

**GETTING STARTED  
WITH  
TAKING DECISIONS**

➤ **DECISION MAKING:**

- we will understand shell decision-making in Unix. While writing a shell script, there may be a situation when you need to adopt one path out of the given two paths. So, you need to make use of conditional statements that allow your program to make correct decisions and perform the right actions.
- The Bourne shell offers four decision making instructions.
  - The if-then-fi statement
  - The if-then-else-fi statement
  - The if-then-elif-else-fi statement
  - The case-esac statement

## THE “IF-THEN-FI” STATEMENT:

**SYNTAX:**    **if** control command  
              **then**  
              do this  
              and this  
              **fi**

## THE “IF-THEN-ELSE-FI” STATEMENT:

**SYNTAX:**    **if** control command  
              **then**  
              do this  
              **else**  
              do this  
              and this  
              **fi**

## THE “IF-THEN-ELIF-ELSE-FI” STATEMENT:

**SYNTAX:**

```
if control command
then
do this
elif control command
then
do this
else
do this
fi
```

## THE “CASE-ESAC” STATEMENT:

**SYNTAX:**    **case** expression **in**  
                 pattern1 )  
                 statements ;;  
                 pattern2 )  
                 statements ;;  
                 **esac**

### ➤ EXAMPLES OF TAKING DECISIONS:

#### 1. Write a script copying from one file to another:

```
$vim demo1.sh
#!/bin/sh
#Example of Copying using if then fi statement
#Author: Raju
echo enter source and target file names
read source target
if cp $source $target
then
echo "File copied Successfully"
fi
```

#### 2. Write a script copying from one file to another:

```
$vim demo2.sh
#!/bin/sh
#Example of copying file one to another
echo Enter source and target file names
read source target
if cp $source $target
then
echo "File copied Successfully"
else
echo "Failed to copy the file"
fi
```

### 3. Write a script given number is even or odd:

```
$vim demo3.sh
#!/bin/sh
echo -n "Enter a number"
read n
if [ `expr $n%2` -eq 0 ]
then
echo "$n is an Even number"
else
echo "$n is an Odd number"
fi
```

### 4. Write a script the given two strings are equal / not:

```
$vim demo4.sh
#!/bin/sh
echo -n "Enter a string1:"
read string1
echo -n "Enter a string2:"
read string2
if [ "$string1" = "$string2" ]
then
echo "Strings are equal"
else
echo "strings are not equal"
fi
```

### 5. Write a program accept and delete a given file:

```
$vim demo5.sh
#!/bin/sh
echo -n "Enter a file name:"
read fname
if rm -i $fname
then
echo "$fname file is deleted"
else
echo "No such file"
fi
```

## 6. Write a program accept a user and check the user is exist or not:

```
$vim demo6.sh
#!/bin/sh
echo -n "Enter a username:"
read $uname
if grep -w $uname /etc/passwd >/dev/null
then
echo "$uname user exist"
else
echo "$uname user doesn't exist"
fi
```

## 7. Example of case-esac statement

```
$vim demo7.sh
#!/bin/sh
#Example of case-esac statement
echo "Enter a character:\c"
read var
case $var in
[a-z])
echo "You entered a lower case Alphabet."
;;
[A-Z])
echo "You entered a upper case Alphabet."
;;
[0-9])
echo "You entered a Digit."
;;
?)
echo "You entered a special Symbol."
;;
*)
echo "You entered more than one Character."
;;
esac
```

## 8. Another Example of case-esac:

```
$vim demo8.sh
#!/bin/sh
echo "Enter a word:"
read word
case $word in
[aeiou]* | [AEIOU]*)
echo "The word begins with a Vowel."
;;
[0-9]*)
echo "The word begins a Digit."
;;
*[0-9])
echo "The word ends with a Digit."
;;
???)
echo "You entered a three-letter word."
;;
*)
echo "I don't know what you have entered."
;;
esac
```