In short, svn is a Centralized Revision Control System, and git is a Distributed Revision Control System (DVCS).  
  
If you're a designer, I'd also recommend that you check out mercurial (hg, for short) and [bitbucket.org](http://bitbucket.org/).  Mercurial is much more user friendly than git, and offers nearly identical features.   
  
When you "check in" or "commit" to svn, you're contacting a central repository, synchronizing (merging & resolving) with the centralized versions, creating a changelist, and then sending that changelist back to the centralized repository.  All users use and share the same repository, possibly with branching.  If 2 users want to share code, the only way they can do so is by checking into the repository and then each doing a sync.   
  
In git, when you check out a repository, you're getting a complete clone of the whole thing -- all revisions of all files, all the metadata, everything.  You make changes locally, check in (commit) to your local copy of the repository, and then "push" those changesets to another repository when you want to publish them or share them with other users.  You can also "pull" to synchronize with a remote repository.  Since every repository is equal (they're all complete copies) you can "push" changes anywhere -- this is how 2 or more users can collaborate on a new feature without impacting the centralized repository. These users just issue push/pull operations with each other when they want to share code.  Once they have a working functional improvement, they can then "push" all the relevant changesets back to the central or production repository.   
  
git is very flexible in the workflows that it supports, and many large projects use complex workflows with multiple repositories and gatekeepers.  The best way to learn about this is to take a look at this great online reference to git: <http://progit.org/book/ch1-1.html>  
  
So, in short, if you're working alone on a small set of files, then you'll see very little difference between svn and git (or mercurial).  If you're collaborating with one or more other users, and you're willing to go through a fairly steep learning curve, I think you should consider git or mercurial.