

If you're reading this, it's because you've been selected after a multi-stage process that likely involved a few hundred candidates from the initial screening stage until your final interview. Welcome.

We don't consider ourselves to be an ordinary team, which means there is a good chance you don't consider yourself average either. Here is a basic survival guide for Dreamers Inc.

### **Culture:**

-We are all here, at least in part because we see the internet as a playground with fun toys, new tools, and a crazy future. In short, we love to learn. So learn from people, and teach people, with an open mind. Approach complex problems from first principles, and try to understand what is really going on whenever practical. Appreciate the power in that method; it got us to the moon and then accomplished nuclear fusion. Share your expertise with others, and they will share it with you. Be light and be kind, even if somebody fucks up. It happens.

-Care about your own work. The most basic principle of our team can be summed up as: **give a damn**. The bare minimum may get you by, but it won't get you better. Life has more to offer than "just okay". Quality of work should be more important to us as a team than to our client as a business objective. We can count our nickels until we are blue in the face, but time is something you can never get back, and taking pride in your work is one of life's simple pleasures.

-The above doesn't mean we have to take ourselves too seriously or suffer from humorless extremism in our approach. If you are criticizing somebody, do it constructively and with kindness. Give credit where credit is due.

-Give a damn about *the team's* work, and not just your own. The difference between building cool hobby projects and executing world changing ideas is the ability to collaborate. If somebody else is doing a poor job, even somebody there before you, speak up about ways we can improve. When you are building something, ask yourself about the actual business case, and try to understand why you are building it, and even whether it is truly necessary. You are here because we value your input, so don't assume that our way is the right way. We are technology agnostic; no tool is the single best tool in the same way a hammer is not superior to a screwdriver.

-On meetings; if it can be summed up with a slack message or an e-mail, we prefer to save valuable time. Respect the time of others by consulting our dear friends [stackoverflow](#), [LLMs](#), and [google](#) before bringing a problem to the chat, but understand it is your responsibility to ask questions whenever you are truly stuck. Many ideas are complex

enough that they require interactive discussion, and sometimes it is necessary to get on the same page on a call. If you feel a meeting is necessary, call one.

-We support the consideration of crazy ideas. We are not a formal team with a very structured hierarchy, and we're all pretty easy going people. If you have something out of the box that may fail, bring it up. Glorious failures are better than dreams locked away in a drawer.

-If you find you are not enjoying the work you are doing, that is an essential problem. No job is completely perfect, and all work on earth contains some amount of tedious tasks. Find your balance, but remember time is the most valuable resource you have, and it is your own responsibility to aim at actuating your massive potential. You should always be challenged in some capacity, and improve your own skills. We do not believe in a high stress work environment. Incredible things can be accomplished with good humor. So if you find the work is increasingly draining or not allowing you to improve, talk to Adam directly.

### **Writing Code:**

-Ask yourself before writing something if it is the best way to do it, whether or not the template was already there when you arrived. Think about a team's culture of convention as valuable outside the project itself and try to enforce best practice in your own work and in the work of others. Keep a light and elegant solution where possible (keeping principles like Do not Repeat Yourself and Keep It Simple Stupid), as there is art in expressing code beautifully.

-Use LLMs responsibly. If you need resources to do your job better, ask for them.

-Be conscious of making whatever you push easy to follow and self-documented, so as to increase the project's [bus factor](#). Do not push directly to master or staging branches without review; instead, assign a merge request. When you do issue an MR, make sure it's tested and clear of conflicts. [Small, frequent commits](#) are better than one large one, especially if you are just starting with a new team. This allows for more efficient testing and review.

-Communicate enough so that people are aware of the status of your task. If you are being asked this a few times a week, you are not communicating enough.

-Write tests where valuable, and QA your own work before considering it done. Coding is only part of the job.

-Be thorough in reading and addressing comments in MRs. It is inefficient if people have to say the same thing twice, and it communicates that you are not valuing their input enough to pay attention to their questions or comments. In the reverse, if somebody assigns you

something to review, try your best to either stand behind their code as your own, or to leave comments that would make it so.

-For the curious among you, tell your bash shell: `echo "import this" | python`