

Part I

Let's start by taking a bit of time to practice the git workflow below. It is so valuable to just practice this workflow a couple times, since you will most likely doing it professionally as well as in your individual projects and open source contributions. Here are some things to do.

1. Create a local repository and add and commit some files
2. Create a remote repository and push your code from the local repo to the remote
3. Create a local branch and add and commit some files
4. Push that local branch to GitHub to create a remote branch

To make a new directory,

```
cd Desktop  
mkdir mem-gen  
cd mem-gen
```

1. Running git init initializes an empty Git repo

```
└─(~/Desktop/mem-game)────────  
└─(17:06:29)→ git init  
Initialized empty Git repository in /Users/berryman/Desktop/mem-game/.git/
```

Running git status confirms that no files have been staged

Running git add FILENAME adds each file individually to the staging area. Running git add . would add every file at once.

```
└─(~/Desktop/mem-game)────────  
└─(17:09:13 on master + ★)→ git add index.html  
└─(~/Desktop/mem-game)────────  
└─(17:09:18 on master + ★)→ git add myScript.js  
└─(~/Desktop/mem-game)────────  
└─(17:09:21 on master + ★)→ git add style.css
```

Running git status shows all files in the staging area.
(See next page)

```
└(~/Desktop/mem-game)─  
└(17:10:15 on master + ★)→ git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
    new file:    gifs/1.gif  
    new file:    gifs/10.gif  
    new file:    gifs/11.gif  
    new file:    gifs/12.gif  
    new file:    gifs/2.gif  
    new file:    gifs/3.gif
```

Running git commit -m “Initial commit” makes a commit

```
└(~/Desktop/mem-game)─  
└(17:12:11 on master + ★)→ git commit -m "Initial commit"  
[master (root-commit) 34a9fe9] Initial commit  
  15 files changed, 578 insertions(+)
```

Running git status again shows that all of the previous files have been committed & the only changes are .DS_Store, which I intentionally left out. In next steps, I added it to .gitignore.

```
└(~/Desktop/mem-game)─  
└(17:12:19 on master ★)→ git status  
On branch master  
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
  .DS_Store  
  
nothing added to commit but untracked files present (use "git add" to track)
```

2. On GitHub, I created a new folder named Memory-Game-Updated

 k-berryman / **Memory-Game-Updated**

The following pushed code from my local repo to my remote repo

```
└─(~/Desktop/mem-game)─
└─(17:20:55 on master *)→ git remote add origin https://github.com/k-berryman/Memory-Game-Updated
fatal: remote origin already exists.
└─(~/Desktop/mem-game)─
└─(17:23:58 on master *)→ git remote remove origin
└─(~/Desktop/mem-game)─
└─(17:24:26 on master *)→ git remote add origin https://github.com/k-berryman/Memory-Game-Updated
└─(~/Desktop/mem-game)─
└─(17:24:28 on master *)→ git push -u origin master
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 8 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (18/18), 29.94 MiB | 894.00 KiB/s, done.
Total 18 (delta 0), reused 0 (delta 0)
```

 k-berryman	Create README.md	✓ 6a4a5a2 1 hour ago	⌚ 2 commits
📁 gifs	Initial commit	1 hour ago	
📄 README.md	Create README.md	1 hour ago	
📄 index.html	Initial commit	1 hour ago	
📄 myScript.js	Initial commit	1 hour ago	
📄 style.css	Initial commit	1 hour ago	

3. I made a new branch called NoCats

```
└─(~/Desktop/mem-game)─
└─(17:25:58 on master *)→ git branch -a
* master
  remotes/origin/master
└─(~/Desktop/mem-game)─
└─(17:26:16 on master *)→ git checkout -b NoCats
Switched to a new branch 'NoCats'
```

I committed some files to the NoCats branch

```
└(~/Desktop/mem-game)─  
└(17:36:28 on NoCats +)→ git status  
On branch NoCats  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    new file:   .gitignore  
    modified:   gifs/1.gif  
    modified:   gifs/2.gif  
    modified:   gifs/4.gif  
    modified:   gifs/5.gif  
    modified:   gifs/6.gif
```

```
└(~/Desktop/mem-game)─  
└(17:36:32 on NoCats +)→ git commit -m "Replace all cat gifs with dog gifs"  
[NoCats 0d6b084] Replace all cat gifs with dog gifs  
 6 files changed, 1 insertion(+)  
  create mode 100644 .gitignore  
  rewrite gifs/5.gif (83%)
```

4. I pushed this local NoCats branch to GitHub to create a new remote branch

```
└(~/Desktop/mem-game)─  
└(17:36:50 on NoCats)→ git push -u origin NoCats  
Enumerating objects: 27, done.  
Counting objects: 100% (27/27), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (26/26), done.  
Writing objects: 100% (27/27), 80.88 MiB | 739.00 KiB/s, done.  
Total 27 (delta 1), reused 0 (delta 0)  
remote: Resolving deltas: 100% (1/1), done.  
remote:  
remote: Create a pull request for 'NoCats' on GitHub by visiting:  
remote:   https://github.com/k-berryman/Memory-Game-Updated/pull/new/NoCats  
remote:  
To https://github.com/k-berryman/Memory-Game-Updated  
 * [new branch]      NoCats -> NoCats  
Branch 'NoCats' set up to track remote branch 'NoCats' from 'origin'.
```

[k-berryman / Memory-Game-Updated](#) Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

NoCats had recent pushes 28 minutes ago

Switch branches or tags

master 2 branches 0 tags

Compare & pull request

Go to file Add file Code

Switch branches/tags

Find or create a branch...

Branches Tags

✓ master default

NoCats

View all branches

style.css

Initial commit 1 hour ago

Create README.md 1 hour ago

Initial commit 1 hour ago

Part II - Put your memory game on GitHub pages

Using GitHub pages, deploy your memory game from the previous unit so that you can share them with anyone!

<https://k-berryman.github.io/Memory-Game-Updated/>

