**# Cheatsheet for making a change to kteha on branch gh-pages**

# Lines beginning with # are comments

# Lines beginning with > are commands

# All other lines are the system's response to commands

**#Summary of key commands:**

>mkdir swproj # optional make new directory to start fresh

>cd swproj # go to directory where you want repository to be

>git init # initialize directory for use by git

#get a copy of all the files on branch gh-pages

>git clone --branch gh-pages --single-branch https://github.com/krcowles/kteha

>cd kteha

## Make changes to index.html ##

>git add index.html # Mark index.html for inclusion in the next commit

>git commit -m "Add hacked up table at the bottom of page." # Commit changes to local repository

>git push # push changes to remote repository

**#The same key commands with additional info included**

>mkdir swproj # optional make new directory to start fresh

>cd swproj # go to directory where you want repository to be

>git init # initialize directory for use by git

Initialized empty Git repository in /home/tom/Documents/FromAnritsuLaptop/Personal/swproj/.git/

#get a copy of all the files on branch gh-pages

>git clone --branch gh-pages --single-branch https://github.com/krcowles/kteha

Cloning into 'kteha'...

remote: Counting objects: 19, done.

remote: Compressing objects: 100% (18/18), done.

remote: Total 19 (delta 0), reused 19 (delta 0), pack-reused 0

Unpacking objects: 100% (19/19), done.

Checking connectivity... done.

>ls -l

total 16

drwxrwxr-x 4 tom tom 4096 Apr 9 17:20 ./

drwx------ 23 tom tom 4096 Apr 9 17:11 ../

drwxrwxr-x 7 tom tom 4096 Apr 9 17:14 .git/

drwxrwxr-x 6 tom tom 4096 Apr 9 17:20 kteha/

# created file cheatSheet.txt

>git status

On branch master

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

cheatSheet.txt

kteha/

nothing added to commit but untracked files present (use "git add" to track)

>cd kteha

>git status

On branch gh-pages

Your branch is up-to-date with 'origin/gh-pages'.

nothing to commit, working directory clean

>ll

total 184

drwxrwxr-x 6 tom tom 4096 Apr 9 17:20 ./

drwxrwxr-x 4 tom tom 4096 Apr 9 17:30 ../

drwxrwxr-x 8 tom tom 4096 Apr 9 17:20 .git/

drwxrwxr-x 2 tom tom 4096 Apr 9 17:20 images/

-rw-rw-r-- 1 tom tom 4593 Apr 9 17:20 index.html

-rw-rw-r-- 1 tom tom 149391 Apr 9 17:20 KTEHA\_DesignSpec.docx

-rw-rw-r-- 1 tom tom 397 Apr 9 17:20 README.txt

drwxrwxr-x 2 tom tom 4096 Apr 9 17:20 styles/

drwxrwxr-x 2 tom tom 4096 Apr 9 17:20 tools/

#Make changes to index.html (adding hacked up table at the bottom)

>git status

On branch gh-pages

Your branch is up-to-date with 'origin/gh-pages'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: index.html

no changes added to commit (use "git add" and/or "git commit -a")

>git add index.html # Mark index.html for inclusion in the next commit

>git commit -m "Add hacked up table at the bottom of page." # Commit changes to local repo

[gh-pages 3840962] Add hacked up table at the bottom of page.

1 file changed, 47 insertions(+), 1 deletion(-)

>git status

On branch gh-pages

Your branch is ahead of 'origin/gh-pages' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working directory clean

>git push # push changes to remote repository

Username for 'https://github.com': tjsandberg

Password for 'https://tjsandberg@github.com':

Counting objects: 5, done.

Delta compression using up to 2 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 746 bytes | 0 bytes/s, done.

Total 3 (delta 2), reused 0 (delta 0)

To https://github.com/krcowles/kteha

61cb5e7..3840962 gh-pages -> gh-pages

>git status

On branch gh-pages

Your branch is up-to-date with 'origin/gh-pages'.

nothing to commit, working directory clean

**#2nd day. Tom's first branch**

>pwd

/home/tom/Documents/FromAnritsuLaptop/Personal/swproj/kteha

>git log --oneline –decorate # See where branch pointers are pointing

3840962 (HEAD, origin/gh-pages, gh-pages) Add hacked up table at the bottom of p

61cb5e7 first commit

>git branch TomsFirstBranch # Create a new branch

>git log --oneline --decorate # See where branch pointers are pointing

3840962 (HEAD, origin/gh-pages, gh-pages, TomsFirstBranch) Add hacked up table a

61cb5e7 first commit

>git checkout TomsFirstBranch # Check out new branch to begin working on it

Switched to branch 'TomsFirstBranch'

>git log --oneline --decorate # See where branch pointers are pointing (unchanged)

3840962 (HEAD, origin/gh-pages, gh-pages, TomsFirstBranch) Add hacked up table a

61cb5e7 first commit

>gedit index.html # Make change

>git log --oneline --decorate # See where branch pointers are pointing (unchanged)

3840962 (HEAD, origin/gh-pages, gh-pages, TomsFirstBranch) Add hacked up table a

61cb5e7 first commit

>git commit -a -m 'Add better table at end. Still not sortable.' # Add/Commit in one

[TomsFirstBranch c724f56] Add better table at end. Still not sortable.

1 file changed, 106 insertions(+), 28 deletions(-)

>git log --oneline --decorate # See where branch pointers are pointing (note new commit)

c724f56 (HEAD, TomsFirstBranch) Add better table at end. Still not sortable.

3840962 (origin/gh-pages, gh-pages) Add hacked up table at the bottom of page.

61cb5e7 first commit

>git status

On branch TomsFirstBranch

nothing to commit, working directory clean

>git push --set-upstream origin TomsFirstBranch # push the change to remote server creating new branch there

Username for 'https://github.com': tjsandberg

Password for 'https://tjsandberg@github.com':

Counting objects: 7, done.

Delta compression using up to 2 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 1.29 KiB | 0 bytes/s, done.

Total 3 (delta 2), reused 0 (delta 0)

To https://github.com/krcowles/kteha

\* [new branch] TomsFirstBranch -> TomsFirstBranch

Branch TomsFirstBranch set up to track remote branch TomsFirstBranch from origin.

>git ls-remote https://github.com/krcowles/kteha

61cb5e7c0fac0b9ad99f94a62118e3a79c35cdc6 HEAD

c724f561e4516746d827fef28de87a15a5a0f86f refs/heads/TomsFirstBranch

3840962ba174b054f7c8e467c5c687be0306d79b refs/heads/gh-pages

61cb5e7c0fac0b9ad99f94a62118e3a79c35cdc6 refs/heads/master

**#What Ken should do. Maybe Tom should clean up repository first? If so, git clone will be simpler**

>mkdir swproj # optional make new directory to start fresh

>cd swproj # go to directory where you want repository to be

>git init # initialize directory for use by git

#get a copy of all the files on branch gh-pages

>git clone --branch gh-pages --single-branch https://github.com/krcowles/kteha

>cd kteha

>ll

>git status

#get a copy of all the files on branch TomsFirstBranch

>git clone --branch TomsFirstBranch --single-branch https://github.com/krcowles/kteha

>cd kteha

>ll

>git status

**#11-Apr-2016 notes on re-reading git manual**

<https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>

git config --global core.editor notepad

git config –list

<https://git-scm.com/book/en/v2/Getting-Started-Getting-Help>

git help config

If the manpages and this book aren’t enough and you need in-person help, you can try the #git or #github channel on the Freenode IRC server (irc.freenode.net). These channels are regularly filled with hundreds of people who are all very knowledgeable about Git and are often willing to help.

<https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository>

git init #should not be necessary if you do the following. Maybe Ken should start a new repo?

git clone https://github.com/libgit2/libgit2

That creates a directory named “libgit2”, initializes a .git directory inside it, pulls down all the data for that repository, and checks out a working copy of the latest version. If you go into the new libgit2 directory, you’ll see the project files in there, ready to be worked on or used.

<https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository>

git status

git add

git add is a multipurpose command – you use it to begin tracking new files, to stage files, and to do other things like marking merge-conflicted files as resolved. It may be helpful to think of it more as “add this content to the next commit” rather than “add this file to the project”.

If you modify a file after you run git add, you have to run git add again to stage the latest version of the file:

git status -s

New files that aren’t tracked have a ?? next to them, new files that have been added to the staging area have an A, modified files have an M and so on. There are two columns to the output - the left-hand column indicates the status of the staging area and the right-hand column indicates the status of the working tree.

.gitignore file #where does it go?

git diff

git diff staged

**#13-Apr-2016 notes on doing fetch and merge**

tom (1571): pwd

/home/tom/Documents/FromAnritsuLaptop/Personal/swproj/ktesa

tom (1575): git remote show origin

\* remote origin

Fetch URL: https://github.com/krcowles/ktesa

Push URL: https://github.com/krcowles/ktesa

HEAD branch: master

Remote branch:

gh-pages tracked

Local branches configured for 'git pull':

AddSomeHikes001 merges with remote AddSomeHikes001

gh-pages merges with remote gh-pages

Local refs configured for 'git push':

AddSomeHikes001 pushes to AddSomeHikes001 (up to date)

gh-pages pushes to gh-pages (local out of date)

tom (1576): git fetch

From https://github.com/krcowles/ktesa

\* branch AddSomeHikes001 -> FETCH\_HEAD

b11b652..9f445fd gh-pages -> origin/gh-pages

tom (1577): git remote show origin

\* remote origin

Fetch URL: https://github.com/krcowles/ktesa

Push URL: https://github.com/krcowles/ktesa

HEAD branch: master

Remote branch:

gh-pages tracked

Local branches configured for 'git pull':

AddSomeHikes001 merges with remote AddSomeHikes001

gh-pages merges with remote gh-pages

Local refs configured for 'git push':

AddSomeHikes001 pushes to AddSomeHikes001 (up to date)

gh-pages pushes to gh-pages (local out of date)

tom (1578): git status

On branch AddSomeHikes001

nothing to commit, working directory clean

tom (1579): git checkout gh-pages

Switched to branch 'gh-pages'

Your branch is behind 'origin/gh-pages' by 1 commit, and can be fast-forwarded.

(use "git pull" to update your local branch)

tom (1580): git diff

tom (1583): git log --oneline --decorate

b11b652 (HEAD, gh-pages) update with main page sortable table

194a748 removed flat structured files

0801769 incorporated complete directory structure

816db12 Added files via upload

d64cf98 Initial commit

tom (1589): git log --oneline --decorate --graph --all

\* 9f445fd (origin/gh-pages) update directory structure with gpsv & maps

| \* 0cc6394 (AddSomeHikes001) Add map as temporary Info Web Page

| \* ed91ec8 Add one hike to table

|/

\* b11b652 (HEAD, gh-pages) update with main page sortable table

\* 194a748 removed flat structured files

\* 0801769 incorporated complete directory structure

\* 816db12 Added files via upload

\* d64cf98 Initial commit

**20160713 – notes on creating a new branch with content Ken had sent me by email in a zip file. Essentials in green**

Note: Just before this I used the web interface to create branch Debug20160713 off of gh-pages

tom (2005): git status

On branch gh-pages

Your branch is up-to-date with 'origin/gh-pages'.

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.html.dbase

mergemessage.txt

pages/AlcoveHouse\_FromEmail.html

pages/diablo.html

pages/tmp.txt

scripts/wpages\_old.js

source/

tools/git\_Process.txt

nothing added to commit but untracked files present (use "git add" to track)

tom (2006): git branch

AddSomeHikes001

Dbug4PagesDir

Debug3rdTry

\* gh-pages

origin/Dbug4PagesDir

tom (2009): git branch Debug20160713

tom (2010): git checkout Debug20160713

Switched to branch 'Debug20160713'

tom (2011): git pull

Already up-to-date.

Note: I used the GUI to copy Ken's files from the zip file he had emailed me. See files highlighted.

tom (2012): git status

On branch Debug20160713

Untracked files:

(use "git add <file>..." to include in what will be committed)

BlueDot.png

MarkerArray.js

TH\_JSON.js

hikeMapTrial.css

hikeMapTrial.html

hikeMapTrial.js

index.html.dbase

mergemessage.txt

pages/AlcoveHouse\_FromEmail.html

pages/diablo.html

pages/tmp.txt

scripts/wpages\_old.js

source/

tools/git\_Process.txt

nothing added to commit but untracked files present (use "git add" to track)

tom (2017): git add -n BlueDot.png MarkerArray.js TH\_JSON.js hikeMapTrial.css hikeMapTrial.html hikeMapTrial.js

add 'BlueDot.png'

add 'MarkerArray.js'

add 'TH\_JSON.js'

add 'hikeMapTrial.css'

add 'hikeMapTrial.html'

add 'hikeMapTrial.js'

tom (2020): git commit -m "New map trial, also experiment using branch"

[Debug20160713 1a7867c] New map trial, also experiment using branch

6 files changed, 466 insertions(+)

create mode 100644 BlueDot.png

create mode 100644 MarkerArray.js

create mode 100644 TH\_JSON.js

create mode 100644 hikeMapTrial.css

create mode 100644 hikeMapTrial.html

create mode 100644 hikeMapTrial.js

tom (2022): git push --set-upstream origin Debug20160713

Username for 'https://github.com': tjsandberg

Password for 'https://tjsandberg@github.com':

Counting objects: 12, done.

Delta compression using up to 2 threads.

Compressing objects: 100% (8/8), done.

Writing objects: 100% (8/8), 6.86 KiB | 0 bytes/s, done.

Total 8 (delta 2), reused 0 (delta 0)

To https://github.com/krcowles/ktesa

76e2819..1a7867c Debug20160713 -> Debug20160713

Branch Debug20160713 set up to track remote branch Debug20160713 from origin.

tom (2023): git remote show origin

\* remote origin

Fetch URL: https://github.com/krcowles/ktesa

Push URL: https://github.com/krcowles/ktesa

HEAD branch: master

Remote branch:

gh-pages tracked

Local branches configured for 'git pull':

AddSomeHikes001 merges with remote AddSomeHikes001

Debug20160713 merges with remote Debug20160713

gh-pages merges with remote gh-pages

Local refs configured for 'git push':

AddSomeHikes001 pushes to AddSomeHikes001 (up to date)

Dbug4PagesDir pushes to Dbug4PagesDir (local out of date)

Debug20160713 pushes to Debug20160713 (up to date)

gh-pages pushes to gh-pages (up to date)

**20160713 Essentials:**

Note: Just before this I used the web interface to create branch Debug20160713 off of gh-pages

tom (2009): git branch Debug20160713

tom (2010): git checkout Debug20160713

Note: I used the GUI to copy Ken's files from the zip file he had emailed me.

tom (2017): git add -n BlueDot.png MarkerArray.js TH\_JSON.js hikeMapTrial.css hikeMapTrial.html hikeMapTrial.js

tom (2020): git commit -m "New map trial, also experiment using branch"

tom (2022): git push --set-upstream origin Debug20160713

**If we want to continue with the above trial – which really is too much of a hack but we could take it to the next step...**

// Ken should do:

git fetch origin // fetch latest from remote

git checkout -b Debug20160713 origin/Debug20160713 // create and checkout a local version of the branch

//The above did not work. The following did work.

git checkout -b Debug20160713

git pull origin Debug20160713

// do some testing, make some changes and push and continue process until done)

**20160714 More correct branch process:**

// Ken (person making change) should do:

git checkout gh-pages // check out gh-pages (if necessary, probably not necessary)

// make sure any unfinished work is committed?

git branch PageTrial1 // Create new branch called PageTrial1 in local repository

git checkout PageTrial1 // checkout the new branch so you can work on it

// shortcut for the above 2 commands: git checkout -b PageTrial1

// make changes in local repository (edit files, add/delete files/directories, do some testing, etc.)

git add <files> // stage the files you want to committed

git commit -m “comments” // commit changes to local directory

git push origin PageTrial1 // push changes to remote, creating remote branch in the process

// or maybe it should be: git push --set-upstream origin PageTrial1

// Tom (person reviewing change) should do:

git fetch origin // fetch latest from remote

git checkout -b PageTrial1 origin/PageTrial1 // create and checkout a local version of the branch

// shortcut for the above command: git checkout –-track origin/PageTrial1

// even shorter cut: git checkout PageTrial1

// do some testing, tell Ken it looks good (or make some changes and push and continue process until done)

// Note that as long as Tom and Ken are working on the PageTrial1 branch, they can push and pull as they have been doing on gh-pages

// Once it is agreed that the changes are ready to go live on gh-pages, Ken (person making change) should do:

git checkout gh-pages // check out gh-pages (if necessary)

git fetch origin // fetch latest from remote

git merge PageTrial1 // merge the changes into gh-pages branch on local

// test to make sure it all works

git push // push merged changes to gh-pages branch on remote

**20160717 Less correct branch process, but the only one we've been able to make work so far:**

// Ken (person making change) should do:

git checkout gh-pages // check out gh-pages (if necessary, probably not necessary)

// make sure any unfinished work is committed

git branch PageTrial1 // Create new branch called PageTrial1 in local repository

git checkout PageTrial1 // checkout the new branch so you can work on it

// shortcut for the above 2 commands: git checkout -b PageTrial1

// make changes in local repository (edit files, add/delete files/directories, do some testing, etc.)

git add <files> // stage the files you want to committed

git commit -m “comments” // commit changes to local directory

git push --set-upstream origin PageTrial1 // push changes to remote, creating remote branch in the process

// Tom (person reviewing change) should do:

git checkout -b PageTrial1 // create and checkout a local branch using the same name

git pull origin PageTrial1 // retrieve latest from remote

// do some testing, tell Ken it looks good (or make some changes and push and continue process until done)

// Note that as long as Tom and Ken are working on the PageTrial1 branch, they can push and pull almost as they have been doing on gh-pages. The difference is that without a tracking branch, the longer commands below must be used:

git push origin PageTrial1 // send latest to remote

git pull origin PageTrial1 // retrieve latest from remote

// Once it is agreed that the changes are ready to go live on gh-pages, Ken (person making change) should do:

git checkout gh-pages // check out gh-pages (if necessary)

git fetch origin // fetch latest from remote

git merge PageTrial1 // merge the changes into gh-pages branch on local

// test to make sure it all works

git push // push merged changes to gh-pages branch on remote

**20161124 New branch process after figuring out how fetch works**

// Tom (person adding branch) should do:

git fetch origin gh-pages # make sure remote-tracking branch is up to date with remote

git checkout gh-pages # check out gh-pages and…

git merge # make sure it is up to date with remote tracking branch

git branch Test1 # Create new branch called PageTrial1 in local repository

git checkout Test1 # checkout the new branch so you can work on it

# shortcut for the above 2 commands: git checkout -b PageTrial1

# make changes in local repository (edit files, add/delete files/directories, do some testing, etc.)

git add <files> # stage the files you want to committed

git commit -m “comments” # commit changes to local directory

git push --set-upstream origin Test1 # push changes to remote, create Test1 branch on remote, and

# add upstream (tracking) reference to local config

git branch -v -v | grep Test1 # note that remote tracking branch and upstream config were created

// Ken (person reviewing branch) should do:

git fetch origin Test1 # fetch the branch and associated objects, and create

# remote-tracking branch origin/Test1 in local repository

git branch –-track Test1 origin/Test1 # create local branch Test1 starting at the same commit as

# Test1 on the remote, and add upstream (tracking) reference

# to local config.

git checkout Test1 # check out Test1

# shortcut for the previous two commands: git checkout –-track origin/Test1

# even shorter cut: git checkout Test1

git branch -v -v | grep Test1 # note that remote tracking branch and upstream config were created

# To Do: Add process for fetch/merge on existing branches