

Git command Notes

Git command	Explanation
<pre>\$ git -version git version 2.37.0.windows.1</pre>	For Checking the git version
<pre>\$ git config --global user.name "Aniket Nigam"</pre>	For setting your name on the git
<pre>\$ git config user.name Aniket Nigam</pre>	For checking your name on git
<pre>\$ git config --global user.email "aniknigam2@gmail.com"</pre>	For setting your email
<pre>\$ git config --global user.email aniknigam2@gmail.com</pre>	For checking email on git
<pre>\$ code .</pre>	For opening vs code
<pre>\$ git init Initialized empty Git repository in C:/Users/Aniket Nigam/Documents/30 Days of Javascript/.git/</pre>	For initializing the repository It is usually the first command you write when you are working in the new project.
<pre>\$ ls -lart total 20 drwxr-xr-x 1 Aniket Nigam 197121 0 Feb 26 00:34 ../ drwxr-xr-x 1 Aniket Nigam 197121 0 Feb 26 00:54 ../ drwxr-xr-x 1 Aniket Nigam 197121 0 Feb 26 00:54 .git/</pre>	It shows the hidden files.
<pre>\$ git status On branch master No commits yet Untracked files: (use "git add <file>..." to include in what will be committed)</pre>	It tells the status of the file and the status is showing that this file is untracked that means it need to be staged.

<code>day1_comment_30_days_of_js.html</code> nothing added to commit but untracked files present (use "git add" to track)	
<pre>\$ git add day1_comment_30_days_of_js.html</pre>	For transferring your file in staging area.
<pre>\$ git status</pre> <p>On branch master</p> <p>No commits yet</p> <p>Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: day1_comment_30_days_of_js.html</p>	Here this command is showing that the file is in staging area and it need to be committed.
<pre>\$ git commit</pre>	This is the initial commit that you make and here the text editor will open and you can only write here after typing "I"and then you will write the message initial commit.
<pre>[esp button]+:wq+[enter button]</pre>	For exiting from the text editor
<pre>\$ git status</pre> <p>On branch master</p> <p>nothing to commit, working tree clean</p>	This shows that your changes has been committed.
<pre>\$ touch day2_variable_30_days_of_js.html</pre>	For creating new file through git
<pre>\$ git add -A</pre>	For transferring multiple files to the staging area at once.
<pre>\$ git commit -m "added variable file in the folder" [master 33dffb1f] added variable file in the folder 1 file changed, 0 insertions(+), 0 deletions(-) create mode 100644 day2_variable_30_days_of_js.html</pre>	Easy way to commitHere the text editor will not open...and it will be easy to use it.
<pre>clear</pre>	For clearing the terminal

<pre>\$ git checkout day1_comment_30_days_of_js.html Updated 1 path from the index</pre>	<p>This command is used when you want to recover your code. This command helps you to recover your code till the last commit.</p>
<pre>\$ git checkout -f</pre>	<p>This command is used to recover all the files till the last commit.</p>
<pre>\$ git log commit 33dff1f47e08af0e0f27e9d573b000619cc6bf26 (HEAD -> master) Author: Aniket Nigam <aniknigam2@gmail.com> Date: Sun Feb 26 01:31:32 2023 +0530 added variable file in the folder commit 1c619084a26a3ef100817d8a5a125fe158c042b4 Author: Aniket Nigam <aniknigam2@gmail.com> Date: Sun Feb 26 01:15:21 2023 +0530 Initial commit</pre>	<p>This command tell you that how many commit you have made.</p>
<pre>\$ git log -p -1 commit 33dff1f47e08af0e0f27e9d573b000619cc6bf26 (HEAD -> master) Author: Aniket Nigam <aniknigam2@gmail.com> Date: Sun Feb 26 01:31:32 2023 +0530 added variable file in the folder diff --git a/day2_variable_30_days_of_js.html b/day2_variable_30_days_of_js.html new file mode 100644 index 0000000..e69de29</pre>	<p>For example you have 100 commit and you want to see the 1 commit and all the details of the commit then you use this command</p> <p>For example you wanted to see the last 5 commits the you will write git log -p -5.</p>
<pre>q</pre>	<p>It is use to quit from the git log</p>
<pre>\$ git diff diff --git a/day1_comment_30_days_of_js.html b/day1_comment_30_days_of_js.html index 4d6b666..b4d37b1 100644 --- a/day1_comment_30_days_of_js.html +++ b/day1_comment_30_days_of_js.html @@ -9,6 +9,6 @@ <body> <h1>Comments in Javascript</h1></pre>	<p>This command is used to compare your code to the staging area... For ex you want to compare your code to the last staging</p>

<pre> <p>hello this is the firest day of usig the js</p> - iodsfsjdsfj +<p>making changes in the code</p> </body> </html> \ No newline at end of file </pre>	area then you use this command.
<pre> \$ git diff -staged diff --git a/day1_comment_30_days_of_js.html b/day1_comment_30_days_of_js.html index 3b7fb2c..4d6b666 100644 --- a/day1_comment_30_days_of_js.html +++ b/day1_comment_30_days_of_js.html @@ -8,5 +8,7 @@ </head> <body> <h1>Comments in Javascript</h1> + <p>hello this is the firest day of usig the js</p> + iodsfsjdsfj </body> </html> \ No newline at end of file </pre>	This command is used for comparing your staging area to the last commit.
<pre> \$ git commit -a -m "I have skipped staging area and directly committed" On branch master nothing to commit, working tree clean </pre>	This is used for directly committing your code and skipping staging area.
<pre> \$ ls day1_comment_30_days_of_js.html day2_variable_30_days_of_js.html </pre>	For listing all your files.
<pre> \$ git rm --cached waste.html rm 'waste.html' </pre>	For removing your file from staging area but the file will still be there in the hardisk
<pre> \$ git rm waste.html </pre>	For deleting the file.
<pre> \$ git status -s M day1_comment_30_days_of_js.html M day2_variable_30_days_of_js.html </pre>	It tells the status more clearly The green one M is in staging area and the red M is in working mode
<pre> \$ git branch * master </pre>	It tells how many branches you have.
<pre> \$ git branch branch2 Git </pre>	It is use to create more branches

<pre>\$ git branch branch2 * master</pre>	Master is the branch we are working in.
<pre>\$ git checkout branch2 Switched to branch 'branch2' M day1_comment_30_days_of_js.html M day2_variable_30_days_of_js.html</pre>	For shifting your branch...here we have shifted to branch2
<pre>\$ git merge branch2</pre>	This is used for merging 2 branches together.
<pre>\$ git push -u origin master</pre>	For pushing your code to your github account
<pre>\$ git remote add origin git@github.com:aniknigam/30_days_of_javascript_by_Aniket_Nigam.git</pre>	For connecting our local files to the github repository
<pre>\$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY</pre>	For cloning the repo from github to your desktop