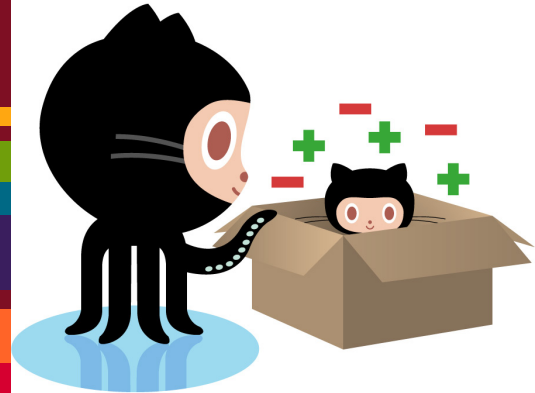


# Week 5

## Git Tools + Resolving Conflicts



# Last Week

- git branch
- git merge
  - 3 way and fast forward

# Review

- Name an advantage of using branching in your workflow
- Difference between three way merge and fast forward merge

# Common Homework Issues

- Some people did not actually resolve merge conflicts
- git will usually tell you when this happens and will sometimes tell you how to resolve



@@ -12,7 +12,11 @@

12	12	* Date: 2014-05-01T17:11Z
13	13	*/
14	14	

+&lt;&lt;&lt;&lt;&lt;&lt; HEAD

15	16	(function( globalization, factory ) {
	17	+=====
	18	+(function( local, factory ) {
	19	+>>>>>> origin/newFeature

16	20	
17	21	if ( typeof module!hihi "object" && typeof module.exports!hihi "object" ) {
18	22	// For CommonJS and CommonJS-like environments where a proper window is present,



@@ -31,7 +35,11 @@

	35	return factory( w )!hi
31	36	hi
32	36	}!hi
33	37	} else {

+&lt;&lt;&lt;&lt;&lt;&lt; HEAD

34	39	factory( globalization )!hi
	40	+=====
	41	+ factory( local )!hi
	42	+>>>>>> origin/newFeature

35	43	}
36	44	
37	45	// Pass this if window is not defined yet



# Git Tools

- [https://git.wiki.kernel.org/index.php/Interfaces,\\_frontends,\\_and\\_tools](https://git.wiki.kernel.org/index.php/Interfaces,_frontends,_and_tools)

# 2-way

LOCAL - current file on the branch

REMOTE - file to be merged into LOCAL

Easier to use

# 3-way merge

LOCAL - current file on the branch

REMOTE - file to be merged into LOCAL

BASE - common ancestor

RESULT - result of the merge



# 3-way vs 2-way merge algorithms

<http://codicesoftware.blogspot.com/2010/11/live-to-merge-merge-to-live.html>

# merge tools

Windows: P4merge

Mac: FileMerge (bundled in command line tools)

Linux: kdiff3 or meld

# Setting a mergetool

```
git config --global merge.tool p4merge  
git config --global mergetool.path <path>
```

# git mergetool

```
Alvins-MacBook-Pro:HW3 Alvin$ git merge newFeature
Auto-merging jquery.js
CONFLICT (content): Merge conflict in jquery.js
Automatic merge failed; fix conflicts and then commit the result.
Alvins-MacBook-Pro:HW3 Alvin$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse ecmerge p4merge araxis bc3 codecompare emerge vimdiff
Merging:
jquery.js

Normal merge conflict for 'jquery.js':
{local}: modified file
{remote}: modified file
Hit return to start merge resolution tool (opendiff): █
```

jquery.js.LOCAL.4456.js - /Users/Alvin/HW3

jquery.js

jquery.js.REMOTE.4456.js - /Users/Alvin/HW3

isArraylike()

isArraylike()

```
* http://jquery.com/
* Includes Sizzle.js
* http://sizzlejs.com/
*
* Copyright 2005, 2014 jQuery Foundation, Inc. and other contributors
* Released under the MIT license
* http://jquery.org/license
*
* Date: 2014-05-01T17:11Z
*/

(function( globalization, factory ) {

    if ( typeof module!=="object" && typeof module.exports!=="object" ) {
        // For CommonJS and CommonJS-like environments where a proper window is present,
        // execute the factory and get jQuery
        // For environments that do not inherently possess a window with a document
        // (such as Node.js), expose a jQuery-making factory as module.exports
        // This accretuates the need for the creation of a real window
        // e.g. var jQuery = require("jquery")(window);
        // See ticket #14549 for more info
        module.exports = global.document ?
            factory( global, true ) :
            function( w ) {
                if ( !w.document ) {
                    throw new Error( "jQuery requires a window with a document" );
                }
                return factory( w );
            }
    } else {
        factory( globalization );
    }

    // Pass this if window is not defined yet
    }(typeof window!=="undefined" ? window : this, function( window, noGlobal ) {

        isArraylike()
```

```
* Includes Sizzle.js
* http://sizzlejs.com/
*
* Copyright 2005, 2014 jQuery Foundation, Inc. and other contributors
* Released under the MIT license
* http://jquery.org/license
*
* Date: 2014-05-01T17:11Z
*/

(function( local, factory ) {

    if ( typeof module!=="object" && typeof module.exports!=="object" ) {
        // For CommonJS and CommonJS-like environments where a proper window is present,
        // execute the factory and get jQuery
        // For environments that do not inherently possess a window with a document
        // (such as Node.js), expose a jQuery-making factory as module.exports
        // This accretuates the need for the creation of a real window
        // e.g. var jQuery = require("jquery")(window);
        // See ticket #14549 for more info
        module.exports = global.document ?
            factory( global, true ) :
            function( w ) {
                if ( !w.document ) {
                    throw new Error( "jQuery requires a window with a document" );
                }
                return factory( w );
            }
    } else {
        factory( local );
    }

    // Pass this if window is not defined yet
    }(typeof window!=="undefined" ? window : this, function( window, noGlobal ) {

        isArraylike()
```

status: 2 differences (2 left, 2 right, 2 conflicts)

Actions

Choose left  
Choose right  
Choose both (left first)  
Choose both (right first)  
Choose neither

Actions



# The Annoyance of Typing

```
Alvins-MacBook-Pro:HW3 Alvin$ git push  
Username for 'https://github.com': dogfalo  
Password for 'https://dogfalo@github.com': █
```

# Setting up SSH on GitHub

# Git GUI

- simple graphical user interface for basic commands
- easy to use
- lacks more advanced commands or at the very least as complicated as command line



# GUI vs command line

## Pros

- easy to use for beginners
- looks pretty
- if inexperienced harder to make mistakes (can't -f)

## Cons:

- takes time to learn
- adds a layer of complexity if you already know the commands
- functionality is limited
- requires more resources
- scripting is hard

# git clean

- removes untracked files from working tree
- useful if your repository has accumulated

# git gc

- cleans up the repo
- compresses revisions
- removes dangling branches and other unreachable objects

# git stash

- Allows you to save your work before it is “commit ready”
- modified tracked files and staged changes
- useful if you are switching branches with uncommitted work
- `git stash list` - shows stash stack
- `git stash apply`
- `git stash clear`

# Emojis

:neckbeard:

:smiling\_imp:

<http://www.emoji-cheat-sheet.com/>

# Next Week on Git

- Workflows Forking
- Pull Requests
- Rebasing

# Cygwin

- unix-like environment for windows