Starz University

Sinkor Airfield, Cheeseman Avenue

P.O. Box 1040

Monrovia, Liberia

ID#: **7222**

Name: **Marilyn Collins**

**GitHub username: marilyncollins**

**Repository Link : https://github.com/marilyncollins/Art.277.git**

**Explain in your words the terms Git and GitHub.**

**GIT** is a version control system, or basically a tool to manage your code, while **GitHub** is a hosting service for GIT repositories.

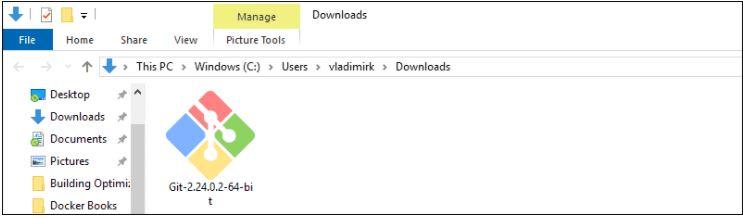
**Explain the steps to download, install and configure Git on your PC**

Step in installing Git on your computer and how to use it:

**Prerequisites:**

* Administrator privileges
* Access to a command-line
* Your favorite coding text editor
* Username and password for the Github website (optional)

* Installing Git prompts you to select a text editor. If you don’t have one, we strongly advise you to install prior to installing Git. Our roundup of the [best text editors for coding](https://phoenixnap.com/kb/best-linux-text-editors-for-coding) may help you decide.
* Download Git for Windows
* Browse to the official Git website: <https://git-scm.com/downloads>Click the download link for Windows and allow the download to complete.
* Extract and Launch Git Installer
* Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.

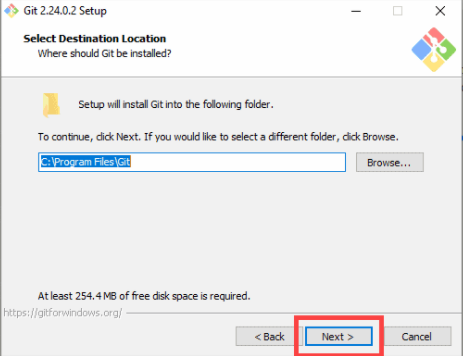


* Allow the app to make changes to your device by clicking **Yes** on the User Account Control dialog that opens. 

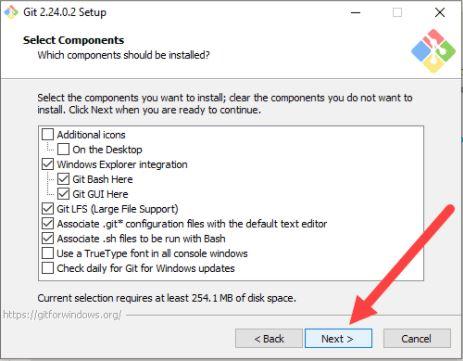
* Review the GNU General Public License, and when you’re ready to install, click **Next**.



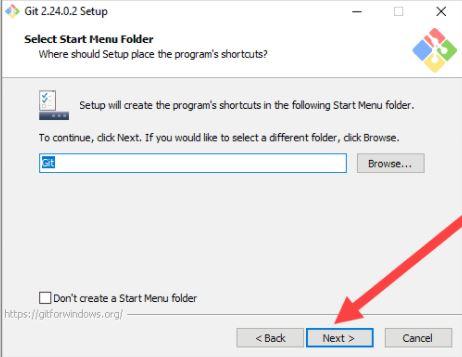
* The installer will ask you for an installation location. Leave the default, unless you have reason to change it, and click **Next**.



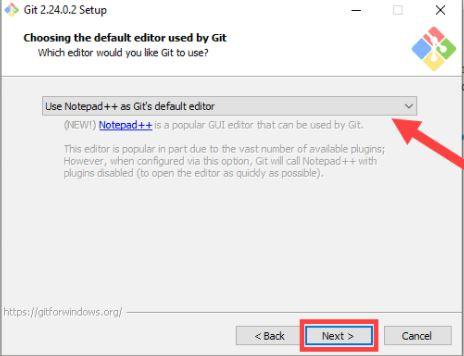
* A component selection screen will appear. Leave the defaults unless you have a specific need to change them and click **Next**.



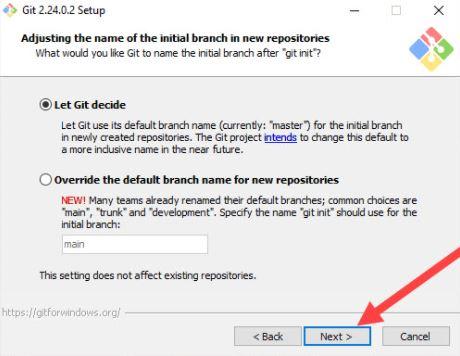
* The installer will offer to create a start menu folder. Simply click **Next**.

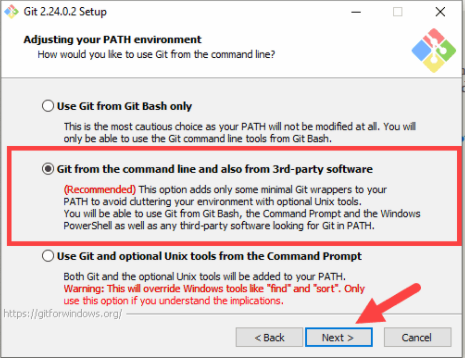


* Select a text editor you’d like to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click **Next**.

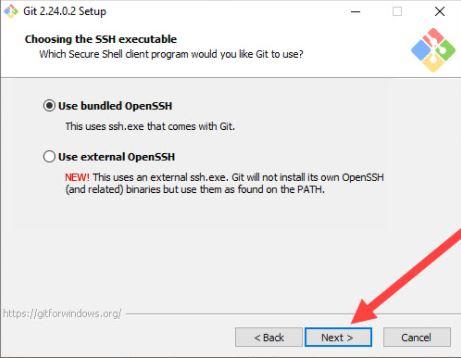


* The next step allows you to choose a different name for your initial branch. The default is 'master.' Unless you're working in a team that requires a different name, leave the default option and click **Next.**

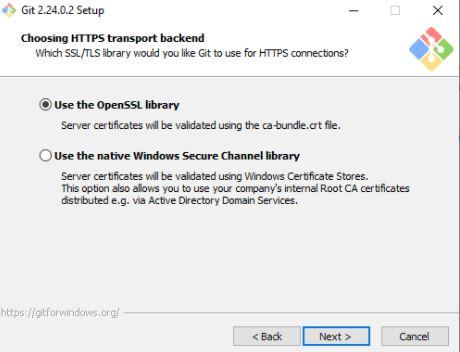


* This installation step allows you to change the **PATH environment**. The **PATH**is the default set of directories included when you run a command from the command line. Leave this on the middle (recommended) selection and click **Next**. 

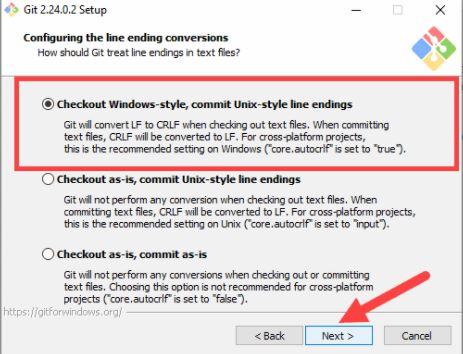
* The installer now asks which SSH client you want Git to use. Git already comes with its own SSH client, so if you don't need a specific one, leave the default option and click **Next.**

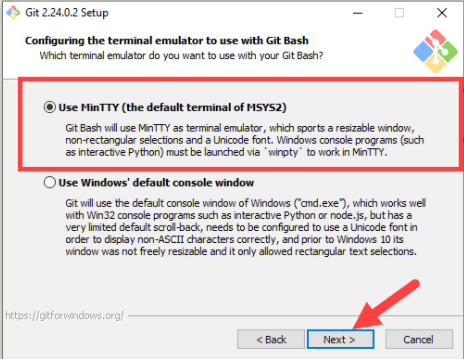


* The next option relates to server certificates. Most users should use the default. If you’re working in an Active Directory environment, you may need to switch to Windows Store certificates. Click **Next**.

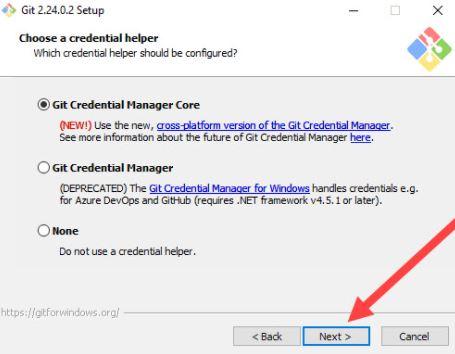


* The next selection converts line endings. It is recommended that you leave the default selection. This relates to the way data is formatted and changing this option may cause problems. Click **Next**.

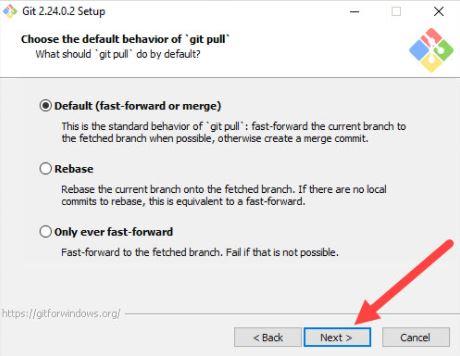


* Choose the terminal emulator you want to use. The default MinTTY is recommended, for its features. Click Next.

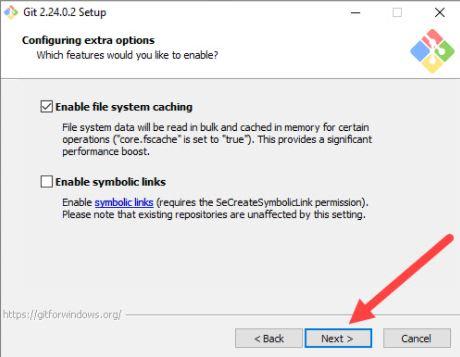
* The installer now asks what the **git pull** command should do. The default option is recommended unless you specifically need to change its behavior. Click **Next**to continue with the installation.



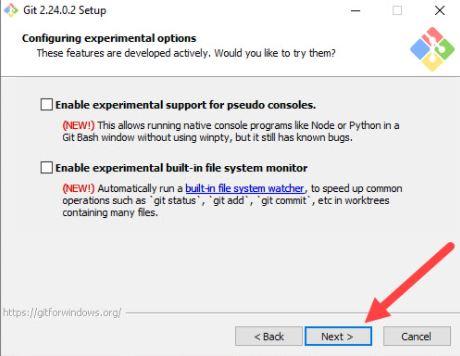
* Next you should choose which credential helper to use. Git uses credential helpers to fetch or save credentials. Leave the default option as it is the most stable one, and click **Next**.



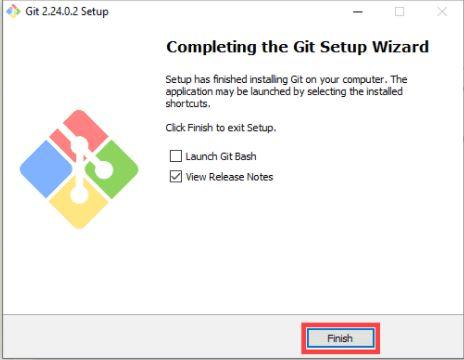
* The default options are recommended, however this step allows you to decide which extra option you would like to enable. If you use symbolic links, which are like shortcuts for the command line, tick the box. Click **Next**.



* Depending on the version of Git you’re installing, it may offer to install experimental features. At the time this article was written, the options to include support for pseudo controls and a built-in file system monitor were offered. Unless you are feeling adventurous, leave them unchecked and click **Install**.



* Once the installation is complete, tick the boxes to view the Release Notes or Launch Git Bash, then click **Finish**.



**Discuss the process to create and initialize a project in Git.**

**Step 1:**

**GitHub account creation**

To create your account, you need to connect on [the main GitHub page](https://github.com/) and to fill in the registration form.

**Git installation**

Now you need to install Git tools on your computer. There are different Git software, but it’s better to install the basic one to start. We will use the command line to communicate with GitHub.

**Step 2:**

**Your first GitHub project!**

Now that you’re ready, you can return to the main GitHub page and click on the “*+*” icon in the menu bar.

Once you click on this button, a new menu appears with a “New repository” entry. Click on it!

[The repository creation page](https://github.com/new) will appear. Choose a cool name for your first repository and put a small description before clicking on the “*Create repository*” button.

Well done! Your first GitHub repository is created. If you want to see all your repositories, you need to click on your profile picture in the menu bar then on “*Your repositories*”.

**Step 3:**

**A good cover**

It’s time to make your first modification to your repository. What do you think about creating a cover for it, a kind of welcome text?

**A local version of your project**

Your first mission is to get a copy of the repository on your computer. To do that, you need to “*clone*” the repository. On the repository page, you need to get the “*HTTPS*” address.

Once you had the address of the repositories, you need to use your terminal (through shell commands) to move in the place where you want to put the directory copy (for example you can move in your “*Documents*” folder). When you are ready, you can enter:

$ git clone [HTTPS ADDRESS]

This command will make a local copy of the repository hosted at the given address.

Now, your repository is on your computer. You need to move in it with:

$ cd [NAME OF REPOSITORY]

*Note: When you clone, Git will create a repository on your computer. If you want, you can access your project with the computer user interface.*

**Explain the steps to pull and push from a local repository to a remote repository.**

**Steps in pull and push from a local repository to a remote repository:**

* [Git PUSH](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#git-push)
* [Using Command line to PUSH to GitHub](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-github)
* [Creating a new repository.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-github2)
* [Open your Git Bash.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-github2)
* [Create your local project in your desktop directed towards a current working directory.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-github3)
* [Initialize the git repository](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-github4)
* [Add the file to the new local repository.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-github4)
* [Commit the files staged in your local repository by writing a commit message.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-github5)
* [Copy your remote repository's URL from GitHub.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-githu6)
* [Add the URL copied, which is your remote repository to where your local content from your repository is pushed.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-github8)
* [Push the code in your local repository to GitHub](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-github9)
* [View your files in your repository hosted on GitHub.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-command-line-to-push-to-github10)
* [Using GitHub Desktop to PUSH to your local content to GitHub.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-github-desktop-to-push-to-your-local-content-to-github-)
* [Click "Set up in a Desktop".](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-github-desktop-to-push-to-your-local-content-to-github-1)
* [Cloning in a GitHub Desktop.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-github-desktop-to-push-to-your-local-content-to-github-2)
* [Copy all the required files from your local computer into the clone folder on your computer.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-github-desktop-to-push-to-your-local-content-to-github-3)
* [Move to GitHub Desktop and commit to master](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-github-desktop-to-push-to-your-local-content-to-github-4)
* [Publish branch in GitHub Desktop to upload your all files to GitHub.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#using-github-desktop-to-push-to-your-local-content-to-github-5)
* [PULL Request](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request)
* [PULL Request through Command Line.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-command-line-)
* [Fork the Repository.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-command-line-1)
* [Open your bash in your computer.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-command-line-1)
* [Make a new branch.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-command-line-2)
* [Make a change by using vim from bash or direct replacement from the original README file.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-command-line-3)
* [Adding and Committing a file to the repository.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-command-line-4)
* [Push the repository to the GitHub.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-command-line-6)
* [PULL request for a specific branch on GitHub.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-command-line-7)
* [Open a Pull request](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-command-line-8)
* [Deleting a Branch after the PULL Request is Merged.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#deleting-a-branch-after-the-pull-request-is-merged-)
* [PULL Request through GitHub Desktop](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-github-desktop)
* [Cloning and Opening to Desktop.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-github-desktop1)
* [Create a new branch.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-github-desktop2)
* [Make a change in the imp file from the text editor.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-github-desktop3)
* [Commit the changes.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-github-desktop4)
* [Publish the branch.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-github-desktop5)
* [Create a PULL Request.](https://www.datacamp.com/community/tutorials/git-push-pull?msclkid=7724e45ca57711ecbc1c6cc8bca295ed#pull-request-through-github-desktop6)

**In your own words, explain the term push, pull, commit, stage and branches in Git**

* Push - The git push command is used to upload local (from your desktop) repository content to a remote repository.
* Pull - git pull is a Git command used to update the local version of a repository from a remote.
* Commit - it records other objects that identifies and specifies the 'snapshot'.
* Stage - is technically a required intermediate step in the process of checking in a file.
* Branches -  is a way to keep developing and coding a new feature or modification to the software and still not affecting the main part of the project.