

Conditional Statement

Condition is a comparison between two values. for comparison you can use test or [expr] statements or even exist status can be also used. expression is defined as - an expression is nothing but combination of values, relational operator (>, <, <>) and mathematical operator (+, -, /). there are the following kinds of statements available in shell programming as decision making statements.

if

simple if is used for decision making in shell script. if the given condition is true then it will execute the set of code that you have allocated to that block.

Syntax

```
if [ condition ]
then
    Execute the statements
fi
```

Example

```
#Check Number is 1

echo "Enter Number:-"
read no

if [ $no -eq 1 ]
then
    echo "Number 1 "
fi
```

if..else

if..else is used for decision making in shell script where the given condition is true then it will execute the set of code that you have allocated to that block otherwise you can execute the rest of code for the false condition.

Syntax

```
if [ condition ]
then
    Execute Statement if Condition is True
elif
```

```
    Execute Statement if Condition is False
fi
```

Example

```
#Check Number is Positive or Not

echo "Enter Number:"
read no

if [ $no -gt 0 ]
then
    echo "Number is Positive"
elif
    echo "Number is Negative"
fi
```

if..elif..else

it is possible to create compound conditional statements by using one or more else if(elif) clause.if the 1st condition is false,then subsequent elif statements are checked.when an elif condition is found to be true,the statements following that associated parts are executed.

Syntax

```
if [ condition ]
then
    Execute Statement if Condition 1
elif [ condition ]
    Execute Statement if Condition 2
elif [ condition ]
    Execute Statement if Condition 3
elif
    Else Condition
fi
```

Example

```
#Find Student Class

echo "Enter Student Mark:-"
read mark

if [ $mark -gt 70]
then
    echo "Distinction"
elif [ $mark -gt 60]
then
    echo "First Class"
elif [ $mark -gt 50]
```

```
then
    echo "Second Class"
elif [ $mark -gt 40]
then
    echo "Pass Class"
elif
    echo "Fail"
fi
```

Nested if

if statement and else statement can be nested in bash shell programming. the keyword "fi" indicates the end of the inner if statement and all if statement should end with "fi".

Syntax

```
if [ condition ]
then
    if [ condition ]
        then
            Execute Statement
        elif
            Execute Statement
        fi
    elif
        Execute Statement
fi
```

Example

```
#Nested if Example

echo "Enter Your Country:"
read cn

if [$cn -eq 'India']
then
    echo "Enter Your State:"
    read st
    if [$st -gt 'Gujarat']
        then
            echo "Welcome to Gujarat"
        elif
            echo "You are Not Gujarati"
        fi
    elif
        echo "Other Country"
fi
```

Case Statement

The case statement is good alternative to multilevel if then else fi statement.it enables you to match several values against one variable.its easier to read and write multiple conditions.

Syntax

```
case $[ variable_name ] in
value1)
    Statement 1
    ;;
value2)
    Statement 2
    ;;
value3)
    Statement 3
    ;;
value4)
    Statement 4
    ;;
valueN)
    Statement N
    ;;
*)
    Default Statement
    ;;
esac
```

Example

```
#Case Statement Example

echo "Enter Country Code:"
read co

case $co in
'IN') echo "India"
    ;;
'PK') echo "Pakistan"
    ;;
*) echo "Enter Vailid Country Code"
    ;;
esac
```