Bash Shell Scripting

Syntax
if

if condition_is_true
    then
    execute commands
else
    execute commands
fi
elif

if condition_is_true
    then
        execute commands
    endif

eelif another_condition_is_true
    then
        execute commands
    else
        execute commands
fi
it -test

- evaluates expression to right
- returns true or false
relational operators

- **eq** Equal to
- **lt** Less than
- **gt** Greater than
- **ge** Greater than or equal to
- **le** Less than or equal to
file related operators

- **-f file** True if file exists and is a regular file
- **-r file** True if file exists and is readable
- **-w file** True if file exists and is writable
- **-x file** True if file exists and is executable
- **-d file** True if file exists and is a directory
- **-s file** True if file exists and has a size greater than zero.
string related tests

-n str True if string str is not a null string
-z str True if string str is a null string
str1 == str2 True if both strings are equal
str1 != str2 True if both strings are unequal
str True if string str is assigned a value and is not null.
compound

-a Performs the **AND** function
-o Performs the **OR** function
test examples

- test $d -eq 0 ; echo $d
- test $s -lt 50; exit
instead of test you can use `[  
if [ $d -eq 0 ]  
  then  
    echo $d  
  fi  
a space must be used after and before `[ and ]
case

case expression in
    pattern1) execute commands ;;
    pattern2) execute commands ;;
    ...
esac

;; terminates a case
case example

echo "Enter your option : "
  read i;

case $i in
  1) ls -l | wc -l ;;
  2) ps -aux ;;
  3) date ;;
  4) who | wc ;;
  5) exit
esac
case example

```
echo "Do you wish to continue? (y/n)"
read ans

case $ans in
  Y|y) ;;
  [Yy][Ee][Ss]) ;;
  N|n) exit ;;
  [Nn][Oo]) exit ;;
  *) echo "Invalid command"
esac
```
compound example

if [ $str1 == $str2 ]
then
  do something
fi

if [ -n "$str1" -a -n "$str2" ]
then
  echo 'Both $str1 and $str2 are not null'
fi