SHELL SCRIPTS

SHELL SCRIPTS

- Used to group a set of UNIX commands in a file.
- These commands are typed and edited using the vi editor.
- A shell variable can contain alphabets numbers and underscore with the first letter being an alphabet or an underscore.

SHELL SCRIPTS

- **VALID SHELL VARIABLE:**

  *n

  *area

  *a1

  *_account

  *a_count

SHELL SCRIPTS

- **ASSIGNING VALUES TO VARIABLE:**

  Variable=value

  Ex: n=10

- **DISPLAYING VALUES OF VARIABLES:**

  $ echo value of n is $n

SHELL SCRIPTS

- **ARITHEMETIC OPERATIONS:**

+	Addition
-	Subtraction
*****	Multiplication
/	Division
%	Modulo

COMPARISON OPERATORS

- **NUMERICAL:**

-eq	equals to
-gt	greater than
-lt	lesser than
-ge	greater than or equal to
-le	lesser than or equal to
-ne	not equal to

- **STRING:**

=	equal to
!=	not equal to

- **LOGICAL OPERATORS:**

  -a and

  -o or

  -! Not

- **READ COMMAND:**

  Used to read the value of the shell variable from a user.

- **COMMENT STATEMENT:**

  #this is a text program.

- **IF STATEMENT:**

```
if condition
then
  ________
  ________
  <block of commands>
  ________
  ________
fi
```

- **ELSE CLAUSE IN IF STATEMENT:**

```
if condition
then
  ________
  ________
  <block of commands>
  ________
else
  ________
  <block of commands>
  ________
  ________
fi
```

- **ELIF CONSTRUCT:**

```
 if condition
    then
     ________
     ________
     <block of commands>
    elif condition2
then
  ________
  ________
  <block of commands>
  ________
  ________
else
 ________
 <block of commands>
 ________
     ________
    fi
```

TEST COMMAND:

- Used to test for equality and it returns a true or false as a result.

  Syntax: **test condition**

- **CASE STATEMENT:**
  Used to execute a block of commands based on a value.

```
Case value in
V1)command1
               .
               .
    Command n
V2)command1
               .
               .
     Command n
               .
               .
               .

Vn) command1
               .
               .
      Command n
esac
```

- **FOR STATEMENT:**

  Used to execute a block of commands based on a list of values for a variable.

```
for variable in list
do
________
 ________
<block of commands>
________
________
done
```

- **WHILE STATEMENT:**

  Used to execute and to repeat a command depending upon a comdition.

```
while condition
do
   ________
   ________
<block of commands>
   ________
   ________
done
```

- **UNTILL CONDITION:**

  Similar to the while statement used to execute a block of commands based upon the condition.

```
until condition
          do
          _______
           _______
          <block of commands>
           _______
           _______
          done
```

Example

```
echo sum and product
echo enter the value of a
read a
echo enter the value of b
read b
c=`expr $a + $b`
echo the sum is $c
d=`expr $a '*' $b`
echo the product is $d
```