Git for the DSI quick reference	
What you want to do	Git commands at command line
Clone a repo on Github locally	git clone <url_from_github></url_from_github>
Check to see what files are changed	git status
Show file differences that haven't been staged	git diff
See what branch you are on	git branch
Add a file to the staging area (for tracking)	git add <filename></filename>
Remove a file from the staging area	git reset <filename></filename>
Show file differences between staged and last commit	git diffstaged
-	git commit -m " <a commit="" descriptive="" message="" short,="">"</a>
Commit changes in staging area to a checkpoint	
See where on Github you will push your commit	git remote -v
Push your commit to Github	git push <name_of_remote> <name_of_branch></name_of_branch></name_of_remote>
Pull your latest changes on Github down locally	<pre>git pull <name_of_remote> <name_of_branch></name_of_branch></name_of_remote></pre>
For pair programming	
Assumes you and your partner are working on the master (defa	nult) branch
After your partner has added you as a collaborator on his/her G	ithub repository:
Add your partner as a remote (for push/pulling)	<pre>git remote add <their_name> <url_of_their_repo_on_github></url_of_their_repo_on_github></their_name></pre>
Pull their changes into your master branch	git pull <their_name> master</their_name>
You do some work (you "drive" aka code)	
Add your changes to the staging area	<pre>git add <filename_1> <filename_2></filename_2></filename_1></pre>
Commit changes in staging area to a checkpoint	git commit -m " <a commit="" descriptive="" message="" short,="">"</a>
Push your commit to your partner's remote	git push <their_name> master</their_name>
	git push origin master
Push your commit to your remote  Now your partner starts to drive (you navigate)	Pre basii oi 18111 maseei
	git pull origin master
First your partner pulls your changes down	git pull origin master
For group case studies	
Assumes you are following the Feature Branch Workflow	
	turo branch warkflow
https://www.atlassian.com/git/tutorials/comparing-workflows/fea	
Assumes you are following the Feature Branch Workflow <a href="https://www.atlassian.com/git/tutorials/comparing-workflows/fea">https://www.atlassian.com/git/tutorials/comparing-workflows/fea</a> In this workflow you never work on the master branch (for fin	al code & code to be shared)
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow <b>you never work on the master branch</b> (for fin One of the group members (A) forks the Galvanize case study,	al code & code to be shared) and then clones it locally.
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow <b>you never work on the master branch</b> (for fin One of the group members (A) forks the Galvanize case study,	al code & code to be shared) and then clones it locally.
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow <b>you never work on the master branch</b> (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo	al code & code to be shared) and then clones it locally. one A's repository (not Galvanize's).
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them	and code & code to be shared) and then clones it locally. one A's repository (not Galvanize's).  git branch <your_name></your_name>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch	al code & code to be shared) and then clones it locally. one A's repository (not Galvanize's).
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch	and code & code to be shared) and then clones it locally. one A's repository (not Galvanize's).  git branch <your_name></your_name>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work	and code & code to be shared) and then clones it locally. one A's repository (not Galvanize's).  git branch <your_name></your_name>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea	and code & code to be shared) and then clones it locally. one A's repository (not Galvanize's).  git branch <your_name></your_name>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so:	and code & code to be shared) and then clones it locally. one A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name></your_name></your_name>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint	and code & code to be shared) and then clones it locally. one A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name></your_name></your_name>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote	<pre>and code &amp; code to be shared) and then clones it locally. one A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git add <filename> git commit -m "<a commit="" descriptive="" message="" short,="">"</a></filename></your_name></your_name></pre>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master	<pre>and code &amp; code to be shared) and then clones it locally. one A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git add <filename> git commit -m "<a commit="" descriptive="" message="" short,="">"</a></filename></your_name></your_name></pre>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch	mal code & code to be shared) and then clones it locally.  one A's repository (not Galvanize's).  git branch <your_name>  git checkout <your_name>  git add <filename> git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master</your_name></a></filename></your_name></your_name>
Inttps://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A pulls down whatever is new on the master branch	mal code & code to be shared) and then clones it locally. One A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git add <filename> git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master</your_name></a></filename></your_name></your_name>
Intips://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A pulls down whatever is new on the master branch A verifies that your branch exists on the remote	mal code & code to be shared) and then clones it locally. one A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git add <filename> git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r</your_name></a></filename></your_name></your_name>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A pulls down whatever is new on the master branch A verifies that your branch exists on the remote A fetches your branch	and code & code to be shared) and then clones it locally.  one A's repository (not Galvanize's).  git branch <your_name>  git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local</your_name></your_name></your_name></a></your_name></your_name>
Intips://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A pulls down whatever is new on the master branch A verifies that your branch exists on the remote A fetches your branch A verifies he/she is on the master branch	and code & code to be shared) and then clones it locally.  one A's repository (not Galvanize's).  git branch <your_name>  git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local git checkout master</your_name></your_name></your_name></a></your_name></your_name>
Intips://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A pulls down whatever is new on the master branch A verifies that your branch A verifies the so on the master branch A verifies he/she is on the master branch A merges your branch into master	and code & code to be shared) and then clones it locally.  one A's repository (not Galvanize's).  git branch <your_name>  git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local git checkout master git merge <your_name></your_name></your_name></your_name></your_name></a></your_name></your_name>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A verifies that your branch exists on the remote A fetches your branch A verifies he/she is on the master branch A merges your branch into master A pushes the master branch up to Github	and code & code to be shared) and then clones it locally.  One A's repository (not Galvanize's).  git branch <your_name>  git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local git checkout master git merge <your_name> git push origin master</your_name></your_name></your_name></your_name></a></your_name></your_name>
Intips://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A pulls down whatever is new on the master branch A verifies that your branch exists on the remote A fetches your branch A verifies he/she is on the master branch A merges your branch into master A pushes the master branch up to Github Others can pull down the master branch to access your code	and code & code to be shared) and then clones it locally.  One A's repository (not Galvanize's).  git branch <your_name>  git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git fetch origin <your_name>:<your_name> # Github:local git checkout master git merge <your_name> git push origin master git merge <your_name> git push origin master git push origin master git push origin master</your_name></your_name></your_name></your_name></your_name></a></your_name></your_name>
In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they close Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A verifies that your branch exists on the remote A verifies theyour branch into master branch A verifies he/she is on the master branch A pulls down whatever is new on the remote A verifies he/she is on the master branch A verifies he/she is on the master branch A pushes the master branch up to Github Others can pull down the master branch to access your code To use it, you and they should make a branch off master	and code & code to be shared) and then clones it locally.  one A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git add <filename> git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local git checkout master git merge <your_name> git push origin master git pull origin master git checkout master</your_name></your_name></your_name></your_name></a></filename></your_name></your_name>
Intips://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A verifies that your branch exists on the remote A verifies that your branch A verifies he/she is on the master branch A werifies he/she is on the master branch A merges your branch into master A pushes the master branch up to Github Others can pull down the master branch to access your code To use it, you and they should make a branch off master Delete the old branch (if all changes have been committed)	and code & code to be shared) and then clones it locally.  Done A's repository (not Galvanize's).  git branch <your_name>  git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">"  git push origin <your_name>  git checkout master  git pull origin master  git fetch origin <your_name>:<your_name> # Github:local  git checkout master  git merge <your_name>  git push origin master  git push origin master  git push origin master  git merge <your_name>  git push origin master  git push origin master  git push origin master  git checkout master  git checkout master</your_name></your_name></your_name></your_name></your_name></a></your_name></your_name>
In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they close Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A verifies that your branch exists on the remote A verifies that your branch into master A guiles they he is on the master branch A verifies he/she is on the master branch A merges your branch into master A pushes the master branch up to Github Others can pull down the master branch to access your code To use it, you and they should make a branch off master Delete the old branch (if all changes have been committed) Make a new branch	and code & code to be shared) and then clones it locally.  one A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local git checkout master git merge <your_name> git push origin master git push origin master git push origin master git checkout master git checkout master git push origin master git push origin master git push origin master git push origin master git branch -d <your_name> git branch -d <your_name> git branch <your_name> git branch <your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></a></your_name></your_name>
https://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area	and code & code to be shared) and then clones it locally.  One A's repository (not Galvanize's).  git branch <your_name>  git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">"  git push origin <your_name>  git checkout master  git pull origin master  git fetch origin <your_name>:<your_name> # Github:local  git checkout master  git merge <your_name>  git push origin master  git push origin master  git push origin master  git merge <your_name>  git push origin master  git push origin master  git push origin master  git push origin master  git checkout master  git checkout master  git push origin master  git push origin master  git push origin master  git checkout master  git checkout master</your_name></your_name></your_name></your_name></your_name></a></your_name></your_name>
In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they close Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so:  Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A verifies that your branch exists on the remote A verifies that your branch into master A guils down whatever is new on the master branch A verifies he/she is on the master branch A verifies he/she is on the master Delete the master branch up to Github Others can pull down the master branch to access your code To use it, you and they should make a branch off master Delete the old branch (if all changes have been committed) Make a new branch Check it out (to work on it)	and code & code to be shared) and then clones it locally.  one A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local git checkout master git merge <your_name> git push origin master git push origin master git push origin master git checkout master git checkout master git push origin master git push origin master git push origin master git push origin master git branch -d <your_name> git branch -d <your_name> git branch <your_name> git branch <your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></a></your_name></your_name>
In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they close Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so:  Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A verifies that your branch exists on the remote A verifies that your branch into master A merges your branch into master A pushes the master branch up to Github Others can pull down the master branch to access your code To use it, you and they should make a branch off master Delete the old branch (if all changes have been committed) Make a new branch Check it out (to work on it)	and code & code to be shared) and then clones it locally.  One A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git push origin master git pull origin master git fetch origin <your_name>:<your_name> # Github:local git checkout master git merge <your_name> git push origin master git checkout master git branch -d <your_name> git branch -d <your_name> git branch <your_name> git checkout <your_name> git checkout <your_name> git checkout <your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></a></your_name></your_name>
Intips://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A verifies that your branch exists on the remote A fetches your branch A verifies theyshe is on the master branch A verifies he/she is on the master Delete he she master branch up to Github Others can pull down the master branch to access your code To use it, you and they should make a branch off master Delete the old branch (if all changes have been committed) Make a new branch Check it out (to work on it)  Fetching a remote branch Clone a repo on Github locally	<pre>mal code &amp; code to be shared) and then clones it locally.  me A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git add <filename> git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local git checkout master git merge <your_name> git push origin master git checkout master git checkout master git checkout master git checkout syour_name&gt; git checkout <your_name> git clone <url_from_github></url_from_github></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></a></filename></your_name></your_name></pre>
Intips://www.atlassian.com/git/tutorials/comparing-workflows/feath this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they close as the member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so:  Add your changes to the staging area  Commit changes in staging area to a checkpoint  Push your commit (on your branch) to A's remote  Now A will pull down your branch, and merge it into master  A gets on the master branch  A verifies that your branch exists on the remote  A verifies that your branch into master  A pushes the master branch up to Github  Others can pull down the master branch to access your code  To use it, you and they should make a branch off master  Delete the old branch (if all changes have been committed)  Make a new branch  Check it out (to work on it)  Fetching a remote branch  Clone a repo on Github locally  Fetch the remote branch	<pre>mal code &amp; code to be shared) and then clones it locally. me A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local git checkout master git merge <your_name> git push origin master git checkout syour_name&gt; git checkout <your_name> git fetch origin <remote_branch>:<remote_branch> # Github:local</remote_branch></remote_branch></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></a></your_name></your_name></pre>
Intips://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A verifies that your branch exists on the remote A fetches your branch A verifies he/she is on the master branch A verifies he/she is on the master Delete the old branch (if all changes have been committed) Make a new branch Check it out (to work on it)  Fetching a remote branch If you want to merge a branch into master (after you've fetch If you want to merge a branch into master (after you've fetch If you want to merge a branch into master (after you've fetch If you want to merge a branch into master (after you've fetch	<pre>mal code &amp; code to be shared) and then clones it locally.  me A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git pull origin master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local git checkout master git push origin master git checkout <your_name> git fetch origin <remote_branch>:<remote_branch> # Github:local check it above)</remote_branch></remote_branch></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></a></your_name></your_name></pre>
Intips://www.atlassian.com/git/tutorials/comparing-workflows/fea In this workflow you never work on the master branch (for fin One of the group members (A) forks the Galvanize case study, On Github, A adds everyone as collaborators, and then they clo Each member (incl. A) makes a branch named after them Now each member (incl. A) checks out their branch Now everyone does their work You have some code you want to share with the group, so: Add your changes to the staging area Commit changes in staging area to a checkpoint Push your commit (on your branch) to A's remote Now A will pull down your branch, and merge it into master A gets on the master branch A verifies that your branch exists on the remote A terifies that your branch exists on the remote A rerifies he/she is on the master branch A werifies he/she is on the master Changes your branch into master A pushes the master branch up to Github Others can pull down the master branch to access your code To use it, you and they should make a branch off master Delete the old branch (if all changes have been committed) Make a new branch Check it out (to work on it)	<pre>mal code &amp; code to be shared) and then clones it locally. me A's repository (not Galvanize's).  git branch <your_name> git checkout <your_name>  git commit -m "<a commit="" descriptive="" message="" short,="">" git push origin <your_name>  git checkout master git pull origin master git branch -r git fetch origin <your_name>:<your_name> # Github:local git checkout master git merge <your_name> git push origin master git checkout syour_name&gt; git checkout <your_name> git fetch origin <remote_branch>:<remote_branch> # Github:local</remote_branch></remote_branch></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></your_name></a></your_name></your_name></pre>