Devtopia steps:

First of all – check the username and password with devtopia account:

$ git config --global user.name

If it shows nothing or a name that’s not your user name then set it:

$ git config --global user.name nobi4775

Similarly, check and/or set user email as follows:

$ git config --global user.email nahmed@esri.com

**Step 1:**

Get the contents from devtopia (https://devtopia.esri.com/ArcGISPro/pythonharness)

If for the first time:

>git **clone** “https://devtopia.esri.com/ArcGISPro/pythonharness.git” to\_folder

(from url to local folder, to\_folder will be created if it does not exist)

# set up

>git **remote add origin** <https://devtopia.esri.com/ArcGISPro/pythonharness.git>

# upload

>git push -u origin master

Second time onward: (stash first?)

>git checkout localBranch (create a new branch)

>git **fetch** origin remoteBranch ???????????

**Step 2:**

Do some editing of a file that we want to modify

>git diff (will show the difference due to changes)

Stage if we are satisfied

>git add file\_name

Create a local branch (or create it after)

>git checkout –b branch\_name

Git switches to the new branch. Now commit the changes:

>git commit –m “write a meaningful comment”

See what is done so far:

>git log

Send the changes to devtopia:

>git push –u origin branch\_name

On the web – go to devtopia and compare. Create a pull request, get it approved … (the rest is like any other process)

fetch or pull:

**Git Pull:**

From what I understand, git pull will pull down from a remote whatever you ask (so, whatever trunk you’re asking for) and instantly merge it into the branch you’re in when you make the request. Pull is a high-level request that runs ‘fetch’ then a ‘merge’ by default, or a rebase with ‘–rebase’. You could do without it, it’s just a convenience.

%> git checkout localBranch

%> git pull origin master

%> git branch

master

\* localBranch

The above will merge the remote “master” branch into the local “localBranch”.

**Git fetch:**

Fetch is similar to pull, except it won’t do any merging.

%> git checkout localBranch

%> git fetch origin remoteBranch

%> git branch

master

\* localBranch

remoteBranch

So, the fetch will have pulled down the remoteBranch and put it into a local branch called “remoteBranch”. creates a local copy of a remote branch which you shouldn’t manipulate directly; instead create a proper local branch and work on that. ‘git checkout’ has a confusing feature though. If you ‘checkout’ a local copy of a remote branch, it creates a local copy and sets up a merge to it by default.