1. What are the differences between Git and GitHub?

Git	GitHub
Git is the distributed version control system. Primarily used by the programmers and others to write code.	GitHub is the website that allows you to upload repositories online.
It runs in the command line on your local machine.	It provides you to backup of your file.
It allows us to keep track of our files and modification to those files in something called repo or repository.	It gives a visual interface for navigating your repo's.
You can use it alone or you can use it with team of people who ever is working on same project.	It gives other people a way to navigate your repo's.
It's uses full in team environment, because everyone can work independently on those files, merge those files together and there is a permanent record who made which change.	It makes repo's collaboration easy.
Test changes to code without losing the original.	

2. What is Git Workflow?
   - A set of guidelines developers can follow when using version control.
   - Referred to as a "Branching Model".
   - Not rules, but guidelines which are not set in stone.

   Works:
   - Central repository.
   - Developers work locally and push their branches.
   - Two branches used to record project history, Master and Develop.
   - Develop serves as an integration branch for features.
   - Master branch store the official release history.

3. How many types of version control systems are there?

There are two types of version control: centralized and distributed.

Centralized version control

With centralized version control systems, you have a single “central” copy of your project on a server and commit your changes to this central copy. You pull the files that you need, but you never have a full copy of your project locally. Some of the most common version control systems are centralized, including Subversion (SVN) and Perforce.

Distributed version control

With distributed version control systems (DVCS), you don't rely on a central server to store all the versions of a project’s files. Instead, you clone a copy of a repository locally so that you have the full history of the project. Two common distributed version control systems are Git and Mercurial.

While you don't have to have a central repository for your files, you may want one "central" place to keep your code so that you can share and collaborate on your project with others. That's where Bitbucket comes in. Keep a copy of your code in a repository on Bitbucket so that you and your teammates can use Git or Mercurial locally and to push and pull code.