



Introduction to Git

Rob Spanton

rspanton@studentrobotics.org

www.studentrobotics.org

www.xgoat.com



Who we are

STUDENT
ROBOTICS





Problem

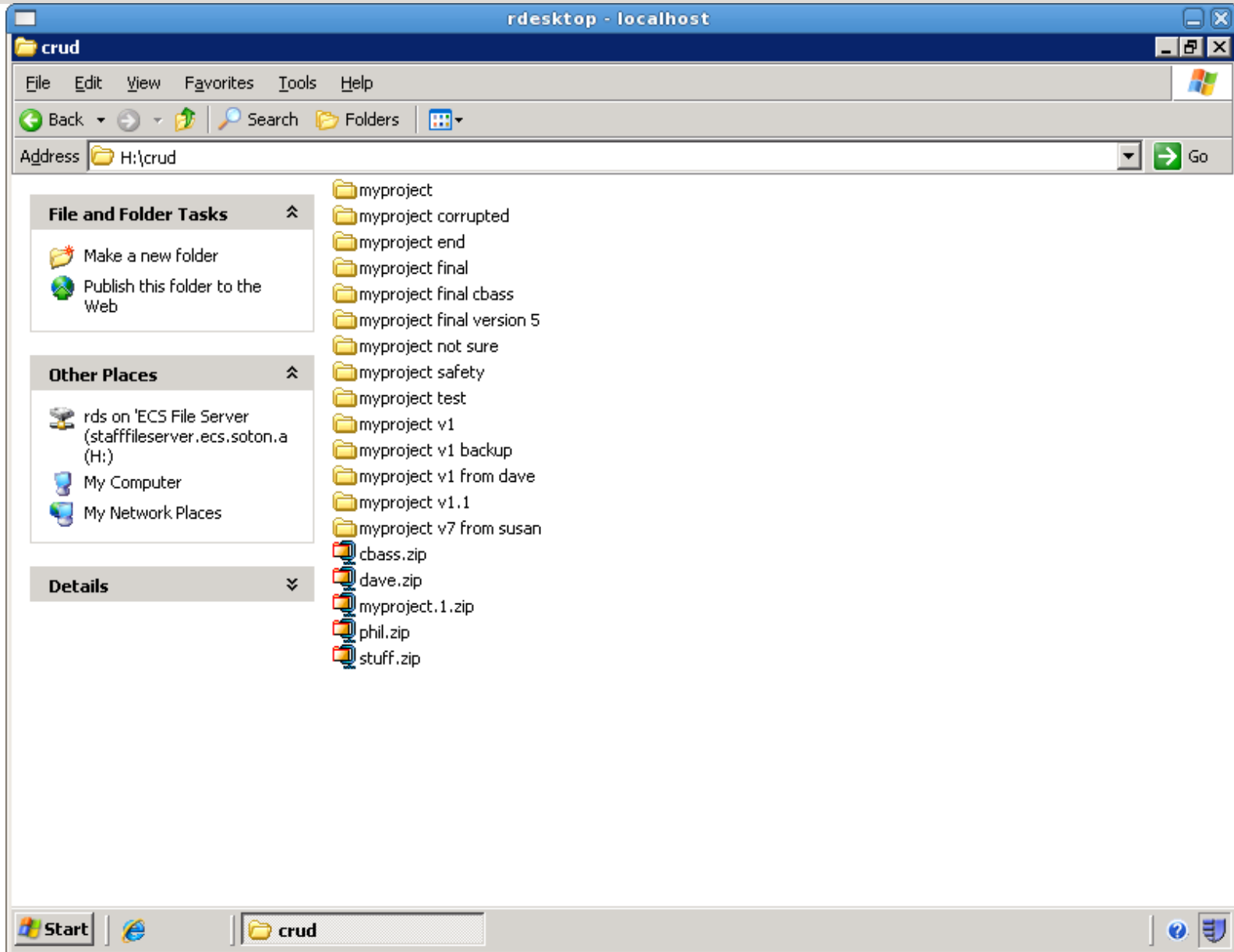
STUDENT
ROBOTICS







Pain

STUDENT
ROBOTICS



ess H:\crud




















File and Folder Tasks

-  Make a new folder
-  Publish this folder to the Web

Other Places

-  rds on 'ECS File Server
(stafffileserver.ecs.soton.ac.uk)
(H:)
-  My Computer
-  My Network Places

Details

-  myproject
-  myproject corrupted
-  myproject end
-  myproject final
-  myproject final cbass
-  myproject final version 5
-  myproject not sure
-  myproject safety
-  myproject test
-  myproject v1
-  myproject v1 backup
-  myproject v1 from dave
-  myproject v1.1
-  myproject v7 from susan
-  cbass.zip
-  dave.zip
-  myproject.1.zip
-  phil.zip
-  stuff.zip



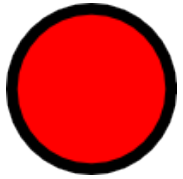
Version Control

1. Git
2. Subversion
3. CVS
4. Monotone
5. Bazaar
6. Microsoft Visual SourceSafe
7. IBM Rational Clearcase
8. Darcs
9. Mercurial
10. Svk
11. Bitkeeper
12. GNU arch



Version Control?

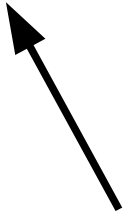
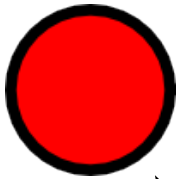
STUDENT
ROBOTICS





Version Control?

STUDENT
ROBOTICS

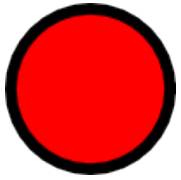


Our initial code.
(A directory of stuff)



Version Control?

STUDENT
ROBOTICS

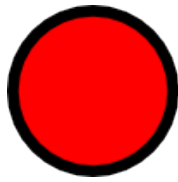
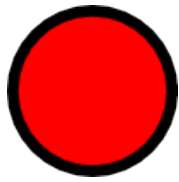


2010/2/20



Version Control?

STUDENT
ROBOTICS



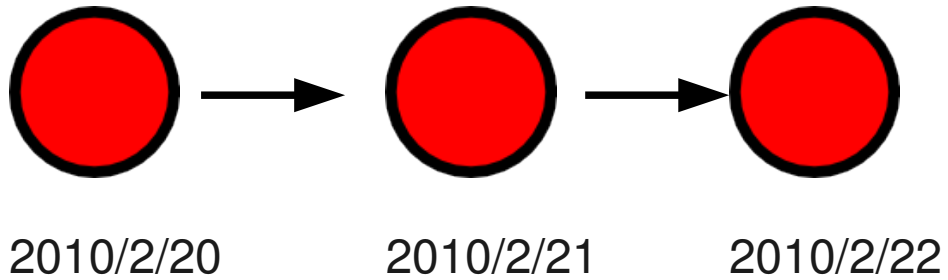
2010/2/20

2010/2/21



Version Control?

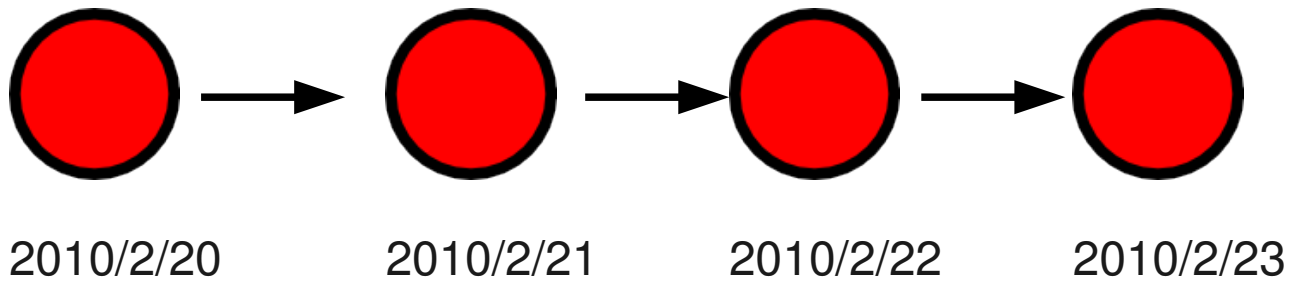
STUDENT
ROBOTICS





Version Control?

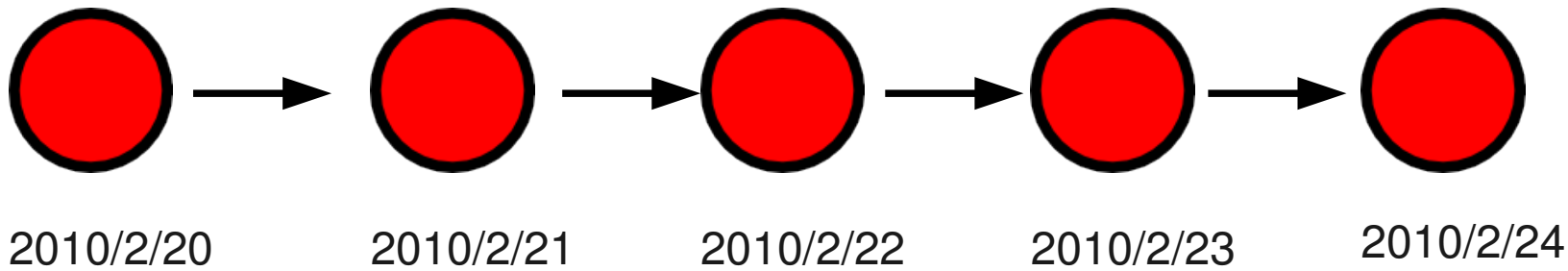
STUDENT
ROBOTICS





Version Control?

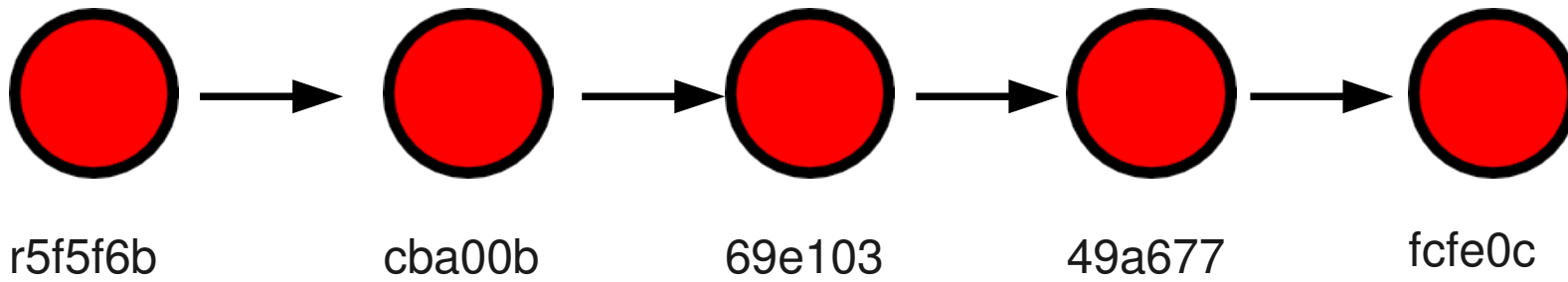
STUDENT
ROBOTICS





Commit Hashes

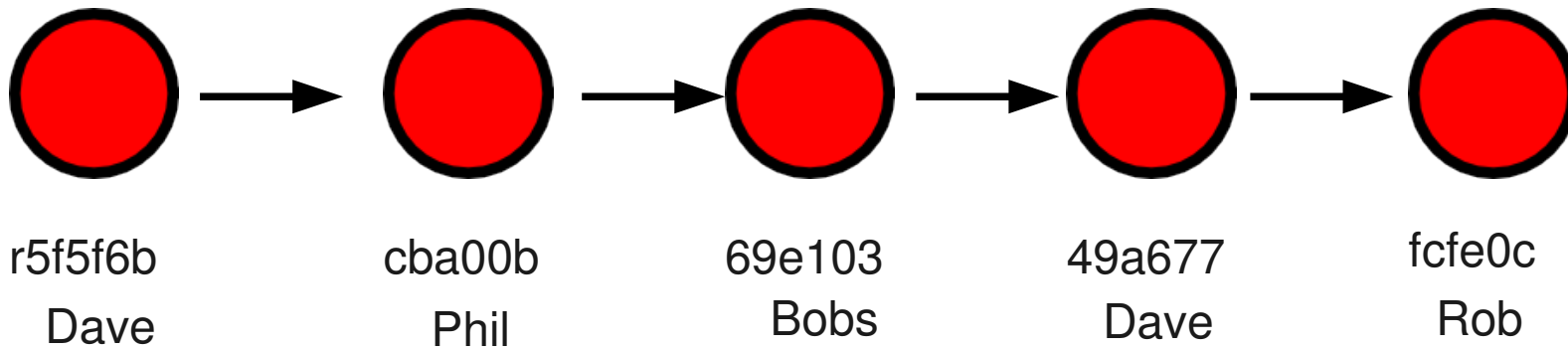
STUDENT
ROBOTICS





Commit Hashes

STUDENT
ROBOTICS





Version Control?

STUDENT
ROBOTICS

Snapshots in time.
(a.k.a. '**revisions**' or '**commits**')



Version Control?

STUDENT
ROBOTICS

Everyone knows:

- What the latest version is.
- Who did what.
- When stuff broke (and who broke it!)
- *Why* stuff broke.



Applications

STUDENT
ROBOTICS

- Group work
 - D4
 - GDP
- Student Robotics
- **All work**
- Used widely by (F)OSS projects and in many companies



Overview

STUDENT
ROBOTICS

git

<http://git-scm.com/>



Example

STUDENT
ROBOTICS

Dave **clones** the SR interconnect controller repository:

```
% git clone git://srobo.org/boards/interconnect-fw.git
Initialized empty Git repository in /tmp/interconnect-fw/.git/
remote: Counting objects: 17, done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 17 (delta 4), reused 0 (delta 0)
Receiving objects: 100% (17/17), done.
Resolving deltas: 100% (4/4), done.

% cd interconnect-fw
% ls
Makefile  README  token_passer.asm
```



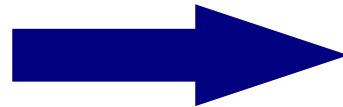
Example

STUDENT
ROBOTICS

Dave then makes some changes to token_passer.asm:

```
...  
Loop:  
    btfss GPIO, TI  
    goto  Loop  
  
    btfss GPIO, RT  
    goto  SendT  
    bsf   GPIO, HT  
...  

```



```
...  
Loop:  
    btfss GPIO, TI  
    goto  Loop  
  
    call  beans  
  
    btfss GPIO, RT  
    goto  SendT  
    bsf   GPIO, HT  
...  

```

Commit b6828340...

Dave's edited copy



What did I do?

STUDENT
ROBOTICS

Dave then looks to see what files he's changed:

```
% git status
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   README
#   modified:   token_passer.asm
#
no changes added to commit (use "git add" and/or "git commit -a")
```



Dave examines what he's done to token_passer.asm:

```
% git diff token_passer.asm
diff --git a/token_passer.asm b/token_passer.asm
index 775cdf9..72a398c 100644
--- a/token_passer.asm
+++ b/token_passer.asm
@@ -23,6 +23,8 @@ TI:    equ        GP3
Loop:  btfss    GPIO, TI
      goto    Loop
+
+      call    beans
+
      btfss    GPIO, RT
      goto    SendT
      bsf     GPIO, HT
```



Example

STUDENT
ROBOTICS

Dave then commits the changes to `token_passer.asm`:
(Not his changes to `README`)

```
% git commit token_passer.asm
```

Add call to beans.

We forgot to put the beans in this code. This patch adds more beans.

```
[master 1faae69] Add call to beans.  
1 files changed, 2 insertions(+), 0 deletions(-)
```

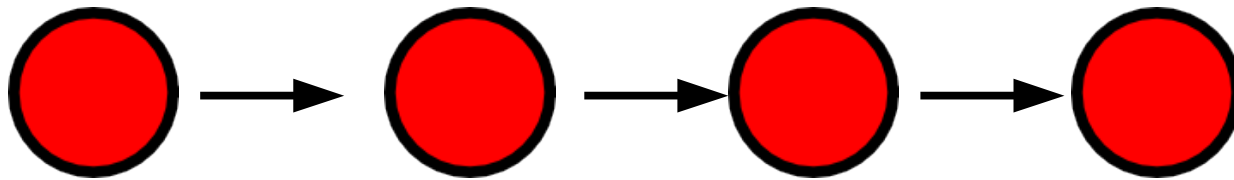
```
%
```




Dave does more

STUDENT
ROBOTICS

Dave makes some more changes, and does a few more commits:

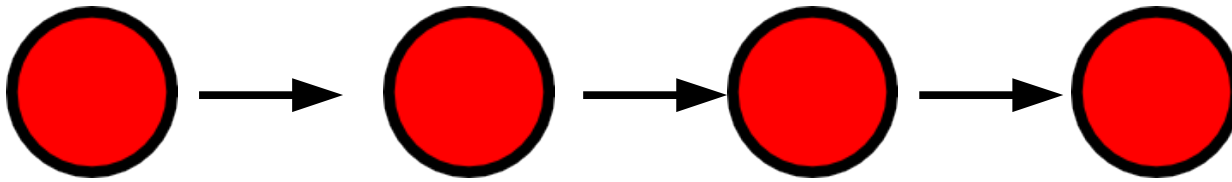




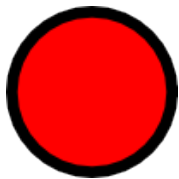
But...

STUDENT
ROBOTICS

What Dave has locally:



What the server has:





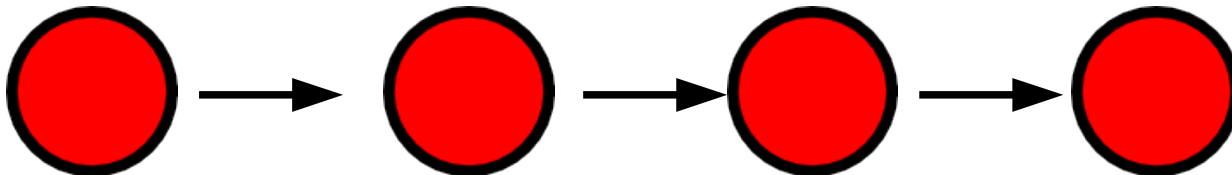
But...

STUDENT
ROBOTICS

Dave needs to **push** his changes to the repository on the server:

```
% git push
Counting objects: 12, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 937 bytes, done.
Total 9 (delta 5), reused 0 (delta 0)
To srobo.org:/srv/git/boards/interconnect-fw
 b682834..2580aad  master -> master
%
```

Now the server has the same as Dave:





git log

STUDENT
ROBOTICS

```
rob@zarniwoop:/tmp/interconnect-fw
commit 2580aad7d4b98204f752a4b93275216a487301ea
Author: Robert Spanton <rspanton@zepler.net>
Date:   Wed Mar 3 11:17:18 2010 +0000

    Remove some nops

commit aec710870645cfddf671bafb36babe6273c5dccb
Author: Robert Spanton <rspanton@zepler.net>
Date:   Wed Mar 3 11:16:53 2010 +0000

    Add a note about cheese to the README

commit 1faae69139dec69521aa0464815d42446d31bab9
Author: Robert Spanton <rspanton@zepler.net>
Date:   Wed Mar 3 11:04:17 2010 +0000

    Add call to beans.

    We forgot to put the beans in this code.  This patch adds more beans.

commit b6828340bdab6f2a8985f5f36f0facf366e2c4ba
Author: Richard Barlow <richard@richardbarlow.co.uk>
Date:   Wed Feb 3 21:53:12 2010 +0000

    Update program to hold HT line until the end of the transaction.
```



:-0

STUDENT
ROBOTICS

```
rob@zarniwoop:/tmp/interconnect-fw
commit 2580aad7d4b98204f752a4b93275216a487301ea
Author: Robert Spanton <rspanon@zepler.net>
Date:   Wed Mar 3 11:17:18 2010 +0000

    Remove some nops

commit aec710870645cfddf671baf36babe6273c5dccb
```

This is **bad**.

Needs to include **why** those nops are being removed.



Version control will save you,
but only if you use it right.

(Think about what you're doing.)



Small, logically ordered, descriptive commits.

Otherwise, you and others will **burn**.



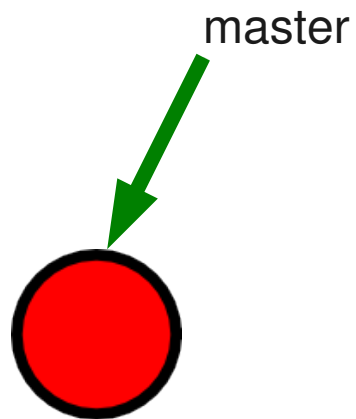
“Life isn't a linear series of commmits.”



Branches

STUDENT
ROBOTICS

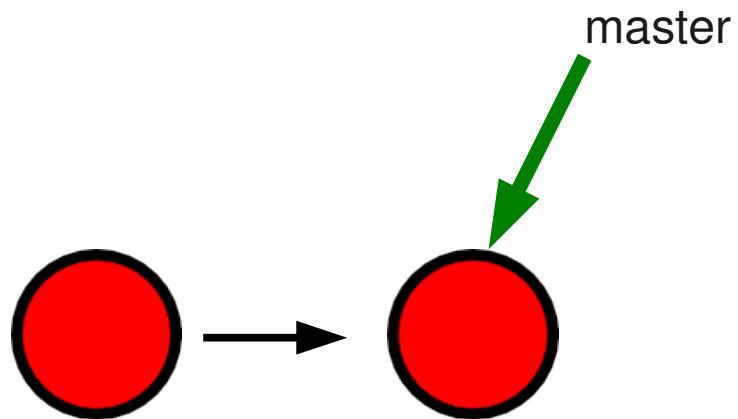
You are always* working on a branch.
The branch's name refers to the head of the branch.





Branches

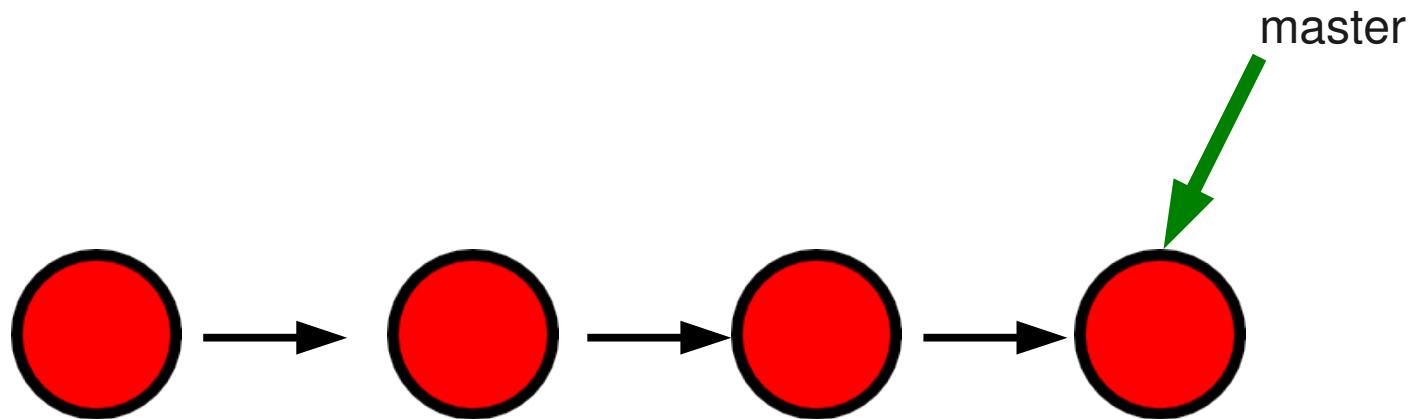
You are always* working on a branch.
The branch's name refers to the head of the branch.





Branches

You are always* working on a branch.
The branch's name refers to the head of the branch.

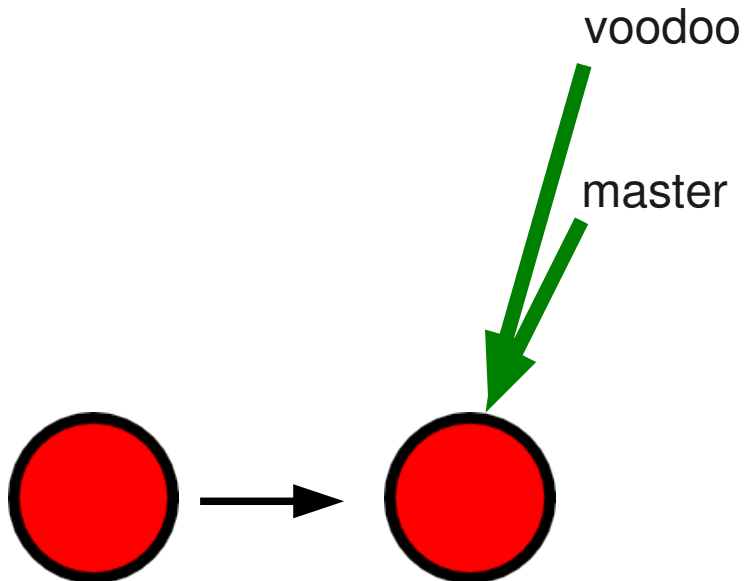




Make a branch

STUDENT
ROBOTICS

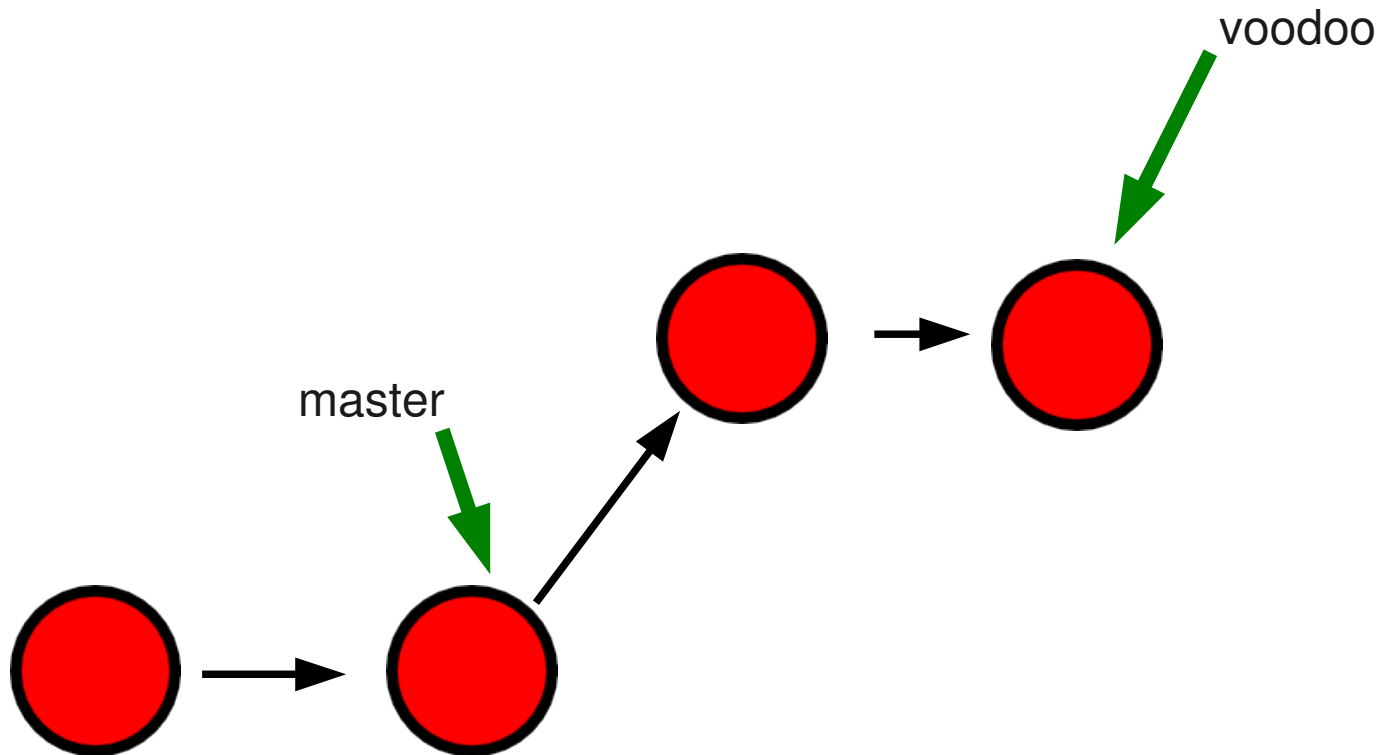
```
% git branch voodoo  
% git checkout voodoo  
Switched to branch 'voodoo'
```





Branches

STUDENT
ROBOTICS

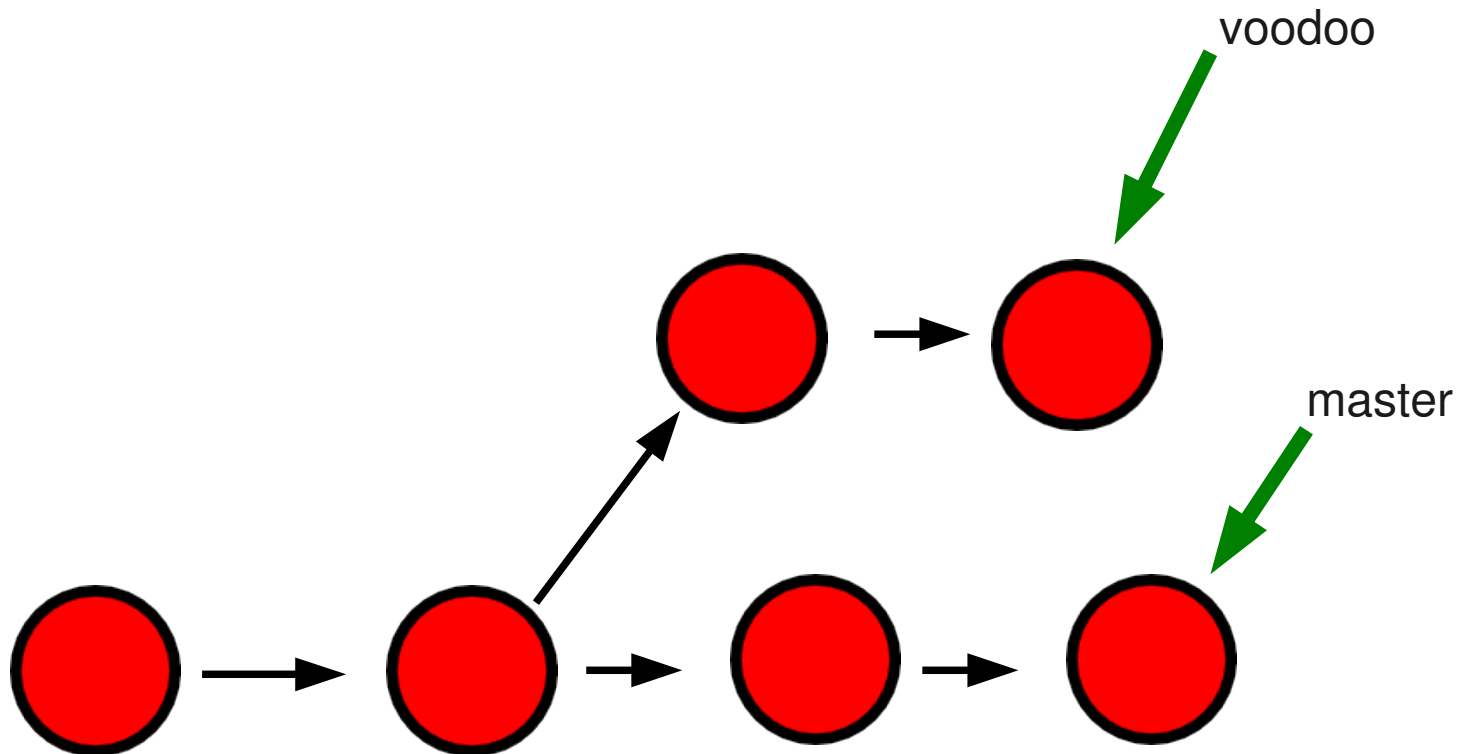




Branches

STUDENT
ROBOTICS

```
% git checkout master  
...
```





Remotes

STUDENT
ROBOTICS

Remotes are other repositories.

```
% git remote  
origin
```

(Could be on the same machine, could be on a different machine – doesn't matter.)

```
% git branch  
master  
* voodoo
```

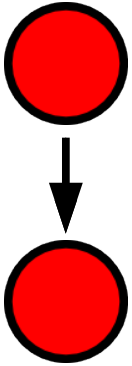
```
% git branch -r  
origin/master
```



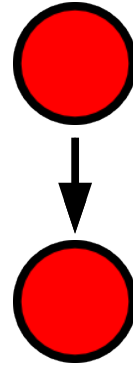
> 1 People

STUDENT
ROBOTICS

Dave



Phil

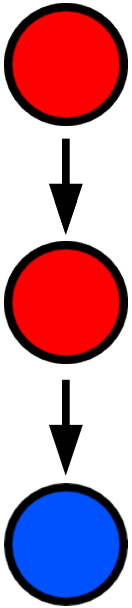




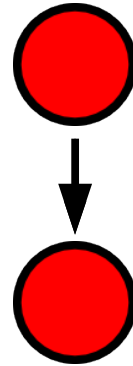
> 1 People

STUDENT
ROBOTICS

Dave



Phil

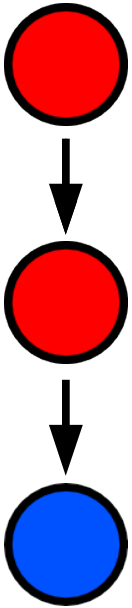




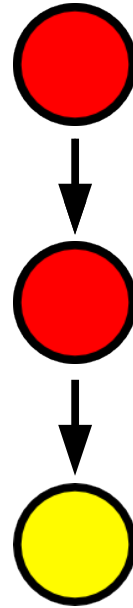
> 1 People

STUDENT
ROBOTICS

Dave



Phil

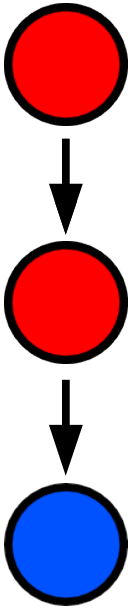




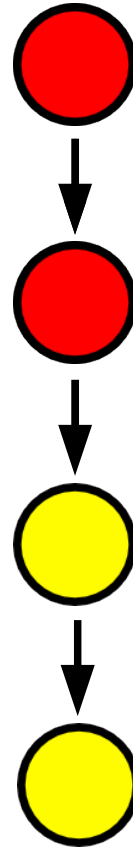
> 1 People

STUDENT
ROBOTICS

Dave



Phil

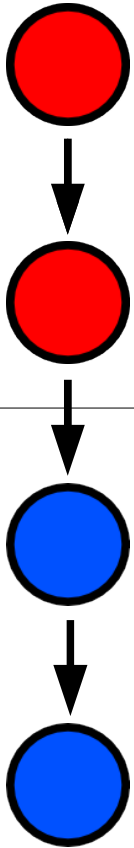




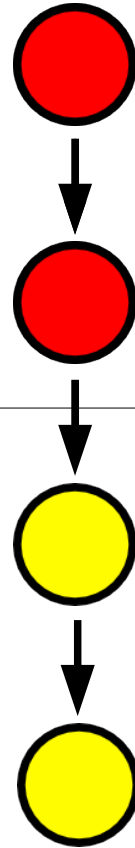
> 1 People

STUDENT
ROBOTICS

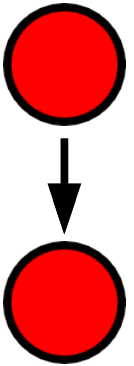
Dave



Phil



Server

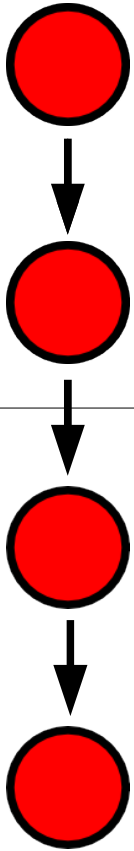




>1 People

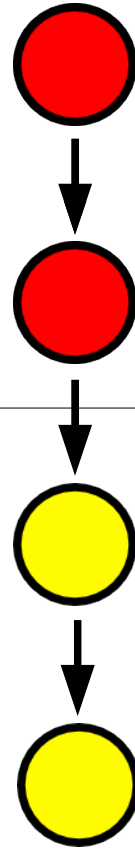
STUDENT
ROBOTICS

Dave

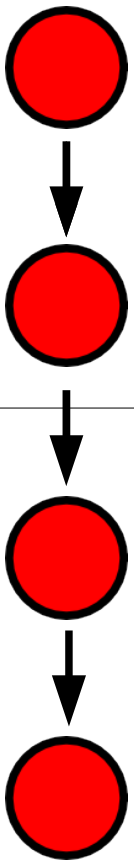


Dave **pushes** his work to the server.

Phil



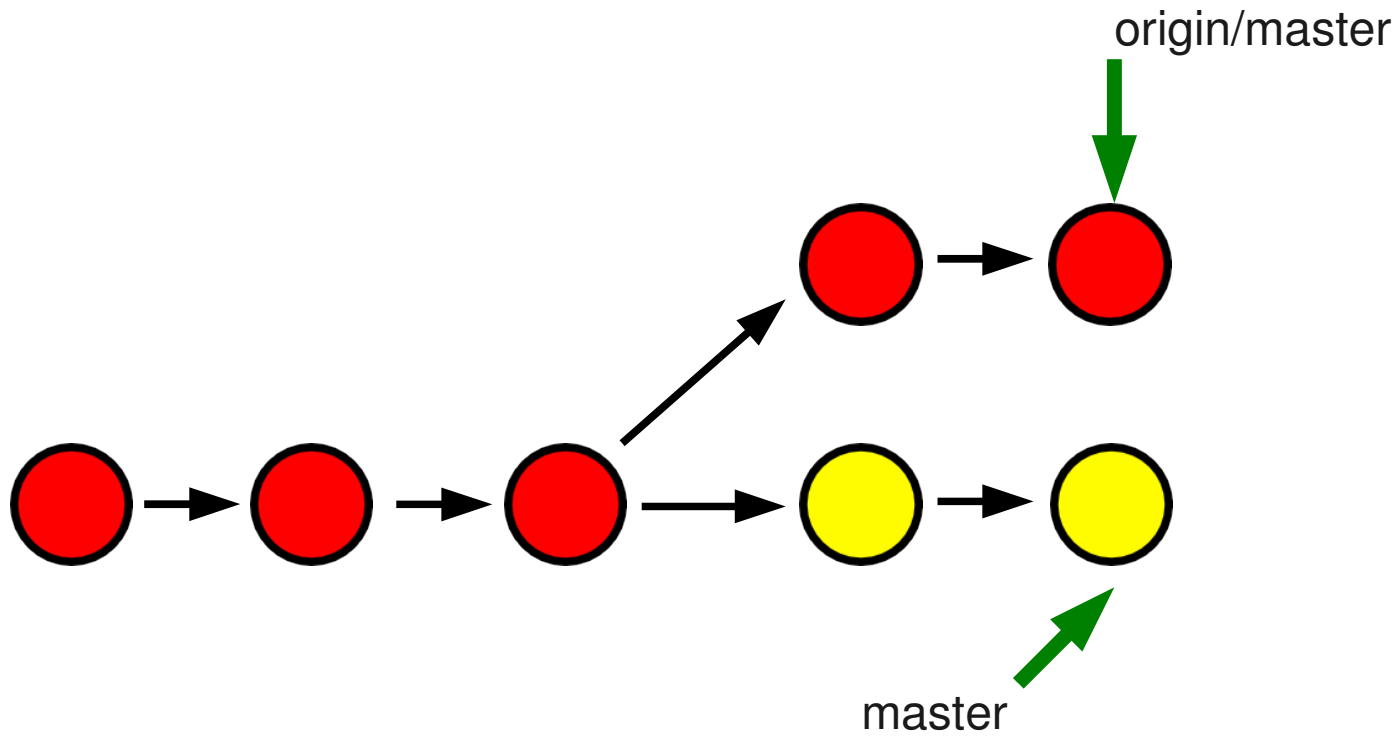
Server





Rotate...

STUDENT
ROBOTICS

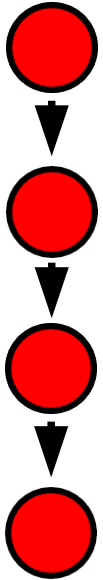




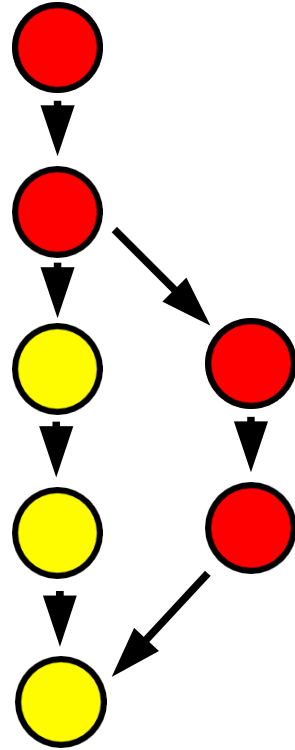
>1 People

STUDENT
ROBOTICS

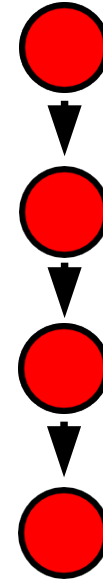
Dave



Phil



Server

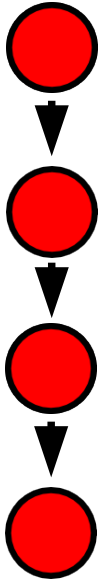




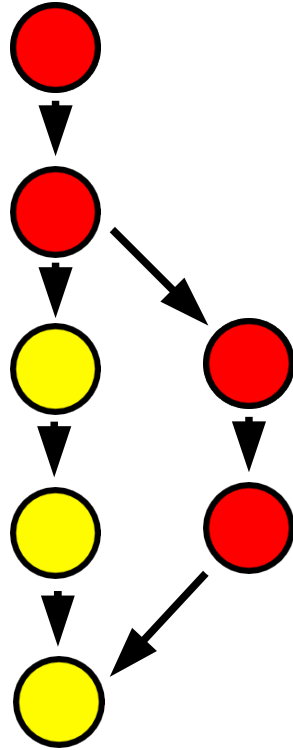
>1 People

STUDENT
ROBOTICS

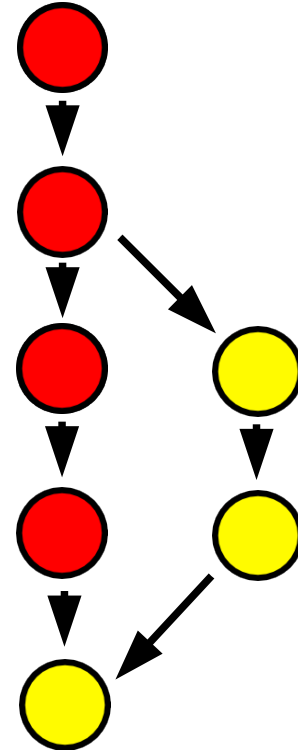
Dave



Phil



Server



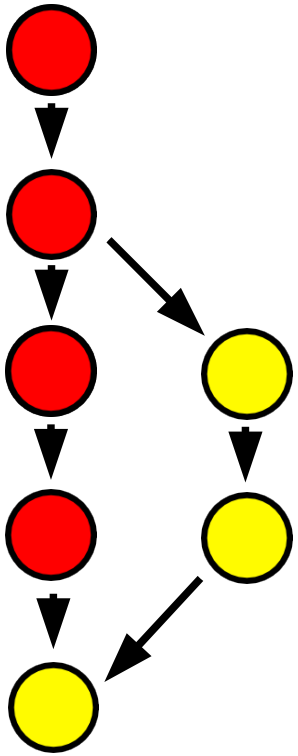
Phil pushes onto the server.



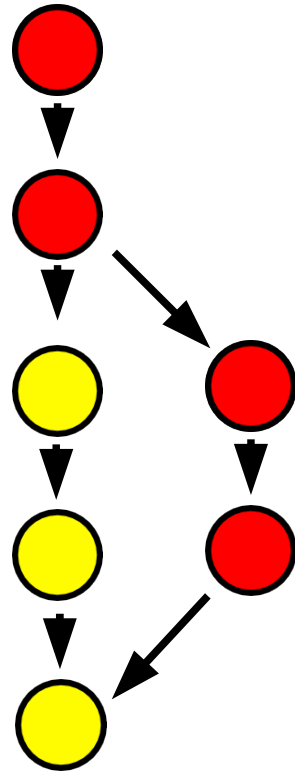
>1 People

STUDENT
ROBOTICS

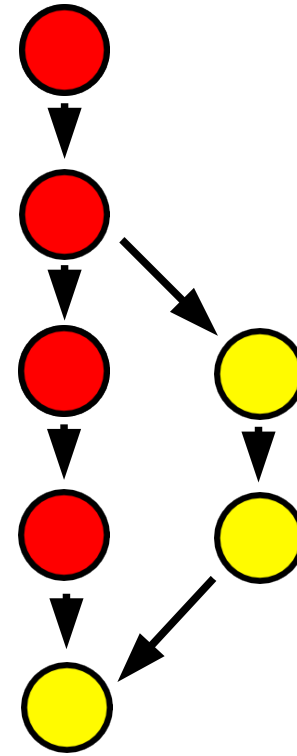
Dave



Phil



Server



Dave **pulls** from the server.

(pull is a fetch followed by a merge)



The Stage

STUDENT
ROBOTICS

Before committing, we **stage** changes to be put into the commit.

(Before, we saw Dave use a shortcut by supplying the changes to be in the commit to **git commit**)

Main tool for this: **git add**



The Stage

STUDENT
ROBOTICS

git status shows us what's staged.
(and we can look at them with **git diff --cached**)

```
% git status
# On branch voodoo
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   modified:   README
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in
working directory)
#
#   modified:   README
#   modified:   token_passer.asm
```



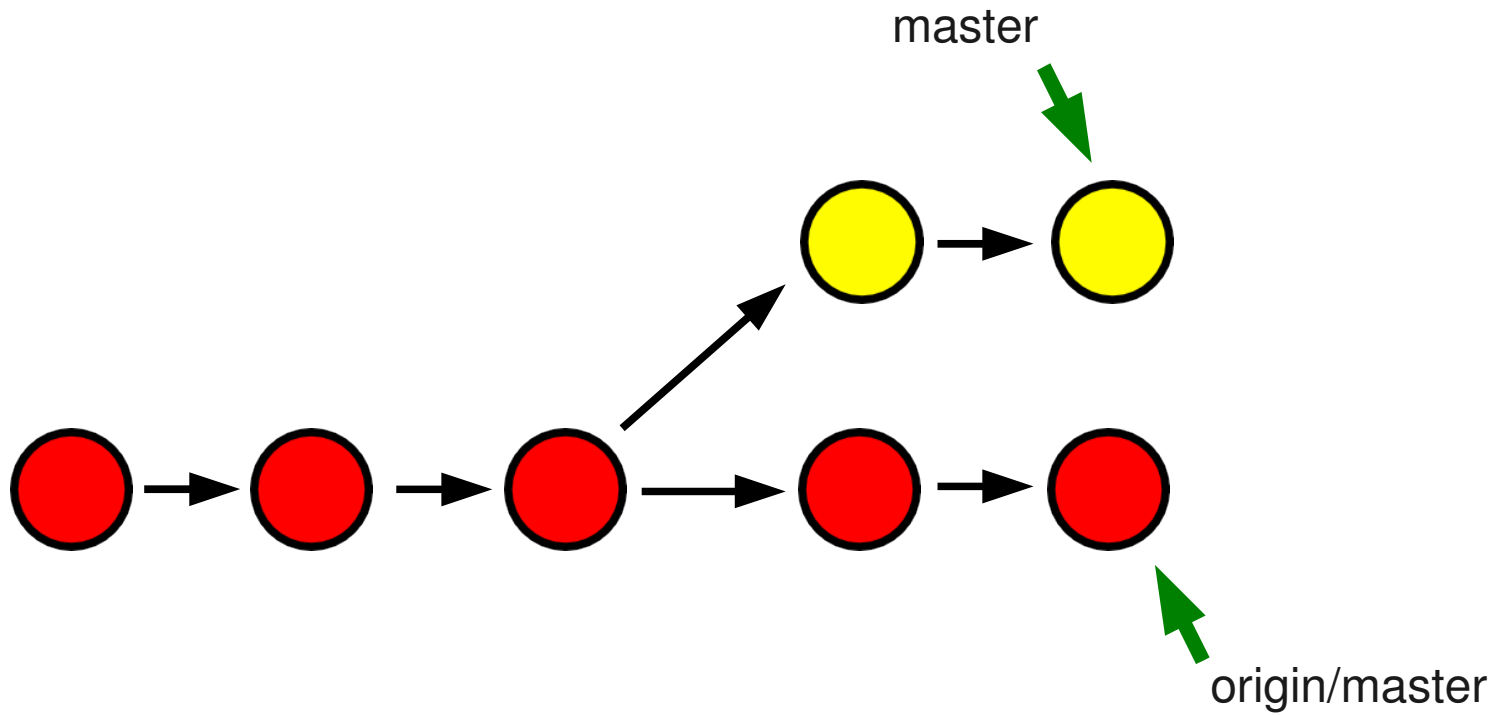
Colours...

```
rob@zarniwoop:/tmp/interconnect-fw
[rob@zarniwoop interconnect-fw(voodoo)]$ git status
# On branch voodoo
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   README
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   README
#       modified:   token_passer.asm
#
[rob@zarniwoop interconnect-fw(voodoo)]$
```



Rebasing

STUDENT
ROBOTICS

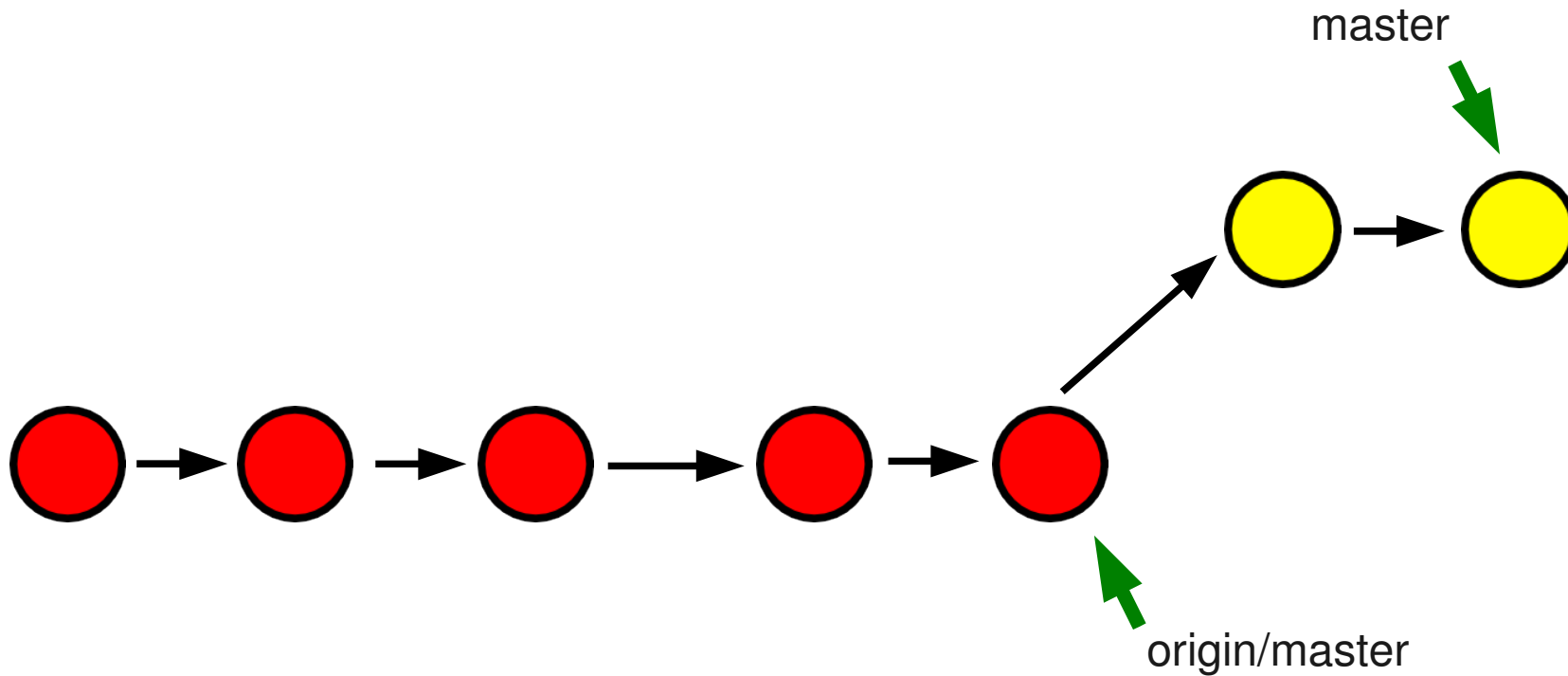




Rebasing

STUDENT
ROBOTICS

```
% git rebase origin/master
```





Rebasing

STUDENT
ROBOTICS

Never rewrite commits that you have already published.

You will annoy everyone!



Old Versions

STUDENT
ROBOTICS

- Use **git log** to look at the history.
- Use **git checkout** to get a given version from the history into your working copy.



Bisecting

STUDENT
ROBOTICS

I know it worked way back here.

OMG PONIES!

I can't go through every revision and test that it'd work. I'm lazy, and need to get on with breaking other things.

It doesn't work anymore :-)



Bisecting

STUDENT
ROBOTICS

Saviour: **git bisect**



```
% git bisect start 78393a HEAD
... test ...
% git bisect good
... test ...
% git bisect bad
... test ...
Repeat
```



Good Practice

(That can save you hours and hours of pain.)



Commit Messages

Make them concise and informative.
Describe the change, and **why** it's happening.



Commit as often as you can.

Small changes.

Easier to find where it broke.

Easier to follow/examine.



Look at what you're committing.

***Before* you commit.**



Be aware.



Documentation

STUDENT
ROBOTICS

`git-scm.com` has loads.

I recommend:

- The “Git User's Manual”
- The “Git Community Book”
- The man pages for each command (e.g. `man git-bisect`)



Learning to use Git

STUDENT
ROBOTICS

You can completely destroy your local situation, and no-one will know.



Questions?

www.studentrobotics.org

Rob Spanton
rspanton@studentrobotics.org