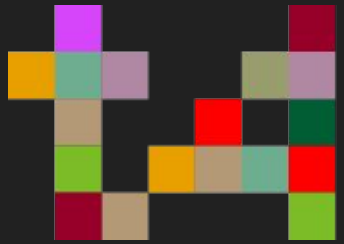


tech

Git Ur Shit together, Debra.

wtf is source control

Source control (or version control) is the practice of tracking and managing changes to code. Source control management (SCM) systems provide a running history of code development and help to resolve conflicts when merging contributions from multiple sources.

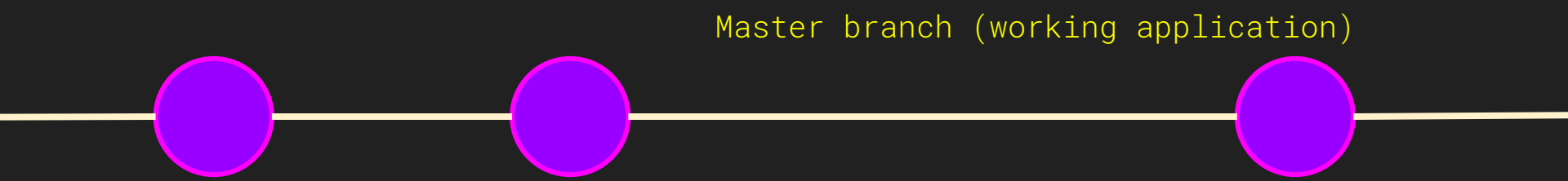
Thanks aws for the definition

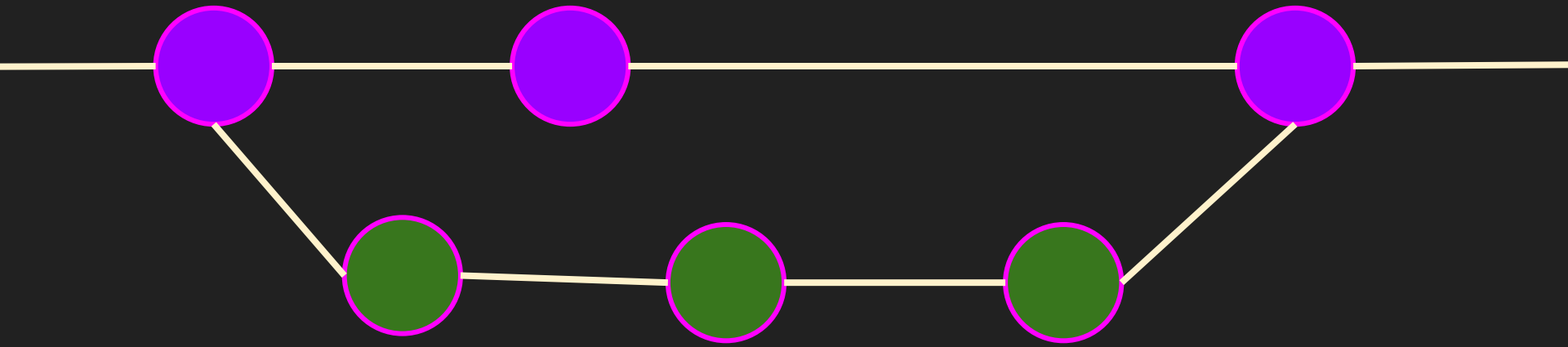
wtf is source control

Git Ur Shit

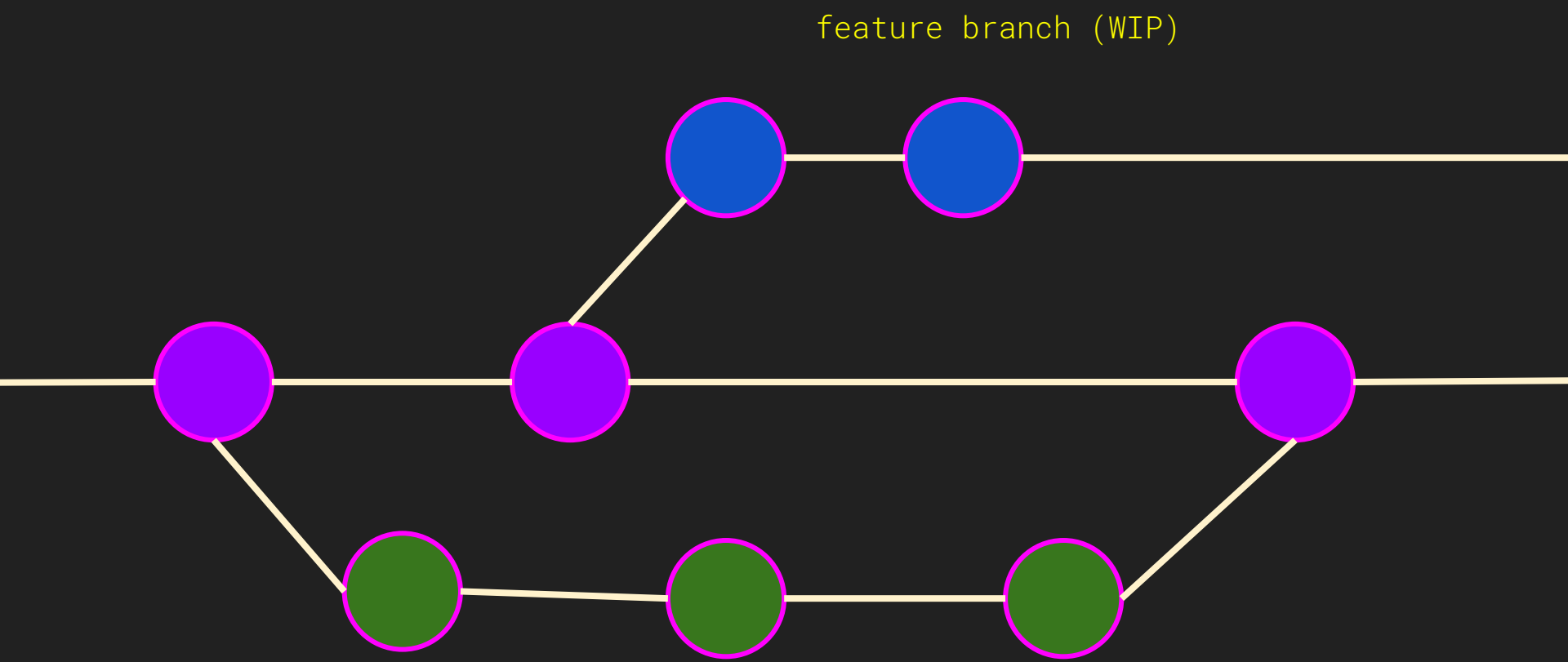
jargon

repository
branch
remote
origin
merge
master
pull request
head


Master branch (working application)



feature branch (merged into master)


feature branch (WIP)

Most common commands (not exhaustive)

`git init`	-	initializes new git repo in current dir
`git branch`	-	makes or shows current working branches
`git checkout`	-	switches HEAD to a different branch
`git log`	-	outputs all recent commits to working dir
`git blame`	-	outputs authors of changes in working dir
`git pull`	-	bring in changes from another source
`git push`	-	add changes from local repo to remote repo
`git add`	-	“stage” code changes to be committed
`git commit`	-	add staged changes to a new commit
`git reset`	-	move the HEAD or abandon changes
`git status`	-	outputs the current state of changes

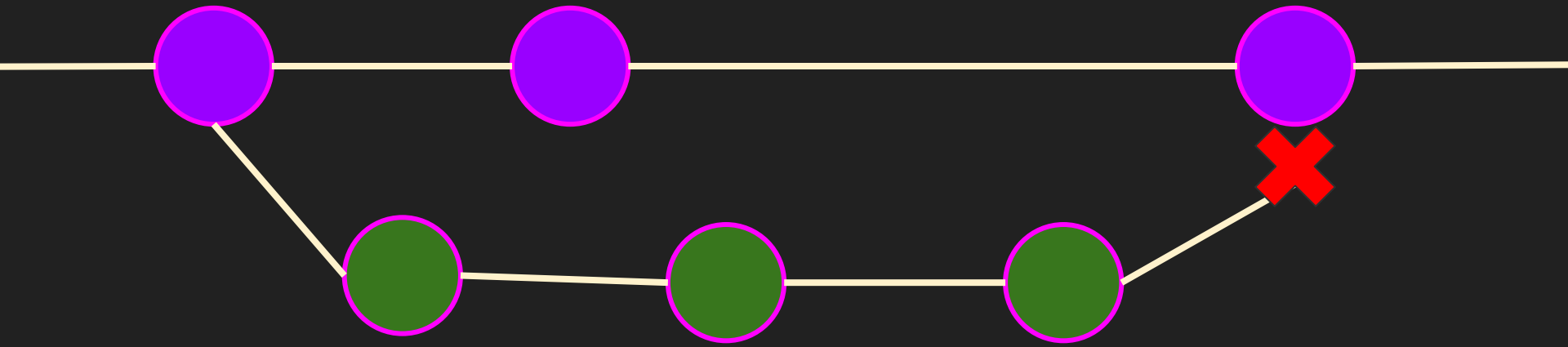
Common safe workflow

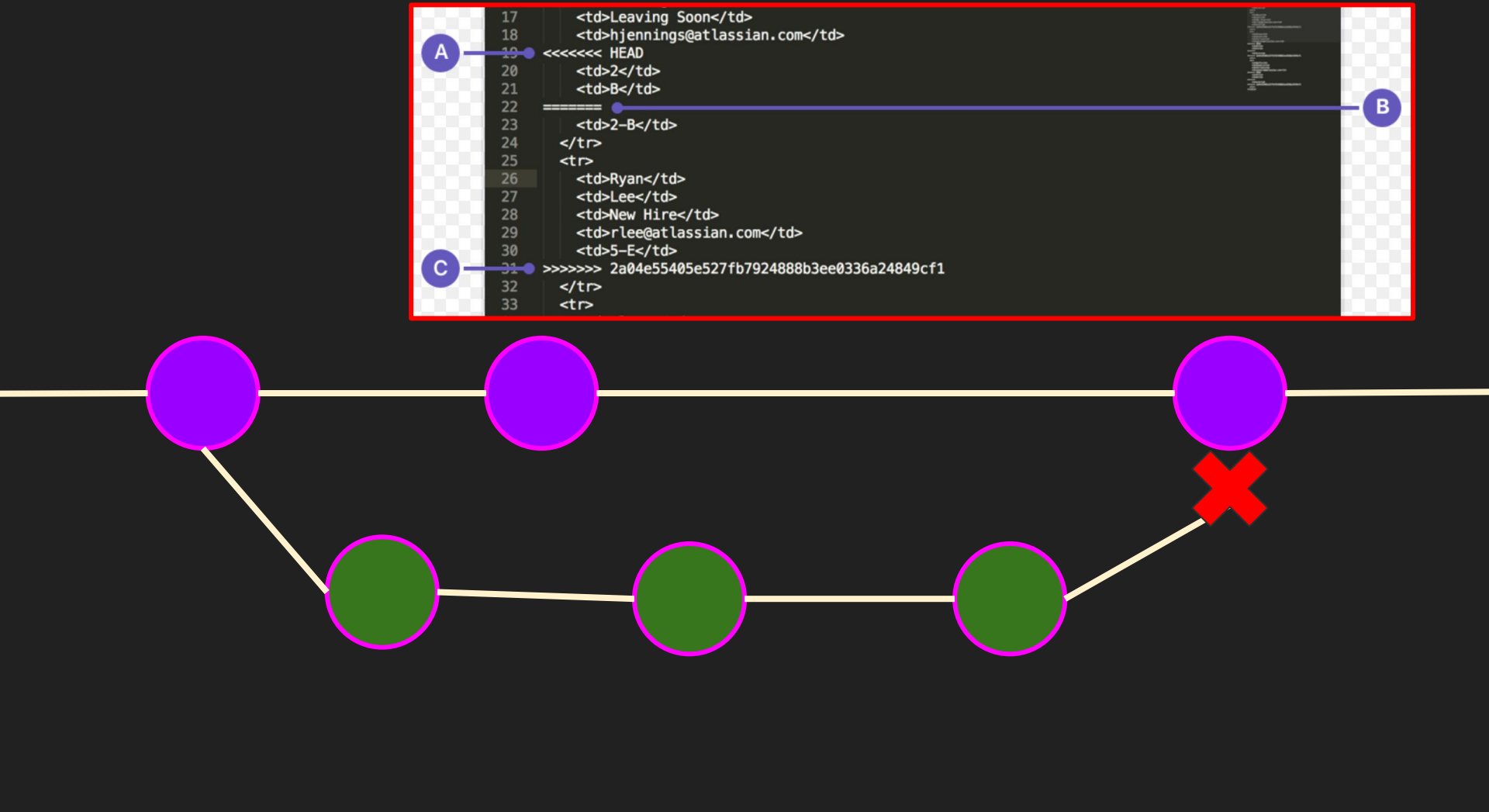
1. Make sure master is up to date
2. Always start a new branch when making changes
3. Only add changes when you want to hold onto them
4. Only commit changes when you've completed a logical step towards achieving the goal
5. Be succinct and descriptive in messaging
6. Always pull from master before pushing local branch for a new pull request
7. NEVER FORCE PUSH
8. NEVER PUSH TO MASTER

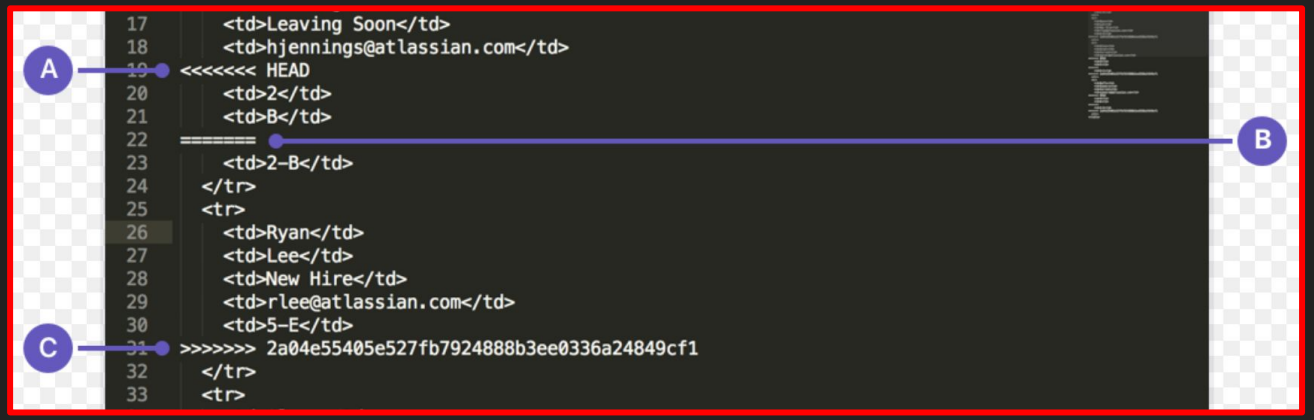
Daly's Daily Workflow Routine

```
git checkout master
git pull origin master
git checkout -b mySweetNewFeature
   [ implements sweet new feature ]
git add .
git commit -m "implemented sweet new feature"
git pull origin master
git push origin mySweetNewFeature
   [ goes to github web app and opens a PR ]
```

THE DREADED MERGE CONFLICT




17 <td>Leaving Soon</td>
18 <td>hjennings@atlassian.com</td>
19 <<<<<<< HEAD
20 <td>2</td>
21 <td>B</td>
22 =======
23 <td>2-B</td>
24 </tr>
25 <tr>
26 <td>Ryan</td>
27 <td>Lee</td>
28 <td>New Hire</td>
29 <td>rlee@atlassian.com</td>
30 <td>5-E</td>
31 >>>>>>> 2a04e55405e527fb7924888b3ee0336a24849cf1
32 </tr>
33 <tr>
A
B
C



1. Choose which affected version to keep
2. Add + commit those changes with a message about resolving the conflict
3. Try again

When in doubt ...

git help

```
These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone      Clone a repository into a new directory
   init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add        Add file contents to the index
   mv         Move or rename a file, a directory, or a symlink
   reset      Reset current HEAD to the specified state
   rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect     Use binary search to find the commit that introduced a bug
   grep       Print lines matching a pattern
   log        Show commit logs
   show       Show various types of objects
   status     Show the working tree status

grow, mark and tweak your common history
   branch     List, create, or delete branches
   checkout   Switch branches or restore working tree files
   commit     Record changes to the repository
   diff       Show changes between commits, commit and working tree, etc
   merge      Join two or more development histories together
   rebase     Reapply commits on top of another base tip
   tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
   fetch      Download objects and refs from another repository
   pull       Fetch from and integrate with another repository or a local branch
   push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
```

Daly's Daily Workflow Routine

```
git checkout master
git pull origin master
git checkout -b mySweetNewFeature
   [ implements sweet new feature ]
git add .
git commit -m "implemented sweet new feature"
git pull origin master
git push origin mySweetNewFeature
   [ goes to github web app and opens a PR ]
```