

1.What is the purpose of the dollar sign (\$) in shell scripting? (Care este scopul semnului dolar (\$) în shell scripting?)

1. What is the purpose of the dollar sign (\$) in shell scripting? (Care este scopul semnului dolar (\$) în shell scripting?)
 - To create a new variable (Pentru a crea o variabilă nouă)
 - To reference the value of an existing variable (Pentru a face referință la valoarea unei variabile existente)
 - To remove a variable (Pentru a elimina o variabilă)
 - To display a list of variables (Pentru a afișa o listă de variabile)

10. What is the purpose of the env command? (Care este scopul comenzii env?)

What is the purpose of the **env** command? (Care este scopul comenzii **env**?)

- To display a list of environment variables (Pentru a afișa o listă de variabile de mediu)
- To create a new variable (Pentru a crea o variabilă nouă)
- To remove a variable (Pentru a elimina o variabilă)
- To set the value of a variable (Pentru a seta valoarea unei variabile)

2. What command is used to display a list of environment variables in the shell? (Ce comandă este folosită pentru a afișa o listă de variabile de mediu în shell?)

What command is used to display a list of environment variables in the shell? (Ce comandă este folosită pentru a afișa o listă de variabile de mediu în shell?)

- echo
- set
- env
- var

3. Which command is used to remove a variable from the shell environment? (Ce comandă este folosită pentru a elimina o variabilă din mediul shell?)

Which command is used to remove a variable from the shell environment? (Ce comandă este folosită pentru a elimina o variabilă din mediul shell?)

- delete
- remove
- unset
- clear

4. How do you create a new variable in the shell? (Cum creați o variabilă nouă în shell?)

How do you create a new variable in the shell? (Cum creați o variabilă nouă în shell?)

- var MyVariable = "value"
- new MyVariable = "value"
- MyVariable = "value"
- set MyVariable = "value"

5. Which type of quotes allows the shell to parse variables? (Ce tip de ghilimele permite shell-ului să analizeze variabilele?)

Which type of quotes allows the shell to parse variables? (Ce tip de ghilimele permite shell-ului să analizeze variabilele?)

- Single quotes (Ghilimele simple)
- Double quotes (Ghilimele duble)
- Both single and double quotes (Atât ghilimelele simple, cât și cele duble)
- Neither single nor double quotes (Nici ghilimelele simple, nici cele duble)

6. Are shell variables case-sensitive or case-insensitive? (Variabilele shell sunt sensibile la majuscule/minuscule sau nu?)

Are shell variables case-sensitive or case-insensitive? (Variabilele shell sunt sensibile la majuscule/minuscule sau nu?)

- \$USER
- \$PATH
- \$HOME
- \$HOMEDIR

7. Which environment variable holds the user's home directory path? (Care variabilă de mediu conține calea directorului home al utilizatorului?)

Which environment variable holds the user's home directory path? (Care variabilă de mediu conține calea directorului home al utilizatorului?)

- \$USER
- \$PATH
- \$HOME
- \$HOMEDIR

8. Which environment variable holds the user's shell program? (Care variabilă de mediu conține programul shell al utilizatorului?)

Which environment variable holds the user's shell program? (Care variabilă de mediu conține programul shell al utilizatorului?)

- \$USER
- \$PATH
- \$HOME
- \$SHELL

9. How can you display the value of the \$HOSTNAME variable in the shell? (Cum puteți afișa valoarea variabilei \$HOSTNAME în shell?)

How can you display the value of the \$HOSTNAME variable in the shell? (Cum puteți afișa valoarea variabilei \$HOSTNAME în shell?)

- echo \$HOSTNAME
- print \$HOSTNAME

- display \$HOSTNAME
- show \$HOSTNAME

1. What is the purpose of the \$PS1 variable in the shell? (Care este scopul variabilei \$PS1 în shell?)

What is the purpose of the \$PS1 variable in the shell? (Care este scopul variabilei \$PS1 în shell?)

- To determine the shell prompt (Pentru a determina promptul shell-ului)
- To set the path for executable commands (Pentru a seta calea către comenzile executabile)
- To export variables to child shells (Pentru a exporta variabile către shell-urile copil/sub-shell)
- To list all environment variables (Pentru a lista toate variabilele de mediu)

10. Which command can you use to set the \$LANG variable for just one instance of the bash shell? (Ce comandă puteți folosi pentru a seta variabila \$LANG doar pentru o singură instanță a shell-ului bash?)

Which command can you use to set the \$LANG variable for just one instance of the bash shell? (Ce comandă puteți folosi pentru a seta variabila \$LANG doar pentru o singură instanță a shell-ului bash?)

- export LANG=value
- env LANG=value bash
- bash --lang=value
- setenv LANG=value

2. Which special character can be used in \$PS1 to display the current working directory? (Ce caracter special poate fi folosit în \$PS1 pentru a afișa directorul de lucru curent?)

Which special character can be used in \$PS1 to display the current working directory? (Ce caracter special poate fi folosit în \$PS1 pentru a afișa directorul de lucru curent?)

- \u
- \w
- \d
- \p

3. What is the purpose of the \$PATH variable in the shell? (Care este scopul variabilei \$PATH în shell?)

What is the purpose of the \$PATH variable in the shell? (Care este scopul variabilei \$PATH în shell?)

- To determine the shell prompt (Pentru a determina promptul shell-ului)
- To set the path for executable commands (Pentru a seta calea către comenzile executabile)
- To export variables to child shells (Pentru a exporta variabile către shell-urile copil/sub-shell)
- To list all environment variables (Pentru a lista toate variabilele de mediu)

4. Which command can be used to display a list of exported variables in the shell? (Ce comandă poate fi folosită pentru a afișa o listă de variabile exportate în shell?)

Which command can be used to display a list of exported variables in the shell? (Ce comandă poate fi folosită pentru a afișa o listă de variabile exportate în shell?)

- set
- env

- export
- readelf -h

5. How can you start a clean shell without any inherited environment? (Cum puteți porni un shell curat fără nicio moștenire/istoric de mediu?)

How can you start a clean shell without any inherited environment? (Cum puteți porni un shell curat fără nicio moștenire/istoric de mediu?)

- shell --clean
- env -i
- bash --new
- newenv

6. Which command is used to export a variable to child shells? (Ce comandă este folosită pentru a exporta o variabilă către shell-urile copil/sub-shells?)

Which command is used to export a variable to child shells? (Ce comandă este folosită pentru a exporta o variabilă către shell-urile copil/sub-shells?)

- set
- env
- export
- variable

7. How can you delineate a variable name in bash to avoid ambiguity? (Cum puteți delimita un nume de variabilă în bash pentru a evita orice ambiguitate?)

How can you delineate a variable name in bash to avoid ambiguity? (Cum puteți delimita un nume de variabilă în bash pentru a evita orice ambiguitate?)

- Use single quotes (Folosind ghilimele simple)
- Use double quotes (Folosind ghilimele duble)
- Use curly braces (Folosind acolade)
- Use parentheses (Folosind paranteze)

8. What happens by default when you try to display the value of an unbound (non-existent) variable in bash? (Ce se întâmplă în mod implicit când încercați să afișați valoarea unei variabile nelegate /inexistente în bash?)

What happens by default when you try to display the value of an unbound (non-existent) variable in bash? (Ce se întâmplă în mod implicit când încercați să afișați valoarea unei variabile nelegate /inexistente în bash?)

- An error is displayed (Este afișată o eroare)
- Nothing is displayed (Nu este afișat nimic)
- The variable name is displayed (Este afișat numele variabilei)
- A default value is displayed (Este afișată o valoare implicită)

9. Which shell option can be used to generate an error when an unbound variable is referenced? (Ce opțiune de shell poate fi folosită pentru a genera o eroare atunci când este apelată o variabilă nelegată?)

Which shell option can be used to generate an error when an unbound variable is referenced? (Ce opțiune de shell poate fi folosită pentru a genera o eroare atunci când este apelată o variabilă nelegată?)

- nounset
- strict
- error
- novalues
- hexdump -C

How can numerical expressions be evaluated in Bash? [Cum pot fi evaluate expresiile numerice în Bash?]

How can numerical expressions be evaluated in Bash? [Cum pot fi evaluate expresiile numerice în Bash?]

- Using double parentheses [folosind paranteze duble] (())
- Using [folosind comanda] eval
- Using [folosind comanda] let
- Using [folosind comanda] case

How can you loop through a list of files in a directory? (Cum puteți itera printr-o listă de fișiere dintr-un director?)

How can you loop through a list of files in a directory? (Cum puteți itera printr-o listă de fișiere dintr-un director?)

- Using a case statement. (Folosind o instrucțiune case)
- Using a for loop with a wildcard character (*). (Folosind o buclă for cu un caracter wildcard (*))
- Directly calling the file names in commands. (Apelând direct numele fișierelor în comenzi)

In the context of shell scripting, what does a case construct allow you to do? (O construcție de tip case, în contextul shell scripting-ului, ce îți permite să faci?)

In the context of shell scripting, what does a case construct allow you to do? (O construcție de tip case, în contextul shell scripting-ului, ce îți permite să faci?)

- Iterate over a list of items (Să iterezi peste o listă de elemente)
- Simplify decision-making structures (Să simplifici structurile de luare a deciziilor)
- Define a loop (Să definești o buclă)
- Execute a function (Să execuți o funcție)

What does the eval command do in shell scripting? (Ce face comanda eval în scripting-ul shell?)

What does the eval command do in shell scripting? (Ce face comanda eval în scripting-ul shell?)

- Executes a program in a new process (Execută un program într-un nou proces)
- Reads arguments as input to the shell and executes them (Citește argumentele ca intrare în shell și le execută)
- Evaluates mathematical expressions (Evaluează expresii matematice)
- Redirects output to a file (Redirecționează ieșirea către un fișier)

What does the exit command do? (Ce face comanda exit?)

What does the exit command do? (Ce face comanda exit?)

- Jumps to a specific line in the script. (Sare la o linie specifică din script)
- Defines a new function. (Definește o funcție nouă)

- Terminates the script execution. (Termină execuția scriptului)
- Reads input from the user. (Citește input-ul de la utilizator)

What does the let command do? (Ce face comanda let?)

What does the let command do? (Ce face comanda let?)

- Lets you define a new variable (Permite definirea unei noi variabile)
- Performs an arithmetic evaluation (Efectuează o evaluare aritmetică)
- Tests a condition (Testează o condiție)
- Assigns string values (Atribuie valori de tip șir)

What is a key difference between directly assigning a variable and using let? (Care este o diferență cheie între atribuirea directă a unei variabile și utilizarea let?)

What is a key difference between directly assigning a variable and using let? (Care este o diferență cheie între atribuirea directă a unei variabile și utilizarea let?)

- let performs an arithmetic evaluation (let efectuează o evaluare aritmetică)
- Direct assignment can execute commands (Atribuirea directă poate executa comenzi)
- let can assign string values (let poate atribui valori de tip șir)
- There is no difference (Nu există nicio diferență)

What is the difference between \$ and \$\$ in shell scripting?

What is the difference between \$ and \$\$ in shell scripting?

- They both represent the same variable. (Ambele reprezintă aceeași variabilă)
- \$ refers to any environment variable, while \$\$ is for specific ones. (\$ se referă la orice variabilă de mediu, în timp ce \$\$ este pentru unele specifice)
- \$ refers to the current process ID, while \$\$ expands to the current shell process ID. (\$ se referă la ID-ul curent al procesului, în timp ce \$\$ se extinde la ID-ul curent al procesului shell)
- \$ is used for arithmetic expressions, \$\$ for string manipulation. (\$ este folosit pentru expresii aritmetice, \$\$ pentru manipularea șirurilor de caractere)

What is the result of using eval with a command stored in a variable? (Care este rezultatul utilizării eval cu o comandă stocată într-o variabilă?)

What is the result of using eval with a command stored in a variable? (Care este rezultatul utilizării eval cu o comandă stocată într-o variabilă?)

- The command is executed successfully (Comanda este executată cu succes)
- The command is printed to stdout (Comanda este afișată pe stdout)
- A syntax error (O eroare de sintaxă)
- The command is written to a file (Comanda este scrisă într-un fișier)

Which construct can simplify nested if statements? (Ce construcție poate simplifica instrucțiunile if înglobate?)

Which construct can simplify nested if statements? (Ce construcție poate simplifica instrucțiunile if înglobate?)

- for loop (bucloa for)
- case exit from loop (iesirea din bucla cu case)

- while loop (bucla while)
- a nested function (o funcție înglobată)

Can shell functions receive parameters? ()

Can shell functions receive parameters? ()

- True
- False

1. What is the purpose of script parameters [like \$1, \$2 ..] in bash? (Care este scopul parametrilor de script [cum ar fi \$1, \$2 ..] în bash?)

What is the purpose of script parameters [like \$1, \$2 ..] in bash? (Care este scopul parametrilor de script [cum ar fi \$1, \$2 ..] în bash?)

- To pass values to the script from the command line (Pentru a transmite valori scriptului din linia de comandă)
- To store the script name (Pentru a stoca numele scriptului)
- To store the process ID (Pentru a stoca ID-ul procesului)
- To store the last return code (Pentru a stoca ultima valoare returnată)

10. What happens if you pass more parameters to a bash script than there are numbered parameters? (Ce se întâmplă dacă transmiteți mai mulți parametri unui script bash decât sunt parametrii numerotați?)

What happens if you pass more parameters to a bash script than there are numbered parameters? (Ce se întâmplă dacă transmiteți mai mulți parametri unui script bash decât sunt parametrii numerotați?)

- The extra parameters are ignored (Parametrii extra sunt ignorați)
- The script will not run (Scriptul nu va rula)
- The extra parameters are accessible through special parameters (Parametrii extra sunt accesibili prin parametri speciali)
- The extra parameters will cause an error (Parametrii extra vor cauza o eroare)

11. What is the purpose of the shift command in Bash scripts? (Care este scopul comenzii shift în script-urile Bash?)

What is the purpose of the shift command in Bash scripts? (Care este scopul comenzii shift în script-urile Bash?)

- To read input from the user (Pentru a citi intrări de la utilizator)
- To parse script parameters one by one (Pentru a analiza parametrii script-ului unul câte unul)
- To source a configuration file (Pentru a încărca un fișier de configurare)
- To set environment variables (Pentru a seta variabile de mediu)

12. Which command is used to read input from the user in a Bash script? (Ce comandă este folosită pentru a citi intrări de la utilizator într-un script Bash?)

Which command is used to read input from the user in a Bash script? (Ce comandă este folosită pentru a citi intrări de la utilizator într-un script Bash?)

- shift
- read
- source

- input

13. What is the purpose of the source command in Bash? (Care este scopul comenzii source în Bash?)

What is the purpose of the **source** command in Bash? (Care este scopul comenzii source în Bash?)

- To read input from the user (Pentru a citi intrări de la utilizator)
- To parse script parameters (Pentru a analiza parametrii script-ului)
- To load and execute commands from a file (Pentru a încărca și executa comenzi dintr-un fișier)
- To set environment variables (Pentru a seta variabile de mediu)

14. Which command is typically used to load a configuration file in a Bash script? (Ce comandă este folosită în mod tipic pentru a încărca un fișier de configurare într-un script Bash?)

Which command is typically used to load a configuration file in a Bash script? (Ce comandă este folosită în mod tipic pentru a încărca un fișier de configurare într-un script Bash?)

- shift
- read
- source
- config

15. How can you shift all command-line arguments in a Bash script? (Cum puteți deplasa toate argumentele din linia de comandă într-un script Bash?)

How can you shift all command-line arguments in a Bash script? (Cum puteți deplasa toate argumentele din linia de comandă într-un script Bash?)

- shift \$*
- shift \$@
- shift \$#
- shift \$0

2. How are script parameters numbered in bash? (Cum sunt numerotați parametrii de script în bash?)

How are script parameters numbered in bash? (Cum sunt numerotați parametrii de script în bash?)

- They are not numbered (Nu sunt numerotați)
- They start from 0 (Încep de la 0)
- They start from 1 (Încep de la 1)
- They start from \$@ (Încep de la \$@)

3. Which special parameter contains the number of parameters passed to the script? (Care parametru special conține numărul de parametri transmis scriptului?)

Which special parameter contains the number of parameters passed to the script? (Care parametru special conține numărul de parametri transmis scriptului?)

- \$0
- \$1
- \$#

- \$@

4. Which special parameter contains a string of all the parameters passed to the script? (Care parametru special conține un șir al tuturor parametrilor transmiși scriptului?)

Which special parameter contains a string of all the parameters passed to the script? (Care parametru special conține un șir al tuturor parametrilor transmiși scriptului?)

- \$0
- \$1
- \$#
- \$@

5. Which special parameter contains the process ID of the current script? (Care parametru special conține ID-ul procesului scriptului curent?)

Which special parameter contains the process ID of the current script? (Care parametru special conține ID-ul procesului scriptului curent?)

- \$0
- \$1
- \$#
- \$\$

6. Which special parameter contains the last return code? (Care parametru special conține ultima valoare returnată?)

Which special parameter contains the last return code? (Care parametru special conține ultima valoare returnată?)

- \$0
- \$1
- \$#
- \$?

7. What does the \$0 parameter contain? (Ce conține parametrul \$0?)

What does the \$0 parameter contain? (Ce conține parametrul \$0?)

- The number of parameters (Numărul de parametri)
- A string of all the parameters (Un șir al tuturor parametrilor)
- The process ID (ID-ul procesului)
- The name of the script (Numele scriptului)

8. How can you access the third parameter passed to a bash script? (Cum puteți accesa al treilea parametru transmis unui script bash?)

How can you access the third parameter passed to a bash script? (Cum puteți accesa al treilea parametru transmis unui script bash?)

- \$0
- \$1
- \$2

- \$3

9. Which command can you use to see a full list of special parameters in bash? (Ce comandă puteți folosi pentru a vedea o listă completă a parametrilor speciali în bash?)

Which command can you use to see a full list of special parameters in bash? (Ce comandă puteți folosi pentru a vedea o listă completă a parametrilor speciali în bash?)

- help
- info
- man bash
- ls
- param

1. What is the purpose of the getopts command in Bash scripts? (Care este scopul comenzii getopts în script-urile Bash?)

What is the purpose of the getopts command in Bash scripts? (Care este scopul comenzii getopts în script-urile Bash?)

- To get command-line options and arguments (Pentru a obține opțiunile și argumentele din linia de comandă)
- To parse command-line options (Pentru a analiza opțiunile din linia de comandă)
- To read input from the user (Pentru a citi intrări de la utilizator)
- To toggle shell options (Pentru a comuta opțiunile shell-ului)

10. How can you handle options that require an argument in the getopts command? (Cum puteți gestiona opțiunile care necesită un argument în comanda getopts?)

How can you handle options that require an argument in the getopts command? (Cum puteți gestiona opțiunile care necesită un argument în comanda getopts?)

- Use the -a option (Folosiți opțiunea -a)
- Use the -o option (Folosiți opțiunea -o)
- Use the -q option (Folosiți opțiunea -q)
- Use the -l option () Folosiți opțiunea -l)

11. What is the syntax for using getopts to parse options? (Care este sintaxa pentru a folosi getopts pentru a analiza opțiuni?)

What is the syntax for using getopts to parse options? (Care este sintaxa pentru a folosi getopts pentru a analiza opțiuni?)

- getopts options argument \$@
- getopts options argument \$*
- getopts options argument "\$@"
- getopts options "\$@" argument

12. Which variable holds the option character parsed by getopts? (Care variabilă conține caracterul opțiunii analizat de getopts?)

Which variable holds the option character parsed by getopt? (Care variabilă conține caracterul opțiunii analizat de getopt?)

- \$0
- \$1
- \$?
- \$OPTARG

13. How can you make getopt stop parsing options after encountering --? (Cum puteți face ca getopt să oprească analizarea opțiunilor după întâlnirea --?)

How can you make getopt stop parsing options after encountering --? (Cum puteți face ca getopt să oprească analizarea opțiunilor după întâlnirea --?)

- Use the -q option (Folosiți opțiunea -q)
- Use the -- option (Folosiți opțiunea --)
- Use the -e option (Folosiți opțiunea -e)
- Use the -n option (Folosiți opțiunea -n)

14. What is the purpose of the OPTIND variable in getopt? (Care este scopul variabilei OPTIND în getopt?)

What is the purpose of the OPTIND variable in getopt? (Care este scopul variabilei OPTIND în getopt?)

- It holds the index of the next argument to be processed (Conține indexul următorului argument care trebuie procesat)
- It holds the index of the last option parsed (Conține indexul ultimei opțiuni analizate)
- It holds the index of the first option parsed (Conține indexul primei opțiuni analizate)
- It holds the index of the current option being parsed (Conține indexul opțiunii curente care este analizată)

15. How can you specify a default option string for getopt if none is provided? (Cum puteți specifica un șir de opțiuni implicit pentru getopt dacă nu este furnizat niciunul?)

How can you specify a default option string for getopt if none is provided? (Cum puteți specifica un șir de opțiuni implicit pentru getopt dacă nu este furnizat niciunul?)

- Use the -d option followed by the default option string (Folosiți opțiunea -d urmată de șirul de opțiuni implicit)
- Use the -o option followed by the default option string (Folosiți opțiunea -o urmată de șirul de opțiuni implicit)
- Use the -s option followed by the default option string (Folosiți opțiunea -s urmată de șirul de opțiuni implicit)
- Use the -f option followed by the default option string (Folosiți opțiunea -f urmată de șirul de opțiuni implicit)

2. Which command is used to toggle shell options in Bash? (Ce comandă este folosită pentru a comuta opțiunile shell-ului în Bash?)

Which command is used to toggle shell options in Bash? (Ce comandă este folosită pentru a comuta opțiunile shell-ului în Bash?)

- getopt

- read
- shopt
- setopt

3. What is the purpose of the shopt command in Bash? (Care este scopul comenzii shopt în Bash?)

What is the purpose of the shopt command in Bash? (Care este scopul comenzii shopt în Bash?)

- To parse command-line options (Pentru a analiza opțiunile din linia de comandă)
- To read input from the user (Pentru a citi intrări de la utilizator)
- To toggle the values of variables controlling optional shell behavior (Pentru a comuta valorile variabilelor care controlează comportamentul opțional al shell-ului)
- To execute commands from a file (Pentru a executa comenzi dintr-un fișier)

4. Which option in the shopt command allows minor spelling mistakes in the cd command? (Ce opțiune în comanda shopt permite mici greșeli de ortografie în comanda cd?)

Which option in the shopt command allows minor spelling mistakes in the cd command? (Ce opțiune în comanda shopt permite mici greșeli de ortografie în comanda cd?)

- cdspell
- spellcheck
- spellerror
- cdmistake

5. How can you check if a particular shell option is set using the shopt command? (Cum puteți verifica dacă o anumită opțiune de shell este setată folosind comanda shopt?)

How can you check if a particular shell option is set using the shopt command? (Cum puteți verifica dacă o anumită opțiune de shell este setată folosind comanda shopt?)

- shopt OPTION_NAME
- shopt -c OPTION_NAME
- shopt -u OPTION_NAME
- shopt -q OPTION_NAME

6. How can you set a particular shell option using the shopt command? (Cum puteți seta o anumită opțiune de shell folosind comanda shopt?)

How can you set a particular shell option using the shopt command? (Cum puteți seta o anumită opțiune de shell folosind comanda shopt?)

- shopt OPTION_NAME
- shopt -s OPTION_NAME
- shopt -u OPTION_NAME
- shopt -q OPTION_NAME

7. How can you unset a particular shell option using the shopt command? (Cum puteți deseta o anumită opțiune de shell folosind comanda shopt?)

How can you unset a particular shell option using the shopt command? (Cum puteți deseta o anumită opțiune de shell folosind comanda shopt?)

- shopt OPTION_NAME
- shopt -s OPTION_NAME
- shopt -u OPTION_NAME
- shopt -q OPTION_NAME

8. Which command can you use to list all available shell options in Bash? (Ce comandă puteți folosi pentru a lista toate opțiunile de shell disponibile în Bash?)

Which command can you use to list all available shell options in Bash? (Ce comandă puteți folosi pentru a lista toate opțiunile de shell disponibile în Bash?)

- getopts -l
- shopt -l
- options -a
- bash --help

9. What is the purpose of the -o option in the getopts command? (Care este scopul opțiunii -o în comanda getopts?)

What is the purpose of the -o option in the getopts command? (Care este scopul opțiunii -o în comanda getopts?)

- To specify the options to be parsed (Pentru a specifica opțiunile care trebuie analizate)
- To specify the options that require an argument (Pentru a specifica opțiunile care necesită un argument)
- To specify the command-line arguments (Pentru a specifica argumentele din linia de comandă)
- To specify the shell options (Pentru a specifica opțiunile de shell)

1. What is the purpose of the test command? (Care este scopul comenzii test?)

What is the purpose of the **test** command? (Care este scopul comenzii test?)

- To test whether something is true or false (Pentru a testa dacă ceva este adevărat sau fals)
- To create files (Pentru a crea fișiere)
- To delete files (Pentru a șterge fișiere)
- To modify files (Pentru a modifica fișiere)

10. What is the purpose of control structures in bash scripting? (Care este scopul structurilor de control în scripting-ul bash?)

What is the purpose of control structures in bash scripting? (Care este scopul structurilor de control în scripting-ul bash?)

- To execute commands sequentially (Pentru a executa comenzi secvențial)
- To dictate the flow of execution based on conditions (Pentru a decide fluxul de execuție în funcție de anumite condiții)
- To create variables (Pentru a crea variabile)
- To print the output (Pentru a afișa output-ul)

2. What is the return value of test when a condition is true? (Care este valoarea returnată de test când o condiție este adevărată?)

What is the return value of **test** when a condition is true? (Care este valoarea returnată de test când o condiție este adevărată?)

- 0
- 1
- 2
- 3

3. What is the return value of test when a condition is false? (Care este valoarea returnată de test când o condiție este falsă?)

What is the return value of **test** when a condition is false? (Care este valoarea returnată de test când o condiție este falsă?)

- 0
- 1
- 2
- 3

4. Which of the following is an alternative way to write the test command? (Care dintre următoarele este o modalitate alternativă de a scrie comanda test?)

Which of the following is an alternative way to write the test command? (Care dintre următoarele este o modalitate alternativă de a scrie comanda test?)

- []
- ()
- {}
- <>

5. How can you combine multiple conditions in the test command? (Cum puteți combina condiții multiple în comanda test?)

How can you combine multiple conditions in the **test** command? (Cum puteți combina condiții multiple în comanda test?)

- Using && for logical AND (Folosind && pentru AND logic)
- Using || for logical OR (Folosind || pentru OR logic)
- Using ! for logical NOT (Folosind ! pentru NOT logic)
- All from here (Toate cele de aici)

6. What is the purpose of the if statement in shell scripting? (Care este scopul instrucțiunii if în shell scripting?)

What is the purpose of the **if** statement in shell scripting? (Care este scopul instrucțiunii if în shell scripting?)

- To create files (Pentru a crea fișiere)
- To delete files (Pentru a șterge fișiere)
- To make decisions based on conditions (Pentru a lua decizii în funcție de condiții)
- To print the output (Pentru a afișa output-ul)

7. What is the purpose of the else statement in shell scripting? (Care este scopul instrucțiunii else în shell scripting?)

What is the purpose of the **else** statement in shell scripting? (Care este scopul instrucțiunii else în shell scripting?)

- To create files (Pentru a crea fișiere)
- To delete files (Pentru a șterge fișiere)
- To execute a block of code when the if condition is false (Pentru a executa un bloc de cod când condiția if este falsă)
- To print output (Pentru a afișa output-ul)

8. What is the purpose of the elif statement in shell scripting? (Care este scopul instrucțiunii elif în shell scripting?)

What is the purpose of the **elif** statement in shell scripting? (Care este scopul instrucțiunii elif în shell scripting?)

- To create files (Pentru a crea fișiere)
- To delete files (Pentru a șterge fișiere)
- To provide additional conditions to test after the if condition (Pentru a oferi condiții suplimentare de testat după condiția if)
- To print output (Pentru a afișa output-ul)

9. What is the syntax for an if statement in shell scripting? (Care este sintaxa pentru o instrucțiune if în shell scripting?)

What is the syntax for an **if** statement in shell scripting? (Care este sintaxa pentru o instrucțiune if în shell scripting?)

- if condition; then commands; fi
- if (condition) { commands }
- if condition then commands endif
- if condition; then commands; else commands; fi

1. What is the basic syntax of the if-then construct in bash? (Care este sintaxa de bază a construcției if-then în bash?)

What is the basic syntax of the if-then construct in bash? (Care este sintaxa de bază a construcției if-then în bash?)

- if <some-condition> then <some-commands> fi
- if (<some-condition>) { <some-commands> }
- if <some-condition> <some-commands> endif
- if <some-condition> then <some-commands> fi

10. What is the purpose of the -d operator in bash when checking files? (Care este scopul operatorului -d în bash când se verifică fișiere?)

What is the purpose of the **-d** operator in bash when checking files? (Care este scopul operatorului -d în bash când se verifică fișiere?)

- To check if a file exists (Pentru a verifica dacă un fișier există)
- To check if a specified entity is a directory (Pentru a verifica dacă o entitate specificată este un director)
- To check if a file is readable (Pentru a verifica dacă un fișier poate fi citit)
- To check if a file is writable (Pentru a verifica dacă un fișier poate fi editat)

11. How can you check if a file exists in an if statement? (Cum puteți verifica dacă un fișier există printr-o instrucțiune if?)

How can you check if a file exists in an if statement? (Cum puteți verifica dacă un fișier există printr-o instrucțiune if?)

- if [-f file]; then commands; fi
- if file exists; then commands; fi
- if [file]; then commands; fi
- if file; then commands; fi

3. What is the purpose of the if-then-else construct in bash? (Care este scopul construcției if-then-else în bash?)

What is the purpose of the **if-then-else** construct in bash? (Care este scopul construcției if-then-else în bash?)

- To execute commands if a condition is true (Pentru a executa comenzi dacă o condiție este adevărată)
- To execute commands if a condition is false (Pentru a executa comenzi dacă o condiție este falsă)
- To execute one set of commands if a condition is true, and another set if it's false (Pentru a executa un set de comenzi dacă o condiție este adevărată și un alt set dacă este falsă)
- To create variables (Pentru a crea variabile)

4. What is the purpose of the if-then-elif construct in bash? (Care este scopul construcției if-then-elif în bash?)

What is the purpose of the **if-then-elif** construct in bash? (Care este scopul construcției if-then-elif în bash?)

- To execute commands if a condition is true (Pentru a executa comenzi dacă o condiție este adevărată)
- To execute commands if a condition is false (Pentru a executa comenzi dacă o condiție este falsă)
- To handle multiple conditions with different outcomes (Pentru a gestiona multiple condiții cu rezultate diferite)
- To create variables (Pentru a crea variabile)

5. What is the purpose of the -gt operator in bash? (Care este scopul operatorului -gt în bash?)

What is the purpose of the **-gt** operator in bash? (Care este scopul operatorului -gt în bash?)

- To check if a number is greater than another number (Pentru a verifica dacă un număr este mai mare decât un alt număr)
- To check if a string is greater than another string (Pentru a verifica dacă un șir este mai mare decât un alt șir)
- To check if a file exists (Pentru a verifica dacă un fișier există)
- To check if a directory exists (Pentru a verifica dacă un director există)

6. What is the purpose of the = operator in bash when comparing strings? (Care este scopul operatorului = în bash când se compară șiruri?)

What is the purpose of the = operator in bash when comparing strings? (Care este scopul operatorului = în bash când se compară șiruri?)

- To check if two strings are equal (Pentru a verifica dacă două șiruri sunt egale)
- To check if a string is greater than another string (Pentru a verifica dacă un șir este mai mare decât un alt șir)
- To check if a string is less than another string (Pentru a verifica dacă un șir este mai mic decât un alt șir)
- To check if a string has a length greater than zero (Pentru a verifica dacă un șir are o lungime mai mare decât zero)

7. What is the purpose of the -f operator in bash when checking files? (Care este scopul operatorului -f în bash când se verifică fișiere?)

What is the purpose of the -f operator in bash when checking files? (Care este scopul operatorului -f în bash când se verifică fișiere?)

- To check if a file exists (Pentru a verifica dacă un fișier există)
- To check if a specified entity is a regular file (not a directory or device) (Pentru a verifica dacă o entitate specificată este un fișier obișnuit (nu un director sau dispozitiv))
- To check if a file is readable (Pentru a verifica dacă un fișier poate fi citit)
- To check if a file is writable (Pentru a verifica dacă un fișier este editabil)

8. What is the purpose of the -r operator in bash when checking files? (Care este scopul operatorului -r în bash când se verifică fișiere?)

What is the purpose of the -r operator in bash when checking files? (Care este scopul operatorului -r în bash când se verifică fișiere?)

- To check if a file exists (Pentru a verifica dacă un fișier există)
- To check if a file is a directory (Pentru a verifica dacă un fișier este un director)
- To check if a file exists and is readable (Pentru a verifica dacă un fișier există și poate fi citit)
- To check if a file is writable (Pentru a verifica dacă un fișier poate fi modificat)

9. What is the purpose of the -nt operator in bash when comparing files? (Care este scopul operatorului -nt în bash când se compară fișiere?)

What is the purpose of the -nt operator in bash when comparing files? (Care este scopul operatorului -nt în bash când se compară fișiere?)

- To check if a file is newer than another file (Pentru a verifica dacă un fișier este mai nou decât un alt fișier)
- To check if a file is older than another file (Pentru a verifica dacă un fișier este mai vechi decât un alt fișier)
- To check if a file exists (Pentru a verifica dacă un fișier există)
- To check if a file is readable (Pentru a verifica dacă un fișier poate fi citit)

1. What is the purpose of a for loop in shell scripting? (Care este scopul unei bucle for în shell scripting?)

What is the purpose of a **for** loop in shell scripting? (Care este scopul unei bucle for în shell scripting?)

- To perform a set of commands for each item in a list (Pentru a executa un set de comenzi pentru fiecare element dintr-o listă)
- To create files (Pentru a crea fișiere)
- To delete files (Pentru a șterge fișiere)
- To modify files (Pentru a modifica fișiere)

10. What is the purpose of the continue statement in a loop? (Care este scopul instrucțiunii continue într-o buclă?)

What is the purpose of the **continue** statement in a loop? (Care este scopul instrucțiunii **continue** într-o buclă?)

- To create a new loop (Pentru a crea o nouă buclă)
- To exit the current loop (Pentru a ieși din bucla curentă)

- To skip the remaining commands in the current iteration and move to the next iteration (Pentru a sări peste comenzile rămase în iterația curentă și a trece la următoarea iterație)
- To execute a set of commands (Pentru a executa un set de comenzi)

2. What is the syntax for a for loop in Bash? (Care este sintaxa pentru o buclă for în Bash?)

What is the syntax for a **for** loop in Bash? (Care este sintaxa pentru o buclă for în Bash?)

- for item in list; do commands; done
- for (item in list) { commands }
- for item from list; commands; next
- foreach item in list; commands; endfor

3. How can you use a for loop to iterate over a range of numbers in Bash? (Cum puteți folosi o buclă for pentru a itera peste un interval de numere în Bash?)

How can you use a **for** loop to iterate over a range of numbers in Bash? (Cum puteți folosi o buclă for pentru a itera peste un interval de numere în Bash?)

- for i in {1..10}; do commands; done
- for i = 1 to 10; do commands; done
- for i from 1 to 10; do commands; done
- for i in range(1, 10); do commands; done

4. What is the purpose of a while loop in shell scripting? (Care este scopul unei bucle while în shell scripting?)

What is the purpose of a **while** loop in shell scripting? (Care este scopul unei bucle while în shell scripting?)

- To create files (Pentru a crea fișiere)
- To delete files (Pentru a șterge fișiere)
- To repeatedly execute a set of commands as long as a condition is true (Pentru a executa în mod repetat un set de comenzi atât timp cât o condiție este adevărată)
- To modify files (Pentru a modifica fișiere)

5. What is the syntax for a while loop in Bash? (Care este sintaxa pentru o buclă while în Bash?)

What is the syntax for a **while** loop in Bash? (Care este sintaxa pentru o buclă while în Bash?)

- while condition; do commands; done
- while (condition) { commands }
- while condition; commands; endwhile
- while condition then commands endif

6. What is the purpose of an until loop in shell scripting? (Care este scopul unei bucle until în shell scripting?)

What is the purpose of an **until** loop in shell scripting? (Care este scopul unei bucle **until** în shell scripting?)

- To create files (Pentru a crea fișiere)
- To delete files (Pentru a șterge fișiere)
- To repeatedly execute a set of commands until a condition is true (Pentru a executa în mod repetat un set de comenzi până când o condiție este adevărată)

- To modify files (Pentru a modifica fişiere)

7. What is the syntax for an until loop in Bash? (Care este sintaxa pentru o buclă until în Bash?)

What is the syntax for an **until** loop in Bash? (Care este sintaxa pentru o buclă until în Bash?)

- until condition; do commands; done
- until (condition) { commands }
- until condition; commands; enduntil
- until condition then commands endif

8. How can you create an infinite loop in Bash? (Cum puteţi crea o buclă infinită în Bash?)

How can you create an infinite loop in Bash? (Cum puteţi crea o buclă infinită în Bash?)

- for (;;) { commands }
- while true; do commands; done
- until false; do commands; done

9. What is the purpose of the break statement in a loop? (Care este scopul instrucţiunii break într-o buclă?)

What is the purpose of the **break** statement in a loop? (Care este scopul instrucţiunii **break** într-o buclă?)

- To create a new loop (Pentru a crea o nouă buclă)
- To exit the current loop (Pentru a ieşi din bucla curentă)
- To skip to the next iteration of the loop (Pentru a sări la următoarea iteraţie a buclei)
- To execute a set of commands (Pentru a executa un set de comenzi)

1. What is the outcome of setting IFS=\$'\n' before iterating over file lines? (Care este rezultatul setării IFS=\$'\n' înainte de iterarea peste liniile fişierului?)

What is the outcome of setting IFS=\$'\n' before iterating over file lines? (Care este rezultatul setării IFS=\$'\n' înainte de iterarea peste liniile fişierului?)

- Ensures that each line from the file is treated as a single item in the loop (Asigură că fiecare linie din fişier este tratată ca un singur element în buclă)
- Causes the loop to skip empty lines (Face ca bucla să sară peste liniile goale)
- Splits lines into words based on whitespace (Împarte liniile în cuvinte, pe baza spaţiilor goale)
- Has no effect on how lines are processed in the loop (Nu are niciun efect asupra modului în care liniile sunt procesate în buclă)

2. How can you decrement a variable in a while loop? (Cum poţi decrementa o variabilă într-o buclă while?)

How can you decrement a variable in a while loop? (Cum poţi decrementa o variabilă într-o buclă while?)

- var1=\$((var1 + 1))
- var1=\$((var1 - 1))
- var1--
- var1-=1

3. Which of the following is an example of iterating over complex values? (Care dintre următoarele este un exemplu de iterare peste valori complexe?)

Which of the following is an example of iterating over complex values? (Care dintre următoarele este un exemplu de iterare peste valori complexe?)

- for var in 1 2 3
- for var in first "the second" "the third"
- for var in \$(1..3)
- for var in {1..5}

4. What effect does placing one loop inside another have on the inner loop? (Ce efect are, asupra buclei interioare, plasarea unei bucle în interiorul alteia?)

What effect does placing one loop inside another have on the inner loop? (Ce efect are, asupra buclei interioare, plasarea unei bucle în interiorul alteia?)

- The inner loop is executed in its entirety for each iteration of the outer loop (Bucula interioară este parcursă în întregime la fiecare iterație a buclei exterioare)
- The inner loop executes only once, regardless of the outer loop (Bucula interioară se execută doar o dată, indiferent de bucla exterioară)
- The inner loop's execution speed is increased (Viteza de execuție a buclei interioare este sensibil mărită)
- The outer loop variables are inaccessible in the inner loop (Variabilele buclei exterioare sunt inaccesibile în bucla interioară)

5. What is the best practice for restoring IFS to its default value after modifying it for a loop? (Care este cea mai bună metodă pentru restaurarea IFS la valoarea sa implicită după modificarea acestuia pentru o buclă?)

What is the best practice for restoring IFS to its default value after modifying it for a loop? (Care este cea mai bună metodă pentru restaurarea IFS la valoarea sa implicită după modificarea acestuia pentru o buclă?)

- Save the original IFS value before changing it, and restore it after the loop (Salvarea valorii originale a IFS înainte de a o schimba și, mai apoi, restaurarea acesteia după buclă)
- Leave IFS modified as it will not affect other parts of the script (Lăsarea IFS modificat, așa cum este, deoarece nu va afecta alte părți ale scriptului)
- Reset IFS by using **unset IFS** (Resetarea IFS folosind **unset IFS**)
- Set IFS to a space character (Setarea IFS corespunzător unui caracter spațiu-gol)

6. What happens if you do not quote complex values in a for loop iteration list? (Ce se întâmplă dacă nu punem între ghilimele valorile complexe în lista de iterare a unei bucle for?)

What happens if you do not quote complex values in a for loop iteration list? (Ce se întâmplă dacă nu punem între ghilimele valorile complexe în lista de iterare a unei bucle for?)

- The script will fail to execute (Scriptul nu va reuși să se execute)
- Each word in the complex value is treated as separate items (Fiecare cuvânt din valoarea complexă este tratat ca elemente separate)
- The loop will only iterate once (Bucula for va itera doar o singură dată)

7. How can you initialize a for loop with the result of a command in bash scripting? (Cum poți inițializa o buclă for cu rezultatul unei comenzi în scripting bash?)

How can you initialize a for loop with the result of a command in bash scripting? (Cum poți inițializa o buclă for cu rezultatul unei comenzi în scripting bash?)

- Using the eval command (Folosind comanda eval)
- Using command substitution (Folosind substituția comezii respective)
- With direct assignment (Prin atribuire directă)
- By writing the command inside the loop (Scriind comanda în interiorul buclei)

8. How do you ensure a bash script exits if any command within fails? (Cum va asigurați că un script bash iese din execuție chiar dacă orice comandă din interior eșuează?)

How do you ensure a bash script exits if any command within fails? (Cum va asigurați că un script bash iese din execuție chiar dacă orice comandă din interior eșuează?)

- By using the **set -e** command at the beginning (Folosind comanda **set -e** la început)
- By using the **set -B** command at the end (Folosind comanda **set -B** la sfarsitul scriptului)
- By manually checking the exit status of each command (Verificând manual starea de ieșire a fiecărei comenzi)
- By wrapping each command with an if statement (Încadrând fiecare comandă într-o instrucțiune if)