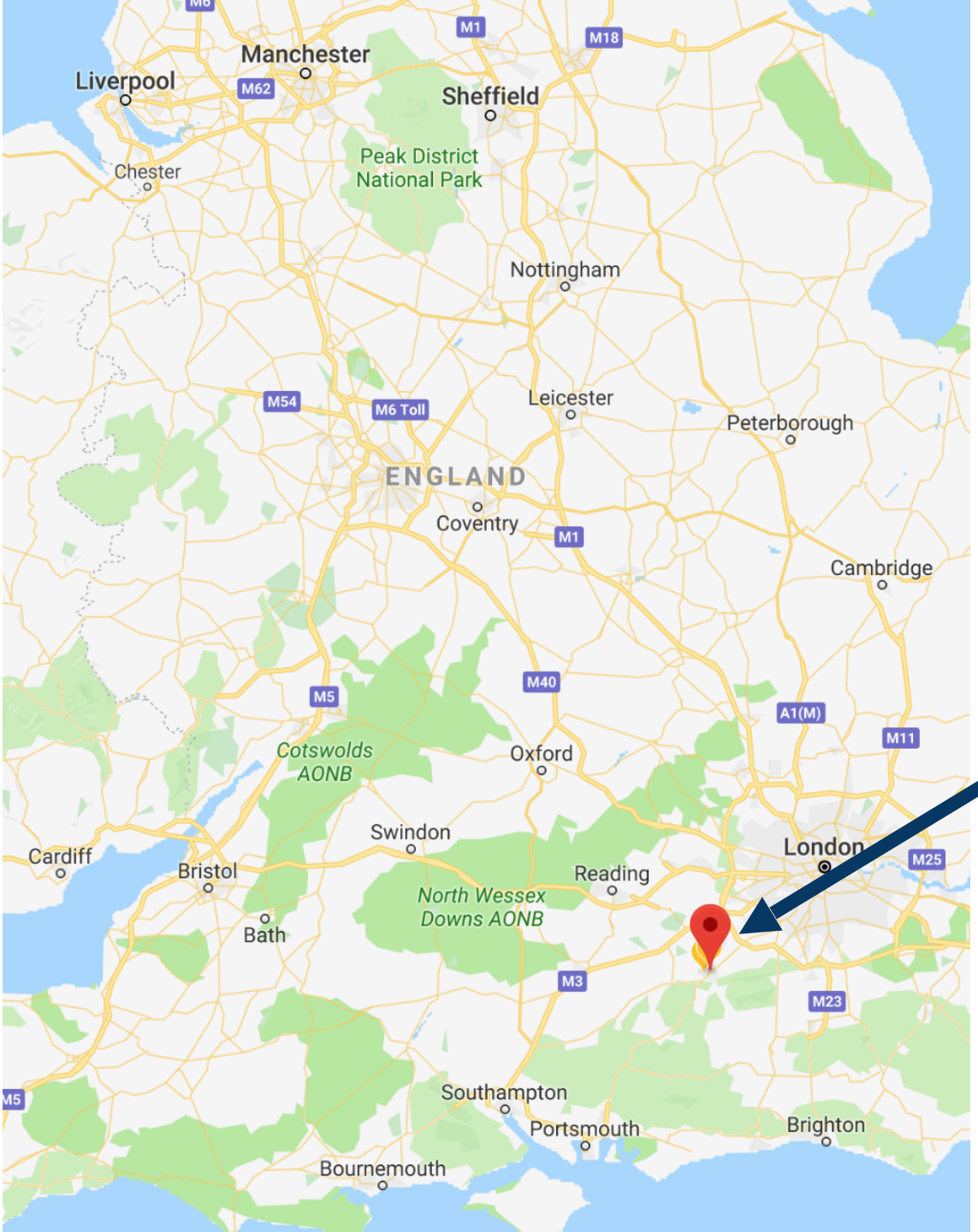
git good

based on slides from @redroxprojects

Daniel Spindelbauer
@sdaniel55



UNIVERSITY OF SURREY

📚 Computer Science

🚩 GitHub Campus Expert

🔨 iOS Developer

@sdaniel55

Being a Campus Expert: The events I have been to in the past 12 months...

2018 - GitHub Constellation (London, UK)

2018 - HackCity18 (London, UK)

2018 - HackSurrey Beta (Guildford, UK)

2018 - Hacktoberfest Hack Night (London, UK)

2018 - HackCon EU (London, UK)

2018 - GitHub Universe (San Francisco, USA)

2018 - Campus Experts Summit (SF, USA)

2018 - DevRelCon (London, UK)

2018 - CraftWork London (London, UK)

2019 - You Got This (London, UK)

2019 - ManMetHacks 1.0 (Manchester, UK)

2019 - Git Merge 2019 (Brussels, BE)

2019 - Royal Hackaway v2.0 (Egham, UK)

2019 - HackSurrey Mk2 (Guildford, UK)

2019 - R. U. Hacking (Reading, UK)

2019 - Code First: Girls, Surrey (Guildford, UK)

Today's Topic:
Version Control with Git

Why Version Control?

- **History:**

  remembering what changed over time, who worked on what and when

- **Backup:**

  keep copies of your work in multiple places to avoid losing it

- **Collaboration:**

  Work concurrently on the same code with other people

What will we cover?

🐣 **Gitting Started:** repositories, staging, committing

👍 **Gitting Good:** branching, jumping around, merging

🆙 **Git Better:** rolling back, using GitHub

💡 **Git Ideas:** how might you use GitHub for your work

Before we begin:

```
$ command line commands are in
  this font & colour
```

My first .git

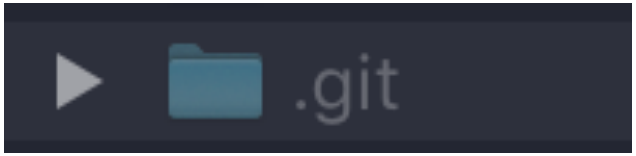
repository creation, staging, committing

@sdaniel55

What is a repository?

A folder containing your code (or docs, thesis, data sets...) **but:**

it has a "**.git**" directory which stores the history of your project



Created with

```
$ git init <repository name>
```

Commits:

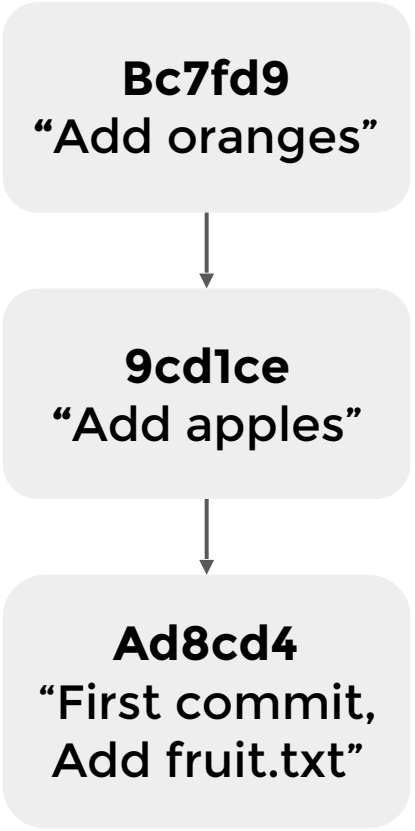
Snapshots of the **contents (e.g. code)**
in your **repository**

Created with

```
$ git add <filename>
```

```
$ git commit -m "<commit description>"
```

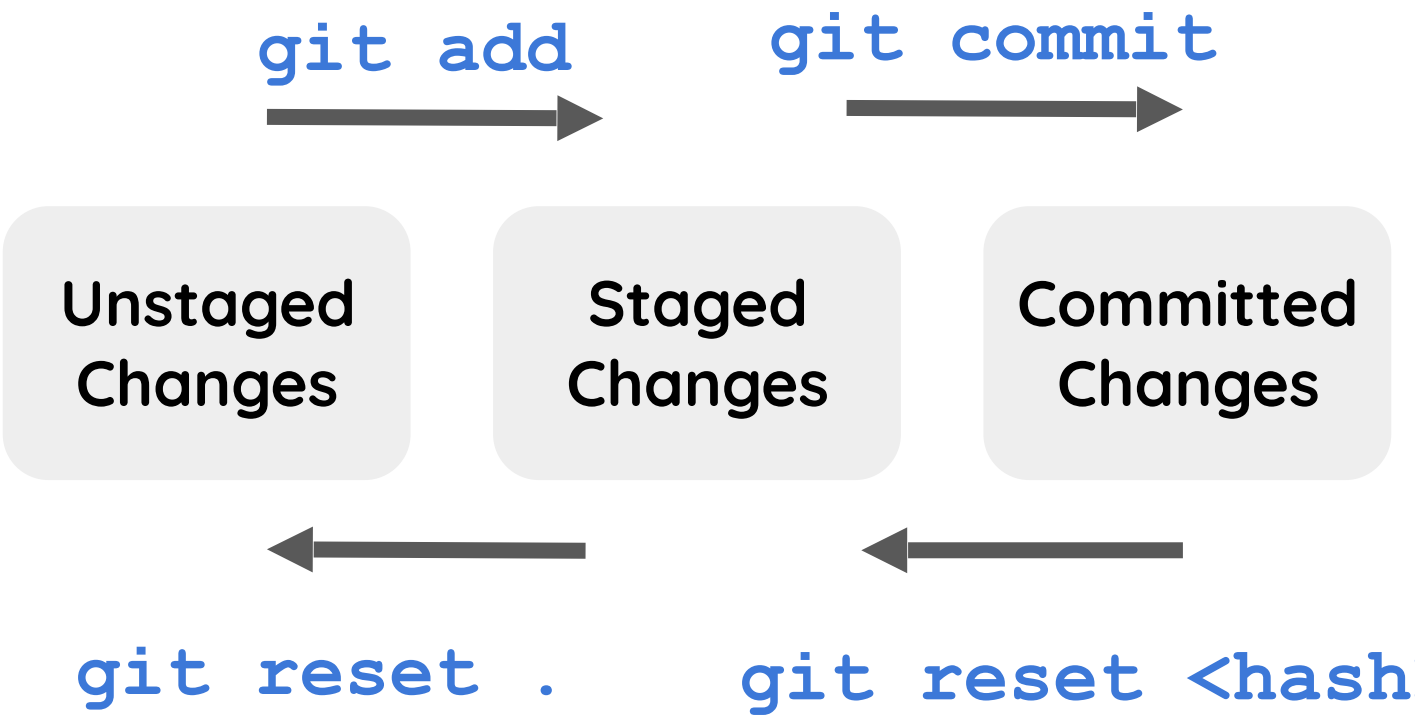
More on Commits



- Commits form a **linked list** structure

- Run `git log` to see every commit in the repository

Staging



(see what's staged: **`git status`**)

Git: Command Recap

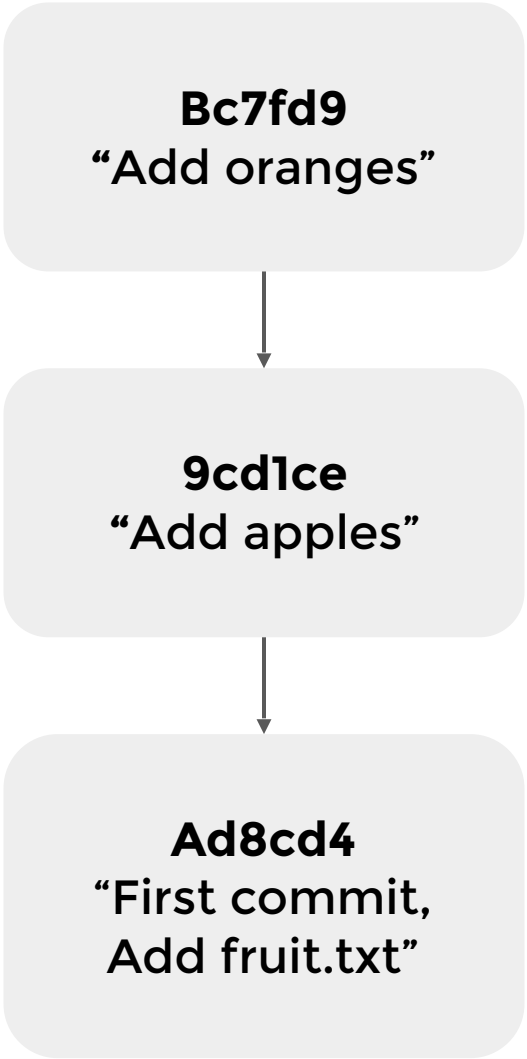
- `git init` - turn any folder into a super smart git folder
- `git add` - stage changes to files
- `git commit` - create a snapshot of staged changes
- `git reset` - unstage, uncommit
- `git log` - see all commits
- `git status` - see what is staged

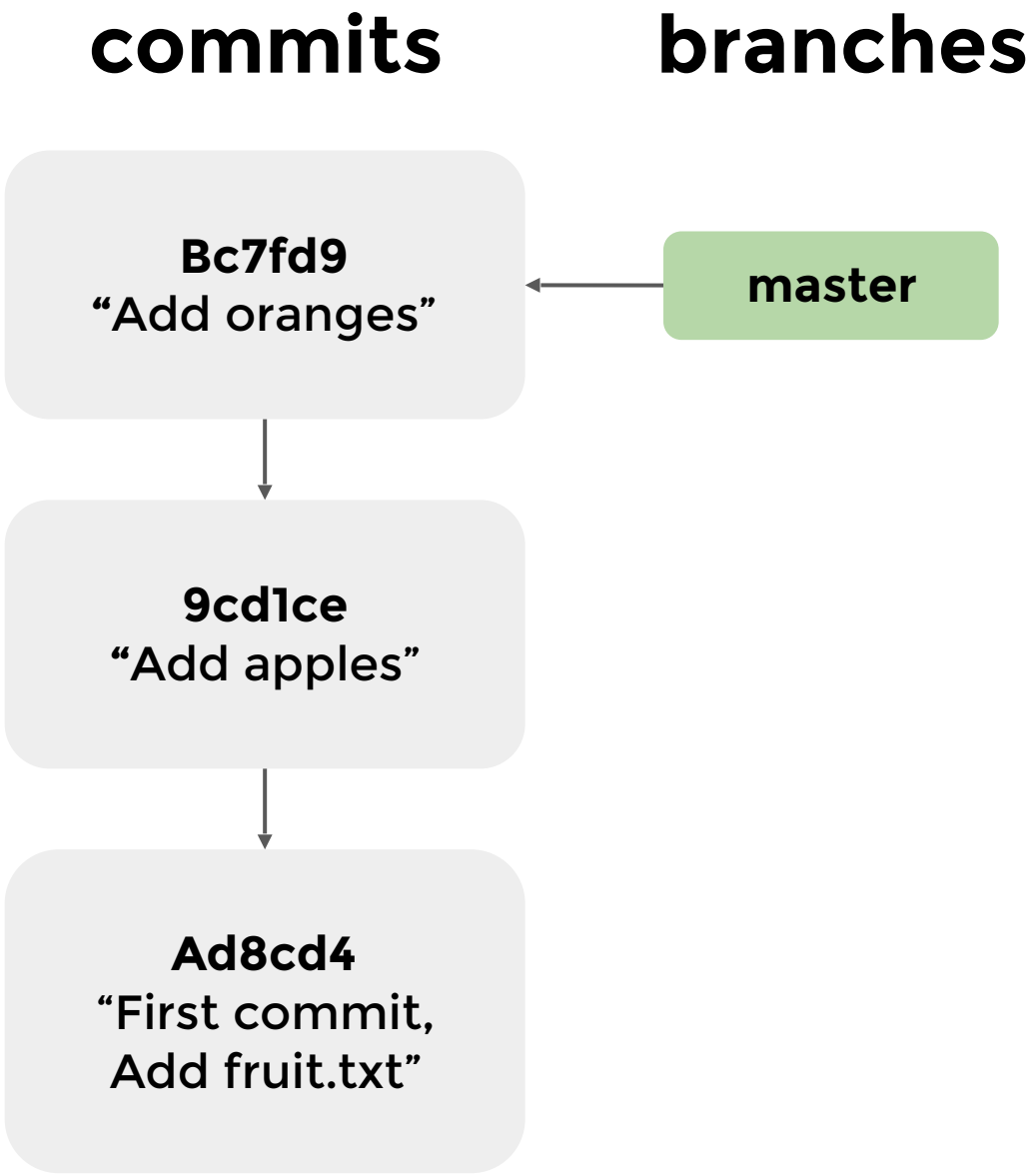
git **good**

branching, context switching, merging

@sdaniel55

commits

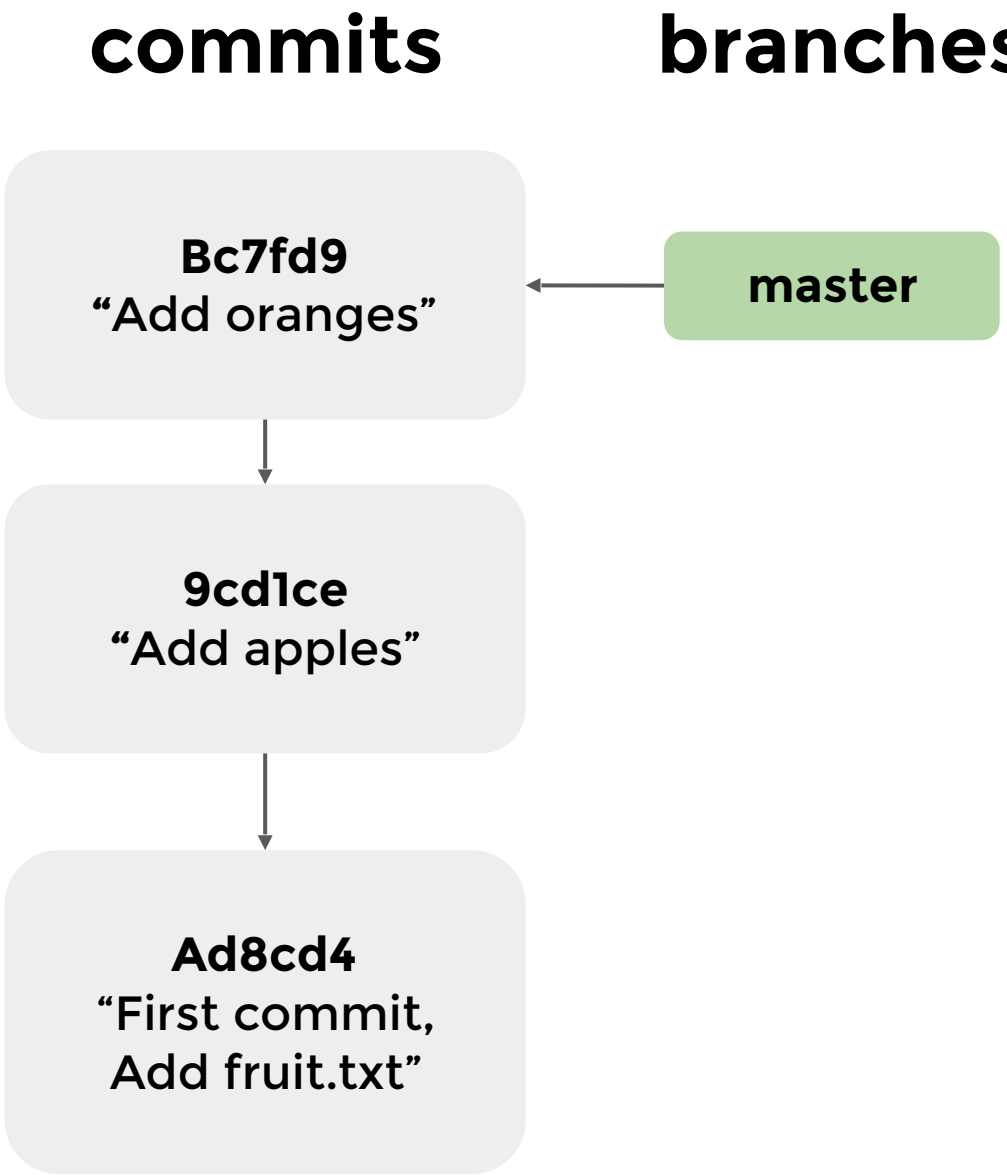

Bc7fd9
“Add oranges”
9cd1ce
“Add apples”
Ad8cd4
“First commit,
Add fruit.txt”


commits
branches
Bc7fd9
“Add oranges”
master
9cd1ce
“Add apples”
Ad8cd4
“First commit,
Add fruit.txt”

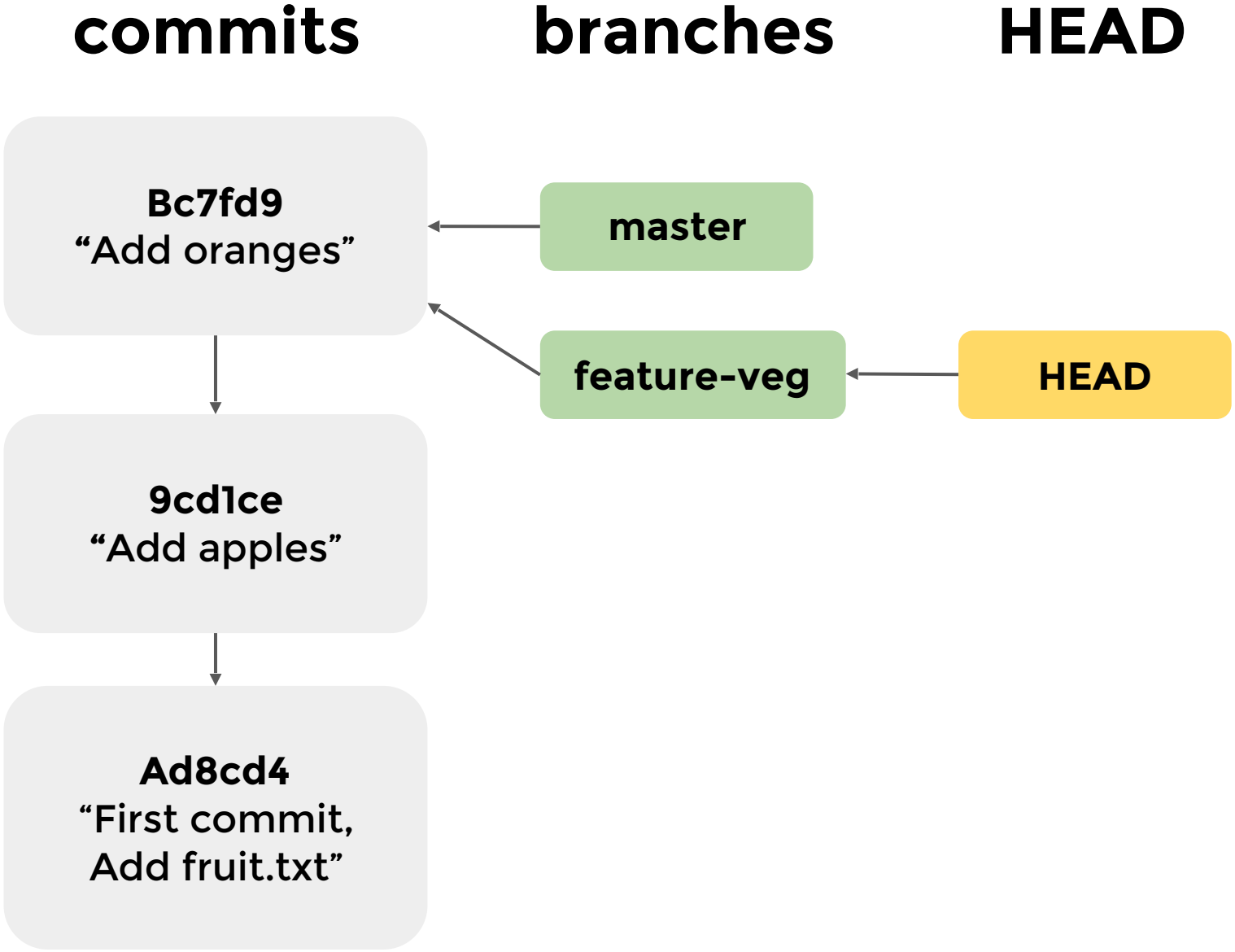
Working with branches

`git branch` - see the names of all available branches

`git branch <branchname>` - create a new branch called “branchname”


commits
branches
Bc7fd9
“Add oranges”
master
9cd1ce
“Add apples”
Ad8cd4
“First commit,
Add fruit.txt”

@sdaniel55


commits
branches
HEAD
Bc7fd9
“Add oranges”
master
feature-veg
HEAD
9cd1ce
“Add apples”
Ad8cd4
“First commit,
Add fruit.txt”

@sdaniel55

Working with HEAD

`git checkout branchname` - move HEAD to point to a branch.

Think of it like this:

“Make my working repository look like <this> branch”

Merging - combining commits

`git merge <branchname>` - create a new commit, combining the latest commit pointed to by **HEAD**, with the latest commit pointed to by the **other branch**.

Gitting Good: Recap

- **`git branch`** - list all branches
- **`git branch <name>`** - create a branch
- **`git checkout`** - jump to a branch
- **`git merge --no-ff branchname`**
  - merge another branch into current

git **better**

rolling back, using GitHub

@sdaniel55

Downloading / Uploading Repos

`$ git clone <url>` - download a copy of a git repository from a website.

`$ git pull` - merge the current branch with the most up-to-date version of this branch on the remote server.

`$ git push` - tell the remote server to to merge your latest version of the branch into its own.

Rolling Back - reflog & reset

`git reflog` - view the history of moves the current branch pointer made.

`git reset --HARD <hash>` - move the branch pointer to a particular hash

Let's Recap

- **Gitting Started:** repository creation, staging, committing
- **Gitting Good:** branching, jumping around, merging
- **Git Better:** cloning, pull & push

Let’s Collaborate on GitHub

@sdaniel55

git.io/cfg-surrey

@sdaniel55

Let's COLLABORATE on GitHub

Go to github.com

Create your account or Log in...

Once you are logged in visit this repository:
git.io/cfg-surrey

Fork this repository using the fork button in the top right of the repository

Now go to **your profile** & click on **your forked repository**
The top of the repository will now have <your username>/**cfg-guestbook**

Add your name & GitHub handle (username) to the **README.md**

Commit your changes - with a little note and by all means add some emoji wherever you like

Let's make a PR

Now I'm going to show you what happens in a PR:

First I need you to go back to the original repository:
git.io/cfg-surrey

We view the Pull Requests tab

Click the Create a new Pull Request button

Find your version of our files by choosing the compare across forks option

The choose the BASE (what you want to merge into) & HEAD forks (the fork you want to take the tasty stuff from)

Once you have done this add a label and make the request
We will merge them TOGETHER! Yay community!

What's **next**?

GitHub Learning Lab

lab.github.com

@sdaniel55

GitHub Pages

pages.github.com

@sdaniel55

Student Developer Pack

Get TONS of FREE STUFF to build cool things with!

education.github.com/pack

@sdaniel55

GitHub Campus Experts

Get FREE training from industry pros & let us help you develop your community on Campus!

githubcampus.expert

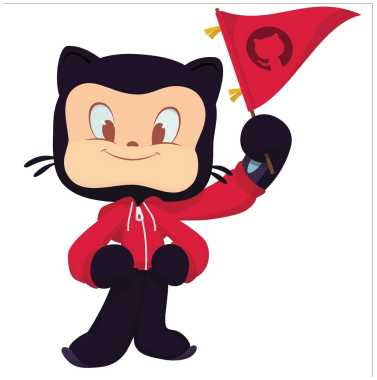
@sdaniel55

Draw an Octocat!

Draw a picture of Mona and tweet it at **@GitHubEducation with both the event's hashtag and #MyOctocat**. The winner at each event will receive an Octocat Statue!

@sdaniel55

Thank You!



Go forth & git good!

Feel free to ask me anything:

@sdaniel55