Essential Git Commands

Contents

[GitHub – Make a repository 2](#_Toc478388618)

[Save a Commit to GitHub 5](#_Toc478388619)

[Making sure you are working on the current commit. 7](#_Toc478388620)

[Branches 8](#_Toc478388621)

# GitHub – Make a repository

|  |
| --- |
|  |
| Click on the green “New repository” |
|  |
| 1. Enter the Repository name 2. Check the “Initialize this repository with a README” 3. Click the green “Create repository” button |

|  |
| --- |
|  |
| Click the green “Clone or download” Button |

|  |
| --- |
|  |
| Click the copy to clip board button (Red arrow pointing at it in this picture) |

|  |
| --- |
|  |
| Open Git Bash |

|  |
| --- |
|  |
| Change to the directory you want to place the file folder from Git Hub, not “cd” is the command to change directory. |

|  |
| --- |
|  |
| Type “git clone “ (make sure there is a space after clone) than paste in the path you copied from Git Hub and enter. |
|  |
| You will see something like this, and your files are cloned and on your local machine ready to use. |
|  |
|  |

# Save a Commit to GitHub

|  |
| --- |
|  |
| Use the command “cd” to go into the folder of your local repository. |

|  |
| --- |
|  |
| Use the command “git add –A” to stage all of the files that have been changed or added. |

|  |
| --- |
|  |
| Use the command ‘git commit –m “comment on what was done” ‘ to create the commit with the message that will be associated with the commit. |

|  |
| --- |
|  |
| Use the command “git push” to push the changes up to GitHub. |

|  |
| --- |
|  |
| After the push you may have to log into your GitHub account. It may be a pop up window like this, or… |

|  |
| --- |
| the UserName might be in the console with the password also in the console or… |
| The UserName might be in the console and the password in a pop up.  If you have logged into the machine before, you might not need to log in at all, depending on your password settings. |

|  |
| --- |
|  |
| When you have finished logging in you will se something like this, which indicates that the commit has been pushed up to GitHub. |

# Making sure you are working on the current commit.

If you are working on a project on different machines, before you start working you should do a pull to make sure you are working on the latest commit.

|  |
| --- |
|  |
| Run Git Bash and go to the folder using the “cd” that has your repository saved on your local machine. |

|  |
| --- |
|  |
| Use the command “git pull” and the latest version will be merged into your repository.cd |

|  |
| --- |
|  |
| If you do “git pull” and there is nothing new to pull you will get this message. |

# Branches

## Creating a new Branch

|  |
| --- |
|  |
| Use the command ‘git checkout –b “new-branch” to create a new branch and go there. |

## Creating a commit to the new branch for the first time.

|  |
| --- |
|  |
| The first time you do a commit to a new branch this is what you see, if you use the standard commands. To fix the problem type in the push command exactly as git tells you to. |

|  |
| --- |
|  |
| If you remember to on your first commit you can type in the push command as ‘git push –set-upstream origin new-branch’ the first time and you will not get the previous fatal error message. It does not hurt if you forget to do the regular ‘git push’ for git will tell you exactly what you need to do for the first commit. |

|  |
| --- |
|  |
| After the first time you do a commit to a new branch, commits will work as normal. |

## Switching Branches

|  |
| --- |
|  |
| Use the command ‘git checkout master’ to switch back to a branch that already exists, in this case master. |

## Merging Branches

|  |
| --- |
|  |
|  |