**What is the difference between export, set and setenv UNIX commands?**

**export, set and setenv** commands are used in UNIX for setting value of a variable or an environment variable. In order to understand the difference between the set, setenv and export UNIX commands, the user should know the difference between a normal **variable** and an **environment variable**.

   Let us  consider an example. In k-shell or bourne shell, a variable is defined as shown below:

|  |
| --- |
| # FILE=”output.txt” |

     This means the variable FILE is assigned a value 'output.txt'. This value can be checked by doing "echo $FILE".  This FILE variable is a normal or local variable.  This assignment makes the scope of this variable  only inside the shell in which it is defined.  Any shell or a process invoked from the original shell will not have the variable FILE defined as shown below.

|  |
| --- |
| #FILE=”output.txt”  #echo $FILE  output.txt  #ksh  #echo $FILE  # |

     There are instances or situations where we would like to define a variable and it should be accessed in all the shells or processes invoked by the original shell. This can be achieved by the **export** command in ksh/sh as shown below.

|  |
| --- |
| #export FILE=”output.txt”  #echo $FILE  output.txt  #ksh  #echo $FILE  output.txt  # |

       This FILE variable is now an environment variable. An environment variable is the one which can be accessed across all the shells or processes initiated from the original shell of the environment. So, in ksh/sh, a variable can be made an environment variable using the export command.

   The example below shows the **set** command usage:

root@linux ~# x=5 # here variable is set without export command

|  |
| --- |
|  |