IO Lab:
More HTML/CSS
Version Control with Git

September 9, 2013

Info 290TA (Information Organization Lab)
Kate Rushton & Raymon Sutedjo-The

More HTML/CSS

Adding CSS

- Inline
- Embedded
- External

Adding CSS

Inline CSS

```
<button id="btn"
style="color:#222;">Save</button>
```

Embedded CSS

```
<head>
    <style>
        button { color: #222 }
    </style>
</head>
```

Adding CSS

External Stylesheet

```
<head>

…
<link rel="stylesheet“ href=“main.css"
/>
</head>
```

Browser CSS

exploring the impact of open data on public policy and democratic governance in the digital age.

September 18
Analytics 3.0: Big Data and Small Data in Big and Small Companies
Dean's Lecture with Tom Davenport of Babson College, co-founder of the International Institute for Analytics

Welcome, New Students!

I School Kicks Off the Year by Welcoming New Students
The four new doctoral students and 56 new MIMS students have a wide range of backgrounds, with previous degrees in cognitive science, studio art, biochemistry, physics, history, computer science, business administration, politics, sociology, psychology, literature, mathematics, graphic design, and more.

Audio & Video

Forbes Interviews Alum Paul Goodman (MIMS '12) on Social Entrepreneurship in Latin America

Breaking News

MIMS Students' Summer Internships
This week, the MIMS class of 2014 is fin summer internships to return to South Hal with seven about what they've been doing summer, how their first year's classes pre for their internships, and what they're pla their second year and beyond.

New Online Master's Degree to Train the Scientists of Tomorrow
Aspiring data scientists have a new oppo opening up to them as the UC Berkeley Sc Information launches the country's first fu Master of Information and Data Science (

Recall: CSS selectors

HTML
```
<button id="save-button" class="btn btn-blue">Save</button>
```

Output (No CSS)

\<tagname>

HTML
```
<button id="save-button" class="btn blue">Save</button>
```

CSS
```
button { margin-bottom: 5px }
```

.class

HTML
```
<button id="save-button" class="btn blue">Save</button>
```

CSS
```
.btn { padding: 5px 20px; }
.blue { background: #2e70db;}
.btn.blue { color: #ffffff;}
```

Output (With CSS)

#ID

HTML
```
<button id="save-button" class="btn blue">Save</button>
```

CSS
```
#save-button { box-shadow: 1px 1px 3px #999; }
```

Output (With CSS) Save

Descendant Selector

HTML
```
<div id="content">
   <button>Save</button>
</div>
```

CSS
```
#content button { … }
```

Other Selectors

CSS

```
#content > button { … }
```
applies immediate children only

```
#content * button { … }
```
all descendants except immediate children

```
#content + button { … }
```
adjacent sibling selector

Attribute Selector

HTML
```
<div id="content">
    <input type="text" name="user">
</div>
```

CSS
```
input[type="text"] { … }
```

Combination Selector

HTML
```
<div id="content">
<button class="btn">Save</button>
</div>
```

CSS
```
div#content button.btn { … }
```

"Cascading" Style Sheets

HTML
```
<div id="content">
<button id="mybutton" class="btn"
style="…">Save</button>
</div>
```

```
button { … }              #mybutton { … }
#content button{ …}       .btn  { … }
div button{ … }           div .button  { … }
```

"Cascading" Style Sheets

"Cascading" Style Sheets

General Rules and Guidelines
1. Inline styles override other styles
2. More specific selectors win
   id > class > element
3. All other things being equal, styles applied last will override earlier styles
4. Start with least specificity, add more as needed
5. Avoid !important

Version Control with Git

Why Use Version Control?

- Save your work
- Undo
- Safe experimentation
- Documentation
- Collaboration

**Abortion
(Wikipedia)**

Viégas, Wattenberg, and Dave

Types of Version Control

- Manual
- Local (rcs)
- Centralized (Subversion, Perforce, etc.)
- Distributed (Mercurial, Git)

Centralized VCS

- Version control managed by a central server
- Users check out files from the server, then check back in with changes
- Optionally can lock files
- Server keeps track of files, and revisions to those files over time

Centralized VCS

Source: ProGit
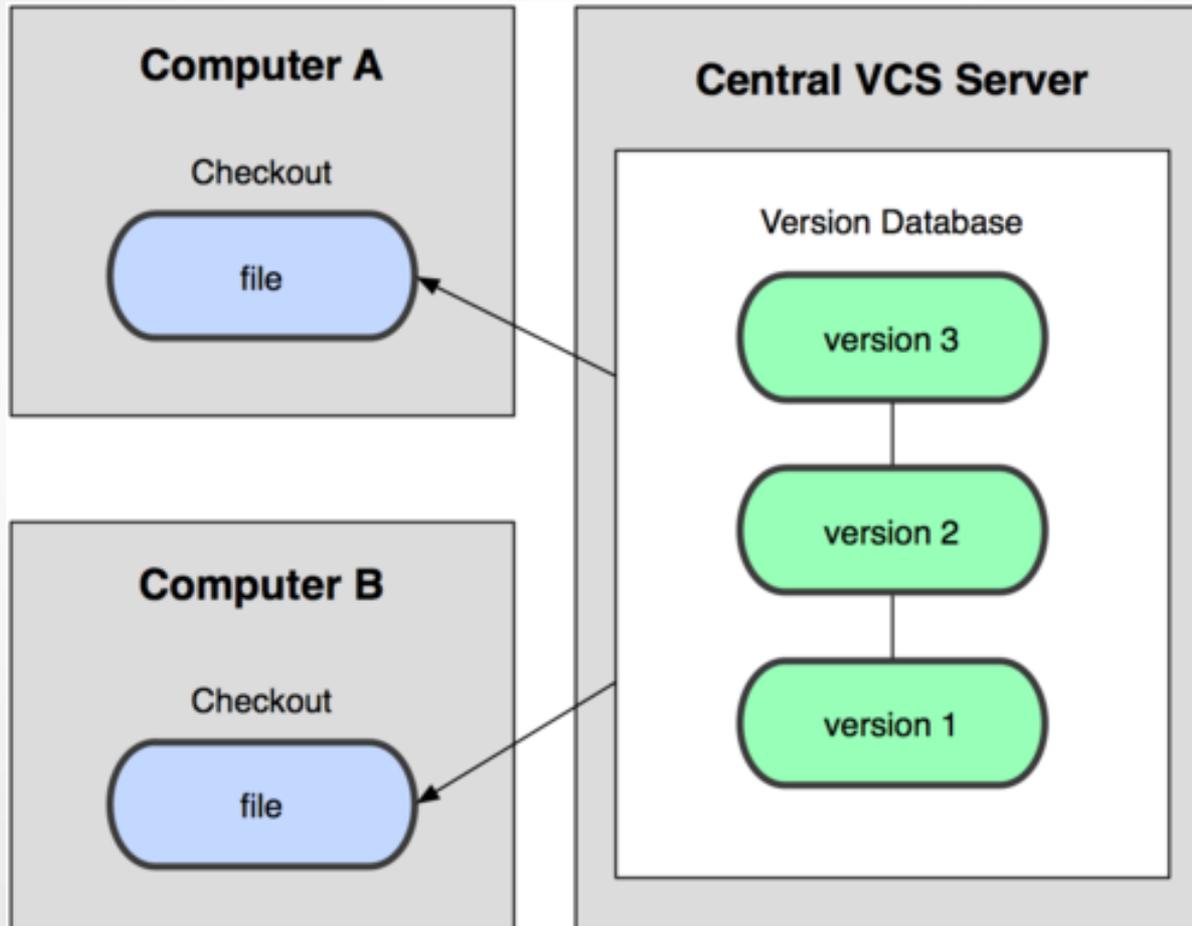
Centralized VCS
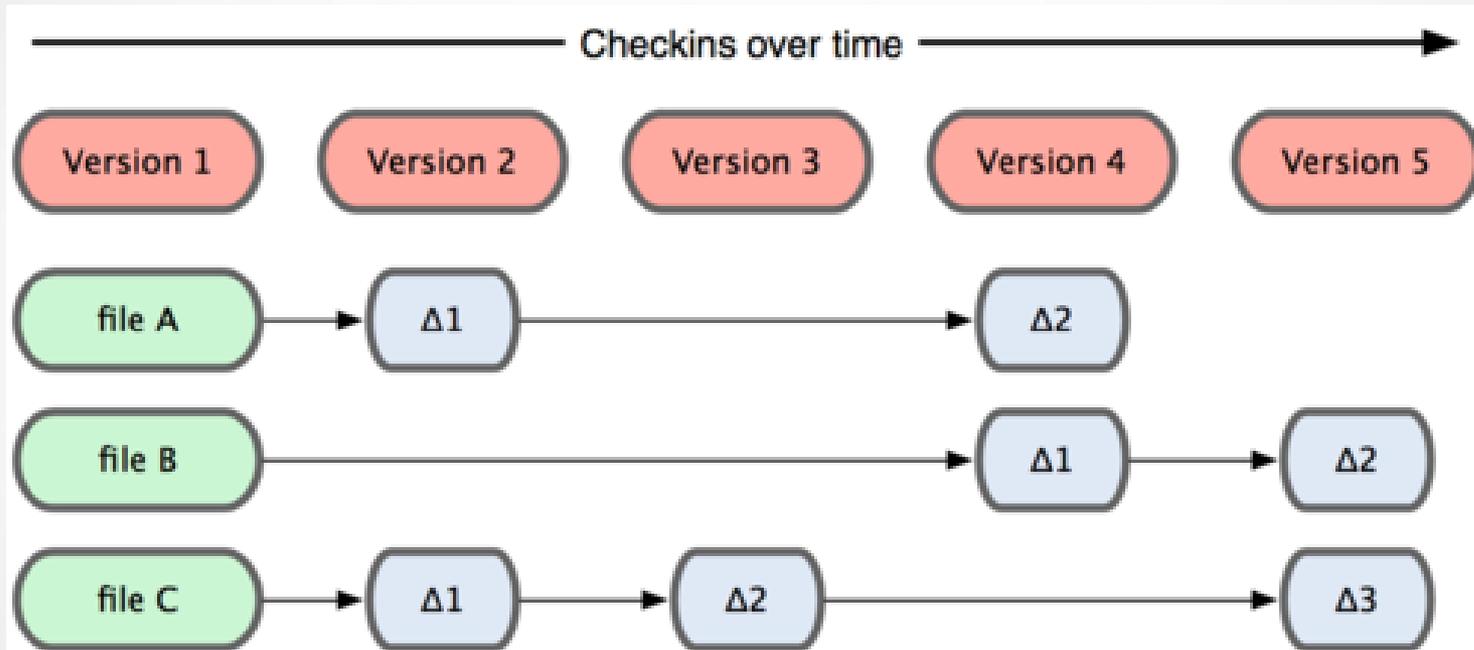
Source: ProGit

Distributed VCS

- Every client has a copy of the full repository
- Users commit files after making changes, which takes a snapshot of the state of the file system, noting which files changed

Distributed VCS
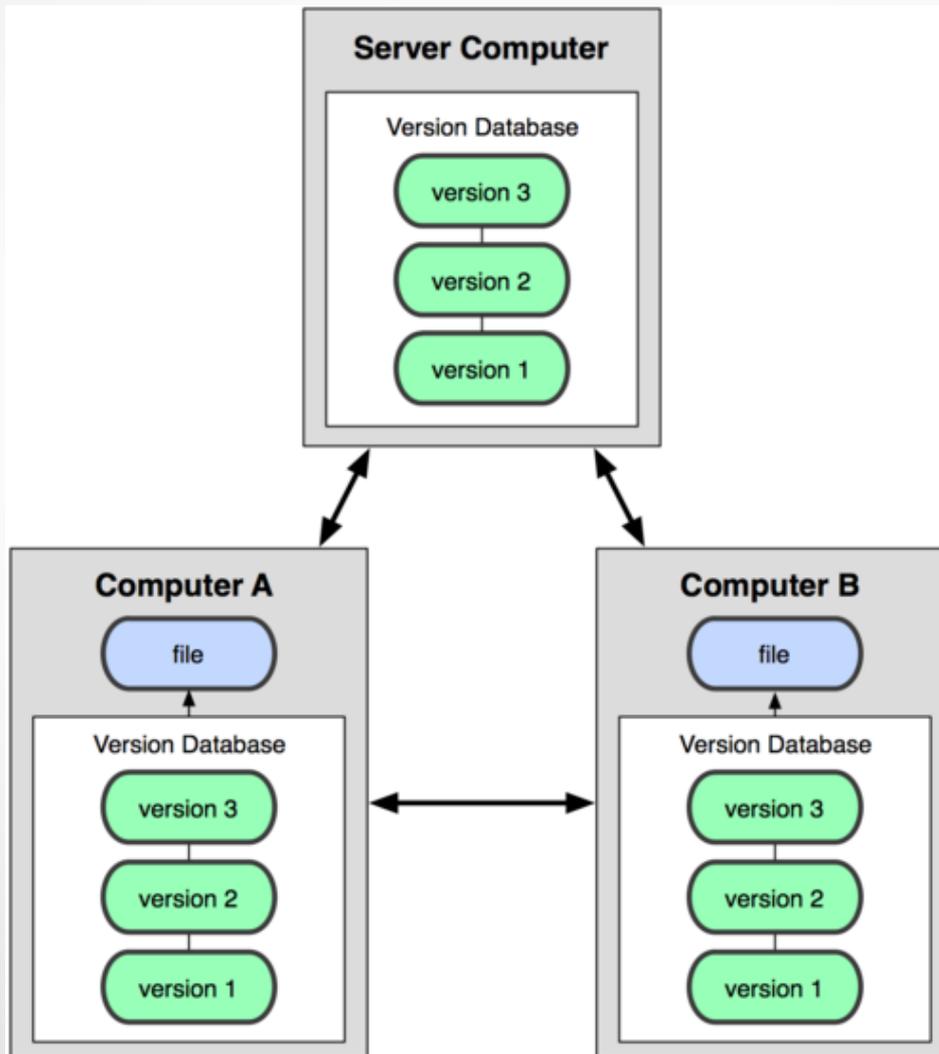
Source: ProGit

Distributed VCS

Source: ProGit
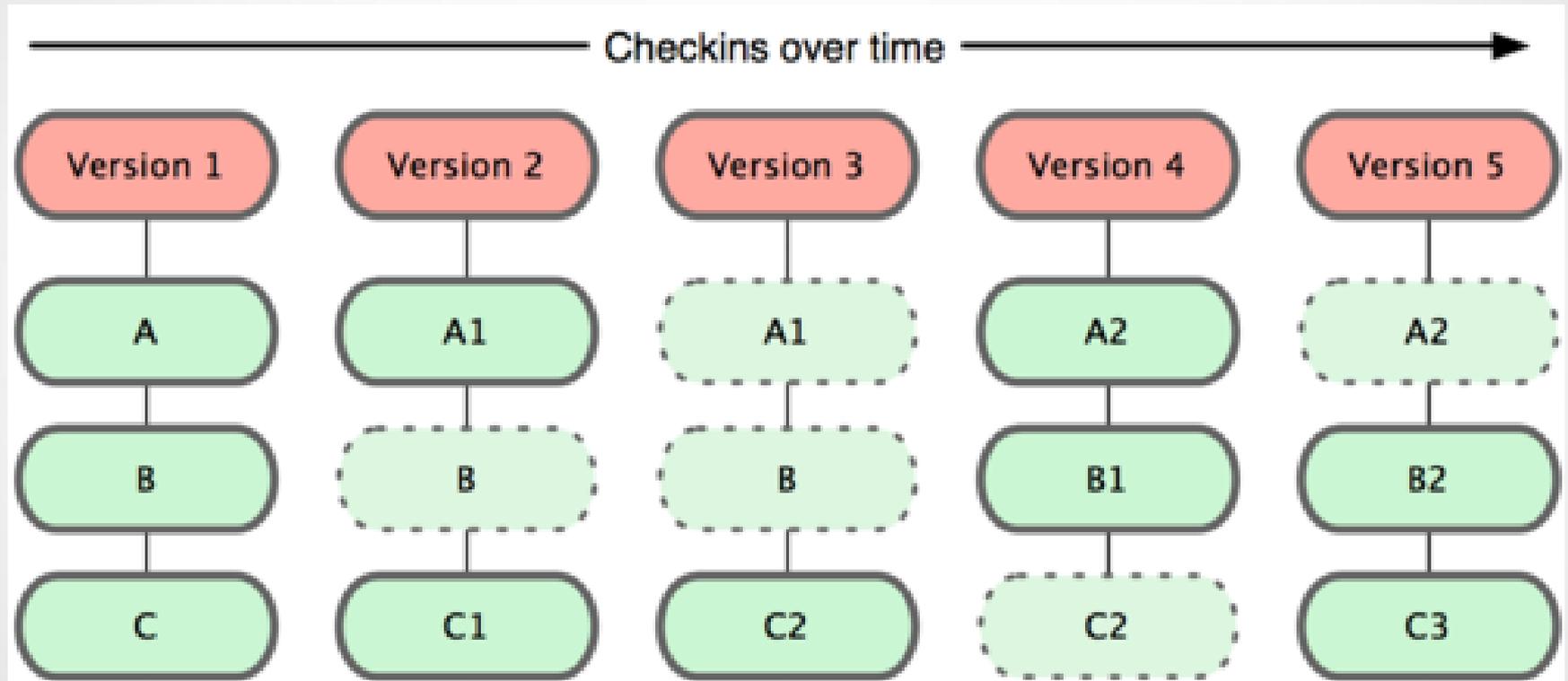
History of Git
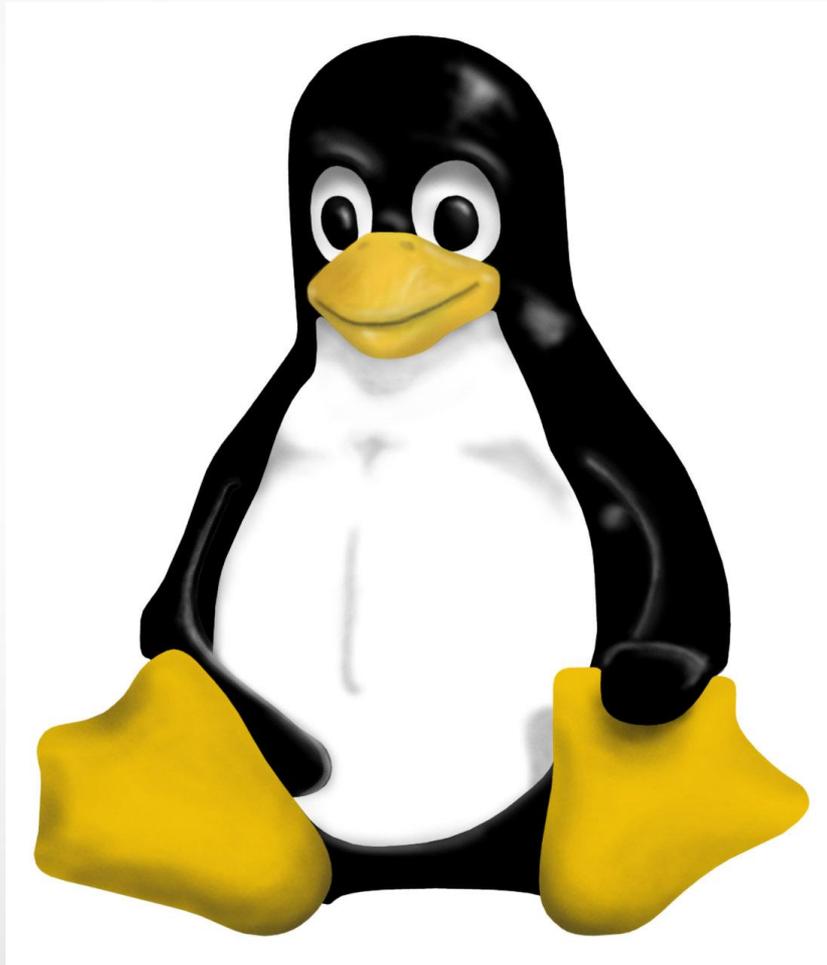
Git States

Files in a git repository can be in 1 of 3 states:

- Committed
- Modified
- Staged

Command Line Git

1. CD into your project directory
```
$ cd myproject
```

2. Initialize a new repository with **init**
```
$ git init
```

3. Edit files, then use **add** to stage
```
$ git add "file.txt"     ← add 1 file
$ git add -A             ← add all
```

Command Line Git

4. Commit the files with **commit** and include a message stating what changed
```
$ git commit -m "fixed bug in
file.txt"
```

Other Helpful Commands

`$ git status`
to see the current status of the repo

`$ git log`
to view commit history.

`$ git checkout <filename>`
removes any changes to unstaged files since the last commit

Other Helpful Commands

```
$ git reset HEAD <filename>
```
unstages a staged file, but keeps the changes made since the last commit

Git GUI - SmartGit

Git GUI - SmartGit

Modified files appear here
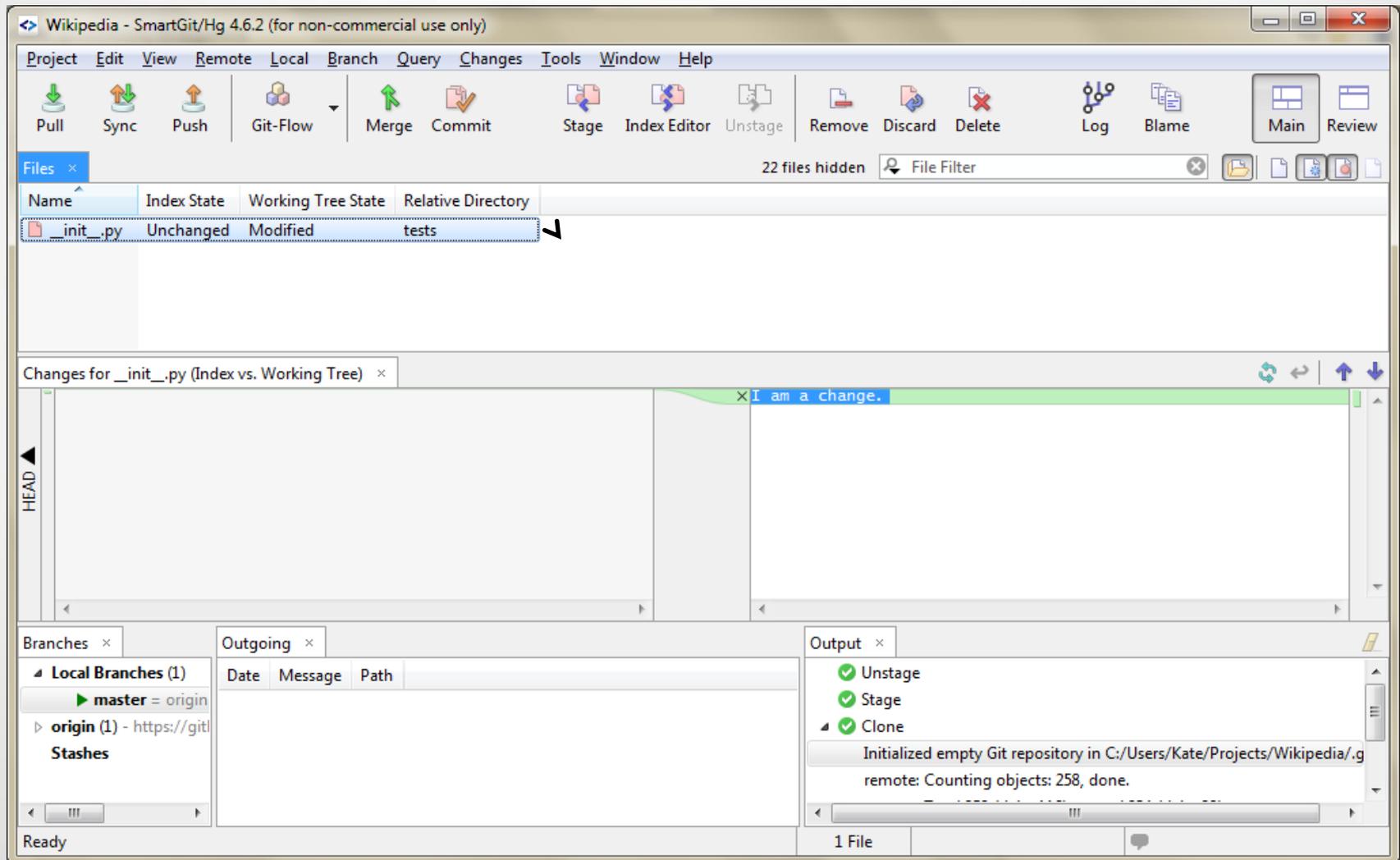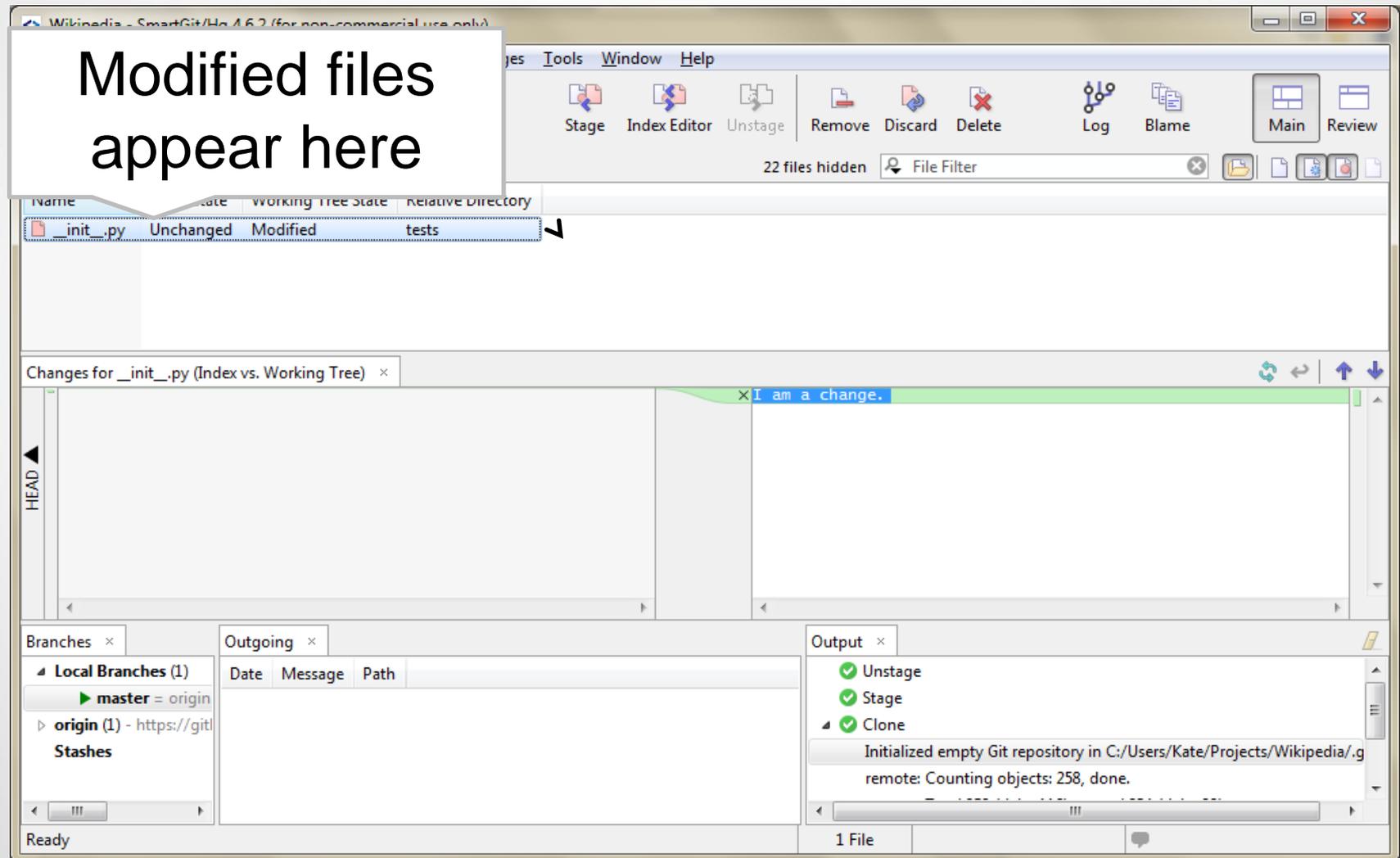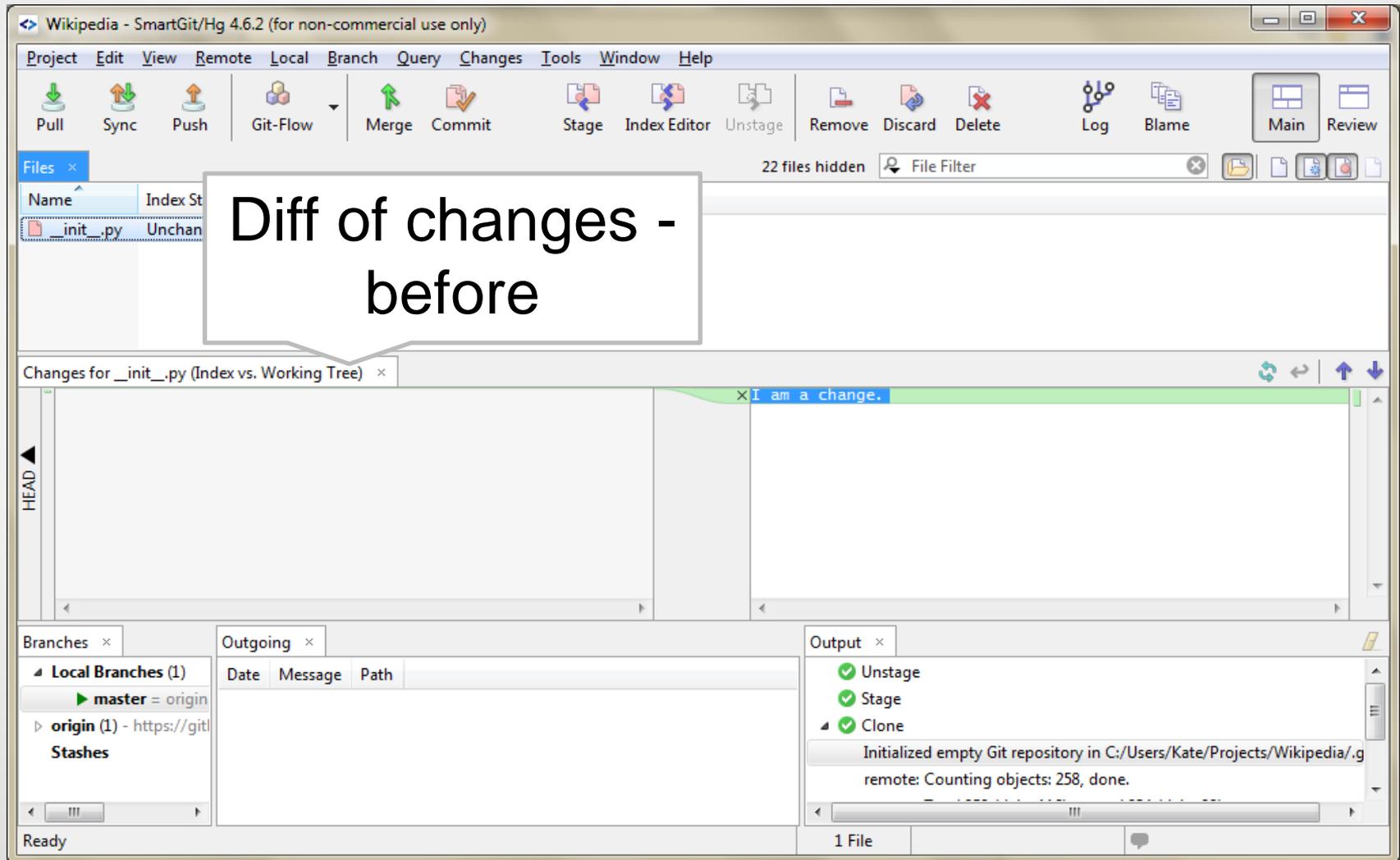
Git GUI - SmartGit
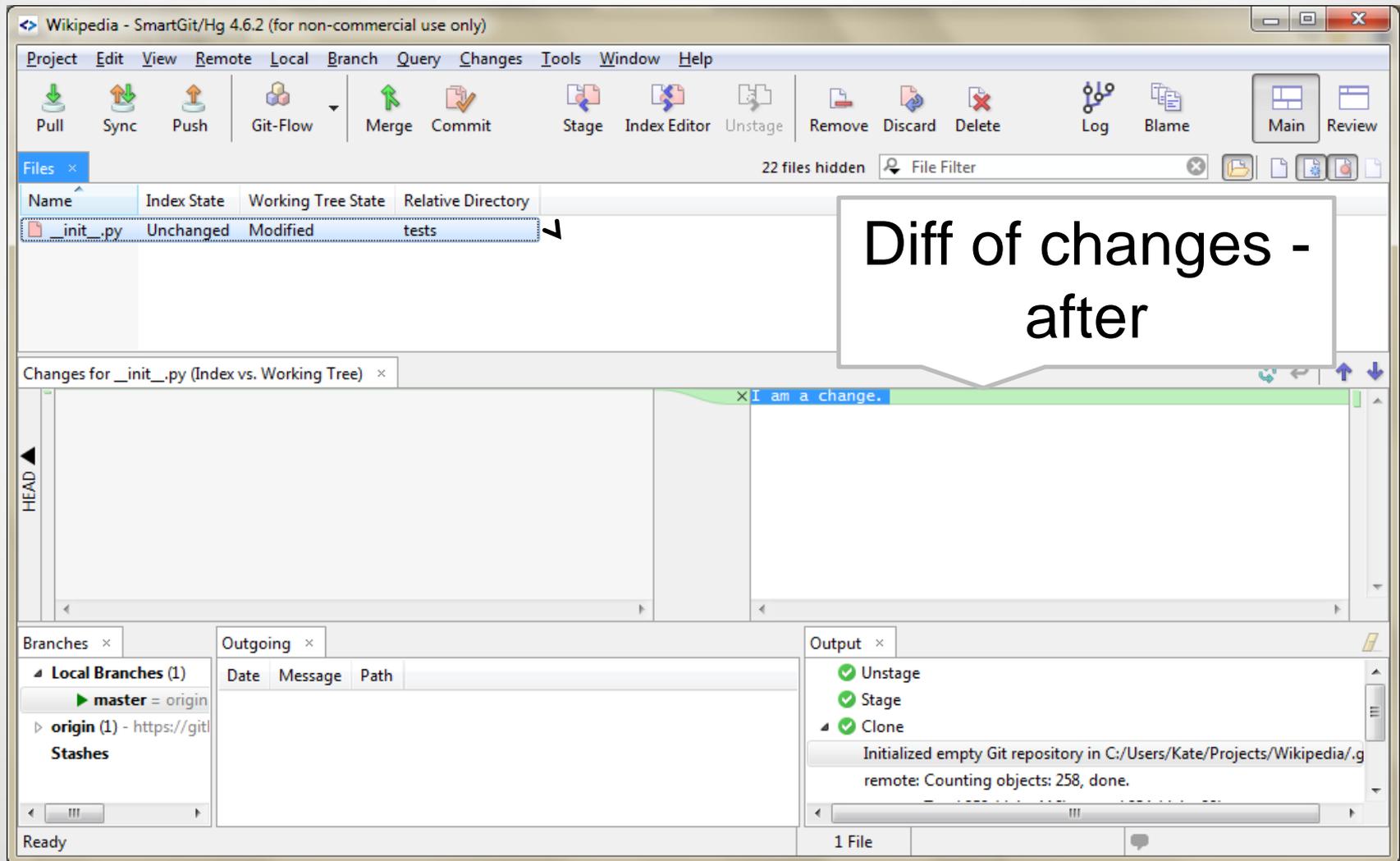
Git GUI - SmartGit
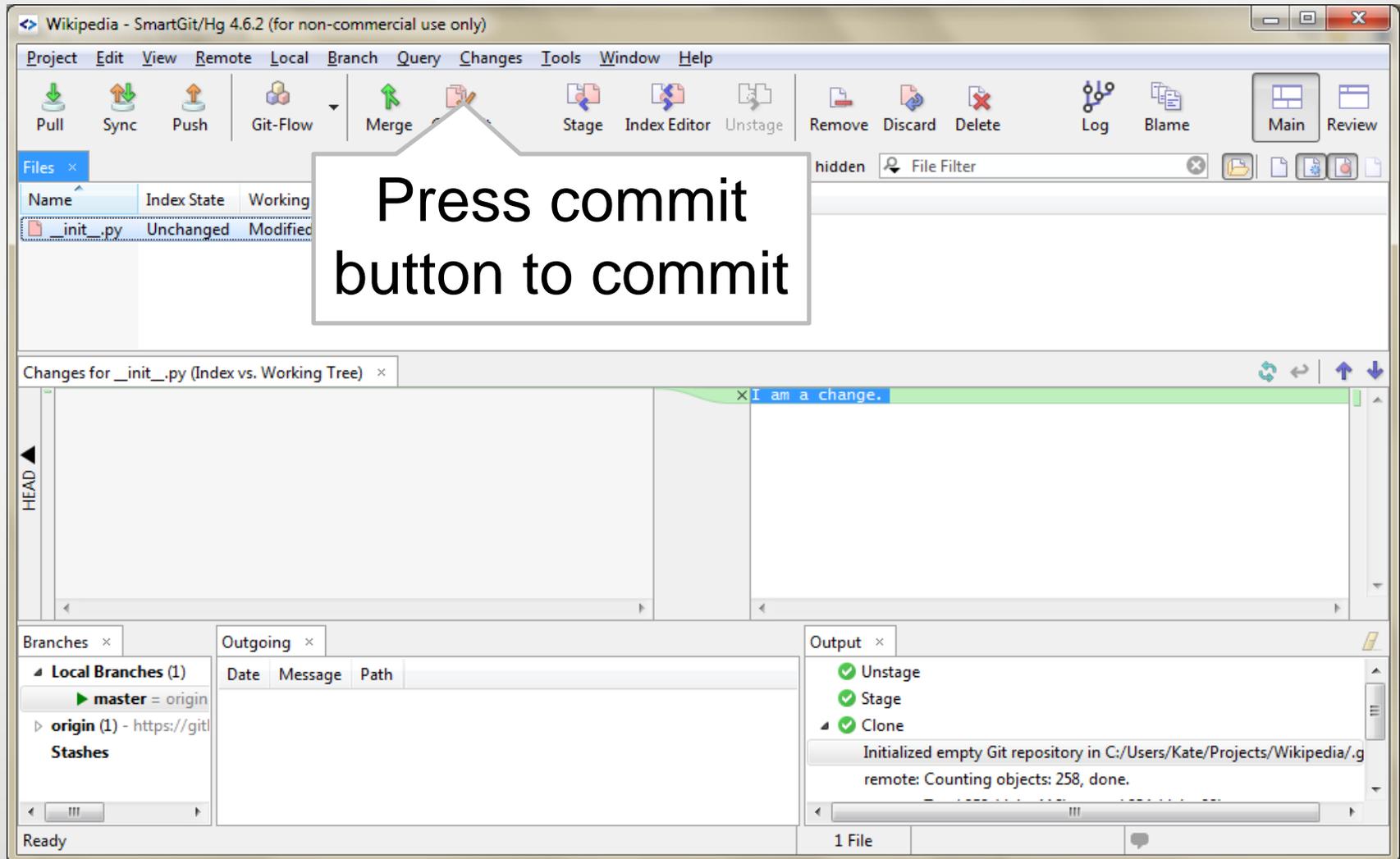
Git GUI - SmartGit

Git GUI - SmartGit
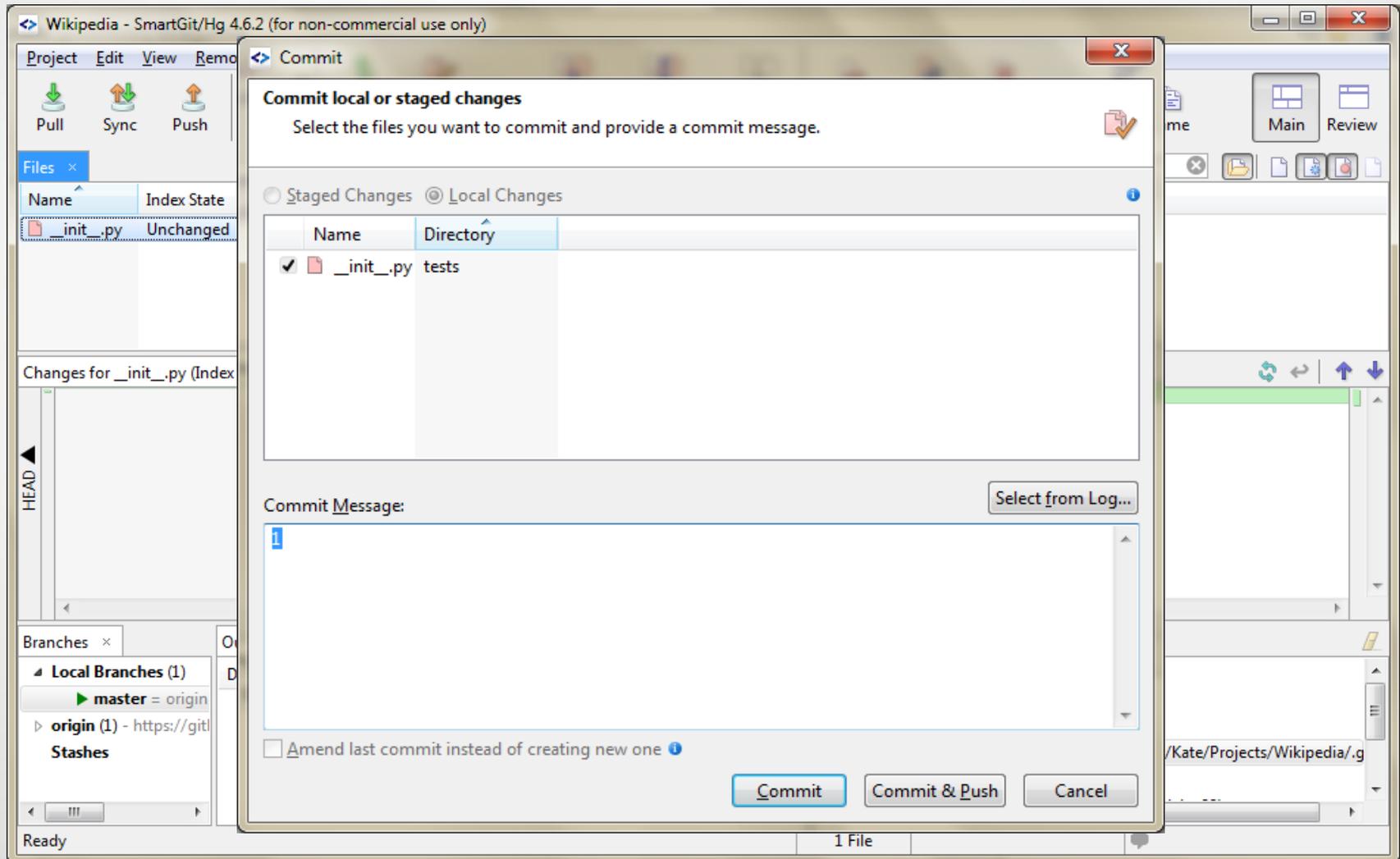
GitHub

- Free web-based hosting for Git projects

GitHub

If you have a repo on your local machine, link it to github with **remote add origin**
```
$ git remote add origin
https://<github filepath>.git
```

Or **clone** an existing repository from Github
```
$ git clone https://<filepath>.git
```

GitHub

GitHub

After making changes and committing, **push** the changes back up to origin
```
$ git push origin master
```

Unless others have made changes, in which case you'll need to resolve on your end first – the simplest way is with **pull**
```
$ git pull
```

http://try.github.io

For Next Time

HW0
1. Install a text editor
2. Install an FTP client
3. Make a test HTML page
4. Install Git