Shell Scripting Interview Questions And Answers:

Q: - What is Shell's Responsibilities?

The shell is responsible for the execution of all programs that you request from your terminal. Each time you type in a line to the shell, the shell analyzes the line and then determines what to do. The line that is typed to the shell is known more formally as the command line. The shell scans this command line and determines the name of the program to be executed and what arguments to pass to the program.

Q: - What is "$#" Variable?

The $# Variable
whenever you execute a shell program, the special shell variable $# gets set to the number of arguments that were typed on the command line.

Q: - Explain "Exit Status" for a shell script?

Whenever any program completes execution under the UNIX system, it returns an exit status back to the system. This status is a number that usually indicates whether the program successfully ran. By convention, an exit status of zero indicates that a program succeeded, and nonzero indicates that it failed. Failures can be caused by invalid arguments passed to the program, or by an error condition detected by the program. For example, the cp command returns a nonzero exit status if the copy fails for some reason (for example, if it can't create the destination file), or if the arguments aren't correctly specified (for example, wrong number of arguments, or more than two arguments and the last one isn't a directory). In the case of grep, an exit status of zero (success) is returned if it finds the specified pattern in at least one of the files; a nonzero value is returned if it can't find the pattern or if an error occurs (the arguments aren't correctly specified, or it can't open one of the files).

Q: - What is "Command Substitution" ?

Command substitution is the process by which the shell runs a command and replaces the command substitution with the output of the executed command. That sounds like a mouthful, but it's pretty straightforward in practice.

Q: - What is " eval" command ?

The eval command exists to supersede the normal command-line substitution and evaluation order, making it possible for a shell script to build up commands dynamically. This is a powerful facility, but it must be used carefully. Because the shell does so many different kinds of substitutions, it pays to understand the order in which the shell evaluates input lines.

Q: - What is awk ?

An awk invocation can define variables, supply the program, and name the input files.

Q: - What is "grep" programe ?

The grep program is the primary tool for extracting interesting lines of text from input datafiles. POSIX mandates a single version with different options to provide the behavior traditionally obtained from the three grep variants: grep, egrep, and fgrep.

Q: - Name a new feature introduced with PHP 5.

PHP 5 introduces (among other things) SQLite support, improved XML support, and a significantly improved object model.

Q: - explain "read" command ?

The read command is one of the most important ways to get information into a shell program:

```
$ x=abc ; printf "x is now '%s'. Enter new value: " $x ; read x

x is now 'abc'. Enter new value: PDQ

$ echo $x

PDQ
```

Q: - What are the two files used by the shell to initialize itself?

/etc/profile
profile

Q: - What is Interactive mode?

Interactive mode means that the shell expects to read input from you and execute the commands that you specify. This mode is called interactive because the shell is interacting with a user. This is usually the mode of the shell that most users are familiar with: you log in, execute some commands, and log out. When you log out using the exit command, the shell exits.

Q: - What is noninteractive mode?

In this mode, the shell does not interact with you; instead it reads commands stored in a file and executes them. When it reaches the end of the file, the shell exits.

Q: - what is local variable?

A local variable is a variable that is present within the current instance of the shell. It is not available to programs that are started by the shell. The variables that you looked at previously have all been local variables.

Q: - What is environment variable?

An environment variable is a variable that is available to any child process of the shell. Some programs need environment variables in order to function correctly. Usually a shell script defines only those environment variables that are needed by the programs that it runs.

Q: - What is shell variable?

A shell variable is a special variable that is set by the shell and is required by the shell in order to function correctly. Some of these variables are environment variables whereas others are local variables.

Q: - Explain the "Exit" command?

Every program whether on UNIX or Linux should end at a certain point of time and successful completion of a program is denoted by the output 0. If the program gives an output other than 0 it defines that there has been some problem with the execution or termination of the problem. Whenever you are calling other function, exit command gets displayed.

Q: - How do you find out what's your shell?

echo $SHELL

Q: - How you will run a process in the background?

./ProcessName &

Q: - How do you write a while loop in shell?

Use While Loop

Q: - How do you read keyboard input in shell scripts?

Use read command

Q: - What is GUI Scripting?

Graphical user interface provided the much needed thrust for controlling a computer and its applications. This form of language simplified repetitive actions. Support for different applications mostly depends upon the operating system. These interact with menus, buttons, etc.

Q: - Explain the term "loops"?

Loops enable you to execute a series of commands multiple times. Two main types of loops are the while and for loops.

Q: - What is "Nested Loops"?

When a loop is located inside the body of another loop it is said to be nested within another loop.

Q: - What is "Infinite Loops"?

Loops that execute forever without terminating.

Q: - What is "File Descriptor"?

An integer that is associated with a file. Enables you to read and write from a file using the integer instead of the file's name.

Q: - Explain "STDIN"?

STDIN Standard Input. User input is read from STDIN. The file descriptor for STDIN is 0.

Q: - Explain "STDOUT"?

STDOUT Standard Output. The output of scripts is usually to STDOUT. The file descriptor for STDOUT is 1.

Q: - Explain "STDERR"?

STDERR Standard Error. A special type of output used for error messages. The file descriptor for STDERR is 2.

Q: - Explain "Escape Sequence"?

An escape sequence is special sequence of characters that represents another character.

Q: - Explain “Output Redirection” in shell scripting?

In UNIX or Linux, the process of capturing the output of a command and storing it in a file is called output redirection because it redirects the output of a command into a file instead of the screen.

Q: Explain “Input Redirection” in shell scripting?

In UNIX or Linux the process of sending input to a command from a file is called input redirection.

Q: - What is “Field separator”?

The field separator controls the manner in which an input line is broken into fields. In the shell, the field separator is stored in the variable IFS. In awk, the field separator is stored in the awk variable FS.

Q: - What is “Library”?

A file that contains only functions is called a library. Usually libraries contain no main code.