Man saprotami funkciju skaidrojumi. Agnese Gūtmane.

1. VirtualBox - - startvm XP - virtuālais draiveris, lai palaistu operētājsistēmas.
2. ls -l - funkcija, kura parāda failus no iekārtas gara saraksta formā.
3. cd -ar šo funkciju var atgriezties sākuma mapē. ( piem., cd DMI). Var mainīt sakumpunktu. Mainīt “mājas”.
4. NAME = “Name” – ar šo funkciju var definēt lietotāja vārdu vai jebkuru vārdu, tas, kas rakstīts pēdiņās”” , tas parādīsies terminālī, kad tiks pieprasīts.
5. echo $NAME- ar šo funkciju varam pieprasīt uzrādīt lietotāja vārdu vai jebkuru vārdu, kurš tika definēts kā NAME..
6. b = x - ar šo funkciju b var raksturot dažādus lielumus, tas atkarīgs no tā, kā nosauc b. ( piem. b=50)
7. echo $b – ar šo funkciju terminālī var pieprasīt uzrādīt , kas ir b.
8. readonly NAME – ar šo funkciju terminālī parādās tikai vārds – NAME- un nevis NAME =... . Nerādās funkcija, tikai ievietotais vārds.
9. name=”Vards Uzvards” – lietotāju var definēt arī šādi.
10. mainiigais\_1=10 – šādi var definēt mainīgos.
11. echo $mainiigais\_1 – ar šo funkciju var pieprasīt uzrādīt no apakštermināla, kurš ir pakārtots galvenajam, pirmajam terminālim, parādīt informāciju, kas ir definēts kā pirmais mainīgais.
12. 1\_mainiigais=20 - tiek parādīts, kas ir pirmais mainīgais.
13. nano shell\_examples.sh – ar šo funkciju var atvērt un ieiet otrā terminālī, kurš ir pakārtots pirmajam terminālim.
14. ls -la – funkcija, kura parāda visus failus no iekārtas gara saraksta formā, arī slēptos ( tos failus, kurus neuzrāda ls -l). Tas ir, paplašinātu sarakstu, pilnīgāku.
15. cat shell\_examples.sh – ar šo funkciju var parādīt ko satur fails/faili.
16. shell\_examples.sh – ar šādu nosaukumu mēs taisījām mapi.
17. PATH=$PATH – strājot terminālī var pieprasīt failu atrašanās vietas, un ar PATH to var darīt.
18. echo $PATH – funkcija, ar kuru var pieprasīt, lai parāda pieprasītā faila atrašanās vietu
19. chmod 744 shell\_examples.sh – ar šo var uzstādīt kādas ir tiesības pret failu vai dokumentu.
20. mv shell\_examples.sh – tiek izmantots , lai pārsauktu vai pārvietotus failus.
21. history > history\_xxxxxxxx.txt – šādi tiek saglabāta vēsture, sākot ar gadu, tad mēnesis , tad datums.
22. nano git-upload – ar šo var izveidot failus vai ierakstus direktorijā.
23. git-upload – var izveidot šādu mapi.
24. chmod 744 git-upload – ar šo var uzstādīt kādas ir tiesības pret failu vai dokumentu.
25. rm git-upload – ar šo funkciju var izdzēst failus no git-upload mapes.
26. skaitliska\_vertiba=`expr x + x` - funkcija, kas parāda skaitlisku izteiksmi , šo funkciju definē apakšterminālī, kas ir pakārtots pirmajam terminālim un rezultātu var pieprasīt pirmajā terminālī.
27. echo ‘’Summas vertiba : ’’$skaitliska\_vertiba – visu nosaka pēdiņas atrašanās vieta – vai tiek rakstīta summas darbība vai rezultāts. – tiek rakstīta darbība.
28. echo “Summas vertiba : $skaitliska\_vertiba” - visu nosaka pēdiņas atrašanās vieta – vai tiek rakstīta summas darbība vai rezultāts. – tiek rakstīts rezultāts.
29. git add . - ar šo funkciju var pievienot failus repositorijā un definēt ar laiku.
30. git commit -m “xxxxxxxx” – definē datumu , kad tiek veikts ierakts github
31. git push origin master – ievadu sevi kā lietotāju.
32. ls -a -visu failu sarkats, iekļaujot paslēptos failus, kas sākas ar “.”
33. pwd- komanda ar kuru var tikt līdz vietai, kur strādā/ atrodas pašreizējā brīdī.
34. cd . . – ar šo funkciju var tikt līdz iepriekšējai atrašanās vietai.
35. $! – kā fons nākamajām darbībām ar $.
36. $$ -piešķirtais numurs skripta izpildei
37. $? – iepriekšējās komaandas izpildes rezultāts
38. $@ - argumenta sarakts. Ar šo funkciju var to attēlot . tas ir attēlošanas un grupēšanas otrs veids.
39. $\* - argumenta sarakts. Ar šo funkciju var to attēlot . tas ir attēlošanas un grupēšanas veids.
40. $# -skriptam nodoto argumentu skaits,
41. $n - tas nozīmē jaunā rindā.
42. $0 – izpilda skripta faila nosaukumu, tas nozīmē, ka caur echo $0, es terminālī ieraudzīšu skripta faila nosaukumu.
43. $1 – pirmā argumenta vērtība
44. $2 – otrā argumenta vērtība