



# The Git Workshop

Getting Started

Download Git Bash  
Windows users:

<https://git-scm.com/downloads>



Make a GitHub Account

Awesome Free Stuff:

<https://education.github.com/pack>



# Assumed Knowledge

## shell commands

cd \_\_\_\_ (change directory)

ls (list everything in the directory)

## terminology

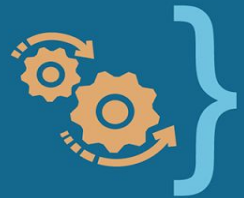
repo: short for repository.





# Topics Being Covered

1. Creating a repository
2. Explaining the git commands used
3. Adding a .gitignore file
4. Branching in git
5. Merging branches in git
6. Common tips and tricks



# Configuring your credentials

```
git config --global user.name "YOUR NAME"
```

```
git config --global user.email "YOUR EMAIL ADDRESS"
```



# Starting Our First Repository





**Pro tip:** updating your profile with your name, location, and a profile picture helps other GitHub users get to know you.

 [Edit profile](#)

 [Contributions](#)

 [Repositories](#)

 [Public activity](#)

Find a repository...

[Search](#)

[All](#) [Public](#) [Private](#) [Sources](#) [Forks](#) [Mirrors](#)

 [New](#)



## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



lolpuddle ▾

/

Repository name

git-workshop-test



Great repository names are short and memorable. Need inspiration? How about **turbulent-octo-giggle**.

Description (optional)



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.



**Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾



Add a license: **None** ▾



Create repository




## Quick setup — if you've done this kind of thing before

 **Set up in Desktop** or **HTTPS** **SSH**  

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


## ...or create a new repository on the command line

```
echo "# git-workshop-test" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/lofpuddle/git-workshop-test.git
git push -u origin master
```



## ...or push an existing repository from the command line


```
git remote add origin https://github.com/lofpuddle/git-workshop-test.git
git push -u origin master
```



## ...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

**Import code**

 **ProTip!** Use the URL for this page when adding GitHub as a remote.



# Steps Involved

1. Go to the directory you wish to create the repository from.
2. "git init"
3. Add a file to the directory if it is empty  
(Preferably a README.md)
4. "git status"
5. "git add \_\_\_\_ (name of files)" or "git add ." to add all files
6. "git commit -m "first commit""
7. "git remote add origin \_\_\_\_\_ (link to your repo)"
8. "git push -u origin master"



# README.md

File is automatically read and used on the page of your repository.

Good example of a README.md  
<https://github.com/robbyrussell/oh-my-zsh/>



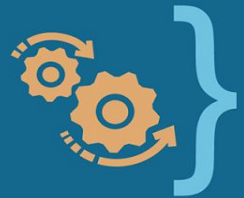
git init

initializes a repository



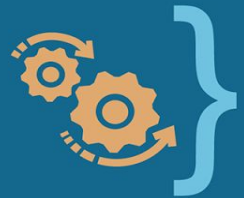
# git status

compares the state of the machine's  
current data with the data from the  
last time it was pulled



git add

Selecting and “adding” files you wish to  
make changes to





# git commit -m "whatever"

\*Include the quotes

taking a "snapshot" of the information on your machine  
and associating a "message" with this snapshot.

If changes have been made to a file, they must be added  
before committed.



git push

uploading the “snapshot” you created  
to the repository



git pull

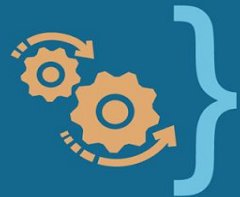
Downloading the data stored on the  
repository



git clone “whatever url”

(no quotes)

Creating a copy of the repository from  
nothing



# git checkout “file or branch”

do not include quotes

Used to say “I don’t want the changes I  
made”

or

Used to switch branches



# git branch "name of branch"

(no quotes)

## Creating a new branch in git



# List of commands covered

git init

git status

git add

git commit -m "some message"

git push

git pull

git clone "some url" (no quotes)

git checkout "some file or some branch" (no quotes)

git branch "name of new branch" (no quotes)

git merge



# Adding a .gitignore file

common .gitignore contents

<https://github.com/github/gitignore>





# Branching + Merging



To create a branch

type

“git branch “some\_name” (no quotes)”



# Working on a new branch

git checkout "branch name" (no quotes)

do whatever you want

git commit -m "whatever"

git checkout master

git merge (name of branch you want to merge)



Why is merging necessary?



# Tips & Tricks



# Getting a clean pull

```
git fetch --all
```

```
git reset --hard origin/master
```

# Getting a clean pull of a branch

```
git reset --hard origin/"name of branch"  
(no quotes)
```



^ ^  
—

