

Recurse Center

# User's Manual

## TABLE OF CONTENTS

1. **Welcome to an unusual experiment**
2. **The environment**
  - a. **The self-directives**
  - b. **Pair programming**
  - c. **The social rules**
3. **Logistics**
  - a. **The time commitment**
  - b. **Batch length**
4. **Advice**
  - a. **Before RC**
  - b. **During your batch**
  - c. **The space**
  - d. **Resources and tooling**
  - e. **Challenges at RC**
  - f. **Never Graduate**
5. **Jobs**
6. **Contact information & employee hours**

This User's Manual was written for admitted Recusers to help them navigate and understand the Recurse Center. We've decided to publish it in the hope that it will give the outside world a clearer picture of what the Recurse Center is like. If you're considering applying to RC, we recommend you check out this manual to get a sense of what to expect.

---

## *Welcome to an unusual experiment*

**The Recurse Center is unlike the rest of the world.** This guide is designed to help you get settled in and get the most out of your time here, and is written for people who have been admitted to RC or are attending a batch.

One of the things that makes the Recurse Center different is that it's largely *self-directed*. This means that while you're here, you won't have someone telling you what to do or how to approach your time. We don't have grades, exams, curricula, or even classes, because we believe that people learn best when they have the freedom to explore what most interests them.

This doesn't mean the Recurse Center is a vacuum, just that the structure we provide is optional, not mandatory. We've been running the Recurse Center since the summer of 2011, and we've learned a good amount about what strategies have and have not worked for different types of people in the past.

The structure we *do* provide is shared space and time: you'll be doing a batch alongside other like-minded programmers from a tremendously diverse set of backgrounds, and in joining RC you'll be joining a community of over 2,700 people who have done a retreat. We only ask that you commit to RC full time, and that you work on open source software while you're here (i.e., don't work on proprietary code or code that you couldn't pair on or share with others).

---

## You belong here

Regardless of how much experience you have, we admitted you because we believed you would be a positive influence on the batch, and we think being at the Recurse Center can help you dramatically improve as a programmer. This is really important, so we'll say it again: Everyone is here because we *want* them to be here, so if you're reading this, *don't worry* about how much more or less other people in your batch know. **You are here because we want you to be here and believe in you.**

---

Everyone has something to learn from their peers at RC, regardless of how much or how little experience they have. No matter how long you've been programming, you don't know everything there is to know about it: people who are relatively new to programming have plenty to share with those with a decade or more of experience. And if you have decades of experience, sharing your knowledge and taking on ambitious work can be a transformative example for other people at RC.

## *The environment*

Regardless of whether you attend online or in-person, the core of RC is the same. You should check out our [About](#) page to learn more about the experience.

## The self-directives

What should you actually do while you're at RC in order to grow and learn the most?

The answer to this question is summed up in three guiding principles called the [self-directives](#). They are:

- Work at the edge of your abilities
- Build your volitional muscles
- Learn generously

We recommend that you think about how to apply the self-directives to your time at RC before your batch begins. Doing the self-directives looks different for everyone, but here are some tips to get you started:

- If you're tackling a big project that's at the edge of your abilities, you might feel stuck sometimes. As one alum put it, try to have **'brain work'** — bigger, challenging stuff that requires deep focus — and **'finger work'** — smaller implementation work you can turn to when you need to rebuild momentum.
- When considering the big picture of what you want to work on at RC, **pay attention to what makes you most excited**. Do you get butterflies thinking about a specific project or avenue of inquiry? Go for that! And remember: it's okay to change course if what you're most excited to do changes. During your batch you'll encounter lots of new people and ideas. Building your volitional muscles means building your capacity to make decisions about what you do based on your motivations, and it's an ongoing process, not something you do once in your first week.
- **Share what you're stuck on**. We have lots of ways for you to share what you're working on, and we encourage you to celebrate victories! But it's also important to ask questions and express uncertainty in public: it's highly likely someone else is struggling with the same thing you are, or struggled with something similar in the past. Posting on [Zulip](#) or sharing a bug that's besting you in check-ins or presentations will help you get advice and new perspectives on your problem.
- **Pair program**. This is a great way to do all three of the self-directives. See the next section for more about that!
- **When in doubt, program**. The more code you write at RC, the more you'll learn. Don't worry about choosing the "perfect" project before you begin. It's much better to just start programming something, anything, that you're excited about. Even something that feels small can help you build momentum.



## Pair programming

One of the most valuable ways to do all of the self-directives at RC is by pair programming. Most alums wish they'd spent more time pair programming during their batch. We learned this after the second batch, and have shared this fact with every batch since. Despite this, lots of people still tell us after their batch that they really wish they'd paired more.

## What is pair programming?

Pairing means two people working on the same code using a single computer, either via screensharing on a video call or by sitting next to one another. Using only one computer is important, because pairing is about actively collaborating and not just two people working on the same project or sitting next to each other on their own computers.

## Pairing quickstart guide

### *Finding someone to pair with*

The best way to do this is to share what you're working on regularly! That will help people find you. Check out the checkins channel on Zulip to learn about what people are working on, and don't be afraid to reach out if something sounds interesting. Check out more specific interest channels on Zulip, too, especially the pairing channel, where you can look for pairing buddies. In both the virtual and physical spaces we have designated

pairing stations: you can use them to pair, or you can work at one by yourself to signal that you're open to pairing.

### *Finding someone to pair with*

It's good to make sure you have similar (or at least compatible) goals before you start pairing. If one person thinks the goal is to learn Python, and the other thinks the goal is to fix a bug as quickly as possible, you can run into friction. Make sure you're on the same page, or at least know where each other are coming from.

A common way to pair is having a "driver" and a "navigator": the driver is the one actually typing the code. The navigator sits next to the driver and reads the same screen, and is there to talk through ideas, direct the driver, and think about how the code currently being written fits into the larger project. If you're pairing this way, try to switch it up sometimes: switching regularly helps with knowledge transfer, ensures that both people get to drive and know what's going on, and helps keep everyone energized.

### *Getting through tough spots*

Pairing can be hard at first, especially if you haven't done it before. You may not think you have any code that is pair-worthy, or any valuable experience to contribute. It might feel uncomfortable programming out loud. You might not be sure if anyone shares your specific interests. But if we can give one piece of advice most strongly from our own experience and that of the thousands of Recursers who came before you, it's that it's worth pushing through these challenges.

### *Why we love pairing*

There are several advantages to pairing. One is that it keeps everyone focused on the task at hand. When you're working alone, it can be easy to get distracted (e.g., by email or a shiny new JavaScript framework), but it's much, much harder to switch to Gmail when someone's sitting by your side and also reading your screen. Successful pairing should yield a productive feedback loop where each person helps keep the other focused.

Another advantage is that the navigator can catch a lot of small errors immediately (typos, missing semicolons, incorrect variable names, etc). It's faster and easier to fix bugs the moment they're introduced than at any subsequent time, so this ends up being a big win. Plus, as the saying goes, with many eyes, all bugs are shallow.



Pairing can help with everything from learning a language to picking up editor or command line tricks. And it's motivating: it can get you excited about your project, stay with a problem when you might otherwise get distracted, or allow you to take on projects of a bigger scope than you could alone. Working on something deeply together can be a rare and special way to connect with another person.

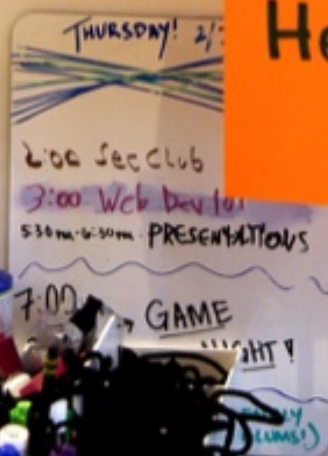
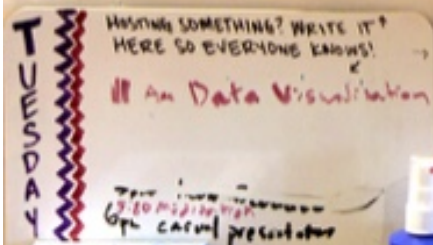
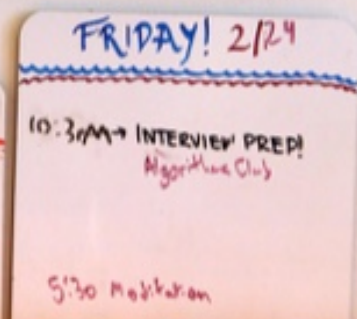
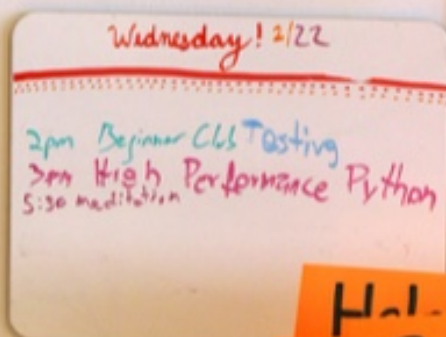
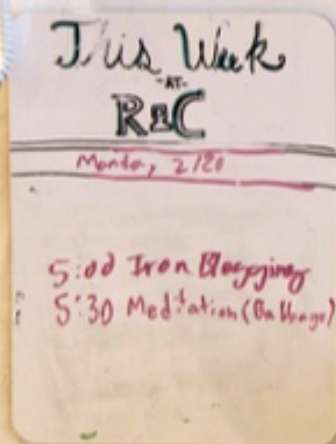
Oh also, it's fun!

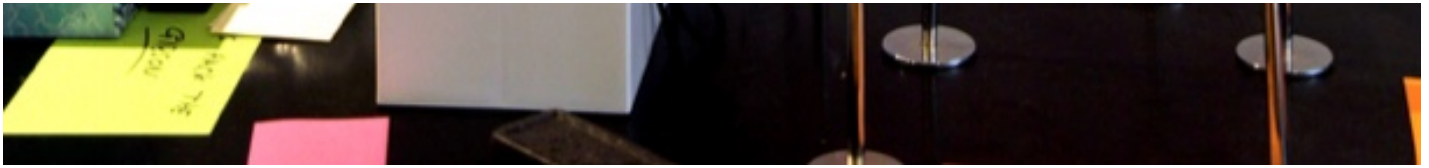
For all these reasons and more, pairing is one of the best ways to practice all three of the self-directives. We think everyone should try it — at least a few times and a few different ways.

## The social rules



To buzz someone in: pick up the phone , then press the lowest  button





We have a small set of lightweight [social rules](#) that make more *explicit* certain social norms that are normally *implicit*. Most of our social rules really boil down to "don't be a jerk" or "don't be annoying." Of course, almost nobody sets out to be a jerk or annoying, so telling people not to be jerks isn't a very productive strategy. That's why our social rules are designed to curtail specific behavior we've found to be destructive to a supportive, productive, and fun learning environment. The social rules are:

### **No feigning surprise**

You shouldn't act surprised when people say they don't know something. This applies to both technical things ("What?! I can't believe you don't know what *the stack* is!") and non-technical things ("You don't know who RMS is?!").

### **No well-actuallys**

A well-actually happens when someone says something that's almost - but not entirely - correct, and you say, "well, actually..." and then give a minor correction. This can be very disruptive when the correction has no bearing on the crux of the conversation.

### **No back-seat driving**

If you overhear, or read on Zulip, people working through a problem, you shouldn't give advice unless you're actively participating in the conversation.

### **No subtle -isms**

Our last social rule bans subtle racism, sexism, homophobia, transphobia, and other kinds of bias. Subtle -isms are small things that make others feel unwelcome, things that we all sometimes do by mistake.

You should read more about the social rules [here](#), including how to gracefully handle it when you unintentionally break them. These rules are designed to help all of us build a pleasant, productive, and fearless community.

### **Code of conduct**

The social rules don't cover harassment or discrimination. For that, we have a [code of conduct](#) enforced by the RC faculty. All members of the RC community are expected to abide by our code of conduct.

You can report code of conduct violations using [our reporting form](#).

## *Logistics*

### **The time commitment**

Our core hours are Monday through Friday, 11 am to 5 pm ET, with many additional events outside of those hours. We don't track attendance down to the hour, but we will check in if we don't see you for a day. Our expectation is that RC is your primary commitment while you're in batch, and that you're committing to joining us during our core hours for the entirety of your batch.

RC doesn't work well for people who aren't ready to make a significant commitment to it. We say six hours a day, but that's a proxy for the commitment we expect you to make rather than a strict requirement in itself. It's difficult to impossible to do RC when you have a full-time or part-time job, or are going to school, and we expect that anyone attending a batch will not have any other significant commitments.

We've seen people try to juggle classes and part-time work with RC, and they invariably end up getting less out of the experience than they want to — most wind up leaving their batches early.

If you miss the first day's orientation events, we'll ask you to join a later batch. If we notice that you aren't participating in your batch, we'll reach out to you, and may ask you to end your batch early. If you're unable to continue doing RC full-time, see the [ending your batch early](#) section below.

## Batch length

Regardless of the batch you apply for, if you're admitted to RC you'll be able to confirm for any open batch and will be able to choose to commit to six or 12 weeks. If you're unsure of whether you'll be able to attend for the full 12 weeks, confirm for a six-week half-batch: you have the option to extend to 12 weeks at any point during your batch if you let a faculty member know you'd like to.

If you need to shorten your batch, that's okay — again, let a faculty member know. We would much rather hear that you're struggling to participate or need to end your time at RC early than notice you've stopped participating and have to track you down!

## *Advice*

### Before RC

Taking time out of your usual routine to attend an RC retreat is an exciting and sometimes daunting prospect. The Recurse Center is as much a social hack as anything. You spend a few months on your own somewhere, programming solo for its own sake, and that's what many Recursers were doing right before RC. Here you'll have the accountability, momentum, and inspiration from a group of peers who care about building and learning as much as you do. (And you'll have an easy answer for your friends and family when they ask what the heck you're doing with your life.)

Pre-batch onboarding focuses on helping you introduce yourself to the RC community and our tools, and meet your batchmates. You'll get an invite to **Zulip**, our chat software, before your batch. You should introduce yourself there, and choose some channels to subscribe to. If you'd like to learn more about how Zulip works, you can [read the docs](#).

**Look for the orange events** on the RC calendar — those are your orientation events! You must attend the first day talks on the first day of your batch. You can come to as many of the other orientation events as look



interesting to you (ideally all of them).

**Look at the Directory to see who's in your batch, and add a photo and bio on your settings page.** After you check out who else is in your batch, think about what you might like to work on, and who you'd like to meet and pair with in your first few weeks.

And finally, if you're attending remotely, adjust your workspace a little bit to change your context: change where you sit, add a little plant to your desk, or rotate your workspace.

## During your batch

### Your first two weeks

Your first day will kick off with a series of welcome talks and meet and greets. The first day events are an important part of your batch, so don't miss them! If you do, we'll likely ask you to attend a later batch. We'll talk about RC, the self-directives, and the social rules, and there will be lots of time to get to know your fellow Recusers.

The first week will be full of fun events for meeting people, learning about what they're working on, and sharing your own ideas. Here's some advice for your first couple of weeks to help you get into the swing of things.

### Meet people!

Get to know your batchmates! The more you invest in building relationships at the beginning, the more you'll be able to take advantage of being surrounded by so many kind, smart, ambitious people who are excited to learn together.

*How?* Sign up for coffee chats, attend events you find on the calendar, and reach out to people who have similar interests.

### Choose an accountability mechanism and then commit to it.

At RC, you're primarily accountable to yourself. You're setting your own goals, working and learning at your own pace, and following your curiosity.

Making your goals and progress public gives you room to reflect on what you're doing and why, support from your batchmates, and (if it's written) a record of what you've done at the end of the batch. Writing or talking about what you're working on helps cement it for you, and can help you see possibilities and pitfalls you otherwise wouldn't. This kind of reflection is at the core of building your volitional muscles.

*How?* Posting checkins on Zulip, attending daily checkins, coming to office hours on a regular basis, pinging a friend at the end of every day, committing to giving a presentation, blogging, or setting up your own sort of structure.

### Create the context you need to do good work.

Our hope is that you're able to fully immerse yourself in RC during your batch by separating this time from the rest of your life. A big part of this immersion comes from the context you put yourself in.

You might have experienced a state of flow while you worked in the past: you have momentum, you're super focused, the minutes fly by, and the difficult things aren't easy but they don't stop you. The feeling has a lot in common with excitement, and if you can get into it, we think it's the optimal state to do RC in.

This extends beyond your physical environment. What do you need to tell yourself about what you're doing — why are you taking the time to do RC? Remind yourself as often as you need to. Are there friends or family members who aren't sure about you taking three months away from work or other obligations to do RC? Take the time to have those conversations, so you have their support.

If you find yourself struggling to do a certain kind of work, you can also ask yourself what kind of work you are in the mood to do. Maybe you're trying to force yourself to do a Leetcode problem when what you really want to do is something inspired by an event you went to earlier in the week.

*How?* If you have experienced flow before, think back to it. What mood were you in? What was the space you were in like? What time of day was it? Try to recreate the situation, and see what happens. If you haven't, pay attention to and try to adjust your environment, keep programming, and see what you learn.

### **Say no to something.**

You'll come into RC with a rough plan or goal for your time, and you'll likely encounter more things than you have time to do in your first week. That makes it necessary to decide what you will and won't spend your time on.

But treating time as a scarce resource makes it scarcer: if you try to over-optimize or overthink, that will take time away from your work! Simplify things for yourself. Say no to the stuff that you feel you should do, but that isn't intrinsically exciting to you.

*How?* You'll likely sign up for a lot of social things in your first week (it's encouraged!) – things like pairing sessions, social events, and working/study groups. You may then need to step back from some of them to balance your working and social time. Take stock of how you feel on day one of week two, and say “no” to the things that aren't serving you.

### **Do one challenging thing, and then do another.**

This is a concrete way to ‘check’ if you're working at the edge of your abilities. The reason to work at the edge of your abilities is to incrementally grow what you're capable of at RC. You'll do that by doing work that's challenging one day, then waking up the next and doing the same.

*How?* Ask yourself if what you're working on is the right level of difficult where you're learning something new. Challenging doesn't mean impossible: the edge of your abilities is a sweet spot of difficult but motivating. If you're pleasantly surprised by the work you've done at the end of a week, it's a good sign.

### **A checklist for getting started:**

---

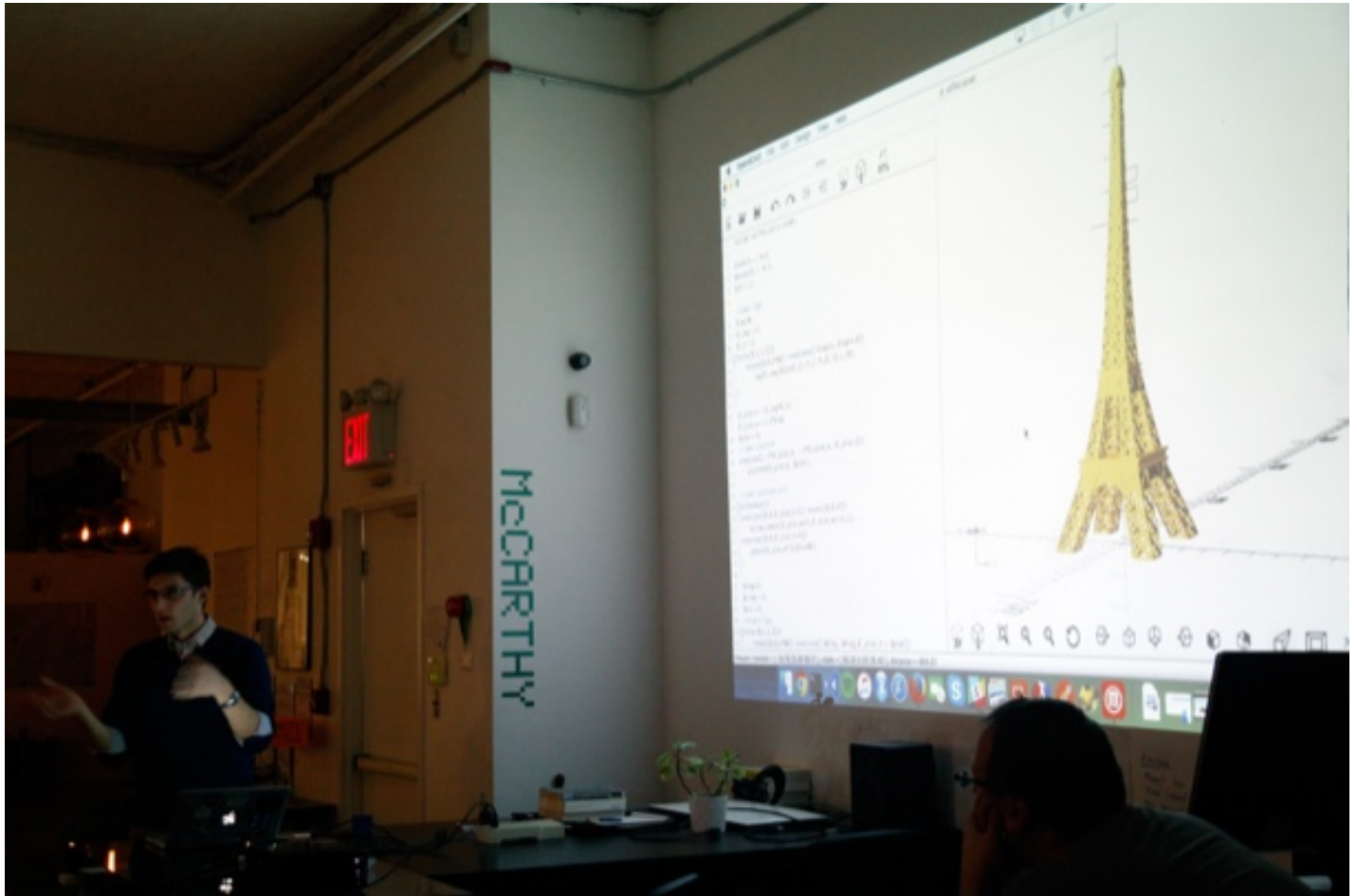
PAPER PRODUCTS!

[Click here](#) for a printable copy of this checklist!

---

- Create a [recurse.com](#) account and fill out your profile on the Directory

- Log in to Zulip and introduce yourself in the welcome stream
- Sign up for coffee chats
- If you're coming to the Brooklyn hub, see [hub details](#) for important space-related notes
- Get settled in your physical environment
- Make a desk on Virtual RC if you're attending an online batch, or pick a cubby if you're attending in-person
- Choose an accountability mechanism, whether that's in-person or Zulip checkins, blogging, or something else
- Attend presentations
- Check the calendar and RSVP to events you're excited about
- If there's an event you wish existed but doesn't, create it!
- Set your intentions/goals for the retreat and plan what you'd like to work on
- Share what you're planning on working on widely and often to find others who are interested in similar things



## The midpoint of your batch

Whether you're doing a six- or a 12-week batch, halfway through is a great point to stop and reflect on how your batch has been going. Take some time to think about whether or not you've been working at the edge of your abilities and about your goals for the remainder of your batch. You may need to change your plans

multiple times – that’s normal. Setting and resetting your expectations for yourself will encourage you to keep learning.

If you’re doing a twelve week batch, this is when one batch ends and another begins. You’ll have to say goodbye to some of the people you’ve been learning and working with in your sixth week (don’t worry – you don’t really have to graduate). You’ll also be saying hello to lots of new people in your seventh. Try to attend the first week events so that you can meet your new batchmates and find new collaborators!

If you are thinking about using RC’s career services we recommend reaching out to a career facilitator in the latter half of your batch on Jobs Chat.

## **The end of your batch**

When your batch ends, you’ll “never graduate,” which means that you’ll become an RC alum. We have a ceremony on your last day where you’ll be able to get your t-shirt and a collection of kind words your batchmates have sent to you.

Alums are still very welcome to participate in RC when they’re not in batch, and many do: as an alum you can attend any event you like in Virtual RC, pair program, participate in conversations on Zulip, run events, present at presentations, apply for another batch, recommend friends, and so on.

## **Ending your batch early**

If your circumstances change and you can’t or no longer want to participate in RC before the last day of your batch, please let us know you’d like to withdraw from your batch. It’s helpful to us if you let us know why you’re withdrawing and we may ask to have a short conversation with you to learn more about your decision. We’ll let you know how to best communicate your change of plans to the rest of your batch, and we’ll deactivate your accounts on Zulip and recurse.com until you return for another batch. If you’re an alum and you’re withdrawing from a batch that isn’t your first, we’ll remove you from your current batch but we won’t deactivate your accounts.

**Please reach out to us as soon as you can.** We value clear, proactive communication, which saves us time and energy checking in with you and hopefully also saves you some guilt and anxiety. This is not a regular school or job, where you face discipline for “failing.” We view honest self-reflection and discernment positively. If you need to withdraw from the batch you’re attending, you’ll be welcome to confirm for another batch that begins within six months, or to re-apply in the future.

## **The physical and online spaces**

Batches are hosted at our NYC location at 397 Bridge St, in Brooklyn. See [hub details](#) to learn more about how our physical space works.

You have the option to attend RC in-person, remotely, or any combination of the two. Regardless of where you do your retreat, all Recursers share access to several shared online spaces: Zulip, Virtual RC, Community, and our Directory.

Our Brooklyn hub has two floors, lots of side rooms, a vintage computer lab, and plenty of hardware and other equipment to experiment with. You’ll need to bring your own laptop and charger, and you’ll get a keyfob and

cubby if you're spending all or most of your batch there.

As an alum you'll be welcome to visit the hub any time!

## **Zulip**

We use Zulip for real-time chat. It's great for asking questions, getting code review, organizing lunch trips, etc. If you're just getting started, you'll be shown a quick-start tutorial to get you up to speed.

You can learn more about how we use Zulip at RC on our Wiki, and how Zulip itself works in [their docs](#). If you have a question about Zulip, you can post it on the small questions stream, or ask any faculty member. RC is primarily a technical space, but you'll find plenty of channels for other interests (read: cats), locations, and fun stuff, too.

When you join RC, you'll be subscribed to a small set of channels, and during your first week you'll get your own checkins topic. You can subscribe to more – we recommend ramping up slowly, so that Zulip doesn't become too overwhelming.

## **Virtual RC**

Our online space is called Virtual RC. It's software we built so that you can see who else is online, join meetings, share photos, join audio calls, chat, and more. We built Virtual RC to mimic the feeling of being at a retreat together in a physical space. You can set your status and share what you're working on, click on rooms to attend events or roll over sticky notes to see what's happening, and pair program at the pairing stations. Make sure that your name on Zoom is set to the same name you use on your RC profile. This allows Virtual RC to show whenever you're in a Zoom room.

Alums are welcome in Virtual RC, at events, and on Zulip 24/7, and we'd love to see you there, regardless of when you attended. If you'd like to host an event in Virtual RC, please do! Just select a location on the calendar that corresponds to a room in Virtual RC, and people will be able to join you there when it's time for your event.

## **The Directory**

The RC Directory is where you'll find profiles for everyone in the RC Community. You can filter it by location, batch, and who's at RC now. You can search for interests to find people you might like to talk to or work with. Make sure to fill out your profile and add a photo on your settings page when you confirm for a batch so people can find you, too!

### *A note about names and faces*

We ask that everyone use a clear photo of their face and a name consistent with their identity outside of RC in the RC Directory. That's because we believe that knowing each other is an important foundation for the culture we want to cultivate in the RC community, where people are kind and accountable to each other, and where it's safe to make mistakes and grow from them. Building a community where people trust each other is hard, and it gets harder as the community gets bigger. One way we work to build this trust is by making it easy to figure out who everyone is. Anonymity – even partial – doesn't foster relationships that support personal or technical growth.



We care that you have a consistent identity within RC – in person and online – which is also consistent with your identity outside of RC. If you've long been known in the programming community by a handle-like name that is clearly associated with you as a person, you can use that name at RC. You should use the name that you're most comfortable with, and that most people know you by. You can upload a new photo at any time on your settings page, whether it's because your appearance has changed or you simply don't like your old one. A small number of Recursers prefer not to have photos of themselves online anywhere, and we respect that as well. If you have questions about this policy or the process to change your name at RC, reach out to us.

Using consistent names and clear pictures makes it easier to trust the people you're talking to and makes people less likely to act maliciously. As a concrete example: In the past, people could change their names on Zulip and started using handles, often as jokes. Zulip was less friendly and more confusing, and we made the "no handles" policy and changed our Zulip instance so that people can't set their own names. This has made Zulip better. We think this small constraint is worth the overall benefit to the community.

## Resources and tooling

If you don't have access to one of these services, please email [ops@recurse.com](mailto:ops@recurse.com) and let us know. You should have been added (or invited) to all of them when your batch started. If you're missing an invitation, please check your spam folder.

### Website

You're here! When you're logged in, you'll find links to everything you'll need at RC in the top menu bar. You'll also be able to access the RC Directory, which can help you find people who share your interests. You should fill out your profile when you confirm for your batch so people know what you're interested in, too!

### Calendar

The RC website has a built in calendar, which you can use to find out what's happening at RC and schedule events. The calendar is not just for things happening in the physical space, but for the RC community at large.

One aspect of the calendar that warrants special attention is that every event is editable by everyone. Feel free to edit events to add useful information without getting special permission to do so. For instance, if the workshop you attended ended up moving from Lovelace to Hopper at the last minute, you can correct the location. And if the person who ran the workshop posted their slides on Zulip afterwards, you can add them to the description to make them easier to find in the future.

Please be respectful when making changes. For instance, don't change the timing or description of an event if you aren't the host and haven't asked them. Or, if you are the host, make sure you check in with people who RSVP'd before making a big change to an event: they won't be notified if you do.

### Wiki

The RC Wiki lives on GitHub and is accessible at [wiki.recurse.com](https://wiki.recurse.com). It has internal documentation about RC, resources for project ideas, interview advice, and tutorials. The wiki is only visible to members of the Recursers team on GitHub. You can get access by authenticating with GitHub on your settings page.

We did a clean-up of the wiki in mid-2024, but RC faculty doesn't regularly maintain it. You are encouraged to update and add to it to keep it up to date!

## Issue Tracker

The Issue Tracker is where we track and discuss the things we're trying to improve about RC. You can use it for general community issues (e.g., how to make the space less noisy) as well as software-related issues (e.g., broken links on recurse.com). If you follow this repo on GitHub you will receive emails when new issues are created, commented on, and closed. Like the wiki, you can get access by authenticating with GitHub on your settings page. Anyone in the community is welcome and encouraged to add issues or comments.

## Community

Community is our discussion forum, and was developed by faculty and other Recursers. Community is open source software.

We don't use Community as much as we used to, but it's a good place to find housing and organize location-based events. Once your batch starts, it's worth browsing Community's subforums and subscribing to those that interest you. You'll receive emails when new threads are made in subforums you're subscribed to.

## Custom subdomains

Recursers can [create recurse.com subdomains](#). This can be useful for:

- Internal tools for the RC community. E.g., `zulip-bots.recurse.com`
- Personal websites and blogs. Please don't register first name-only subdomains. E.g., use `zachallaun.recurse.com` instead of `zach.recurse.com`.
- Other things we haven't thought of. Please use your best judgement, and ask a faculty member if you're unsure or have any questions.

## Sharing guidelines

Zulip and Community are private. The only people with access are RC faculty and members of the RC community. Content is not publicly searchable. While they may be online, these conversations are akin to private talk amongst friends. Please get explicit permission before you share things people write on Zulip or Community outside of RC. You're welcome to share links to public content (e.g. a link to someone's blog post).

Please don't post job listings for programming roles on Zulip or Community. This makes it harder for RC to succeed as a business and adds to the noise on our forums. The one exception to this policy is the contracting stream on Zulip, where you're welcome to post about freelance and part-time contracting work, as well as internships, since RC only recruits for full-time jobs.



## Challenges at RC

RC can sometimes appear to be a place where everyone is productive, totally comfortable in the environment, and feels great about themselves and their work. But that isn't always the case: you'll face challenges at RC, but likely won't see everyone else wrestling with their own. This section exists to acknowledge challenges some Recursers struggle with.

If you're struggling at RC, you aren't alone: it's likely that someone else has dealt with what you're dealing with. Reach out for help.

### **Lack of structure**

You might be coming to RC from more structured environments, like work or school. Not having imposed structure and being accountable to yourself can be a big adjustment. You might struggle to decide what to work on, feel unproductive, or feel like you're imposing by asking others for help. Or you might have too much going on: RC is very social, especially in the first week, and your attention will likely be divided between what you came in wanting to work on and all of the cool people and events around you. A big part of RC is finding a balance that works for you.

You'll have to work through these feelings. If you feel overwhelmed by options, don't be too precious about what you decide to do on a given day: it's better to make a bit of progress and build momentum on an imperfect thing rather than spend days deciding on the most optimal way to spend your time. You can also offer to pair with someone on their work if you're feeling stuck on yours. And don't be afraid to say no to things or to change

your plans, either: surprising yourself by what you end up working on at RC is about as universal an experience as we have.

## **Uncertainty**

This is an uncertain time for many people. Maybe your schedule has changed dramatically, you feel guilty for stepping away from family obligations, you're facing difficulties at home, or you're worried about what comes after your batch.

There's change inherent in RC, too: six weeks into each batch, the next batch of Recursers shows up and the previous batch leaves. This transition can be jarring. You might struggle with half your RC friends being suddenly gone and replaced by a bunch of new faces, or with the culture of RC changing.

But there are things you do have control over: you can choose how to spend each day, and how to show up for yourself and your batchmates. Take RC as an opportunity to become more comfortable with uncertainty and change alongside other people who are experiencing the same things.

## **Fear**

We think fear is one of the most pernicious impediments to education, and we do our best to remove it from RC. In most of the world, but especially school and work, people are afraid of looking stupid. This could be because they've had bad experiences in the past, they're experiencing impostor syndrome, or they're afraid of looking like they don't understand something in a new group of people. This fear frequently keeps us from asking important questions like "how does that work?" or even just "why?" Worse, it keeps us from saying "I don't understand." That means many of us muddle on with a half-baked or entirely incorrect understanding of core concepts. This is particularly bad with programming, because these misunderstandings compound, and over time become harder and more embarrassing to admit to and address.

We can say that you shouldn't be afraid at RC, but you'll need to experience it to understand and trust that that's true. Try reading the small questions and checkins streams on Zulip to see how often Recursers ask questions and how generously everyone helps each other learn and grow.

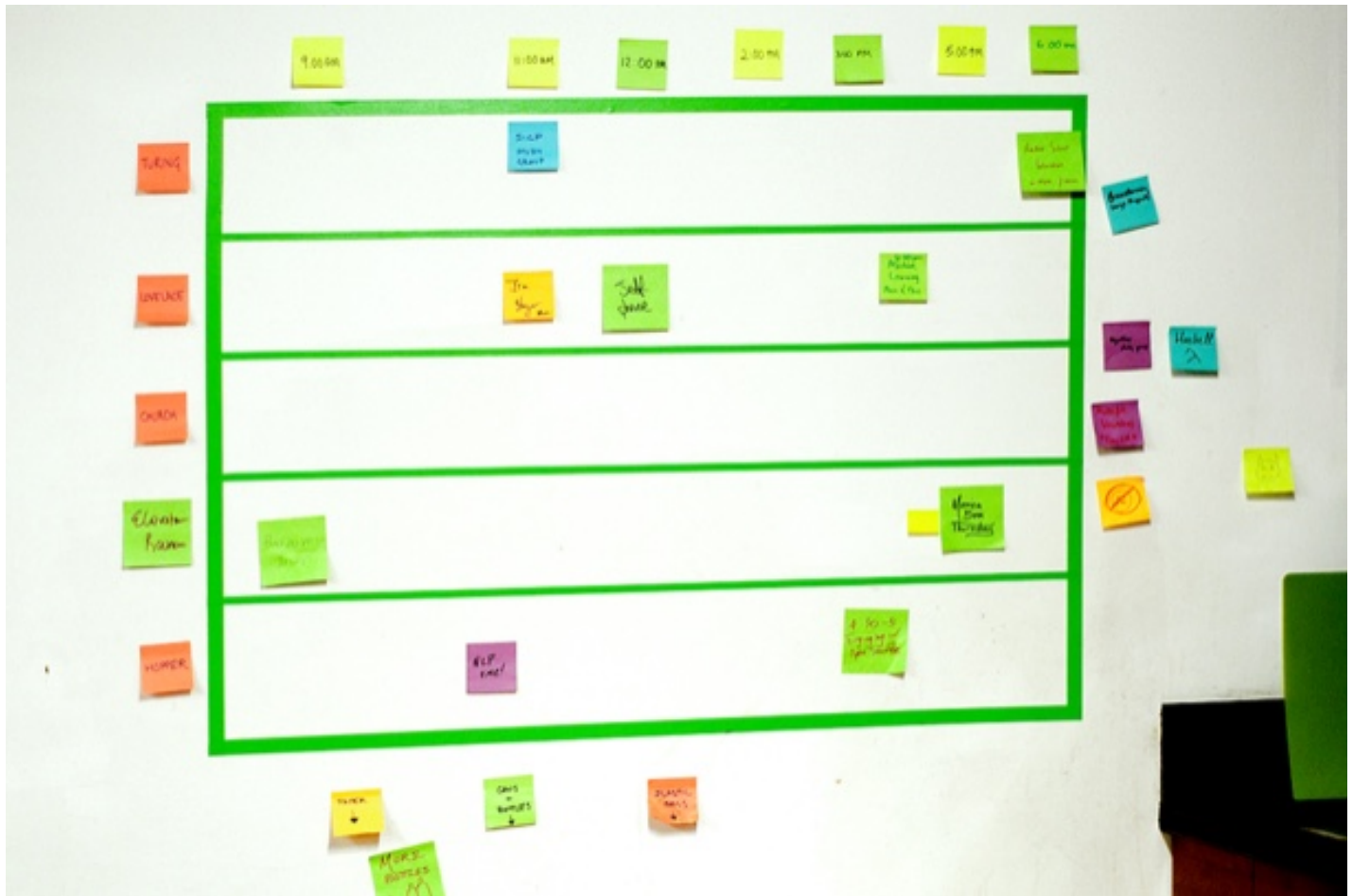
## **Burnout**

You might be coming to RC to rediscover a love of programming you lost at work or school. That can be a lot of pressure to put on yourself! Try to especially follow the things that excite you at RC – don't do anything you feel like you 'should' do because of some extrinsic reason. Take lots of walks, and give yourself lots of time and space to think and engage with the work you're doing. Regardless of how you feel at the end of your batch, you'll have learned something valuable about yourself and will be better prepared to take the right next steps for you.

## **Mental health**

Some Recursers struggle with mental health issues, which can add an extra level of complexity to navigating their time at RC. Smart, self-motivated people can have mental health issues, and there is no shame in getting support in dealing with them. Don't hesitate to reach out to us if you need help. While we don't have anyone on staff who is a mental health professional, we will do what we can to connect you to people who are, and other

resources you may need. And if you need to take a break from your batch, that's okay — you should take care of yourself, and you can come back in the future.



## Never Graduate

Alums are welcome to program, chat, host and attend events at RC all day, every day, regardless of when you attended RC! Many folks choose to stick around after their batches and continue to attend checkins, present, and have coffee chats — this means you can effectively do a 14-week batch, if that lines up better with your plans.

We would love to see lots of alums at the hub programming, pairing, and collaborating with Recursers in batch. Please do not use the hub to do work for your day job, or to take meetings.

We recently got rid of our limited 'alum hours' in favor of allowing alums to visit any time, since we have capacity at the RC hub. Depending on our plans and demand for the space, this may be subject to change.

## Re-admissions

You're always welcome to **apply for another batch**. You are welcome to come back for another batch of RC after you finish your first — the current record is four full batches! You'll need to fill out a written application, but won't need to interview. If you'd like to do 18 or 24 weeks at RC, you can apply for a second batch during your first.



## Recommend an applicant

Most Recusers find out about RC from friends and colleagues! If you'd like to put in a good word for someone, there are two ways to do it:

1. You can submit a recommendation (or anti-recommendation) for people who have applied or who you think may apply. Your recommendations are only visible to RC faculty and relevant alumni interviewers and will be taken into account when we do application review.
2. If you've programmed with someone before, you can give them a Golden Ticket. The Golden Ticket you write will allow the person to skip the conversational interview, which will speed up the admissions process for them!



## Jobs

RC makes money by recruiting. We see recruiting and education as integrated parts of a whole, each of which gets better as the RC community gets bigger, more diverse, and more active.

Recruiting is a lifelong service for Recursers – we’re a resource during your batch, after, and even years into the future.

To learn more about getting a job through RC, you can read our jobs advice and get in touch with us on Jobs Chat. Here's what we offer:

- We'll work with you to understand what you're looking for and your goals, and provide personalized recommendations for roles with our partner companies
- We'll only connect you with companies you're genuinely interested in
- We'll be here to help guide you throughout your interview process and provide advice, be a rubber duck, help craft communications, and help you negotiate
- We can connect you to alumni mock interviewers who are experienced at giving tech interviews, so you can practice before the real thing
- We can connect you with alums at companies to help you learn about what it's like working there.

We have partnerships with over 100 companies who pay us to hire Recursers for programming jobs.

If you're hired by a partner company, they pay us a fee (usually 25% of your first year salary) provided you stay at the company at least 90 days. This fee is paid out of the company's recruiting budget and does not come out of your salary. There's no obligation to get a job through RC or to only interview at RC partner companies but it's worth noting that recruiting fees are RC's only source of revenue. If you are interested in a programming job after your batch or at any point in the future, please get in touch!

## Want to hire from RC?

Some of our best partnerships have come from introductions Recursers have made. Learn more about how to get your company to hire from RC (not a Recurser? [Learn more about hiring from RC here](#)).

## Jobs software

- **Jobs Chat:** A private jobs dashboard where you can chat with RC faculty to keep track of the companies you're interviewing at or considering.
- **Jobs profile:** Update your jobs profile to share your background, projects, skills, and work preferences with companies.
- **Company browser:** Browse and research companies RC works with.
- **Jobs settings:** Set your visibility and contact preferences for companies.

## Advertising jobs to the RC community

Please don't post job listings for programming roles on Zulip or Community. This makes it harder for RC to succeed as a business and adds to the noise on our forums.

The one exception to this policy is the contracting stream on Zulip, where you're welcome to post about freelance and part-time contracting work, as well as internships, since RC only recruits for full-time jobs.

If you're an alum working for a company we don't recruit for, we also ask you not recruit Recursers to work with you. If you work at a non-RC company and would like to start recruiting Recursers, or if you work at an RC company and want to know how you can help hire more Recursers, please email us.

# Contact information & employee hours

---

## Editor's note

This section contains the email addresses and phone numbers for all Recurse Center employees, and is only visible to members of the Recurse Center community.

---

---

[About](#) [FAQ](#) [Code of conduct](#) [Blog](#) [Team](#) [Hire](#) [Who comes to RC](#)

[Apply](#)

---

 [LinkedIn](#)  [Twitter](#)  [Instagram](#)