What is GitHub? When was it created? Why? By who? What similar platforms exist? Why would you use such a platform? (Answer between 5 and 10 lines)

* GitHub is like Wikipedia for programmers. You can edit files, see who changed what, view old versions of files, and access it from anywhere in the world – except you’re working with source code instead of encyclopedia data. Companies use it to build software and websites, while hobbyist programmers use it to find and share projects.
* The business model is simple: if you want to share your source code with everyone and make it public, you don’t pay anything. If you need to hide your source code because it’s private and runs your business, you pay.
* Development of the GitHub platform began on 1 October 2007. The site was launched in April 2008 by [Tom Preston-Werner](https://en.wikipedia.org/wiki/Tom_Preston-Werner), [Chris Wanstrath](https://en.wikipedia.org/wiki/Chris_Wanstrath), and PJ Hyett after it had been made available for a few months prior as a beta release.
* Tom Preston-Werner is the founder of Github.
* Bitbucket, Sourceforge, Gitlab are alternatives to Github.
* GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

Define the following terms in the context of Git (2 lines maximum):

* Repository: A repository is simply a place where the history of your work is stored. It often lives in a .gitsubdirectory of your working copy - a copy of the most recent state of the files you're working on.
* Commit: Every time you commit, or save the state of your project in Git, it basically takes a picture of what all your files look like at that moment and stores a reference to that snapshot. To be efficient, if files have not changed, Git doesn’t store the file again, just a link to the previous identical file it has already stored. Git thinks about its data more like a **stream of snapshots**.
* Push: to push commits made on your local branch to a remote repository.
* Branch: In Git, the repository refers to your entire project. Within a single Git repository, you have at least one branch. You can use git branch new Feature to create a new branch within your repository to track your changes to the changes to your code base that pertain to a particular new feature. Anytime you clone a repository you are creating a copy of the project repository on your local machine. In addition, you will also be able to git pull any new changes that other developers have pushed up to the public repository.
* Fork: A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea.
* Merge: Join two or more development histories together
* Clone: Clones a repository into a newly created directory, creates remote-tracking branches for each branch in the cloned repository (visible using git branch -r), and creates and checks out an initial branch that is forked from the cloned repository’s currently active branch.
* Pull: Fetches from and integrates with another repository or a local branch
* Pull request: Generates a summary of pending changes. The request, printed to the standard output, begins with the branch description, summarizes the changes and indicates from where they can be pulled.