# Testing Git Branch Command Line Commands

## 0 - Overview

* Testing Git Branch Command Line Commands by using Git Bash and using the “Learning-Github” repository.
* <https://github.com/theCivilCoder/Learning-Github>

## 1 - Setup

* Clone a repository
* Text

  Description automatically generated
* Cd into the git repository on the local machine
  + Text

    Description automatically generated
* Check current branches available in repository
* Text

  Description automatically generated

## 2 – Create New Branch

* Text

  Description automatically generatedgg
* Text

  Description automatically generated

### Switch to New Branch

* Text

  Description automatically generated

### Renaming the Current branch

* A screenshot of a computer

  Description automatically generated with medium confidence
* Look at the current branches on this local repo
* Text

  Description automatically generated

### Renaming other branches

* Text

  Description automatically generated
  + I renamed “new-branch1-RENAMED” (this was a mistake, this was supposed to be a new branch3)

## 3 – Publish to Remote Repo

* Currently on github; current branches
  + Graphical user interface, text, application, email

    Description automatically generated
* I tried pushing my new branch to origin and this is what I got
* Text

  Description automatically generated
  + Text

    Description automatically generated
  + Git push –set-upstream origin new-branch1a-RENAMED
  + New branch has no upstream branch so we have to use this command to set origin as the upstream branch within the push command
* Text

  Description automatically generated
* Graphical user interface, text, application, email

  Description automatically generated

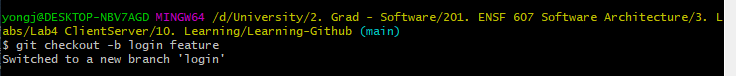
## 4 – Delete a local branch

* Text

  Description automatically generated

## 4a - Various

### Create a branch based on another branch

* 
  + Create the local branch “login” based on the current branch “feature”

### Doing some work then publishing branch

* While under login branch, I created a new py file called “test\_child2.py”
* A screenshot of a computer

  Description automatically generated with medium confidence
* Text

  Description automatically generated
* Text

  Description automatically generated
* Now back on github
* Clicking on the login branch
* Graphical user interface, text, application, email

  Description automatically generated
* As opposed to the feature branch, it does not have the new third py file
* Graphical user interface, text, application

  Description automatically generated
* Notice that login is its own branch even though my earlier
* Graphical user interface, application, Teams

  Description automatically generated
* I created another branch based off login
* Then I pushed the branch to origin
* Text

  Description automatically generated Text

  Description automatically generated
* I ended up deleting the branch from the remote repo
* Text

  Description automatically generated
* I merged “login\_child” to main
* Text

  Description automatically generated
* Then pushed it
* Now “main” branch has all of the commits from the login\_child branch
* Graphical user interface, text, application

  Description automatically generated
* Graphical user interface, application

  Description automatically generated with medium confidence

## 5 – switching between branches

* Branch: login
  + Graphical user interface, text

    Description automatically generated
* Branch: login\_child
  + Graphical user interface, text

    Description automatically generated

## 6 – Merging 2nd branch to main

* Origin/main (main branch on remote repo) currently only has “test\_child2a\_\_.py”
  + Local branch “login\_child2b\_\_) has “test\_child2bbb.py”
* Text

  Description automatically generated
  + After your message is done. Press “esc”, write “:wq” (appears at the bottom) then hit enter
  + <https://stackoverflow.com/questions/19085807/please-enter-a-commit-message-to-explain-why-this-merge-is-necessary-especially>
  + Text

    Description automatically generated
* Now “main” has the files from both “login\_child” branch and “login\_child2b\_\_” branch
  + Note that only commits that are ahead of the main are merged into main.
  + Note how it says the local repo’s main branch is ahead of the “origin/main” branch by 2 commits.
    - One commit was the commit “created the test\_child2bbbb”
    - 2nd commit was the commit to merge the “login\_child2b\_\_” branch with the “origin/main” branch
  + Graphical user interface, text, application, email

    Description automatically generated

## 7 – Create branch based on revision (old commit)

* Using the hashtag found in the commit;
  + Graphical user interface, text, application, email

    Description automatically generated
* Text

  Description automatically generated
* Graphical user interface, text

  Description automatically generated
  + Can see that this “mainOld” branch is a copy of the main branch from the original commit which add in “test\_child.py”. None of the new test\_child2a or test\_child2b.py files are included
* a