Table of Contents

Welcome to Mozilla Science Lab's Study Group Orientation!

Hello! Thanks for dropping by.

You're reading an online handbook for the Study Group Lead Orientation. **A Mozilla Study Group is an informal, in-person meet-up for scientists and researchers who want to know more about code and open research practice.** This training will help you start your own Study Group.

How to Use this Training and What to Expect

This series is designed for completion online, as a solo training experience, done at your own pace. There are five sections, some with videos and assignments or activities to read and work through. You'll probably need ~4 to 6 hours to complete this training. If you'd like to connect with experienced Study Group Leads or other newcomers like you, to network and compare notes, see the below, in "How to Connect and Get Help"!

Here's what's in the book:

- In the first section, you'll get a sense of what Study Groups are all about, and why you might want to lead one.
- In Section 2 you'll get tips on how to facilitate great Group meetings and learn about formats for sessions.
- In Section 3, we'll walk you through how to work with the collaboration platform GitHub. You'll set up your Group project's GitHub-powered website, and a collection of files called a repository. These will enable you to work together and share materials and code.
- In Section 4, you'll get a whirlwind introduction to some of the key issues and resources related to "open" research.
- In Section 5, you'll learn more about Mozilla, the Mozilla Science Lab, and why we're so excited about Study Groups and open research.
- And, finally, in Section 6 you'll get tips on how to grow your group, from communication and recruiting strategies to planning sessions and meetings over the

long term.

Before you get started, check our map to see if there's already a Study Group in your area.

whereWeAre.geojson rendered with by **GitHub**

If there's a Group nearby, get in touch with them and attend a meeting. If not, keep reading!

How to Connect, and Get Help!

We encourage Study Group Leads from different groups to connect and share resources! We aim to gather a group of aspiring Study Group Leads to work through this training twice a year, roughly aligned with the Fall-Spring Academic Calendar.

If you're joining us at one of those times, please join us for **Office Hours** (check the schedule here). A Mozilla Staff memeber and a seasoned Study Group Lead will be available by video chat or phone to answer your questions.

You can also join our **Chat room on Gitter mozillascience/studyGroup**, a web-based chat client that's associated with the Study Group Rep. Sign-up using your GitHub ID to join! The Gitter chatroom is open 24 hours a day, and it's a great place to ask questions, share news and ideas, and just hang out with other Study Group Leads. (If you've never used GitHub before, that's OK! You'll learn all about it in Section 3 of this training!)

We hold regular **Study Group Calls** to share and discuss new resources, materials and ideas for Study Groups, to plan Sprints and other inter-group collaborations, and to network and connect! These are video conference calls, but you can also dial in toll-free by phone. Check our Events page for the details about the next call.

Finally, if you have a question or problem, you can reach out to **Zannah Marsh, Mozilla's Learning Strategist, or Aurelia Moser, Mozilla's Community Lead**. The best way to do this is file a GitHub issue in this Study Group Orientation repository, or ping us on the Gitter Chat. This is basically a comment or question that we'll see and respond to. (Again, it's OK if you're not familar with GitHub and GitHub issues-- all will be explained in Section 3.)

Help us grow and improve this book!

Check out ourissues (a list of items that need help or fixing), to see what you can do to improve or add to these materials. If you're new to GitHub, again, a quick look at section 3 should help explain how this works. **We'd love your help!**

About Study Groups

A Mozilla Study Group is a regular, in-person, informal meet-up where people get together to learn to code and to explore open research practice-- strategies for maximizing the quality and impact of research in any field, from astrophysics to sociology. Coding skills in particular are increasingly important in data-rich research environments, but many of us don’t have formal training in computer science... and the tools and technologies we need are all constantly changing. **Mozilla Study Groups are peer-to-peer learning environments for skill-sharing, idea discovery, community support, and collaboration.**

Here’s what a few different study groups look like in real life:

From Sara, at the University of Wisconsin Study Group in Madison, Wisconsin, in the USA:

From Tom, at Boston University's Study Group in Boston, Massachusetts in the USA:

Why join a Study group?

- Learn and share new coding skills you need, learn and problem-solve faster with others
- Super-power your research through collaboration and open science best practices
- Be part of the discussion around transforming research practice
- Connect to a network of research leaders across disciplines
- Make friends and become part of a supportive learning community

Why start a Study Group?

- Learn teaching and facilitation skills, community building skills
- Get support and mentoring from Mozilla Science Lab and the Mozilla network
- Show leadership in your community, gain skills to advance your career
- Access resources like our beautiful website template and library of coding lessons
- Add some open source sparkle to your work by association with the Mozilla brand

There are Mozilla Study Groups all around the world, many at academic institutions and research organizations. Anyone can start a group. The format and schedule for study groups is flexible: you can modify it to fit the needs of your members. It’s all up to you!

Mozilla Science Lab helps out by providing this training, an easy-to-use template for a website, templates for advertising materials, and by managing an awesome collection of lessons created by Study Group members from around the world. We organize regular calls where study group members can connect and share what they’ve been doing with others. MSL also supports other community events and gatherings (you’ll learn more about these in Section 6).

Continue to the next section to learn more about the Study Group Lead.

1.1 The Study Group Lead Role

The work you'll do as a Study Group lead will ramp up slowly, because most groups start small. Your tasks will change over time as your group grows and changes. As Study Group Lead, you'll do a mix of the following tasks:

- **Outreach and communications**, as you find your first few members and getabout the group, and as you publicize each of the group's meetings and events;
- **Logistics and event coordination**, as you organize group meetings and events;
- **Strategy and planning**, as you create a schedule and plan for your group over a semester or year, and tweak that plan to respond to the needs of your group;
- **Teaching**, as you prepare some of the materials and facilitate skill-sharing sessions for your group;
- **Community-building**, as you find ways to invite new members in and expand your group;
- **Mentoring**, as you create a group that leverages the skills and abilities of all members in co-teaching, co-facilitating, and co-organizing-- in other words, you get lots of help!

This last point about sharing responsibility and mentoring others to help lead is very important for the long-term survival of your Study Group! Bringing on others to co-lead will help you create a sustainable group, one that will live on if you decide to move on to a new job or a different institution.

You're not required to be an expert coder to lead a Study Group! Some coding experience is helpful, but you mostly need to be curious, motivated, and willing to jump in, get your hands dirty, and learn and share with other group members. To lead a successful Group, you'll also need to be consistent (showing up on time, holding regular meetings). You'll need to be patient and persistent while your Group gets off the ground, and willing to dedicate time to the project-- usually about a few hours a week.

In return you'll get valuable experience in communications, event planning, teaching, and working collaboratively with other researchers, not to mention improving your own research practice! And you'll expand your professional network and connections into a broader research community.

1.2 Assignment: Write a README file to Welcome All Members

The README file (whose name by convention is written in all-caps, and represents a request to one and all--- “read me!”) is one of the first things that potential members will encounter when learning about your Group. This file will live on the web, as part of a collection of working documents on your Group, called a “repository” (you'll set this up in Section 3).

The aim of the README is to welcome, orient, and encourage newcomers to participate. You’ll likely start out with a few members you know well, people from your lab or department. It’s fine to start in familiar territory-- most groups do. But your Study Group should welcome anyone who’s interested and shares your goals of learning to code together, and improving research through open practice.

If you make your group welcoming to researchers in other departments, labs, and disciplines, you’ll likely find that researchers of all stripes want to learn the same or similar tools, and share the same kinds of research problems around data analysis and visualization, data management, collaborative workflows, and much more. **Don’t feel limited by your field or area of expertise-- this is a great opportunity to make interdisciplinary connections!** As you write your Group's README, keep a diverse audience in mind.

Write your README

IN your README, Be sure to:

- **Say hello!** Welcome people to the Group. It’s great to introduce yourself here, so people know they’re dealing with a person or group of real people. Let potential members know you’re excited that they’re here to learn more.
- **Write a SHORT group description.** In your own words. Try to phrase this so it’s understandable and appealing to a wide variety of people, not just those in your field. Maybe add sentence or two about your focus for the semester or year, etc. If you're having trouble getting started, here's some basic text-- feel free to copy-paste and edit: *"This Mozilla Study Groups is a fun, informal meetups of friends and*

colleagues to share skills, stories and ideas on using code for research, and exploring open research practices. The goal is to create a friendly, no-pressure environment where people can share their work, ask for help on a coding problem, and learn and work together with their peers."

- **Include some key details.** Note any regular meeting times and places, contact people, etc.
- **Ask for what you need!** If your group wants to learn or work on a certain topic, but doesn't have the expertise, mention that here-- that way, perspective members can see if their skills are needed!
- **Test your README for jargon** When you're working in any field, whether it's software engineering or astrophysics, you'll learn and use jargon -- terms that have a special meaning to your field but likely won't make sense to anyone who isn't part of that field. Too much jargon can confuse newcomers, so use simple langauge and define all potentailly confusing terms here.
- **Spread the word about the program** While you repo will be used mostly by your own Group's members, some visitors may be interested in starting a group of your own. We'd appreciate it if you'd include the following text, or something like it, to encourage new groups to form. *"If you're interested in starting a group of your own, you can check out the Study Group Repository on GitHub or our Study Group Orientation Guide."*

Hang on to the README you just wrote! You'll use it in Section 3, when you get your Group's Website and Repo online.

1.3 Assignment: Writing a Code of Conduct

Mozilla Study Groups should be welcoming environments for all participants, regardless of their gender, ethnic background, sexual orientation, age, skill level, preference for text editor or coding langauge, etc. Study Groups should get everyone learning together!

Take some time to think the learning environment you'd like to create by writing a **Code of Conduct**, also known sometimes as **Participation Guidelines.** This document describes your ideal Group culture, and defines social norms and responsibilities for members of your Group. The guidelines apply to in person interactions and those that happen online, in forums, chat, via email and in your Study Group repositories.

It may seem unnecessary to have these, but when working in a collaborative group, it is always possible (and maybe inevitable) that tensions may arise! You may have seen this in your own research labs or classes. A Code of Conduct puts people in the right mindset to work together.

A Code of Conduct should:

- Establish the positive behaviors encouraged by the community
- Make clear which behaviors are unacceptable and discouraged
- Define a process by which any problems will be handled.

Write your Code of Conduct

1. **Brainstorm.** Reflect on the following questions.
2. **What core words would you associate with the Group you hope to create?** These could be values, ideals, or characteristics, and behaviors. Try to keep these to one word answers, if possible. (Examples: friendly, productive, relaxed sharing, helping, asking questions, etc).
3. **What behaviors to discourage?** Be specific enough to be useful here. (interrupting, put-downs, unwanted physical contact, sexist comments)
4. **What should someone do if they have an issue or problem with a behavior in the group?** This may be as simple as speaking to the group leader, or sending a

email or text. Be explicit in these steps. It’s OK (and encouraged) to have a few different options for reporting issues.

5. **What consequences are there for violating the code?** It’s important the consequences are real and appropriate to the situation. People might be warned, or asked to leave the session, or they may be asked to leave the Group.
6. **Who decides what does and does not violate the code? What's an example of how this might be done?** As Group Lead, you're primarily responsible for making sure the group goes well, but as you add members, you might ask another trusted member to help you make these decisions.

7. **Refine and Remix** Using the information you’ve collected in the brainstorm above, write your own Code of Conduct. Remember, Codes of Conduct are about creating a welcoming and safe, and postive environment as much as they are about discouraging bad behavior. As you write be sure to highight the desired values of the group. You're also welcome to refer to or remix our Participation Guidelines.

8. **Share and Discuss** If you already have a few members, share these and discuss them, and revise as needed.

Teaching and Facilitation for Study Groups

The most successful study groups are welcoming, friendly, personable environments. **Complete newcomers to coding should feel comfortable, and a bit (or sometimes a lot) of confusion is expected as you work through new concepts and tools-- it's part of the learning process.**

This isn't a space to show off your own skills or an opportunity to outshine others, it's a mini-community where you're all getting to know each other, and working together to help each other learn. It helps for Study Group Leads to create this space with intention, as it may not be typical of some research and work environments. Even if another member is presenting material, the Study Group Lead should be in a back-seat facilitator role, gently and thoughfully helping to ensure the meeting goes well.

In this video Madeleine Bonsma-Fisher, of the Unviersity of Toronto Coders group, talks about how she worked to create a welcoming Group environment.

Here are a few more tips for facilitating Group meetings:

- **Arrive Early.** Get to the meeting space 10 minutes before the start time so you can set up, and be sure to give arriving members some time to greet each other and get settled before you jump into the session.
- **Greet newcomers** Say hello! Introduce yourself to any newcomers and welcome them to the Group, or have someone assigned to do so at each session.
- **Make Introductions.** Ask participants to introduce themselves to the Group. Ask them to briefly describe their research practice, tools they use, and any skills they have, or want to gain. This is a great way to discover common interests among members.
- **Be relaxed and friendly.** Make sure your tone of voice and body language are open and positive.
- **Encourage questions!** Making time for dialog and questions speeds learning for all members as you come to answers together. There are no stupid questions!
- **Slow down.** Slow the pace of lessons and presentations down if people seem to be struggling-- there’s no rush.
- **Share successes AND failures.** If something isn’t working, encourage members to share! It’s an opportunity to practice debugging skills together. The more eyes on the problem, the better!
- **Orient newcomers to Group Systems** Make sure newcomers know where the website is, and tell them about the GitHub repo, email lists, social media feeds and groups, and any other systems you use.

2.1 Leading a Session

The most successful, sustainable Study Groups leverage the experience, skills, and interest of the entire group.

Remember, as Study Group Lead you are *not* expected to teach all or even most of the sessions. A member doesn't have to an expert in a topic to lead a session-- they just need to be motivated, have a bit of background and a willingness to explore the topic with others. The relaxed, non-competitive tone you create for your study group should help members feel comfortable stepping up to lead sessions. If you think a member might have a certain skill or technology to share, invite them to lead a session on it. **Leading a session is a great way for those new to teaching to get some practice in a low-stakes environment-- experience that can be very valuable in an academic career!**

To help you plan sessions and organize your materials, and to ensure that all is usable and useful for others, we've created this **Lesson Template** (Indonesian translation.) It's part of the Study Group Repository) for you to use to structure any notes, guides, challenges, or examples you share in session. Once you've created your lesson notes using the template, and tested out your lesson with your Study Group, we hope you'll add it to the Study Group Lesson repository. This repo enables skills-sharing among Study Groups worldwide. Anyone can access these open resources; it's a way of maximizing the learning potential and power of all our Groups, worldwide. **Explore the Lesson Repository here**.

In additon to the template, we've detailed formats and provided tips for new session leaders in the next section. Share these with any prospective leads. Remind members that it's most useful and authentic to see how they use a tool or skill in their research practice; they don't need to present an in-depth, advanced 90 minute lecture on a topic.

Each session will be a bit different, but here are three excellent guiding principles for leading a session, developed by the University of Toronto Study Group:

- **Keep the code you're working with as simple as possible while still covering the concept.**
- **When making the code, make no or few assumptions about the knowledge of the audience.**

- **Keep the code generalizable. Members may come from diverse fields of research. What we share is the need to code.**

Do you have tips or suggestions for running sessions we haven't mentioned here? Tell us about it here!

2.2 Options for Study Group Sessions

As mentioned in module 1, “Introducing Mozilla Study Groups,” the format and schedule for study groups is flexible. You can modify it to fit the needs of your members, and revise or change it depending on what you find works best for your group. Here some formats that have worked well in Mozilla Study Groups to date. Got a great idea for another format? Tell us about it here!

Format: Work-Along
Description: A hands-on guided tour of a piece of code, tool, or other skill.
Time Required: 1+ hours to prepare; event time: 1 hour.
How To Do It: In this skill-sharing model, one group member demonstrates how they use a particular piece of code, tool, or other skill in their research, while everyone else follows along. It’s best if all members have laptops to work on, and the person who’s demonstrating can project their own screen for easy viewing. Work-alongs help your community discover new tools and see novel solutions to common problems. The presenter should have a plan, a set of notes, and some ideas for basic challenges or problems. it’s great if this documentation can follow the Study Group Template (see below), but remember this is not a lecture, it’s really all about learners trying out the skill themselves. See our list of tips for Work-Along presenters.

Format: Co-Working session
Description: An open session where members work on separate projects, but help each other with problems and questions, and share successes.
Time Required: no time to prepare; event time: 2 hours
How To Do It: Get your Study Group together for a couple of hours. Start the session by having each member share (very briefly) what they are working on. After intros, everyone gets to work on their own project, but each member is encouraged to ask questions if they’re stuck, and share successes while they work. Co-working sessions are easy to run, and the format encourages collaboration, skill-sharing, and hands-on problem-solving in a casual, low-pressure environment… all while members to make progress on their own work! See our list of tips for co-working facilitators here.

Format: Strategy Circle
Description : A go-around where members describe a particular problem or issue they are facing, and the group together talks through solutions to each problem.
Time Required: minimal time to prepare; event time: 1 to 1.5 hours

How to Do it: Let members know in advance that they can come with a challenge or problem. One by one, invite members to describe their problem, and then open the floor for suggestions and a discussion of what kind of packages, tools or strategies they could try out to solve their problem. After each problem is discussed, spend the remaining time experimenting with those new ideas and getting help in a co-working session with the entire group.

Format: Hacky Hour
Description: A pub night or cafe meeting for researchers who are curious about code and open practice.
Time Required: minimal time to prepare (sending out invites); event time: flexible
How To Do It: A Hacky Hour is an informal, social get-together where community members meet one another, discuss new ideas, and build the personal ties that help strengthen any community. Hacky Hours are the easiest events to organize, since all that's required is an open cafe or pub! Hacky Hours are great “first-time” events for those curious about Study Groups. Holding a few Hacky Hours is an excellent way to recruit new Group members. As host, you’ll be focused on welcoming attendees and making introductions and connections. See our list of tips for hosting a successful Hacky Hour.

Format: Lightning Demo Series
Description: A series of short (10-15 minute) introductions to a tool, programming library, or piece of code.
Time Required: time to prepare: minimal (identify/invite presenters from your group); event time: about an hour
How To Do It: Ask four or five members to prepare short (10-15 minute) demonstrations of their favorite tool, library, or piece of code. These demos are meant only as short introductions to allow the audience to decide if they’re interested in learning more. Unlike work-a-longs, the audience isn’t expected to try out the tool or code during the demo, they’re just engaged, interested spectators. The presenter should encourage questions, and be sure to provide context for the tool, explaining how they use it in their own research. Switching between potentially wildly different ideas in a lecture format can get exhausting for the audience after more than an hour, so limit your number of presenters. If you have lots of willing presenters, consider throwing a Lightning Demo Party.

Format: Lighting Demo Party
Description: A mixer event with a science-fair set-up where the audience can walk around to get (5 minute) introductions to a tool, programming library, or piece of code.
Time Required: time to prepare: minimal (identify/invite presenters from your group,

invite a lot of guests) event time: 1.5 hours.
How To Do It: Like the Hacky Hour, the Lighting Demo Party is an informal, social get-together where community members meet one another, discuss new ideas, and make friends. The main event is a series of 8 to 10 demos, set up around the room, with presenters at each station, ready to share and explain their tool or skill and chat with anyone who stops by. As above, presenters should be sure to introduce their research projects and the context in which they use the tool or code. These demos are shorter than those in the Demo Series, so people have time to experience each one during the event. Setting demos up in a big horse-shoe pattern lets people easily circulate, socialize, and talk about how they'll use all these great new tools and ideas. Serve snacks and drinks if you like, and be sure to invite lots of people, so you have a good audience for your presenters.

These are just a few possible formats for your Study Group Meetings. If there's another format you've used successfully or plan to test out, tell us about it here!

2.3 Planning your first meeting

Now that you've got a sense of what a Study Group session can look like, you're ready to plan your first meeting! Here's a list of planning tasks to help you get started.

- **Do a bit of audience research.** Poll some potential attendees to see if there's a topic that most people want to learn (like R or Python). If so, you can start with that. Or, check out our list of suggested first sessions.
- **Find a meeting space.** You can use a free classroom space or conference room, or you can meet in a common area like a student center, library, or even a local bar or cafe. Keep in mind the noise level and access to power and wi-fi.
- **Pick a time.** Schedule meetings when people are most likely to be around and free-- perhaps there's an obvious space in the schedule (after a lab meeting or a certain required course) when most people are on campus.
- **Decide how long you'll meet.**Depending on schedules, and what you want to get done, your study group could meet for an hour, an hour and a half, or more.
- **Send reminders.**Send out a quick email and/or tweet the day before each meeting, so people know it's on!
- **Be ready to welcome everyone!** A successful Study Group requires a set of members who know eachother and feel comfortable together-- start creating a postitve, friendly environment from the first meeting.
- **Meet regularly.** Mention your next meeting at the close of the first meeting! It may help to hold meetings on the same day of the week and at the same time each time you meet, so members can plan in advance and are more likely to remember meetings.

See Section 6 for more about scheduling over the longer term, and for tips for good Group communication.

3. Using GitHub

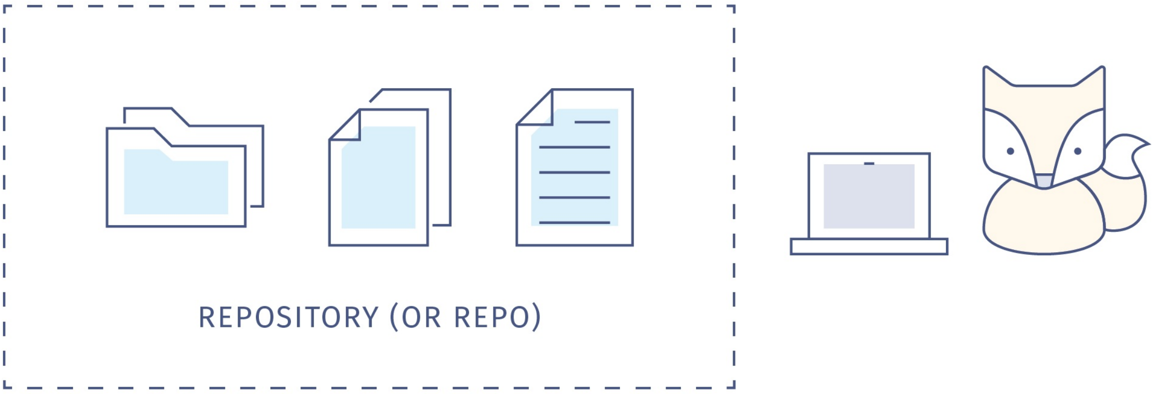
In addition to being a physical space for collaboration and learning, your Study Group will likely generate some content-- code, lessons, materials, as well as discussions, questions, announcements, planning documents, and more. For all these materials to be as useful and accessible to all your members as possible--- and to enable your members to contribute to them--they should live on the world wide web. **We recommend using the web-based software platform called GitHub, which is designed for collaboration and version control. GitHub is a place to discuss changes and issues related to that content, and collaborate on creating more content together.** You'll store the collection of files related to your Study Group, called a repository, on GitHub. GitHub also is the software that will power your Study Group Website. While GitHub was developed by and for software engineers, you **don't** have to be an expert coder to use it. This section will take you through some of the basics.

3.1 Collaboration, Version Control, & GitHub

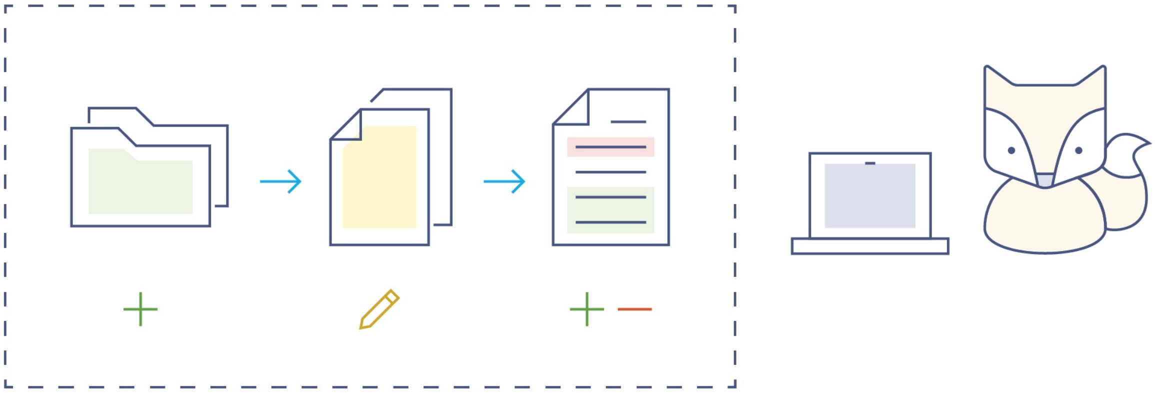
One of the main challenges in working with many people on a single project-- whether it's planning and running multi-institution research initiative to collect and analyze data or the process of co-creating, sharing, and updating learning materials for your study group-- is **"version control," the task of managing the many contributions your group makes to shared working documents.**

Your contributors may be spread around the world or working in the same room; they may be working simultaneously or asynchronously. No matter how your group is organized, the work of many contributors needs to be wrangled into a single project. Version control manages this process: it stores a history of changes and who made them, allowing you to revert or go back to earlier versions of those documents, and understand how contributions by different contributors have changed the project over time. You may have used word processing software that has a "changes," "history" or "revisions" feature, which also allows you to see and revisit any changes to the document: this, too, is a form of version control. **If you've never used GitHub or any other specialized version control software before, it may help to look at some diagrams and define some new terms before you get started.**

When we code, write text, or create any kind of content using computers, we end up with **a collection of files in a folder or directory, also known as a repository, or "repo."**



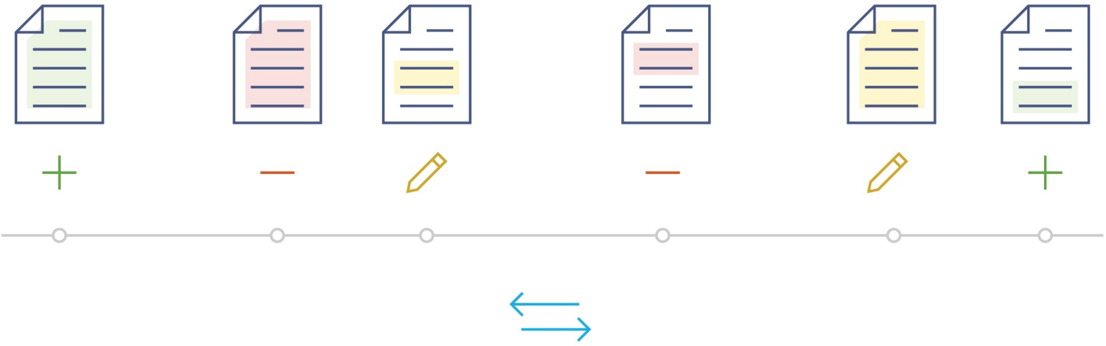
Even if you're working alone, you're probably going to make lots of changes to your content or code as you go-- you'll change some wording or functionality and leave others untouched, you'll make mistakes while you experiment with new ideas.



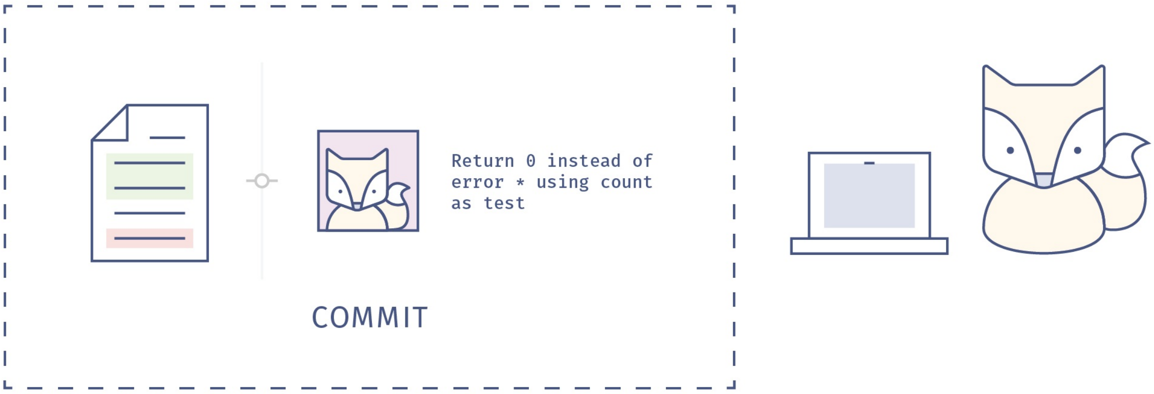
You might make multiple copies of your files to preserve a version that's working while you try to improve it or add functionality, but keeping track of all these versions and the differences between them becomes difficult.

?
work-copy.js
work-copy-filter.js
work-copy-count.js

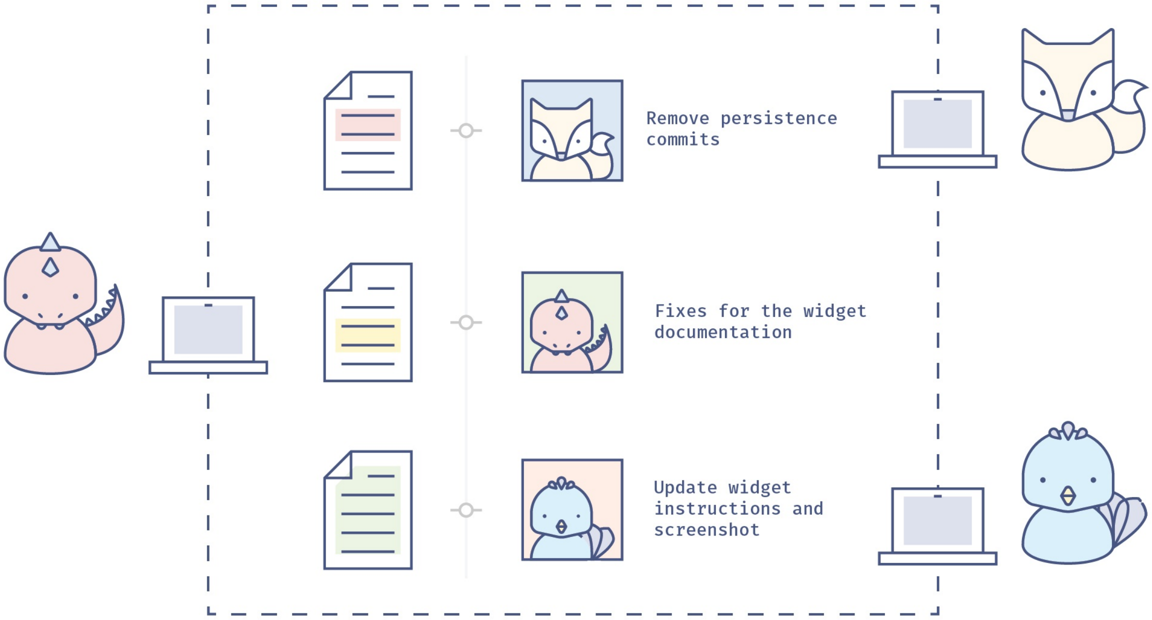
You need version control. Imagine your document has a life story. Version control is like a time machine, it can take you back to the moment your document was born, or any other point in time when you or a collaborator saved that document. **You don’t save copies of your document, you just save the the life story of the document, or its timeline.**



What's on the timeline? For our purposes, **let's call any saved change a commit.** The life story of your document is a timeline of commits.



When we share and work on projects with collaborators, managing the changes, or commits that multiple collaborators working in different places at different times make to a single set of documents becomes very, very important.


Remove persistence commits
Fixes for the widget documentation
Update widget instructions and screenshot

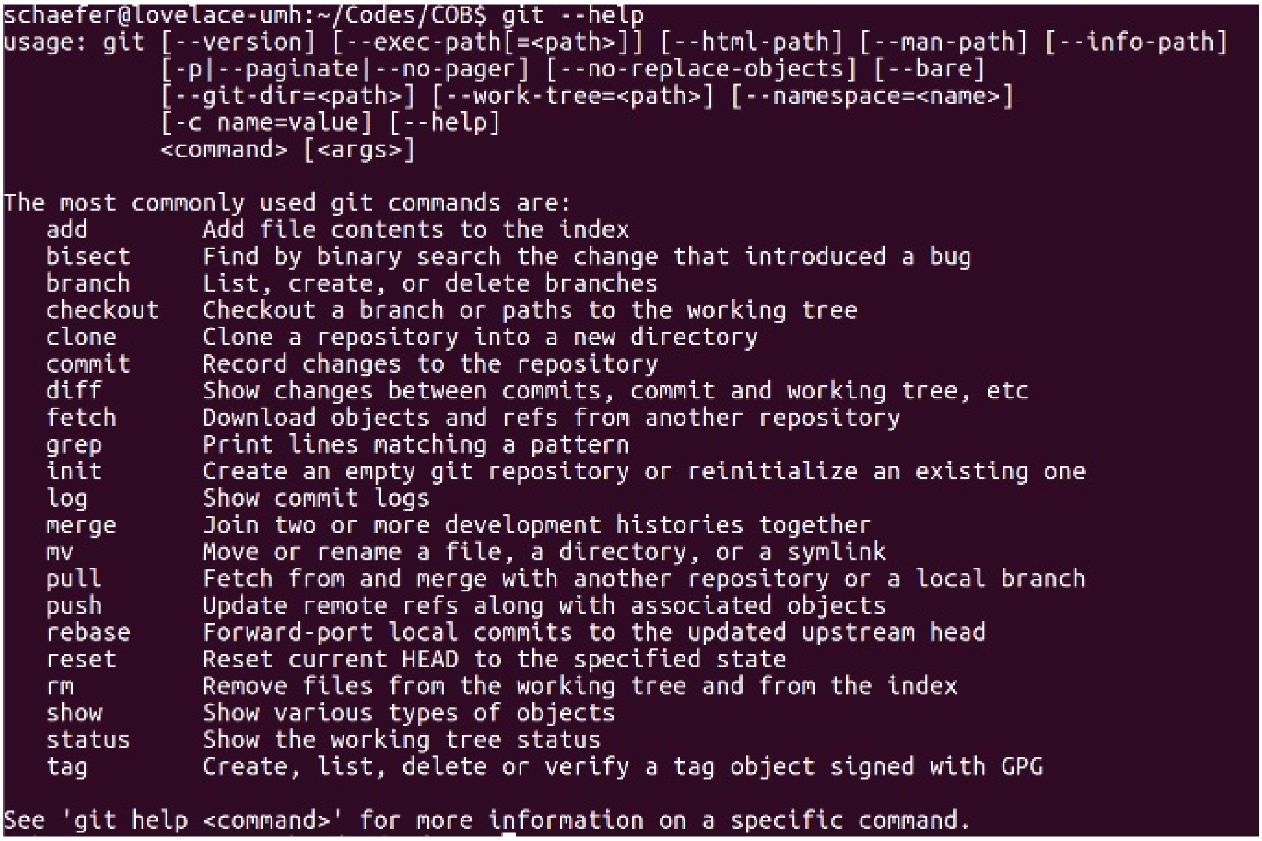
And when we’re working with multiple collaborators, everybody needs to know and understand what commits are being incorporated into the repository and why, so good communication becomes very very important. The great news is that there’s a piece of version control software to help us both manage and communicate with our collaborators about commits to our project-- that software is called Git, and it’s the basis for GitHub, the platform your study group repository and website will live.

3.2 Getting to Know GitHub

There's lots to know about GitHub-- it has terrific project management features, a social platform, and communication tools that are useful for any project where a group of people are working together on the same set of documents.

Because GitHub is online, it’s designed to share your work, and allows other people to “fork” your project-- meaning they can create an independent copy of your work to test, modify, remix and reuse it. That’s exactly what you’ll do to make a website for your group-- you’ll fork the Study Group Repo and to make a copy in your own space on GitHub (detailed instructions are below, in the Assignment section).

Before we go any further, we need to talk for a moment about Git. As mentioned above, Git is the command line software that powers GitHub and actually handles the version control work. When you use GitHub, Git is working behind the scenes. You can also use Git without ever using GitHub, and many software engineers and research coders do. This is what Git looks like to a user:

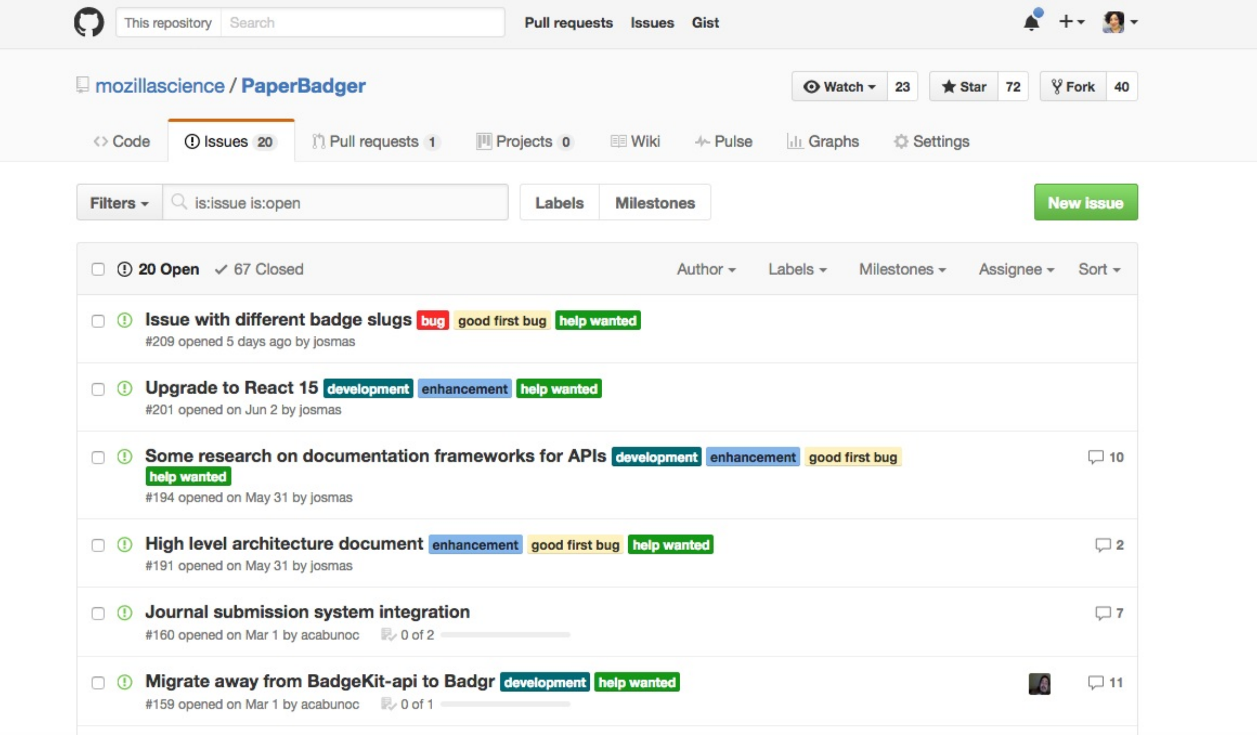


```
schaefer@lovelace-umh:~/Codes/COB$ git --help
usage: git [--version] [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [-c name=value] [--help]
           <command> [<args>]

The most commonly used git commands are:
   add        Add file contents to the index
   bisect     Find by binary search the change that introduced a bug
   branch     List, create, or delete branches
   checkout   Checkout a branch or paths to the working tree
   clone      Clone a repository into a new directory
   commit     Record changes to the repository
   diff       Show changes between commits, commit and working tree, etc
   fetch      Download objects and refs from another repository
   grep       Print lines matching a pattern
   init       Create an empty git repository or reinitialize an existing one
   log        Show commit logs
   merge      Join two or more development histories together
   mv         Move or rename a file, a directory, or a symlink
   pull       Fetch from and merge with another repository or a local branch
   push       Update remote refs along with associated objects
   rebase     Forward-port local commits to the updated upstream head
   reset      Reset current HEAD to the specified state
   rm         Remove files from the working tree and from the index
   show       Show various types of objects
   status     Show the working tree status
   tag        Create, list, delete or verify a tag object signed with GPG

See 'git help <command>' for more information on a specific command.
```

But, for nearly all your initial collaboration needs-- and to set up your study group website-- you don’t won’t use the command line and Git directly. You’ll use the GitHub web interface (with Git running in the background). That interface looks a bit friendlier, like this:



□ Issue with different badge slugs bug good first bug help wanted
#209 opened 5 days ago by josmas

□ Upgrade to React 15 development enhancement help wanted
#201 opened on Jun 2 by josmas

□ Some research on documentation frameworks for APIs development enhancement good first bug help wanted 10
#194 opened on May 31 by josmas

□ High level architecture document enhancement good first bug help wanted 2
#191 opened on May 31 by josmas

□ Journal submission system integration 7
#160 opened on Mar 1 by acabunoc 0 of 2

□ Migrate away from BadgeKit-api to Badgr development help wanted 11
#159 opened on Mar 1 by acabunoc 0 of 1

Here’s a quick tour of the GitHub interface by Abby Cabunoc Mayes, Mozilla’s Developer Engagement Manager and open source/open research advocate.

Now that you’ve got a sense of what GitHub can do, and where things are in the interface, let’s get your Study Group Repository and website up and running!

3.3 Assignment: Get your Study Group Website and Repo Up and Running!

Everything you need to set up your own Mozilla Study Group website for organizing events is right here - follow the following steps and you'll be up and running soon.

1. Make yourself an account on GitHub. The free one is just fine.
2. Fork this repository. Up in the top right corner of this page, there's a button that says 'Fork'; press it! This makes your very own copy of all this stuff in your space on GitHub; when the copy is done, GitHub will automatically take you there.
3. Turn on the Issue Tracker. An issue tracker is a message board GitHub sets up for every repository; you'll use it to post information about your event and talk to your participants. To turn it on:
   - click on 'Settings' in the right sidebar of your repo;
   - click the checkbox beside 'Issues'
   - head back to your repo by clicking on the big studyGroup at the top of the page.
4. Edit your README file. Remember the text about your group that you wrote back in Section 1? In your directory of files, click on README.md and edit the generic text to include all the specific info about your group and what it's all about. If you'd like to do a bit of fancy formatting (headings, bold, italics, etc) you can use the built-in [GitHub Markdown toolbar] (https://help.github.com/articles/about-writing-and-formatting-on-github/).
5. Edit the _config.yml file in your new repository:
   - click on _config.yml;
   - click on the little pencil near the top right;
   - follow the instructions in the file on how to edit it;
   - when you're done, press the green 'Commit Changes' button at the bottom of the page.
6. That's it, you're done! You can see your new website at https://yourUserName.github.io/studyGroup/, where yourUserName is the user name you signed up for GitHub with. If this is your first time making a webpage on GitHub, it might take 30 minutes for things to percolate through GitHub's computers - don't worry, it's all good! Check back later and your website should be up and running.

If you have any trouble with the above steps, let us know on the Gitter chat mozillascience/studyGroup or in an issue, and we'll help you out!

3.5 Assignment: Add an Event!

When you're ready to list a new event for your Study Group, follow these steps, or watch this video where we walk you through event listing.

1. Make a new Issue to describe your event.
2. Click on 'Issues' over on the right sidebar of your repo,
3. Click the green 'New Issue' button near the top right
4. You'll then see a form where you can give your event a title and a description - fill these out with all the relevant information:
   - Where will your event be? Include a link to a map.
   - When will it be? Date and time.
   - What people should do to prepare beforehand (install any dependencies, set something up, etc)
5. Go to the _posts directory. It'll be at https://github.com/yourUserName/studyGroup/tree/gh-pages/_posts - or you can click on _posts in your repo.
6. Make a new file by clicking on the + sign beside _posts/ Name it like the following: YYYY-MM-DD-word.markdown, where YYYY-MM-DD is the date of your event, and word is anything you want.

   title: Study Group Meetup
   text: a one sentence description of your event
   location: Hacky Hour Stadium
   link: https://github.com/yourUserName/studyGroup/issues/1234
   date: 2016-01-04
   startTime: '15:00'
   endTime: '16:00'

7. Change all the fields to describe your event; make sure the link is the address of the issue you created When you're done, click 'Commit Changes' at the bottom.

That's it! Your event is now listed on your webpage, and there's a discussion thread where people can ask questions and discuss the details. Events will be automatically removed from the schedule on the webpage when they're more than a week in the past - but the issue you created will always be there as a record of what you've done.

Event Listing Gotchas! Here are a few things to look out for when listing an event:

- Did you remember to include the --- above and below? The website builder needs those.
- Can't find the issue tracker? Remember to turn it on under the 'Settings' menu on the right.
- The seven fields need to be on exactly one line each; some text editors will insert line breaks into lines that are too long; remove these if so.

3.5 Assignment: Feature Your Community in the 'Who We Are' Section

Your website includes a gallery of participants in your Study Group; adding people here is a great way to show off your community and highlight your new friends and colleagues. To add someone to the list:

1. Edit the _data/members.yml file by adding the following section for them:
   - name: their human name
   - affiliation: school, lab, department, business....
   - github: their GitHub handle
   - interests (list one to three different interests)
2. Commit the file and your community members should appear in the gallery!

3.6 Optional Assignment: Set up a Google Calendar for your Study Group

If you'd like to offer your community a calendar of events they can import into their own calendars, try using a Google Calendar. To set up, make a new google account, and update the variables in _config.yml under the heading 'Setup Google Calendar'.

You can add events to your calendar by hand, but if you'd like to manage it automatically, there's a script to do so in scripts/updateCalendar.py; instructions for use are at the top of that file.

Open Science/Open Research 101

Open science is a different approach to the practice of scientific research, aimed at making research more understandable, useful, and impactful. When you’re doing open science you:

- **Clearly document all research methods, processes, observations, code, and data; make these freely, publicly available for reuse and redistribution**
- **Collaborate with researchers from within your discipline and across disciplines, often via the world wide web**
- **Make research communications widely available and accessible to the public, encouraging public engagement and participation in research.**

These open practices are not just specific to science-- in fact, they are applicable to any kind of research!

4.1 Why Open Science and Open Research?

Whether you're studying the human genome, black holes, deep-sea ecology, or the cultural implications of climate change, research is the practice and process of learning and creating knowledge. Researchers always build on (or transform) our existing understanding of the world. **When a researcher shares an insight or discovery, makes her data available on the web, or makes the details of a new experimental technique or tool public so others can use and reuse it, she empowers both fellow researchers and citizens, furthering our collective knowledge...** knowledge that can be used to solve problems, save lives, and inspire and amaze us all. The more research data, knowledge, methods, tools and skills made widely and openly available to all, the better.

While the idea of “open” (drawing inspiration from the open-source movement in software engineering) has taken root in the sciences recently, it is relevant and useful to any kind of research: economics, psychology, sociology, the humanities-- you name it.

Making your research open will likely require some shifts and changes in how you’re currently doing research, but you don’t have to rush in and do it all at once! The following sections provide a bit more information about some key concepts in open science, and links and resources for you to explore and learn more, and decide which of these open practices make sense for you and your work.

As you read through this module, you might encounter some unfamiliar terms. This handy Open Research Glossary https://docs.google.com/document/d/1uXZzyXPHNcjCPiR5qkzEuB5u2PUIYQzq0mrG9BtD-Qo/edit#heading=h.tsneh02k2pc8 should help with definitions!

4.2 Open Publication and Open Access

Currently, publication in a peer-reviewed journal is the primary measure of any researcher’s work and career. This system has been around for centuries, and is intended to rigorously vet research and provide a platform for sharing discoveries-- a very important aim! But the intense pressure to “publish or perish,” the revenue-focused business model of academic publishers, and the rapid decline of print publishing has resulted in a system that’s flawed and problematic, and in many ways out-of-date.

The **Open Access movement, which asserts that all research outputs be freely available online with no restrictions, and available for reuse with minimal or no restrictions,** has emerged in response to the current system. The following video, from PHD comics, provides an introduction to Open Access and critique of the traditional publishing system.

The Open Publication movement seeks to answer the following questions: How can the current system be transformed so it’s more effective and open? What are some new strategies and platforms to share and find research findings, for maximum accessibility, reuse and impact? What can emerging researchers do to make their publications as freely available as possible to other reserachers, and the public?

4.3 Reproducibility

You’re preparing your research for publication and the temptation may be to focus on the results and discussion sections of your paper-- after all, that’s what will make the biggest splash! **But consider how to use publication to make your work reproducible, so that other researchers successfully recreate your results using your data, code and methods.** (Reproducing the results of a study is a bit different than replicating a study, where another researcher uses your methods and your code but collects or generates a new data set. Both replication and reproduction are things another researcher may try to verify the results of a published study. For more on the reproducibility versus replicability, see “A Statistical Definition for Reproducibility and Replicability,” by Patil et al.)

By making your work reproducible, you:

- **Increase the usefulness of your research by enabling others to easily build on your results, and re-use your research materials**
- **Ensure validity and trust in your results, and help to support the validity of future studies that are based on your work**
- **Increase accuracy, trust, and confidence in your field broadly.**

Publishing studies that can be reproduced or replicated may seem like a no-brainer. But it’s not an inevitable outcome of every publication. In 2012, cancer researchers Begley and Ellis published a comment in the journal Nature, called “Drug development: Raise standards for preclinical cancer research.” The article describes a crisis in the quality of scientific literature in cancer research. Working over a period of 10 years, Begley and his team at Amgen labs attempted to replicate the results of 53 known “landmark” studies in the field, but were only able to confirm results in 6 of those studies (11%).

Some of these non-replicable studies had resulted in hundreds of secondary publications, building on unconfirmed results and likely leading to the development and eventually the testing of ineffective drugs in cancer patients. Certainly, drug development is a complex problem, with models and technologies that are challenging to work with. **But the intense pressure to publish early and often can result in the submission of studies without the level of documentation that allows for either reproduction or replication of results, and doesn’t tell the full story of the research.** A glance at the website Retraction Watch, a project of the Center for Scientific Integrity, shows that the problem of publishing unverifiable results isn’t confined to oncology research. For one perspective on how this plays out in different fields, see Roger Peng’s blog post A Simple Explanation for the Replication Crisis in Science.

In their comment, Begley and Ellis call for more rigorous documentation practices, such as the inclusion of all experimental methods and data from all trials of a given drug in a published paper about that drug-- not just the few trials that succeeded. **A truly reproducible study should contain a complete narrative of the research, and include well-documented methods, code, and data.**

There are a number of tools and practices that can help you tell a coherent research story, without gaps or fuzzy areas. See biostatistician Karl Broman’s terrific tutorial, “Initial Steps Toward Reproducible Research” for more on how you might get started. Another great resource is anthropologist Ben Marwick’s presentation “Reproducible Research: A View from the Social Sciences.” As mentioned in the introduction to this section, **you don’t have to adopt every best practice in reproducibility at once! Find the ones that seem most promising for your work, and give them a try.**

4.4 Code Review

As research becomes more and more powered by data and focused on data analysis, the ability to write and skillfully use code (whether in R, python, or your language of choice) to process that data is key. Documenting and vetting the code you use is a critical part of ensuring your research is reproducible by others. As genomics researcher Titus Brown says:

“...as soon as "theory" touches "real data" there is a gulf of unknown size between the theory and the data. Code is what bridges that gap, and specifies how edge cases, weird features of the data, and unknown unknowns are handled or ignored.”

Because the outcome of your analysis is dependent on the design and performance of your code, it’s critical that your code is validated. A code review is when you allow someone who has programming expertise to read over and comment on the code you’ve written. By getting another pair of eyes on your code, you can:

- **Catch any errors or bugs, verify assumptions and logic, and get feedback to improve your documentation**
- **Ensure that your code can be understood by others**
- **(Eventually) minimize the amount of time you coding by participating in the collaborative development and sharing of valid code with other researchers.**

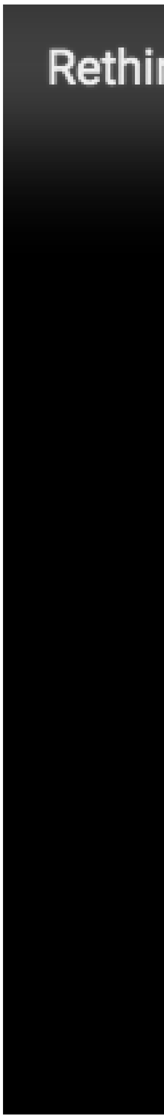
The friendly, peer to peer learning environment of a Mozilla Study Group is the perfect setting for informal code review.

4.5 Open Data and Data Sharing

In the section above, we mentioned that sharing is a key aspect of reproducible research, and helps to ensure the validity and trustworthiness of results. The notion of data sharing and “open data” are central to open research. The Open Knowledge Foundation, an organization dedicated to bringing “openness” to the mainstream, defines the following key factors that make data “open”:

- **Access & availability** - data is available to all in a convenient and modifiable form
- **Re-use & redistribution** - terms of use allow for reusing, remixing and redistributing the data
- **Universal participation** - there are no restrictions on who may do any of the above with the data

In a nutshell, open data is data that is made freely and easily available to anyone to use, reuse and distribute. But why should you take your carefully collected, hard-earned data, and set it free on the internet, for strangers to reuse, remix, and redistribute? There are so many reasons. This TED Talk from Data Librarian Kristin Briney covers just a few of them:



In addition, many research funders require grantees to share their data! So it’s not only great open research practice, it’s also the law. Opening your data requires some careful planning, great documentation, and a good repository (or online storage site where users can freely access your data). The process can be daunting. Luckily, we’ve created a series of primers to help you understand more about why open data is so great, and to help you easily and successfully share your data. Mozilla Science Lab also offers materials and guidance for those interested in leading hands-on, in-person workshops on data sharing and data management.

4.6 Collaboration in Open Research & Open Source

The idea of “open research” is inspired in part by the open source movement in software engineering. Open source software is often developed collaboratively by engineers and designers who are inspired and motivated by the project’s vision to contribute time and expertise. **Collaboration increases the skills and resources devoted to achieving a common goal, and diversifies the perspectives and problem-solving capabilities of the group working toward that goal.** Collaboration in the open source world has elements in common with formal collaboration that’s common in academic research. In both models, collaborators must effectively:

- **Define a common goal (this may take some negotiation)**
- **Create good working relationships among collaborators**
- **Share resources (expertise, equipment, materials, data, code, work hours)**
- **Communicate regularly about process and progress**
- **Share decisionmaking about the project’s direction**
- **Give clear, fair credit to all contributions to the project.**

In the open source world, barriers to collaboration can be very low. Here are some of the things that open source projects do to make sustained collaboration easier:

- **Post source code and clear documentation in an open repository**
- **Choose version control software (like GitHub) that enables logging and documenting every contribution, so it’s clear who did what**
- **Use an online platform (like GitHub) for collaboration, so anyone with a web browser and some skills can communicate with project leaders and contribute**
- **Leverage the project management, communication, and discussion features of online tools (again, like GitHub!) to streamline work**
- **Build mentorship into the project plan, so new contributors can make connections in the field, sharpen their skills, and “level up”-- these are great motivators for participation!**

With encouragement, on open source projects we’ve seen groups of contributors coalesce into sizable communities that advocate for the project and persevere through difficulties.

While research collaborations have their own constraints and particular requirements, many collaboration tools and practices from open source can useful in open research. Collaboration is not without its challenges and pitfalls: lots of researchers struggle with questions of authorship, who will take the role of lead or champion, and which technologies to use for collaboration and communication. **Mozilla Science Lab offers support, resources and learning materials, such as our Working Open Workshops, and the Open Leadership Training Series, to teach best practices for working open and help foster effective collaboration.**

In your Mozilla Study Group you can boost collaboration skills by sharing expertise, reviewing each other’s code, working in GitHub to facilitate discussion and organize group meetings, events, and content. Groups may also take on a project together, like creating a specialized code library, a series of lessons on a certain tool or topic, or even working to publish a paper together.

In 2015, Mozilla fellow Dr. Christie Bahlai developed a course called Reproducible Quantitative Methods that emphasizes collaboration skills and data re-use.

Here's more from Christie about that course.

A comprehensive instructor guide is available for the course, and free to be remixed and reused. We encourage Study Groups to use these materials in whatever way works best for their group!

4.7 Public Communication of Research

You cannot truly practice openness in research without considering the public communication of your work (sometimes also called “public engagement” or “outreach”). Public engagement is not just about communicating the conclusions of your research with interested audiences. It also involves engaging with members of wider society regarding the social, ethical and philosophical aspects of your research – sometimes even before you have begun your project! But why spend time discussing your research with those outside your field, people who aren’t even researchers? **Public communication of your research makes it more impactful and useful to others. And it also fulfils the implicit obligation we have as researchers to share the knowledge we acquire on behalf of humanity at large.**

The motivations for public communication of research and the methods of doing so have evolved over the years. While older models featured one-way communication – from scientists to the public, in the form of lectures and popular articles – the recent model of “public engagement” recommends a two-way approach. Leaving aside the ideological, political and sociological reasons for participating in public communication of research, let’s look at how public communication can benefit your research. Public communication can:

- **Make your research more responsive and actionable** Research is a complex process-- especially in fields closely related to human lives, like medicine or climate science-- and researchers may not fully understand all the social and environmental factors that may affect their work. Discussions with patient groups or local disaster-response teams, for example, can help researchers confirm assumptions or discover issues that might need further investigation. Also, not everyone follows research news, even in fields that may have a significant impact on their lives. By reaching out and engaging the public in conversations related to your work, you can help bring new results to those audiences who may be directly affected by them.
- **Help measure the impact of your research** Researchers are now acknowledging that the traditional metrics of measuring impact of research — citations — are not particularly meaningful in the modern age, where content may be consumed over the web, discussed via social media and written about in blogs. Alternative metrics are now being given their due. By being part of the conversation using the tools the web has made available, you can increase the impact of your research and also

measure the audiences it reaches.

- **Ensure your research has sustained support and funding** This is a slightly more self-centred reason, but your brightest ideas may not see the light of day without the funding to back your work. In a democratic society, an engaged public plays a crucial role in determining funding policies for research. Engagement can help you reach out to interested groups and cultivate new connections. Furthering public interest in research benefits not just science as a whole, but also your own endeavours.

You can lend your voice to public communication of research in a variety of ways. You might organize public visits to your research facility or laboratory, or hold public demonstrations of research technologies and methods. Consider speaking at a local school about research practice in general, to inspire future researchers. Or you could work with teachers on developing school curricula. You can reach remote audiences on the web, by blogging about your research and its ups and downs, by discussing your subject on social media sites like Twitter and Facebook, or even hosting your own YouTube channel.

Citizen-science initiatives such as Zooniverse invite active partiicpation and bring research into people’s homes — find out how you can crowd-source your latest data collection or analysis. And you might engage with local politicians, members of industry, artists, and museums to discuss possible collaborations on public events, exhibitions, etc. You will likely find the possibilities very rewarding!

4.8 Challenges to Open (& Responses!)

Research culture and practice is changing, as technologies, institutions, and funder requirements change. We think open research practice is the best approach for doing better, more useful, more powerful research. But like any change, it’s controversial.

Many senior researchers came up in and succeeded the old system, and are skeptical of anything new. Many publishers rely on a business model that is predicated on closed, paywalled content. And the fact that many researchers feel they must “publish or perish” can discourage collaboration and the open sharing of useful data and methods.

Have you heard some other challenges? Got your own great arguments for open research practice? Tell us about them here.

Finally, as noted at the start of this section, it may not be possible or be practical for you to run out and make all your research open and reproducible immediately. Start at a pace that makes sense for you: think about your own particular research situation and your institution, find an open practice or two that makes sense for your work, try them out and see how things go. Christie’s Bahlai’s blog post on “Baby Steps for the Open Curious” discusses how you can get started.

What’s the Mozilla Science Lab? And who's Mozilla?

Meet the Mozilla Science Lab

The Study Group Orientation materials were developed by the Mozilla Science Lab (MSL), with lots of help from members of the MSL community. The Mozilla Science Lab’s mission is to help scientists and researchers (anyone from students to established researchers to citizen scientists) to work openly and do better research, more research, and make that research more useful by sharing it widely.

Although our name suggests otherwise, MSL is not a brick and mortar laboratory, nor do MSL staff regularly do scientific research. We do work in an iterative, experimental way, building on learning and constantly evaluating our outcomes and fine-tuning our programs. **Our community includes scientists, researchers, designers, developers, and librarians, all working to making research open and accessible.**

MSL seeks to promote open research through:

- **Fellowships**. We offer a 10-month, funded fellowship for researchers who want to influence the future of open science and data sharing within their communities. Fellows receive training around open source, data sharing, open science policy, and licensing. Fellows work on their own open projects and create code, learning resources, and trainings to support open practice at their institutions and in their local communities.
- **Mentorship**. In our mentorship programs, participants connect with experienced mentors to gain open leadership skills and advance their research projects
- **Project-based learning**. In addition to our Study Group program, we produce learning materials and workshops on Open Data and open practice designed specifically for researchers and project leads; our online Projects directory is a curated collection of open source, web-based tools to help you do better science.

In addition to the Study Group calls, MSL also has regular Community calls open to all--drop in to meet fellow community members and learn about exciting topics in the world of open research. Explore the MSL website to learn more about our offerings.

Oh by the way, we’re from Mozilla!

The Mozilla Science Lab is a program of the non-profit Mozilla Foundation. You may recognize the name Mozilla-- we’re the group of people, organized as a non-profit, public charity-- best known for making the Firefox web browser. Mozilla’s Firefox is an open source project, which means the source code is freely, publicly available and anyone who would like to can help develop the project. Firefox was created and is sustained by the Mozilla staff and a collaborative community of volunteers, called “contributors.” Currently, 1/5th of the web's traffic comes through the Firefox browser, giving some of the big corporate browsers a bit of competition! **Many open practices-- such as the use of the web as a collaboration tool and a space to share code, resources, and data-- were created, tested, and refined by software developers working on open-source projects like Firefox.**

The open source movement showed that scrappy, self-organized groups of passionate volunteers could collaborate to develop viable alternatives to proprietary software, and even compete against huge corporations! The principles of participation, self-organization, flexibility, and open sharing of resources are at the heart of Mozilla’s Study Group Program, and the open source ethos drive’s Mozilla’s commitment to open science and its Science Lab program.

In addition to developing Firefox and supporting the Science Lab, Mozilla fights for what we call the “open web.” We’re doing this by championing:

- **Inclusion**: an accessible, free Internet for all
- **Privacy**: user control of their own data on the web
- **Web literacy**: the ability for everyone to meaningfully
    - Read: navigate the web, and find, evaluate, and synthesize web content
    - Write: code, design, compose, revise and remix content for the web
    - Participate: connect with online communities, share, contribute, and work collaboratively on the web
- **Decentralization**: internet technologies and platforms based on open standards, allowing for the seamless flow and transfer of information and content
- **Open Innovation**: the web as a space for new ideas and creativity, where all users can create, innovate, and compete online without asking permission.

To drive this work, Mozilla coordinates and staffs programs to work directly with interested, motivated community members on Learning, Advocacy, Science, the emerging Internet of Things, and Women in Web Literacy. We think of this constellation of programs and the amazing community leaders (like you) who work with us as the Mozilla Leadership Network. Members of the network are leading projects as diverse as improving civic engagement, creating better educational experiences, building more human technologies, and making scientific research more collaborative and efficient (through data sharing, open projects, and Study Groups, just like yours!).

To see what’s exciting and new in Mozilla’s network, check out Network Pulse, a space for updates and information on network projects and activities. As a Study Group lead, you’re a member of our Network, and encouraged to take advantage of resources, trainings, and make connections with others in the Network.

Growing Your Group

It’s OK to start small-- in the beginning a Study Group can be just few people working together. You don’t need dozens of members to have your first meeting. On the flip side, some Study Groups start of big, with lots of buzz and excitement, but attendees lose interest after a few weeks.

Your Study Group will stabilize and grow-- it may take some time. Here are some tips on outreach for new and growing Groups.

- **Start with people you know.** Invite people from your lab or your classes. A face to face conversation is the best way to connect and get someone excited about the Group.
- **Send an email blast.** Send out an announcement to your lab email list, departmental lists, and those of related departments. If there’s a campus-wide graduate student association, that’s a great place to advertise, too.
- **Communicate clearly.** This means making sure all event listing and emails include time, place (with link to a map) and a short description of what the event is about.
- **Send updates and reminders.** Part of your role as Study Group lead is to be a cheerleader for the group, keep people excited and engaged, and most importantly let them know that meetings are happening!
- **Make friends with your librarians and department administrators.** These people can refer others to the group, may help you find meeting space and A/V equipment, and can assist if you deicde to hold a larger event or afternoon workshop.
- **Contact like-minded groups.** Your institution may have a research computing department or group with common interests. Reach out to someone from this group to see if they might be interesting in attending or leading sessions in your Study Group.
- **Get the word out to newly arriving colleagues.** Ask your department administrator if your group can be mentioned in welcome packages for new students or researchers.
- **Advertise around campus.** Distribute flyers and put up posters announcing the group.
- **Look outside your field of study to find potential members.** Researchers from all fields (life and physical sciences, humanities, social sciences) work with data and encounter similar challenges. A diverse group will make for lively, surprising

discussions.

- **Plan a Fun Event!** Throw a Demo Party or Hacky Hour-- these low pressure, social events are great ways to introduce potential members to the Group.
- **Persist! Keep reaching out and encouraging people to come!** Set a goal to personally invite a new person each time, and encourage other members to bring friends or labmates.

Communication Channels for Study Groups

You can direct newcomers and regular members to your awesome Study Group website for basic info and a calendar of events, but you'll want to actively reach out to members and potential members on a regular basis. You have many possible tools and channels for reaching your members: emails and email lists, social media platforms like Twitter or Facebook, your GitHub repo… and, of course face to face conversations! Use the tools that work best for your community of members. **You might want to ask your members what communication channels they prefer, and use those favored by most people in your group.**

If lots of your members are active on Twitter, setting up an account for your Study Group and Tweeting out event reminders or requests for session ideas, or just Study-Group relevant news (like articles or blog posts about coding or open science) might be the best way to reach members and create buzz. If more of your members are active on Facebook, start a page for your Group there. Email can be a really effective way to reach your members, if they check it regularly and their inboxes aren't already to crowded. If email works for your Group, maintain a mailing list and send out announcements. A personal email is always a great way to encourage a member who's missed a session or two. **Personal messages and conversations also work really well when you're requesting help running an event or finding a session facilitator.**

Nobody will come to your events if you don't let people know about them! Follow these steps to effectively communicate about each individual event (this list is GitHub-focused, but if your members tend to gravitate towards an other mode of communication, post or send reminders where your members will find them!)

- As early as possible and at least one week in advance, list your event on your website (detailed instructions are in Section 3 of this training).
- Focus on the issue tracker. There can be a lot of links and information for an event - location, lesson notes, examples to download - stay organized by putting all the relevant information in an issue for that event, and direct all attention to this issue in subsequent communications.
- At the start of the week, send a reminder email to all the mailing lists you have access to, pointing out the events happening that week (remember to include links

to the issues describing the events, as well as a link to your Code of Conduct, and tell people to bring their friends!).

- The morning of the event, comment on the issue describing the event - even a simple 'Looking forward to this today!' is enough, since that will send a notification to everyone watching your repo.
- Consider webcasting your demos, so people can participate remotely. The easiest way to go about this is using Google Hangouts on Air, which directly streams your webcast to YouTube. *Don't forget to ask people to watch your GitHub repo! That way, they'll get updates about everything your Study Group is doing.

Scheduling, Planning, and Rhythm

It feel be daunting to plan for an entire semester of Study Group sessions. Remember, you don’t have to map out the content for an entire semester of sessions and before you begin-- you can plan a few sessions, see what works, what the needs and skills are in the Group. Here are a few tips for planning:

- **Survey your Group.** Find out what people most want and need to learn, and tailor your sessions and content to meet those needs.
- **Be consistent.** Regular meetings will ensure that your Group has a chance to stabilize and grow. Depending on the availability of your members, weekly or every other week meetings may work best.
- **Be realistic.** Give yourself plenty of time in sessions and over the semester to tackle topics, troubleshoot, and work through examples. It’s easy to overestimate what you can cover.
- **Start at the right level.** Take on activities that are targeted for your group-- don’t dive into advanced material if most of your members are at a beginner or intermediate level.
- **Use existing resources.** Explore the Study Group lessons repo, and ask other Leads what's worked best for them.
- **Listen to feedback.** Be sure to ask group members, or send out a quick survey to find out what people want to learn. Find out how people feel about sessions and formats you've already tried. Be flexible and responsive to requests.
- **Expect ebbs and flows.** Your Group will probably follow the rhythm of your institution’s schedule-- during busy times, you might not get many attendees.
- **Mix it up!** If you can, bring in guest speakers or teachers to get the members working on a new topic.
- **End the period (semester, term, or quarter) with a party.** Thank members for participating and ask them what they'd like to do next term; this party is a hacky hour with lots of advertising push, and a celebratory end of semester/term/quarter feel.

Using Personas to Grow Your Group

A “persona” is a tool commonly used in the design world, to help create products and experiences that work for real world users (aka “user-centered design”). You’d use a persona when you want to make your group experience better for members, attract an audience you're not yet bringing in, or to try to understand why people may be drifting away from your group.

To recap, the persona is a description of a user. Personas should:

- describe an imaginary user
- be based on observations or understandings of actual potential or current users
- be detailed enough so it feels real to the designer, so they can imagine the persona’s needs and responses to a product (or in this case, process of contributing)

Sample Persona

Here is a sample persona created by MSL staff to think about users of the Study Group Program.

Rashid is a PhD student in astronomy at a university in Southern England. He’s outgoing and a snappy dresser, favoring skinny jeans and colorful cardigans. He lives in on-campus housing and after a long day at the lab he usually rushes home to see his wife and infant son. Rashid took an intro Java programming course long ago, as an undergrad, but his research now demands Python skills. Because of the competitive nature of his lab, he’s reluctant to ask colleagues for help. He follows Mozilla Science Labs on Twitter, has some exposure to and interest to Open Science, but is hesitant to share his data for fear of being “scooped” on an important discovery.

About Pathways

Once you have created a persona, youcan imagine how they might interact with our project-- the pathway they to get involved and to sustain involvement. This process of engagement most likely has a few phases.

- **Discovery** - How they first hear about the group.
- **First Contact** - How they first engage with the group, their initial interaction.
- **Participation** - How they first participate or contribute.
- **Sustained Participation** - How their contribution or involvement can continue.
- **Networked Participation** - How they may network within the community.
- **Leadership** - How they may take on some additional responsibility in the group, or begin to lead.

If you are working with a good persona, you can clearly see a progression of steps. Here’s an example (using Rashid).

- Discovery - Early in the semester, Rashid sees poster advertising study group around campus.
- First Contact - He attends the third meeting of the Group, and is encouraged to return personally in a follow up email from the Study Group Lead.
- Participation -He asks and answers questions during the help session.
- Sustained Participation - He attends several co-working sessions throughout the semester.
- Networked Participation - He invites some of his colleagues from his lab to a session.
- Leadership - He agrees to present an intro session on Java, and creates a learning resource to contribute to the group's repo.

When you think through a pathway, you should begin to realize what needs to be in place to move your persona forward. **You’ll begin to see potential pitfalls for your persona, in terms of skills, time, and motivation.** In Rashid’s story, some of the challenges might be:

- a lack of awareness about the group,
- some reluctance to continue attending, not feeling quite at home since he's joined mid-way through
- a lack of knowledge of GitHub
- family time and scheduling conflicts
- a lack of awareness on the part of the Study Group lead of Rashid’s skills and what he can contribute

Once you have a sense of this story, you can begin to list solutions to those challenges:

- Publicize group meetings via posters around campus as well as on twitter and via email blasts.

- Collect emails of new group attendees for follow up messages-- personal messages may work best for new attendees who are unsure
- Offer an online intro to GitHub for those who join mid-semester and missed the first sessions, and be sure to explain Group communications systems to newcomers
- Schedule Group meetings for daytimes and early evenings to avoid conflicts with family schedules
- Talk to newcomers to discover what skills they have to offer the group

You’ll likely need more than one persona for your project, as different groups of users will have different needs and motivations.

Assignment: Create a Persona & Pathway

1. **Brainstorm.** Read through the two following questions and come up with answers. You can do this individually or as a group. If working in a group, brainstorm individually and share afterwards. Sticky notes are great for this task, for recording ideas during the brainstorm, and sharing and organizing the group's ideas. Spend about 3 minutes on each question.
    - Who is the person you'd like to attract to your Study Group? Think of skills and attributes-- but, since this is an imaginary person, also give them identifying details, a brief life story, etc.
    - What are that person's motivations and needs? Think of what might draw them to your Group, what value they would gain from it, how it fits into their long term goals.
2. **Create a short written description of your persona.** Spend about 4 minutes on this task. If you like, draw a picture of this person!
3. **Plan a Pathway.** Using the structure above (Discovery → Leadership), describe a pathway for your persona. What are the steps to move through the group? What could be stumbling blocks for user?
4. **List your Solutions.** For each potential stumbling block or barrier your user might encounter, list a solution that you'll work into your design of your group or project.

Whew! You did it!

Thank you for completeing the Study Group Lead Orientation, and congratulations!

We've got lots planned for the Study Group program-- more resources are coming your way, and we hope soon to begin to organize regional events, workshops, and meet-ups where different groups can get together. Please come to the Study Group calls to learn more. And please let us know how you're doing... say hello and give us an update about your group (successes and struggles both welcome) on the mozillascience/studyGroup Gitter chat.

Feel free to add an issue to this repo if you have comments, ideas, or feedback on this training series, and please jump in and help out with some of our issues-- we're always looking for help and contributions to improve these materials.