output;
do
    usage=$(echo $output | awk '{print $1}' | sed
's/%//')
    partition=$(echo $output | awk '{print $2}')
    if [ $usage -ge $THRESHOLD ]; then
        echo "Warning: Disk usage for $partition is
at $usage%"
```

# Real-World Use Case: Automation Script



## Disk Space Monitoring Script



A script to monitor disk space and send alerts if usage exceeds a threshold.

```
[root@localhost Test2]# df
Filesystem      1K-blocks   Used   Available  Use% Mounted on
devtmpfs        1914728     0    1914728   0% /dev
tmpfs          1930524     8    1930516   1% /dev/shm
tmpfs          1930524  12748   1917776   1% /run
tmpfs          1930524     0    1930524   0% /sys/fs/cgroup
/dev/sda3      23796376 12403964  11392412  53% /
/dev/sdal       303780   297376     6404  98% /boot
tmpfs          386108     32    386076   1% /run/user/1000
[root@localhost Test2]# ./diskMonitor.sh
Warning: Disk usage for /dev/sdal is at 98%
[root@localhost Test2]#
```

# Recap

- Introduction to Shell Scripting
- Basic Shell Script Structure
- Shell Variables and Input/Output
- Conditional Statements (if-else)
- Loops in Shell Scripts
- Functions in Shell Scripts
- Real-World Use Case: Automation Script



ਪੰਨਵਾਦ ਸ਼ੁਕਧਿ ਪ੍ਰਸ਼ਾਸਨ  
ਤਹਿਤ ਕਾਰੋਬਾਰ ਧਨ੍ਯਵਾਦ  
ਆਮਾਰ ਧਨ੍ਯਵਾਦ ਲਈ ਧਨ੍ਯਵਾਦ  
**Thank You**  
ਧਨ੍ਯਵਾਦ ਧਨ੍ਯਵਾਦ ਧਨ੍ਯਵਾਦ ਮੌਜੂਦੀ

NPTEL