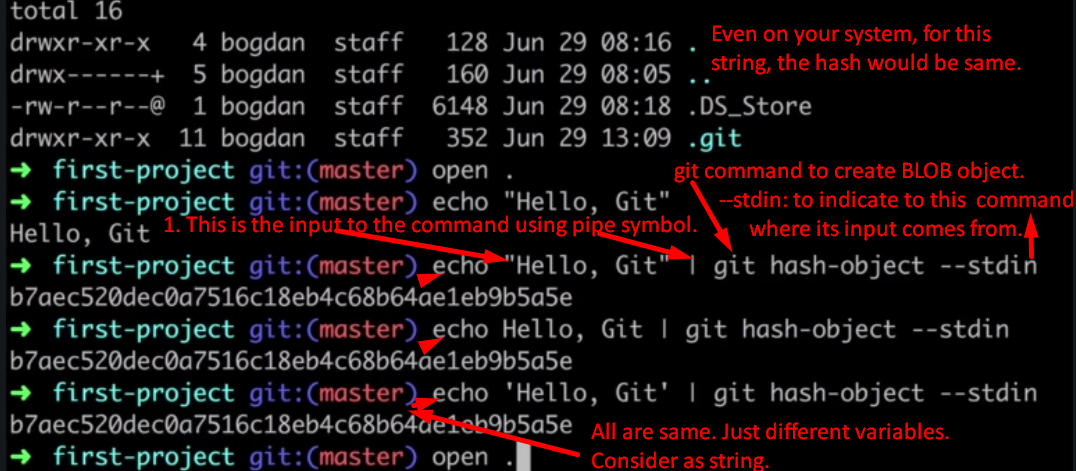
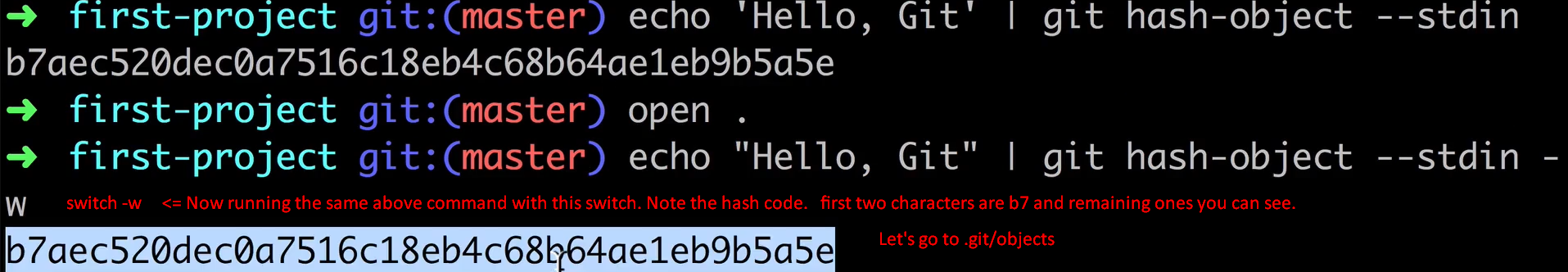
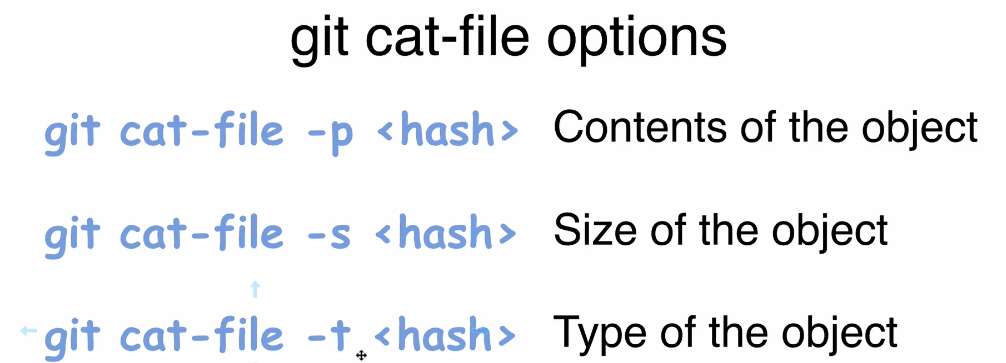
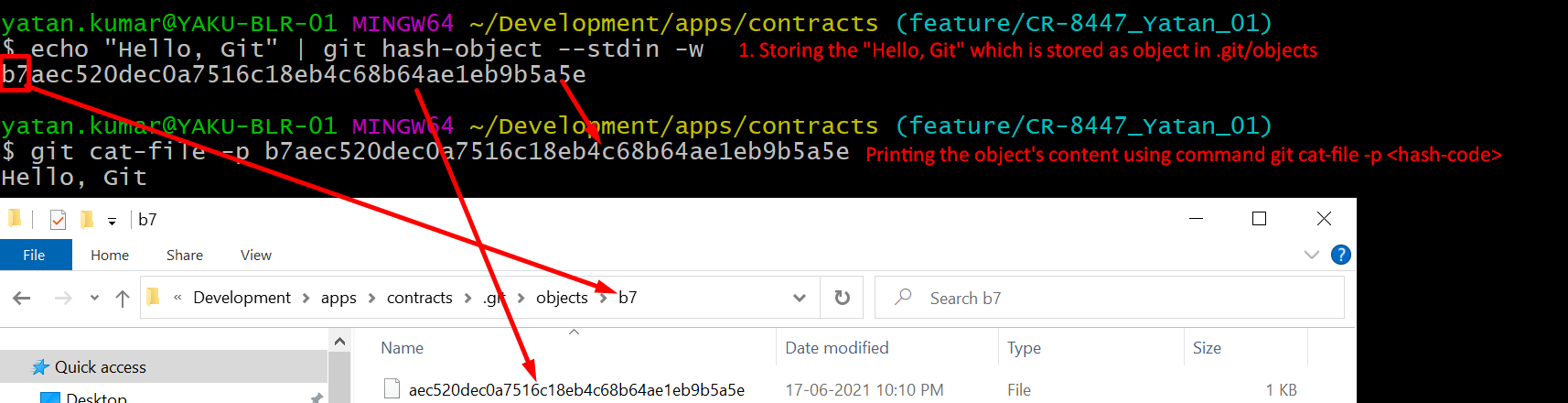
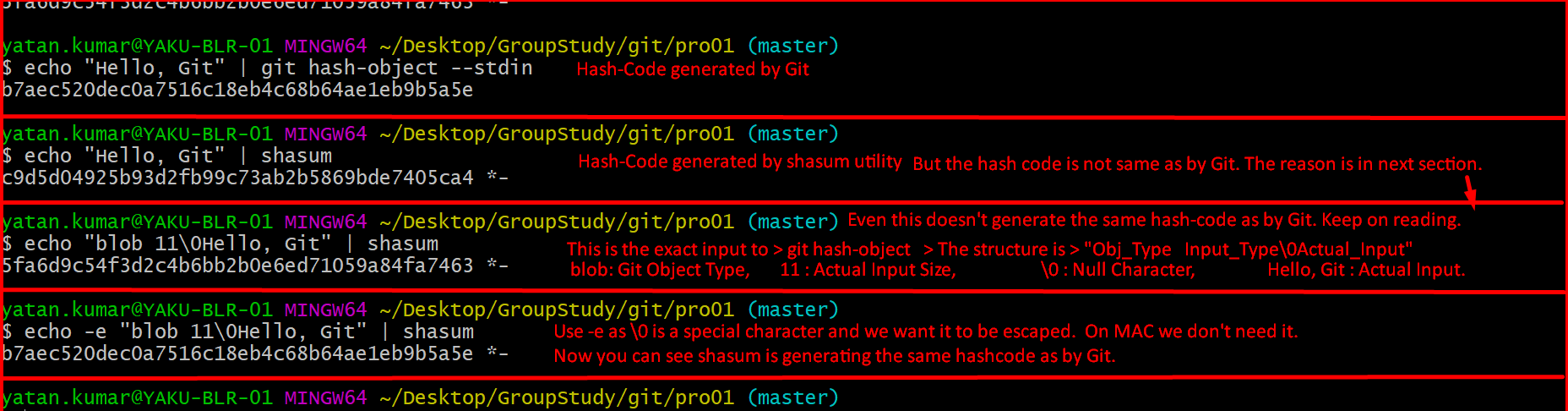
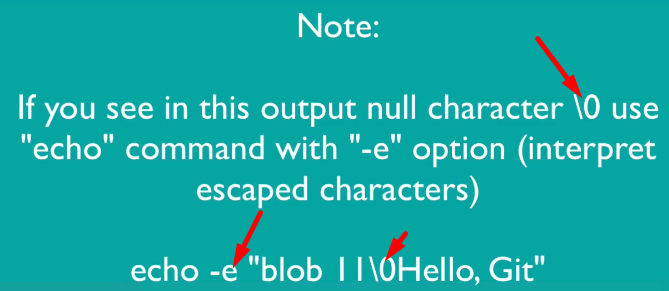
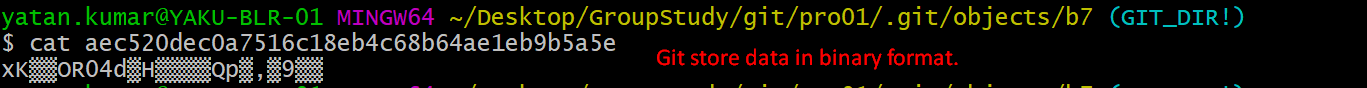
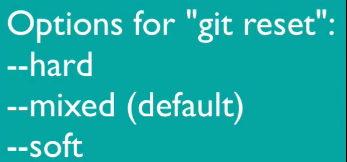
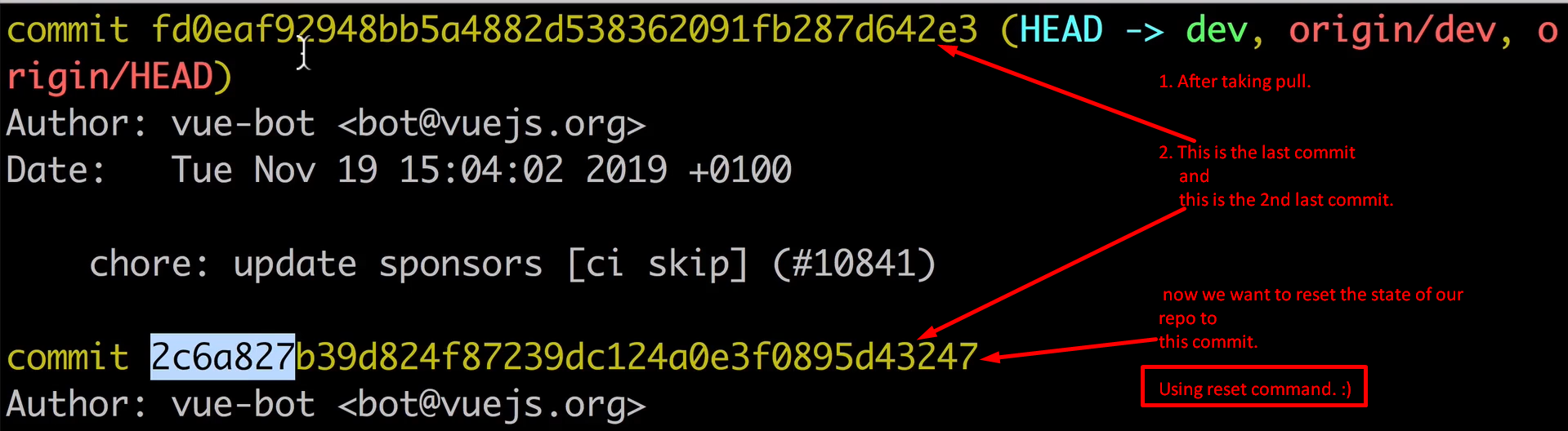
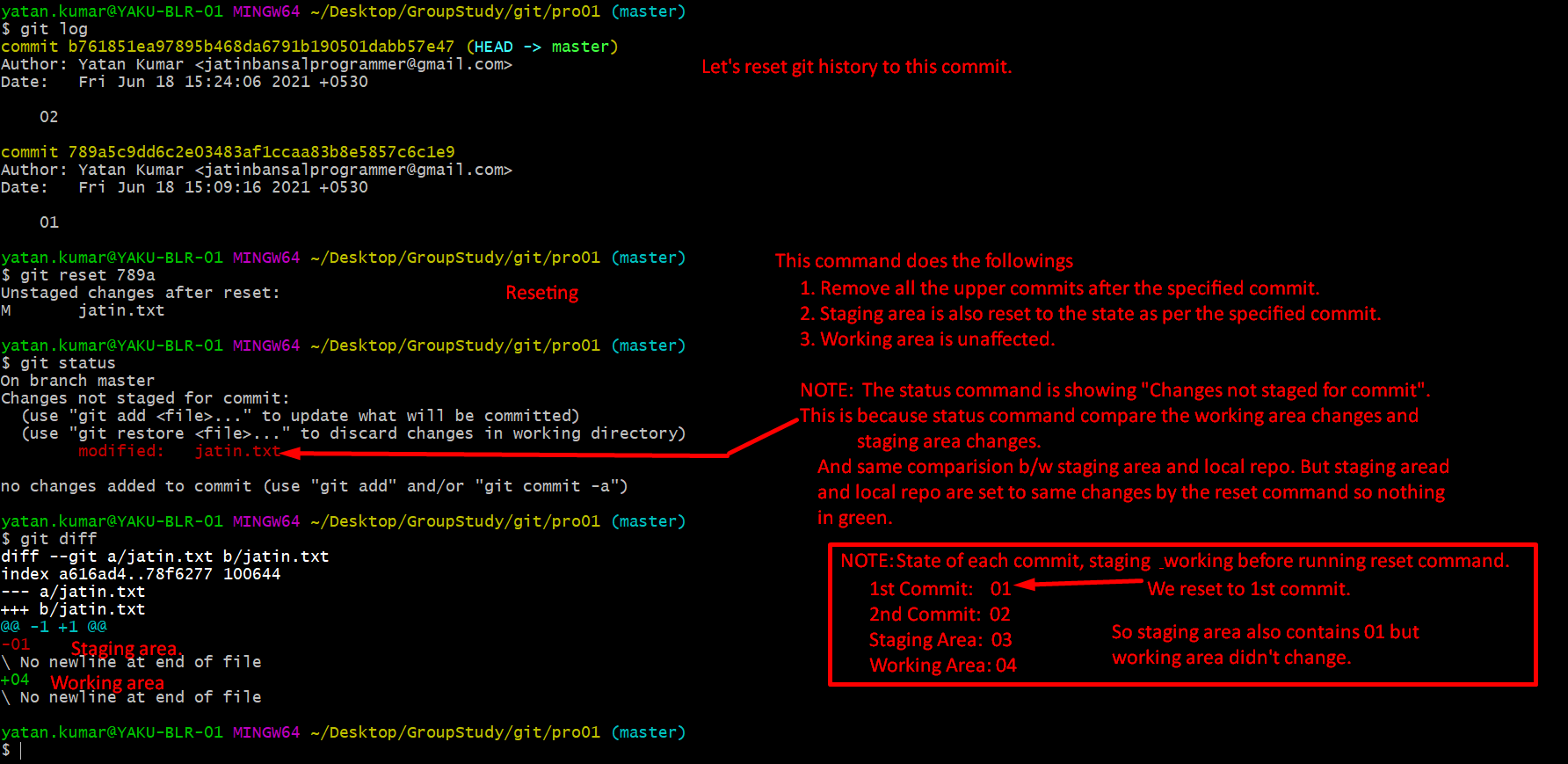
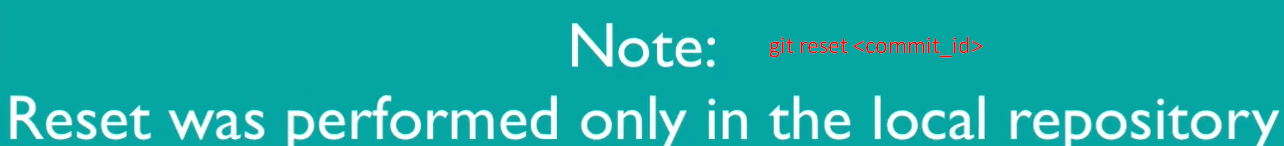
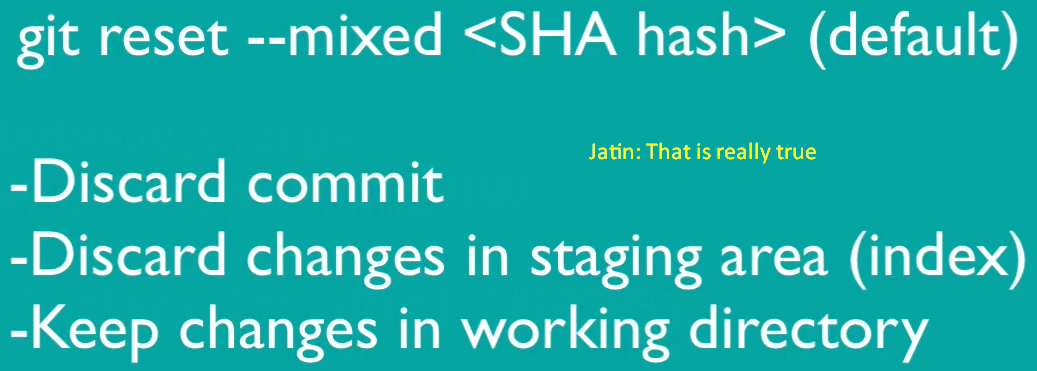
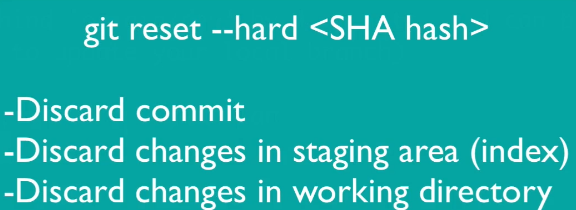
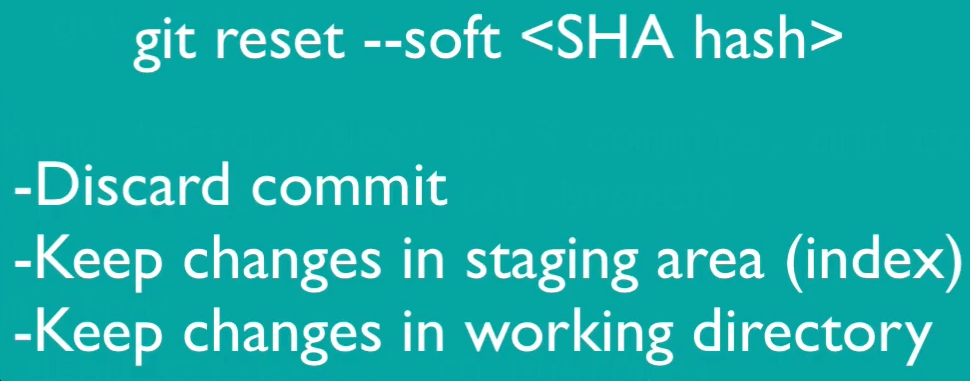
1. **git init**:
   1. Creates .git (hidden folder) which is git repo.
2. **mkdir <dir\_name>**
3. **ls -la**
   1. **a:** To display hidden folders.
4. **start . or explorer .**
5. **Low Level Commands**:
   1. **git hash-object**: To create git object such as BLOG
   2. **git cat-file**: To read git object.
   3. **git mktree**: To create git tree object.
6. blob, tree, commit, Annotated Tag, hash
7.   
   **Mistake**: git cat-file is used to read the content of a git object (blog, tree, commit, annotated tag).
8. 
   1. echo “Hello, Git” | git **hash-object** --stdin -w
9. 
10. key:value 🡺 hashcode:file’s content 🡺 hashcode based on file’s content so file’s content can’t be same.   
    Jatin🡺If they’re same, the same object would be referred.
11. Variable Length input 🡪 Hash Function 🡪 Hash Code  
    MD5, SHA1, SHA256, SHA384, SHA512
12. **shasum**:
    1. Utility to create SHA1 code for text or file.
    2. Example:
    3. echo -n “text”: -n will not add line break at the end of text when printing.
       1. **Output**: text% 🡸 % is indication that no line break. 🡸 On windows, %is not displayed but on MAC.
    4. echo “text!” : Exclamation will not work on MAC but widows.
13. 
14. 
15. 
16. 
17. 
18. 
19. 
20. 
21. 
22. 
23. **Git reset**: Everything about reset commands.
24. 
25. 
26. 
27. 
28. 
29. 
30. 
31. 
32. **Missing Section 4**