git good

Amy Dickens
@RedRoxProjects

(Hacknotts 2015)

The University of **Nottingham**

UNITED KINGDOM · CHINA · MALAYSIA

UoN

🎓 **2015 - PhD Computer Science**

(Mixed Reality Laboratory)

🚩 **2016 - GitHub Campus Expert**

(GitHub Campus Rep)

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

2016 - HackCon EU (Eindhoven, Netherlands)

2016 - JacobsHack (Bremen, Germany)

2016 - Hackference (Birmingham, UK)

2016 - BrumHack (Birmingham, UK)

2016 - AstonHack (Birmingham, UK)

2016 - HackNotts *

2016 - GNM Design Hack (Newcastle, UK)

2016 - DevRel Con (London, UK)

2016 - SexTech Hack (London, UK)

2017 - DurHack (Durham, UK)

2017 - Inspire WiT Conference *

2017 - Hack UPC (Barcelona, Spain)

2017 - Anvil Hack (London, UK)

2017 - CodeMotion (Amsterdam, Netherlands)

2017 - GitHub Satellite (London, UK)

2017 - UK GovHack (London, UK)

2017 - SciFi Hack (London, UK)

2017 - HackCon EU (Birmingham, UK)

2017 - Grace Hopper Celebration (Orlando, USA)

2017 - GitHub Universe (San Francisco, USA)

2017 - Lovelace (London, UK)

@redroxprojects

I ❤️ community

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

Before we begin:

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

Don't Panic

There's always a cheat sheet!

🐣

My first .git

repository creation, staging, committing

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

Let's take a look at GitHub
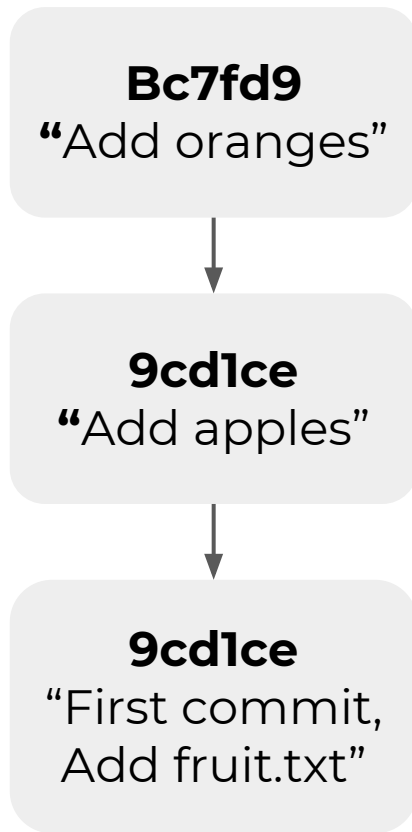
Created with

`$ git init <repository name>`

Commits :

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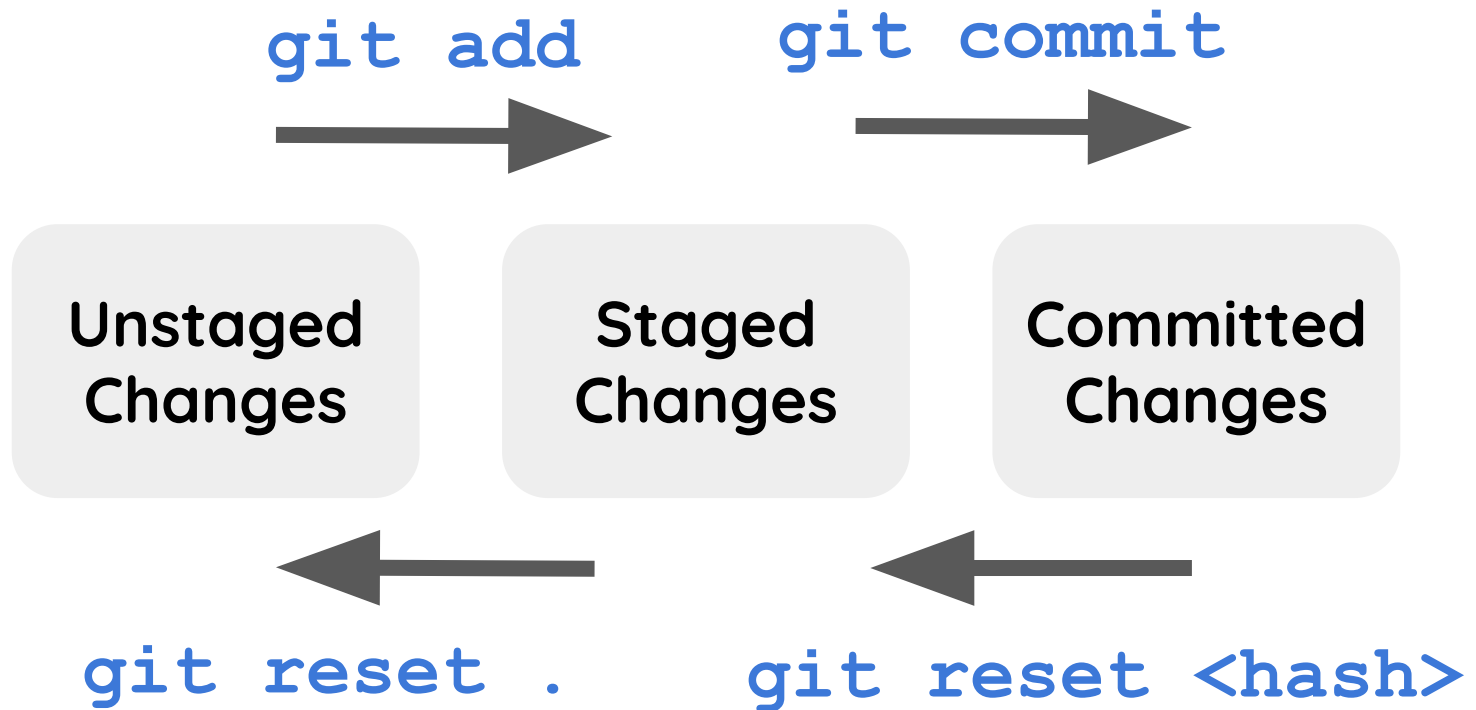
Created with

```
$ git add .
```

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

More on Commits

Bc7fd9
"Add oranges"

↓

9cd1ce
"Add apples"

↓

9cd1ce
"First commit,
Add fruit.txt"

- Commits form a **linked list** structure

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

Staging

git add   git commit

Unstaged
Changes   Staged
Changes   Committed
Changes

git reset .   git reset <hash>

(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
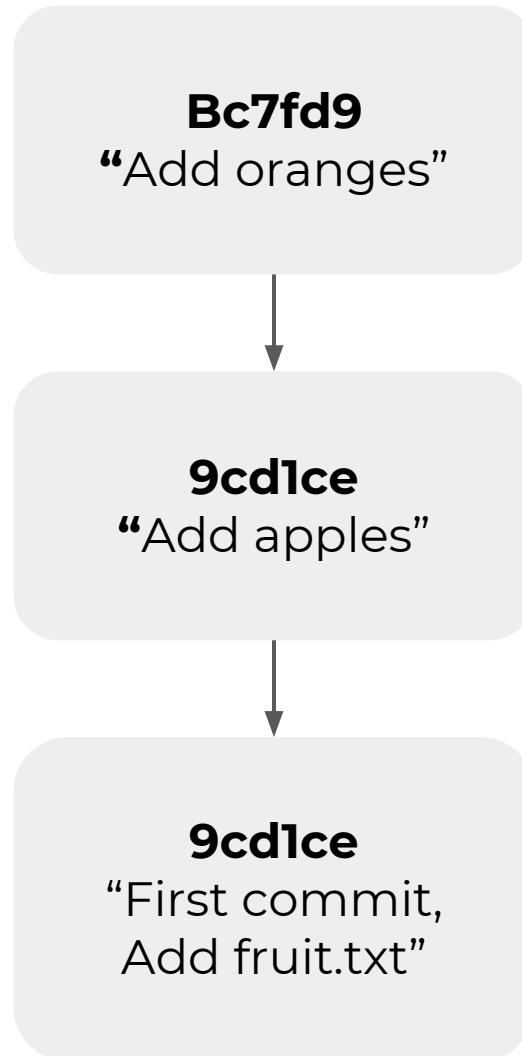- **git log** - see all commits
- **git status** - see what is staged

@redroxprojects
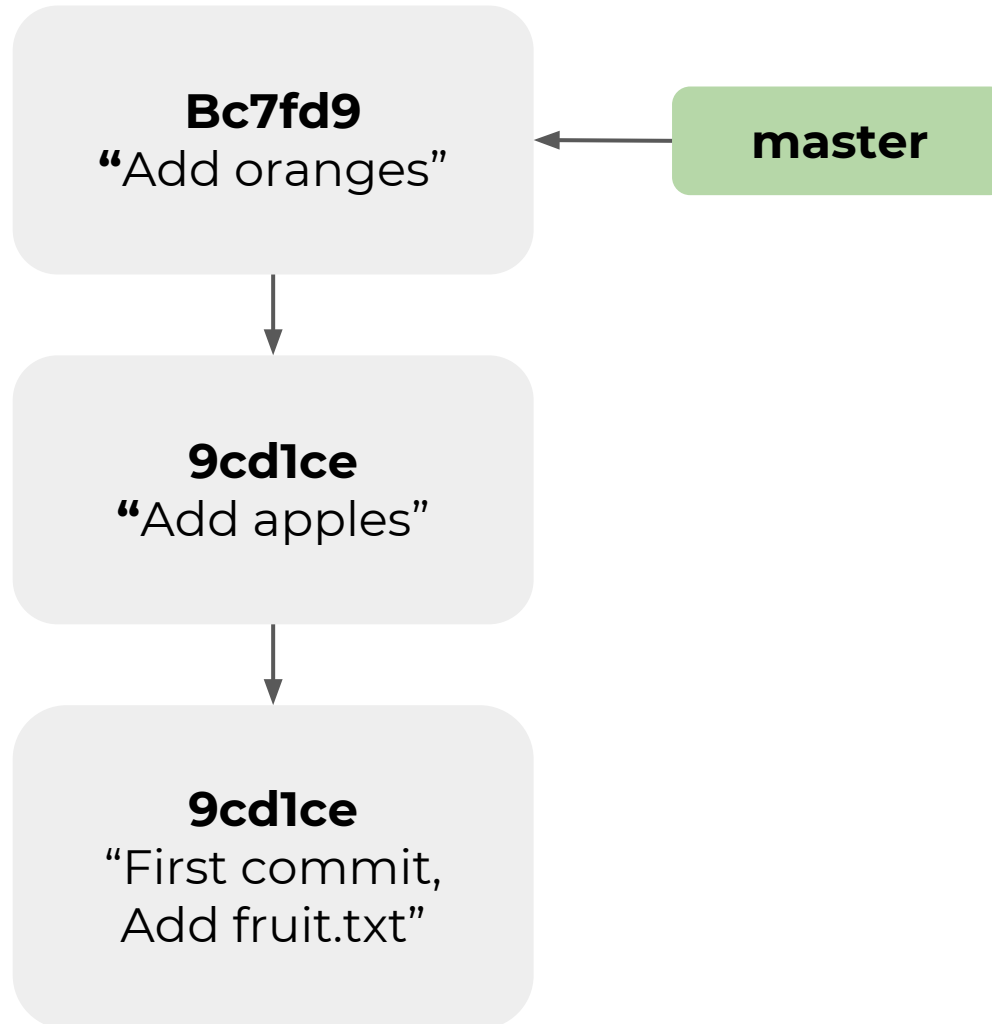
Let's take a look at GitHub

👍

git **good**

branching, context switching, merging

commits

Bc7fd9
"Add oranges"

9cd1ce
"Add apples"

9cd1ce
"First commit,
Add fruit.txt"

commits   branches

Bc7fd9
"Add oranges"

master

9cd1ce
"Add apples"

9cd1ce
"First commit,
Add fruit.txt"
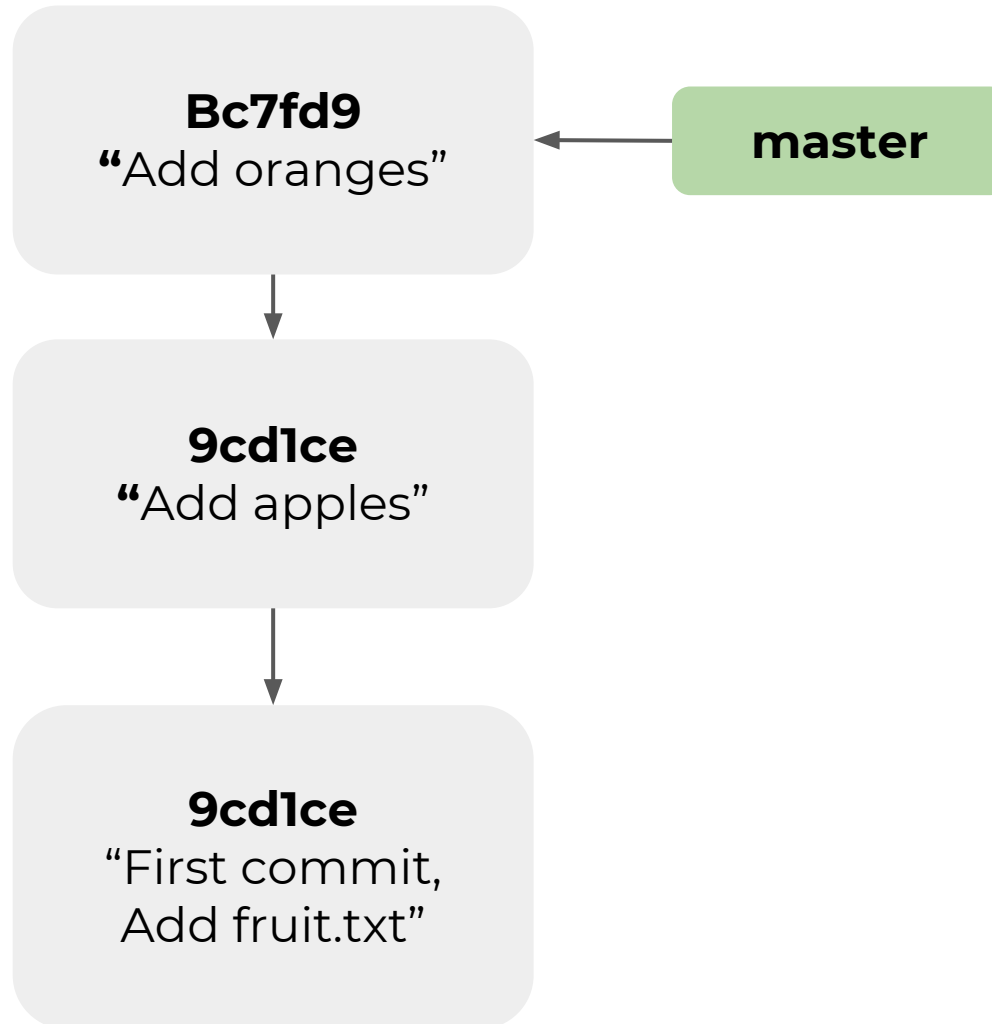
@redroxprojects

Working with branches

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

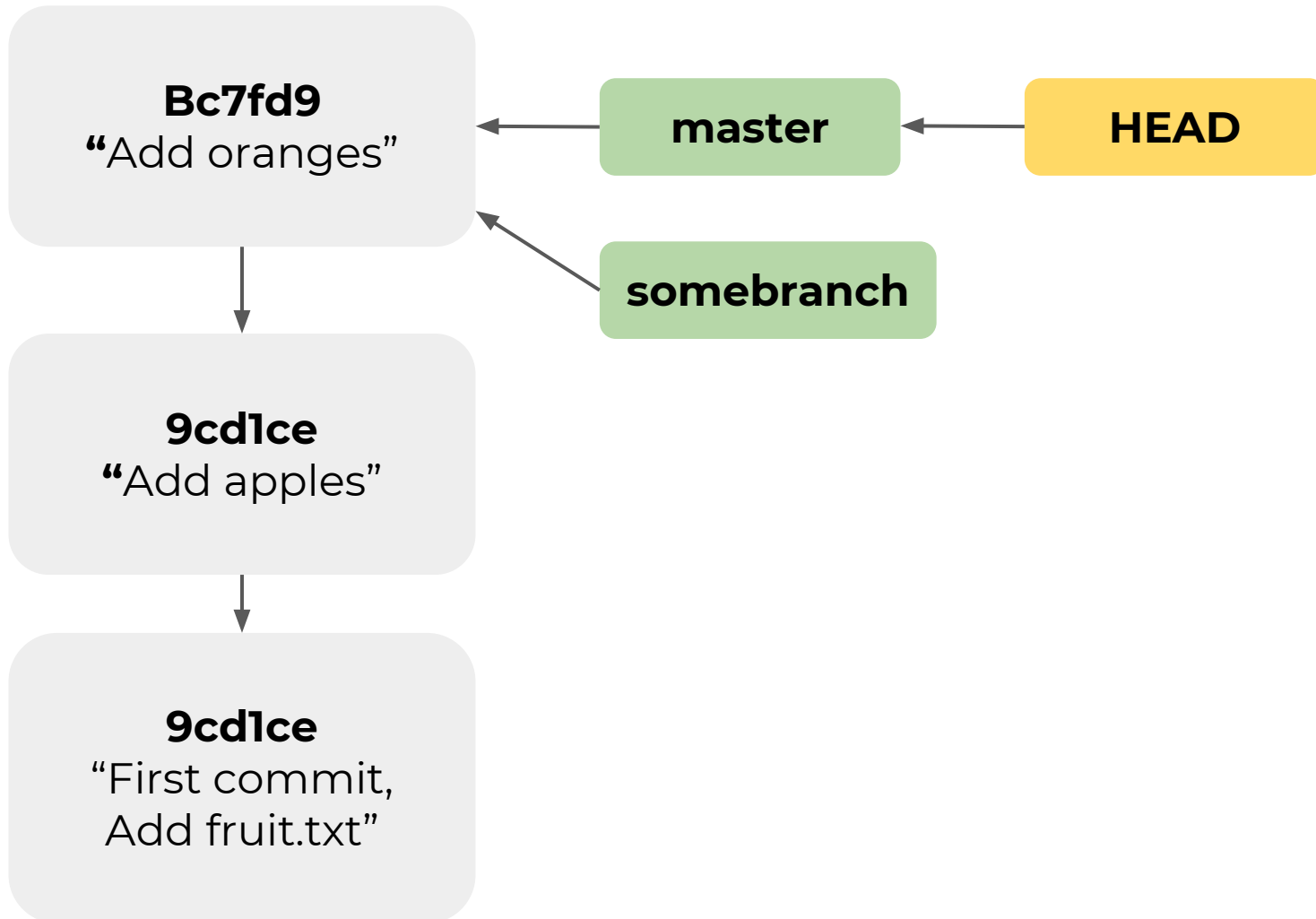**`git branch <branchname>`** - create a new branch called "branchname"

commits   branches

Bc7fd9
"Add oranges"   ← master

9cd1ce
"Add apples"

9cd1ce
"First commit,
Add fruit.txt"

@redroxprojects

commits   branches   HEAD

Bc7fd9
"Add oranges"

master ← **HEAD**

somebranch

9cd1ce
"Add apples"

9cd1ce
"First commit,
Add fruit.txt"

@redroxprojects

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

Let's Collaborate on GitHub

Let's COLLABORATE on GitHub

Go to www.github.com

Create your account or Log in...

Once you are logged in visit this repository:
github.com/redroxprojects/24-pull-requests

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>/**24-pull-requests**

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 your first PR

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

First I need you to go back to the original repository:
github.com/redroxprojects/24-pull-requests

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!

Recap

- **Baby steps:** repository creation, staging, committing

- **Basic git:** creating branches, jumping between branches, merging

- **A spoonful of community:** collaborating on GitHub by forking & making a PR

git **ideas**

A quick look at what other folks are doing on GitHub

[Software for Science on GitHub](#)

Go & Explore the things that are on GitHub…. I think you'll be surprised what you find!

Collections on GitHub

Go & Explore the things that are on GitHub.... I think you'll be surprised what you find!
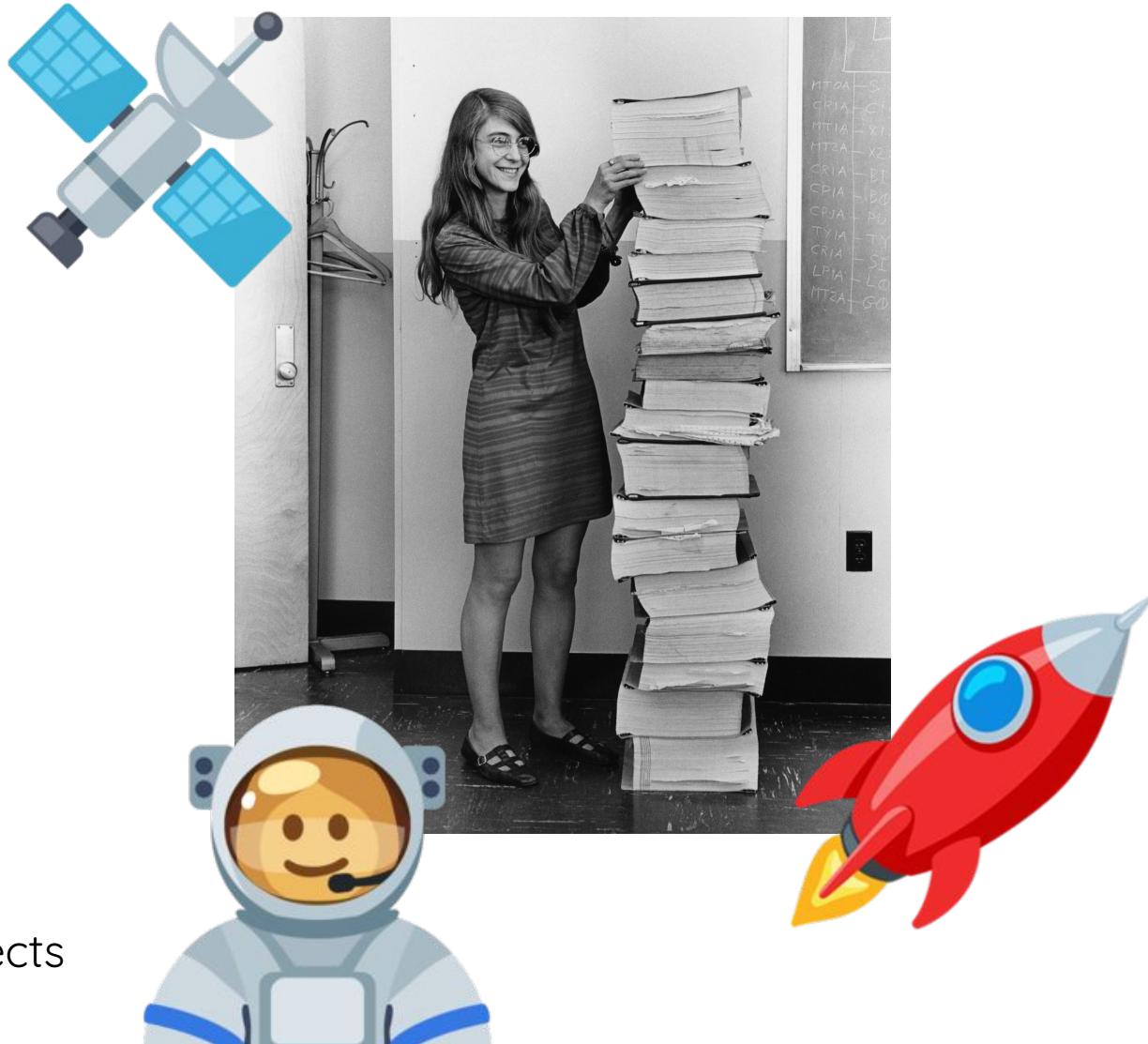
Social Impact   Learn to Code   Organisations   Open Journalism

And SO MUCH more!

@redroxprojects

The Original Apollo 11 code on GitHub

@redroxprojects

git **more**

More awesome things you can do on GitHub

Organisations & Community Groups

Build & Host your own website with Pages

@RedRoxProjects
An Open Source Community Checklist

- ❏ Be Patient - Maintainers are real people not bots!

- ❏ Be Open - Share your knowledge

- ❏ Be Observant - Check the issues first

- ❏ Be Mindful - Consider your tone when communicating online

- ❏ Be Aware - Know the CoC & Processes

- ❏ Be Respectful - Treat people as you wish to be treated

- ❏ Be Curious - Ask the community

@redroxprojects

P. O. O. M. A. R. C

some inspiration

Thank You!

Get your STICKERS...

Go forth & git good!

I'm here to help:

@RedRoxProjects

amy.dickens@nottingham.ac.uk